# 1 Guides

1.1 Setting Up The ASK SDK .................................................. 3
1.2 Developing Your First Skill .............................................. 6
1.3 Sample Skills ................................................................. 16
1.4 Alexa Capabilities supported by SDK ................................... 17
1.5 Setting Up The ASK SDK .................................................. 18
1.6 Developing Your First Skill .............................................. 18
1.7 Sample Skills ................................................................. 18
1.8 SDK Supported Alexa Capabilities ..................................... 18

# 2 SDK Features

2.1 Request Processing .......................................................... 19
2.2 Response Building .......................................................... 28
2.3 Configuring Skill Instance ............................................... 31
2.4 Skill Attributes ............................................................... 33
2.5 Alexa Service Clients ..................................................... 37
2.6 Alexa Out-Of-Session Service Clients ................................. 51
2.7 Host a Custom Skill as a Web Service ................................. 55
2.8 Request Processing .......................................................... 59
2.9 Response Building .......................................................... 59
2.10 Skill Attributes ............................................................... 59
2.11 Skill Builders ................................................................. 59
2.12 Alexa Service Clients ..................................................... 59
2.13 Alexa Out-Of-Session Service Clients ................................. 59
2.14 Hosting Skills as Webservice ............................................ 59

# 3 Got Feedback?

# 4 Additional Resources

4.1 Community ................................................................. 271
4.2 Tutorials & Guides .......................................................... 271

# 5 Indices and tables

Python Module Index .......................................................... 275
Index ................................................................. 283
The **ASK SDK for Python** makes it easier for you to build highly engaging skills, by allowing you to spend more time on implementing features and less on writing boiler-plate code.

To help you get started with the SDK we have included the following guides. In the future, we plan to include more documentation and samples too.
1.1 Setting Up The ASK SDK

1.1.1 Introduction

This guide describes how to install the ASK SDK for Python in preparation for developing an Alexa skill.

1.1.2 Prerequisites

The ASK SDK for Python requires **Python 2 (>= 2.7)** or **Python 3 (>= 3.6)**. Before continuing, make sure you have a supported version of Python installed. To show the version, from a command prompt run the following command:

```
$ python --version
Python 3.6.5
```

You can download the latest version of Python [here](link).

1.1.3 Adding the ASK SDK for Python to Your Project

You can download and install the ASK SDK for Python from the Python Package Index (PyPI) using the command line tool `pip`. If you are using **Python 2 version 2.7.9 or later** or **Python 3 version 3.4 or later**, `pip` should be installed with Python by default.

Many Python developers prefer to work in a virtual environment, which is an isolated Python environment that helps manage project dependencies and package versions. The easiest way to get started is to install the SDK in a virtual environment. See the section *Set up the SDK in a virtual environment*.

Another option is to install the ASK SDK for Python to a specific folder. This ensures that you have the required dependencies and makes it easy to locate and deploy the required files for your finished skill. See the section *Set up the SDK in a specific folder*. 
Tip: The following steps showcase the installation process for the standard SDK distribution. The standard SDK distribution `ask-sdk` is the easiest way to quickly get up and running with the SDK. It includes the core SDK package, the model package, and the package for the Amazon DynamoDB persistence adapter that enables storing skill attributes in DynamoDB.

If you do not need everything in the standard distribution `ask-sdk`, you can install the core package and expand with individual add-on packages later.

For doing that, change the package name in the `pip install <package name>` commands in the following sections with `ask-sdk-core` for the core package and `ask-sdk-dynamodb-persistence-adapter` for the DynamoDB persistence adapter add-on.

### Option 1: Set up the SDK in a virtual environment

This option requires you to install the `virtualenv` package. `virtualenv` is a tool to create isolated Python environments. To get started, from a command prompt, use the following command to install the package:

```
$ pip install virtualenv
```

Next, create a new folder for your Alexa skill and navigate to the folder:

```
$ mkdir skill
$ cd skill
```

Next, create a virtual environment called `skill_env` by issuing the following command:

```
$ virtualenv skill_env
```

Next, activate your virtual environment and install the sdk.

**MacOS / Linux**

Run the following command to activate your virtual environment:

```
$ source skill_env/bin/activate
```

The command prompt should now be prefixed with `(skill_env)`, indicating that you are working inside the virtual environment. Use the following command to install the ASK Python SDK:

```
(skill_env)$ pip install ask-sdk
```

Depending on the version of Python you are using, the SDK will be installed into the `skill_env/lib/Python3.6/site-packages` folder. The `site-packages` folder is populated with directories including:

```
ask_sdk
ask_sdk_core
ask_sdk_dynamodb
ask_sdk_model
boto3
...
```

**Windows**

Run the following command to activate your virtual environment:

```
```

Run the following command to install the ASK Python SDK:

```
```

Depending on the version of Python you are using, the SDK will be installed into the `skill_env/lib/Python3.6/site-packages` folder. The `site-packages` folder is populated with directories including:

```
ask_sdk
ask_sdk_core
ask_sdk_dynamodb
ask_sdk_model
boto3
...
```
$ skill_env\Scripts\activate

The command prompt should now be prefixed with (skill_env), indicating that you are working inside the virtual environment. Use the following command to install the ASK Python SDK:

```
(skill_env)$ pip install ask-sdk
```

The SDK will be installed into the `skill\Lib\site-packages` folder. The site-packages folder is populated with directories including:

```
ask_sdk
ask_sdk_core
ask_sdk_dynamodb
ask_sdk_model
boto3
...
```

**Option 2: Set up the SDK in a specific folder**

To get started, from a command prompt create a new folder for your Alexa skill and navigate to the folder:

```
$ mkdir skill
$ cd skill
```

Next, install the ASK SDK for Python using pip. The `--target` option targets a specific folder for installation:

```
$ pip install ask-sdk --target skill_env
```

This creates a folder named `skill_env` inside your `skill` folder and installs the ASK SDK for Python and its dependencies. Your `skill` directory should now contain the folder `skill_env`, which is populated with directories including:

```
ask_sdk
ask_sdk_core
ask_sdk_dynamodb
ask_sdk_model
boto3
...
```

**Note:** If using Mac OS X and you have Python installed using Homebrew, the preceding command will not work. A simple workaround is to add a `setup.cfg` file in your `ask-sdk` directory with the following content:

```
[install]
prefix=
```

Navigate to the `skill_env` folder and run the `pip install` command:

```
$ cd skill_env
$ pip install ask-sdk --target .
```

More on this can be checked on the [homebrew docs](https://docs.brew.sh/Homebrew-Cheatsheet.html)
1.1.4 Next Steps

Now that you’ve added the SDK to your project, you’re ready to begin developing your skill. Proceed to the next section Developing Your First Skill, for instructions on getting started with a basic skill.

1.2 Developing Your First Skill

The Getting Started guide showed how to set up and install the ASK SDK for Python into a specific directory or into a virtual environment using virtualenv. This guide walks you through developing your first skill with the ASK SDK for Python.

1.2.1 Prerequisites

In addition to an installed version of the ASK SDK for Python you need:

- An Amazon Developer account. This is required to create and configure Alexa skills.
- An Amazon Web Services (AWS) account. This is required for hosting a skill on AWS Lambda.

1.2.2 Creating Hello World

You’ll write your Hello World in a single python file named hello_world.py. In the skill folder that you have created earlier, use your favorite text editor or IDE to create a file named hello_world.py.

1.2.3 Implementing Hello World

Request handlers

A custom skill needs to respond to events sent by the Alexa service. For instance, when you ask your Alexa device (e.g. Echo, Echo Dot, Echo Show, etc.) to ‘open hello world’, your skill needs to respond to the LaunchRequest that is sent to your Hello World skill. With the ASK SDK for Python, you simply need to write a request handler, which is code to handle incoming requests and return a response. Your code is responsible for making sure that the right request handler is used to process incoming requests and for providing a response. The ASK SDK for Python provides two ways to create request handlers:

1. Implement the AbstractRequestHandler class under ask_sdk_core.dispatch_components package. The class should contain implementations for can_handle and handle methods. This is described under Implementation using classes section.

2. Use the request_handler decorator in instantiated skill builder object to tag functions that act as handlers for different incoming requests. This is described under Implementation using decorators section.

The implementation of the Hello World skill explores using handler classes first and then shows how to write the same skill using decorators. The functionality of these is identical and you can use either.

Exception handlers

Sometimes things go wrong, and your skill code needs a way to handle the problem gracefully. The ASK SDK for Python supports exception handling in a similar way to handling requests. You have a choice of using classes or decorators. The following implementation sections explore how to implement exception handling.
Tip: You may use either Implementation using classes or Implementation using decorators options to write a skill.

Warning: We strongly recommend you to choose one of the options and use it consistently throughout your skill, for better code structure.

1.2.4 Option 1: Implementation using handler classes

Start by creating a skill builder object. The skill builder object helps in adding components responsible for handling input requests and generating custom responses for your skill.

Type or paste the following code into your `hello_world.py` file.

```python
from ask_sdk_core.skill_builder import SkillBuilder
sb = SkillBuilder()
```

To use handler classes, each request handler is written as a class that implements two methods of the AbstractRequestHandler class; can_handle and handle.

The can_handle method returns a Boolean value indicating if the request handler can create an appropriate response for the request. The can_handle method has access to the request type and additional attributes that the skill may have set in previous requests or even saved from a previous interaction. The Hello World skill only needs to reference the request information to decide if each handler can respond to an incoming request.

LaunchRequest handler

The following code example shows how to configure a handler to be invoked when the skill receives a LaunchRequest. The LaunchRequest event occurs when the skill is invoked without a specific intent.

Type or paste the following code into your `hello_world.py` file, after the previous code.

```python
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.utils import is_request_type, is_intent_name
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_model import Response
from ask_sdk_model.ui import SimpleCard

class LaunchRequestHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_request_type("LaunchRequest")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speech_text = "Welcome to the Alexa Skills Kit, you can say hello!"
        handler_input.response_builder.speak(speech_text).set_card(SimpleCard("Hello World", speech_text)).set_should_end_session(False)
        return handler_input.response_builder.response
```

The can_handle function returns True if the incoming request is a LaunchRequest. The handle function generates and returns a basic greeting response.

1.2. Developing Your First Skill
HelloWorldIntent handler

The following code example shows how to configure a handler to be invoked when the skill receives an intent request with the name HelloWorldIntent. Type or paste the following code into your hello_world.py file, after the previous handler.

```python
class HelloWorldIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("HelloWorldIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speech_text = "Hello World"
        handler_input.response_builder.speak(speech_text).set_card(
            SimpleCard("Hello World", speech_text)).set_should_end_session(
                True)
        return handler_input.response_builder.response
```

The can_handle function detects if the incoming request is an IntentRequest, and returns True if the intent name is HelloWorldIntent. The handle function generates and returns a basic “Hello World” response.

HelpIntent handler

The following code example shows how to configure a handler to be invoked when the skill receives the built-in intent AMAZON.HelpIntent. Type or paste the following code into your hello_world.py file, after the previous handler.

```python
class HelpIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("AMAZON.HelpIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speech_text = "You can say hello to me!"
        handler_input.response_builder.speak(speech_text).ask(speech_text).set_card(
            SimpleCard("Hello World", speech_text)).set_should_end_session(
                True)
        return handler_input.response_builder.response
```

Similar to the previous handler, this handler matches an IntentRequest with the expected intent name. Basic help instructions are returned, and .ask(speech_text) causes the user’s microphone to open up for the user to respond.

CancelAndStopIntent handler

The CancelAndStopIntentHandler is similar to the HelpIntent handler, as it is also triggered by the built-in AMAZON.CancelIntent or AMAZON.StopIntent Intents. The following example uses a single handler to respond to both intents. Type or paste the following code into your hello_world.py file, after the previous handler.

```python
class CancelAndStopIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("AMAZON.CancelIntent")
```
The response to both intents is the same, so having a single handler reduces repetitive code.

### SessionEndedRequest handler

Although you cannot return a response with any speech, card or directives after receiving a `SessionEndedRequest`, the `SessionEndedRequestHandler` is a good place to put your cleanup logic. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```python
class SessionEndedRequestHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_request_type("SessionEndedRequest")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        # any cleanup logic goes here
        return handler_input.response_builder.response
```

### Implementing exception handlers

The following sample adds a *catch all* exception handler to your skill, to ensure the skill returns a meaningful message for all exceptions. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```python
from ask_sdk_core.dispatch_components import AbstractExceptionHandler

class AllExceptionHandler(AbstractExceptionHandler):
    def can_handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> bool
        return True

    def handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> Response
        # Log the exception in CloudWatch Logs
        print(exception)

        speech = "Sorry, I didn't get it. Can you please say it again?!"
        handler_input.response_builder.speak(speech).ask(speech)
        return handler_input.response_builder.response
```

### Creating the Lambda handler

1.2. Developing Your First Skill
Note: For a custom skill, you can host your service in AWS Lambda or as a web service hosted on your own endpoint. Generally, hosting the skill code on AWS Lambda is the easiest way. The below sections provide information on how to achieve this.

However, if you wish to host it with any other cloud hosting provider, the SDK provides some support packages (ask-sdk-webservice-support, flask-ask-sdk, django-ask-sdk). You can find more information on this configuration here.

The Lambda handler is the entry point for your AWS Lambda function. The following code example creates a Lambda handler function to route all inbound requests to your skill. The Lambda handler function creates an SDK skill instance configured with the request handlers that you just created. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```python
sb.add_request_handler(LaunchRequestHandler())
sb.add_request_handler(HelloWorldIntentHandler())
sb.add_request_handler(HelpIntentHandler())
sb.add_request_handler(CancelAndStopIntentHandler())
sb.add_request_handler(SessionEndedRequestHandler())
sb.add_exception_handler(AllExceptionHandler())
handler = sb.lambda_handler()
```

### 1.2.5 Option 2: Implementation using decorators

The following code implements the same functionality as above but uses function decorators. You can think of the decorators as a replacement for the `can_handle` method implemented in each handler class above.

Start by creating a skill builder object. The skill builder object helps in adding components responsible for handling input requests and generating custom responses for your skill.

Type or paste the following code into your `hello_world.py` file.

```python
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()
```

#### LaunchRequest handler

The following code example shows how to configure a handler to be invoked when the skill receives a `LaunchRequest`. The LaunchRequest event occurs when the skill is invoked without a specific intent.

Type or paste the following code into your `hello_world.py` file, after the previous code.

```python
from ask_sdk_core.utils import is_request_type, is_intent_name
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_model import Response
from ask_sdk_model.ui import SimpleCard

@sb.request_handler(can_handle_func=is_request_type("LaunchRequest"))
def launch_request_handler(handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "Welcome to the Alexa Skills Kit, you can say hello!"
```

(continues on next page)
Similar to the `can_handle` function for the `LaunchRequestHandler` in the Class pattern, the decorator returns `True` if the incoming request is a `LaunchRequest`. The `handle` function generates and returns a basic greeting response in the same way the `handle` function works for the Class pattern.

**HelloWorldIntent handler**

The following code example shows how to configure a handler to be invoked when the skill receives an intent request with the name `HelloWorldIntent`. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```python
@sb.request_handler(can_handle_func=is_intent_name("HelloWorldIntent"))
def hello_world_intent_handler(handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "Hello World!"

    handler_input.response_builder.speak(speech_text).set_card(
        SimpleCard("Hello World", speech_text)).set_should_end_session(
            True)
    return handler_input.response_builder.response
```

**HelpIntent handler**

The following code example shows how to configure a handler to be invoked when the skill receives the built-in intent `AMAZON.HelpIntent`. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```python
@sb.request_handler(can_handle_func=is_intent_name("AMAZON.HelpIntent"))
def help_intent_handler(handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "You can say hello to me!"

    handler_input.response_builder.speak(speech_text).ask(speech_text).set_card(
        SimpleCard("Hello World", speech_text))
    return handler_input.response_builder.response
```

Similar to the previous handler, this handler matches an `IntentRequest` with the expected intent name. Basic help instructions are returned, and `.ask(speech_text)` causes the user’s microphone to open up for the user to respond.

**CancelAndStopIntent handler**

The `CancelAndStopIntentHandler` is similar to the `HelpIntent` handler, as it is also triggered by the built-in `AMAZON.CancelIntent` or `AMAZON.StopIntent` intents. The following example uses a single handler to respond to both Intents. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```python
@sb.request_handler(can_handle_func=is_intent_name("AMAZON.CancelIntent"))
def cancel_intent_handler(handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "Goodbye!

    handler_input.response_builder.speak(speech_text).ask(speech_text)
    return handler_input.response_builder.response
```
In the above example, `can_handle` needs a function to be passed. `is_intent_name` returns a function, but we need to check if the request is either `AMAZON.CancelIntent` or `AMAZON.StopIntent`. We achieve this by creating an anonymous function on the fly using Python’s built-in `lambda` function.

The response to both intents is the same, so having a single handler reduces repetitive code.

### `SessionEndedRequest` handler

Although you cannot return a response with any speech, card or directives after receiving a `SessionEndedRequest`, the `SessionEndedRequestHandler` is a good place to put your cleanup logic. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```python
@sb.request_handler(can_handle_func=is_request_type("SessionEndedRequest"))
def session_ended_request_handler(handler_input):
    # type: (HandlerInput) -> Response
    # any cleanup logic goes here
    return handler_input.response_builder.response
```

### Implementing exception handlers

The following sample adds a catch all exception handler to your skill, to ensure the skill returns a meaningful message in case of all exceptions. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```python
@sb.exception_handler(can_handle_func=lambda i, e: True)
def all_exception_handler(handler_input, exception):
    # type: (HandlerInput, Exception) -> Response
    # Log the exception in CloudWatch Logs
    print(exception)
    speech = "Sorry, I didn't get it. Can you please say it again!!"
    handler_input.response_builder.speak(speech).ask(speech)
    return handler_input.response_builder.response
```

### Creating the Lambda handler

**Note:** For a custom skill, you can host your service in AWS Lambda or as a web service hosted on your own endpoint.

Generally, hosting the skill code on AWS Lambda is the easiest way. The below sections provide information on how to achieve this.
However, if you wish to host it with any other cloud hosting provider, the SDK provides some support packages (ask-sdk-webservice-support, flask-ask-sdk, django-ask-sdk). You can find more information on this configuration here.

The Lambda handler is the entry point for your AWS Lambda function. The following code example creates a Lambda handler function to route all inbound requests to your skill. The Lambda Handler function creates an SDK skill instance configured with the request handlers that you just created.

Type or paste the following code into your hello_world.py file, after the previous handler.

```python
handler = sb.lambda_handler()
```

When using decorators, your request handlers and exception handlers are automatically recognized by the Skill Builder object instantiated at the top of the code.

### 1.2.6 Full source code

The full source code for hello_world.py can be found here.

### 1.2.7 Preparing your code for AWS Lambda

Your code is now complete and you need to create `.zip` files that contain the files ready to upload to Lambda.

When you upload your code to AWS Lambda, you must include your skill code and its dependencies inside a zip file as a flat file structure, so you’ll place your code in the same folder as the ASK SDK for Python, before zipping it.

SDK Setup in Virtual Environment

If you set up the SDK using a virtual environment, the dependencies are installed in the `site-packages` folder in your virtual environment. So, navigate to the `site-packages` folder in `skill_env`.

**Note:** On Windows the `site-packages` folder is located inside the `skill_env\Lib` folder.

**Note:** For MacOS/Linux the `site-packages` folder location depends on the version of Python you are using. For instance Python 3.6 users will find `site-packages` inside the `skill_env/lib/Python3.6` folder.

Copy the `hello_world.py` file into the `site-packages` folder and create a `.zip` file of the contents of the folder (not the folder itself). Name the file `skill.zip`. You can check the AWS Lambda docs to get more information on creating a deployment package.

SDK Setup in specific folder

If you set up the SDK in a specific folder, the dependencies are installed in that specific folder. That would be `skill_env` folder if you followed the steps mentioned in the getting started guide.

Copy the `hello_world.py` file into the `skill_env` folder and create a `.zip` file of the contents of the folder (not the folder itself). Name the file `skill.zip`. You can check the AWS Lambda docs to get more information on creating a deployment package.

Before uploading the code to AWS Lambda, you need to create an AWS Lambda function and create the skill on the Alexa Developer Portal.
1.2.8 Creating an AWS Lambda function

Refer to Hosting a Custom Skill as an AWS Lambda Function for a walkthrough on creating an AWS Lambda function with the correct role for your skill. When creating the function, select the Author from scratch option and select the Python 2.7 or Python 3.6 runtime.

Once you’ve created your AWS Lambda function, it’s time to give the Alexa service the ability to invoke it. To do this, navigate to the Triggers tabs in your Lambda’s configuration, and add Alexa Skills Kit as the trigger type. Once this is done, upload the skill.zip file produced in the previous step and fill in the handler information with module_name.handler which is hello_world.handler for this example.

1.2.9 Configuring and testing your skill

Now that the skill code has been uploaded to AWS Lambda, you can configure the skill with Alexa.

- Create a new skill by following these steps:
  1. Log in to the Alexa Skills Kit Developer Console.
  2. Click the Create Skill button in the upper right.
  3. Enter “HelloWorld” as your skill name and click Next.
  4. For the model, select Custom and click Create skill.

- Next, define the interaction model for the skill. Select the Invocation option from the sidebar and enter “greeter” for the Skill Invocation Name.

- Next, add an intent called HelloWorldIntent to the interaction model. Click the Add button under the Intents section of the Interaction Model. Leave “Create custom intent” selected, enter “HelloWorldIntent” for the intent name, and create the intent. On the intent detail page, add some sample utterances that users can say to invoke the intent. For this example, consider the following sample utterances, and feel free to add others.

  - say hello
  - say hello world
  - hello
  - say hi
  - say hi world
  - hi
  - how are you

- Since AMAZON.CancelIntent, AMAZON.HelpIntent, and AMAZON.StopIntent are built-in Alexa intents, you do not need to provide sample utterances for them.

- The Developer Console allows you to edit the entire skill model in JSON format. Select JSON Editor from the sidebar. For this sample, you can use the following JSON schema.

```json
{
  "interactionModel": {
    "languageModel": {
      "invocationName": "greeter",
      "intents": [
        {
          "name": "AMAZON.CancelIntent",
          "samples": []
        },
        {
          "name": "AMAZON.HelpIntent",
          "samples": []
        }
      ]
    }
  }
}
```

(continues on next page)
• Once you are done editing the interaction model, be sure to save and build the model.

• Next, configure the endpoint for the skill. To do this, follow these steps:

1. Under your skill, click the **Endpoint** tab, select AWS Lambda ARN, and copy the **Skill ID** of the skill you just created.

2. Open the AWS Developer Console in a new tab.

3. Navigate to the AWS Lambda function created in the previous step.

4. From the **Designer** menu, add the **Alexa Skills Kit** trigger menu, and scroll down to paste the skill ID into the **Skill ID Verification** configuration. Click **Add and save** once completed to update the AWS Lambda function.

5. Copy the AWS Lambda function **ARN** from the top right corner of the page. An ARN is a unique resource number that helps Alexa service identify the AWS Lambda function it needs to call during skill invocation.

6. Navigate to the Alexa Skills Kit Developer Console, and click on your **HelloWorld** skill.

7. Under your skill, click **Endpoint** tab, select **AWS Lambda ARN** and paste in the ARN under **Default Region** field.

8. The rest of the settings can be left at their default values. Click **Save Endpoints**.

9. Click **Invocation** tab, save and build the model.

• At this point you can test the skill. In the top navigation, click **Test**. Make sure that the **Test is enabled for this skill** option is enabled. You can use the Test page to simulate requests, in text and voice form.

• Use the invocation name along with one of the sample utterances as a guide. For example, *tell greeter to say hello* should result in your skill responding with “Hello World” voice and “Hello World” card on devices with display. You can also open the Alexa app on your phone or at [https://alexa.amazon.com](https://alexa.amazon.com) and see your skill listed under **Your Skills**.

• Feel free to start experimenting with your intents as well as the corresponding request handlers in your skill’s code. Once you’re finished iterating, optionally move on to getting your skill certified and published so it can
be used by customers worldwide.

1.3 Sample Skills

This section provides sample skills that demonstrate the usage of ASK SDK for Python to build engaging Alexa Skills.

1.3.1 Hello World (using Classes)

This code sample will allow you to hear a response from Alexa when you trigger it. It is a minimal sample to get you familiarized with the Alexa Skills Kit and AWS Lambda. This sample shows how to create a skill using the Request Handler classes. For more information, check the Request Processing documentation.

1.3.2 Hello World (using Decorators)

This code sample will allow you to hear a response from Alexa when you trigger it. It is a minimal sample to get you familiarized with the Alexa Skills Kit and AWS Lambda. This sample shows how to create a skill using the Request Handler Decorators. For more information, check the Request Processing documentation.

1.3.3 Color Picker

This is a step-up in functionality from Hello World. When the user provides their favorite color, Alexa remembers it and tells the user their favorite color. It allows you to capture input from your user and demonstrates the use of Slots. It also demonstrates use of session attributes and request, response interceptors.

1.3.4 Fact

Template for a basic fact skill. You’ll provide a list of interesting facts about a topic, Alexa will select a fact at random and tell it to the user when the skill is invoked. Demonstrates use of multiple locales and internationalization in the skill.

1.3.5 Quiz Game

Template for a basic quiz game skill. Alexa quizzes the user with facts from a list you provide. Demonstrates use of render template directives to support displays on Alexa-enabled devices with a screen.

1.3.6 Device Address

Sample skill that shows how to request and access the configured address in the user’s device settings. Demonstrates how to use the alexa APIs using the SDK. For more information, check the documentation on Alexa Service Clients.

1.3.7 Fact with In-Skill Purchases

Sample fact skill with in-skill purchase features, by offering different packs of facts behind a purchase, and a subscription to unlock all of the packs at once. Demonstrates calling monetization alexa service and using ASK CLI to enable in-skill purchasing.
1.3.8 City Guide

Template for a local recommendations skill. Alexa uses the data that you provide to offer recommendations according to the user’s stated preferences. Demonstrates calling external APIs from the skill.

1.3.9 Pet Match

Sample skill that matches the user with a pet. Alexa prompts the user for the information it needs to determine a match. Once all of the required information is collected, the skill sends the data to an external web service that processes the data and returns the match. Demonstrates how to prompt and parse multiple values from customers using Dialog Management and Entity Resolution.

1.3.10 High Low Game

Template for a basic high-low game skill. When the user guesses a number, Alexa tells the user whether the number she has in mind is higher or lower. Demonstrates use of persistence attributes and the persistence adapter in the SDK.

1.3.11 AudioPlayer SingleStream and MultiStream

Sample skills that show how to use AudioPlayer interface and PlaybackController interface in Alexa, to build audio-player skills. The SingleStream skill sample demonstrates how to create a live radio skill, along with localization support. The MultiStream skill sample demonstrates how to create a basic podcast skill that can play multiple, pre-recorded audio streams.

1.3.12 Pager Karaoke

This sample demonstrates 3 features of APL: the Pager Component, SpeakItem Command, and accessing device characteristics in the skill code.

1.4 Alexa Capabilities supported by SDK

This section provides all the Alexa Capabilities that are currently supported in the SDK.

1.4.1 Stable

- Amazon Pay
- Audio Player
- Display – Body templates for devices with a screen
- GadgetsGame Engine – Echo Buttons
- Directive Service (Progressive Response)
- Messaging
- Monetization
- Video
- Device Address
• Lists
• Request for customer contact information
• Obtain customer settings information
• Account Linking
• Entity Resolution
• Dialog Management
• Location Services
• Reminders
• Proactive Events
• Dynamic Entities
• Skill Messaging
• Connections

1.4.2 Preview

Note: The following capabilities are released as Preview. The interfaces might change during a stable release.

• Alexa Presentation Language
• Name-free Interactions

1.5 Setting Up The ASK SDK

This guide will show you how to include the SDK as a dependency in your Python project.

1.6 Developing Your First Skill

Walks you through step-by-step instructions for building the Hello World sample.

1.7 Sample Skills

Provides a comprehensive list of skill samples using the SDK.

1.8 SDK Supported Alexa Capabilities

Provides a list of alexa capabilities, that are supported in the ASK SDK.
2.1 Request Processing

2.1.1 Standard Request

Alexa communicates with the skill service via a request-response mechanism using HTTP over SSL/TLS. When a user interacts with an Alexa skill, your service receives a POST request containing a JSON body. The request body contains the parameters necessary for the service to perform its logic and generate a JSON-formatted response. The documentation on JSON structure of the request body can be found here.

Though Python can handle JSON natively as dict objects, for providing type support, they are deserialized into model objects (ask-sdk-model package) for skill consumption.

2.1.2 Handler Input

Request Handlers, Request and Response Interceptors, and Exception Handlers are all passed a global HandlerInput object during invocation. This object exposes various entities useful in request processing, including:

- **request_envelope**: Contains the entire request body sent to skill, session information and some context information about the input request. Interface details: ask_sdk_model.request_envelope.RequestEnvelope
- **attributes_manager**: Provides access to request, session, and persistent attributes. Interface details: ask_sdk_core.attributes_manager.AttributesManager
- **service_client_factory**: Constructs service clients capable of calling Alexa APIs. Interface details: ask_sdk_model.services.service_client_factory.ServiceClientFactory
- **response_builder**: Contains helper function to build responses. Interface details: ask_sdk_core.response_helper.ResponseFactory
- **context**: Provides an optional, context object passed in by the host container. For example, for skills running on AWS Lambda, this is the context object for the AWS Lambda function.
2.1.3 Request Handlers

Request handlers are responsible for handling one or more types of incoming Alexa requests. There are two ways of creating custom request handlers:

- By implementing the `AbstractRequestHandler` class.
- By decorating a custom handle function using the `SkillBuilder request_handler` decorator.

**Warning:** You may use either implementation using **classes** or **decorators** to write a skill.

We strongly recommend you to choose **one** of the options and use it consistently throughout your skill, for better code structure.

**Interface**

**AbstractRequestHandler Class**

If you plan on using the `AbstractRequestHandler` class, you will need to implement the following methods:

- **can_handle**: `can_handle` method is called by the SDK to determine if the given handler is capable of processing the incoming request. This function accepts a `Handler Input` object and expects a boolean to be returned. If the method returns `True`, then the handler is supposed to handle the request successfully. If it returns `False`, the handler is not supposed to handle the input request and hence not executed to completion. Because of the various attributes in `HandlerInput` object, you can write any condition to let SDK know whether the request can be handled gracefully or not.

- **handle**: `handle` method is called by the SDK when invoking the request handler. This function contains the handler's request processing logic, accepts `Handler Input` and returns a `Response` object.

```python
class AbstractRequestHandler(object):
    @abstractmethod
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        pass

    @abstractmethod
    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        pass
```

**RequestHandler Decorator**

The `request_handler` decorator from `SkillBuilder` class is a custom wrapper on top of the `AbstractRequestHandler` class and provides the same functionality to any custom decorated function. However, there are couple of things to take into consideration, before using the decorator:

- The decorator expects a `can_handle_func` parameter. This is similar to the `can_handle` method in `AbstractRequestHandler`. The value passed should be a function that accepts a `Handler Input` object and returns a boolean value.

- The decorated function should accept only one parameter, which is the `Handler Input` object and may return a `Response` object.

```python
class SkillBuilder(object):
    ....
    def request_handler(self, can_handle_func):
        ....
```
def wrapper(handle_func):
    # wrap the can_handle and handle into a class
    # add the class into request handlers list
    ....
    return wrapper

Code Sample

The following example shows a request handler class that can handle the HelloWorldIntent.

AbstractRequestHandler Class

def can_handle(self, handler_input):
    return is_intent_name("HelloWorldIntent")(handler_input)

def handle(self, handler_input):
    speech_text = "Hello World"

    return handler_input.response_builder.speak(speech_text).set_card(
        SimpleCard("Hello World", speech_text)).response

The can_handle function detects if the incoming request is an IntentRequest and returns true if the intent name is HelloWorldIntent. The handle function generates and returns a basic “Hello World” response.

RequestHandler Decorator

The is_intent_name function accepts a string parameter and returns an anonymous function which accepts a HandlerInput as input parameter and checks if the incoming request in HandlerInput is an IntentRequest and returns if the intent name is the passed in string, which is HelloWorldIntent in this example. The handle function generates and returns a basic “Hello World” response.

Registering and Processing the Request Handlers

The SDK calls the can_handle function on its request handlers in the order in which they were provided to the Skill builder.

2.1. Request Processing
AbstractRequestHandler Class

If you are following the AbstractRequestHandler class approach, then you can register the request handlers in the following way

```python
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# Implement FooHandler, BarHandler, BazHandler classes
sb.add_request_handler(FooHandler())
sb.add_request_handler(BarHandler())
sb.add_request_handler(BazHandler())
```

RequestHandler Decorator

If you are following the request_handler decorator approach, then there is no need to explicitly register the handler functions, since they are already decorated using a skill builder instance.

```python
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# decorate foo_handler, bar_handler, baz_handler functions
```

**Note:** In the above example, the SDK calls request handlers in the following order:

1. FooHandler class / foo_handler function
2. BarHandler class / bar_handler function
3. BazHandler class / baz_handler function

The SDK always chooses the first handler that is capable of handling a given request. In this example, if both FooHandler class / foo_handler function and BarHandler class / bar_handler function are capable of handling a particular request, FooHandler class / foo_handler function is always invoked. Keep this in mind when designing and registering request handlers.

### 2.1.4 Exception Handlers

Exception handlers are similar to request handlers, but are instead responsible for handling one or more types of exceptions. They are invoked by the SDK when an unhandled exception is thrown during the course of request processing.

In addition to the Handler Input object, the handler also has access to the exception raised during handling the input request, thus making it easier for the handler to figure out how to handle the corresponding exception.

Similar to Request Handlers, custom request interceptors can be implemented in two ways:

- By implementing the AbstractExceptionHandler class.
- By decorating a custom exception handling function using the Skill Builder exception_handler decorator.
Warning: You may use either implementation using classes or decorators to write a skill. We strongly recommend you to choose one of the options and use it consistently throughout your skill, for better code structure.

Interface

AbstractExceptionHandler Class

If you plan on using the AbstractExceptionHandler class, you will need to implement the following methods:

- **can_handle**: can_handle method, which is called by the SDK to determine if the given handler is capable of handling the exception. This function returns True if the handler can handle the exception, or False if not. Return True in all cases to create a catch-all handler.

- **handle**: handle method, which is called by the SDK when invoking the exception handler. This function contains all exception handling logic, and returns a Response object.

```python
class AbstractExceptionHandler(object):
    @abstractmethod
def can_handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> bool
        pass

    @abstractmethod
def handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> Response
        pass
```

ExceptionHandler Decorator

The exception_handler decorator from SkillBuilder class is a custom wrapper on top of the AbstractExceptionHandler class and provides the same functionality to any custom decorated function. However, there are couple of things to take into consideration, before using the decorator:

- The decorator expects a can_handle_func parameter. This is similar to the can_handle method in AbstractExceptionHandler. The value passed should be a function that accepts a Handler Input object, an Exception instance and returns a boolean value.

- The decorated function should accept only two parameters, the Handler Input object and Exception object. It may return a Response object.

```python
class SkillBuilder(object):
    ....
def exception_handler(self, can_handle_func):
    def wrapper(handle_func):
        # wrap the can_handle and handle into a class
        # add the class into exception handlers list
        ....
    return wrapper
```

Code Sample

The following example shows an exception handler that can handle any exception with name that contains “AskSdk”. AbstractExceptionHandler Class

2.1. Request Processing
class AskExceptionHandler(AbstractExceptionHandler):
    def can_handle(self, handler_input, exception):
        return 'AskSdk' in exception.__class__.__name__
    def handle(self, handler_input, exception):
        speech_text = "Sorry, I am unable to figure out what to do. Try again later!!"
        return handler_input.response_builder.speak(speech_text).response

The handler’s can_handle method returns True if the incoming exception has a name that starts with “AskSdk”. The handle method returns a graceful exception response to the user.

ExceptionHandler Decorator

from ask_sdk_core.skill_builder import SkillBuilder
sb = SkillBuilder()
@sb.exception_handler(can_handle_func = lambda i, e: 'AskSdk' in e.__class__.__name__)
def ask_exception_intent_handler(handler_input, exception):
    speech_text = "Sorry, I am unable to figure out what to do. Try again later!!"
    return handler_input.response_builder.speak(speech_text).response

Registering and Processing the Exception Handlers

AbstractExceptionHandler Class

from ask_sdk_core.skill_builder import SkillBuilder
sb = SkillBuilder()
# Implement FooExceptionHandler, BarExceptionHandler, BazExceptionHandler classes
sb.add_exception_handler(FooExceptionHandler())
sb.add_exception_handler(BarExceptionHandler())
sb.add_exception_handler(BazExceptionHandler())

ExceptionHandler Decorator

from ask_sdk_core.skill_builder import SkillBuilder
sb = SkillBuilder()
# decorate foo_exception_handler, bar_exception_handler, baz_exception_handler
functions

Note: Like request handlers, exception handlers are executed in the order in which they were registered to the Skill.

2.1.5 Request and Response Interceptors

The SDK supports Global Request and Response Interceptors that execute before and after matching RequestHandler execution, respectively.
Request Interceptors

The Global Request Interceptor accepts a Handler Input object and processes it, before processing any of the registered request handlers. Similar to Request Handlers, custom request interceptors can be implemented in two ways:

- By implementing the AbstractRequestInterceptor class.
- By decorating a custom process function using the Skill Builder global_request_interceptor decorator.

**Warning:** You may use either implementation using classes or decorators to write a skill.

We strongly recommend you to choose one of the options and use it consistently throughout your skill, for better code structure.

Interface

AbstractRequestInterceptor Class

The AbstractRequestInterceptor class usage needs you to implement the process method. This method takes a Handler Input instance and doesn’t return anything.

```python
class AbstractRequestInterceptor(object):
    @abstractmethod
    def process(self, handler_input):
        # type: (HandlerInput) -> None
        pass
```

GlobalRequestInterceptor Decorator

The global_request_interceptor decorator from SkillBuilder class is a custom wrapper on top of the AbstractRequestInterceptor class and provides the same functionality to any custom decorated function. However, there are couple of things to take into consideration, before using the decorator:

- The decorator should be invoked as a function rather than as a function name, since it requires the skill builder instance, to register the interceptor.
- The decorated function should accept only one parameter, which is the Handler Input object and the return value from the function is not captured.

```python
class SkillBuilder(object):
    ....
    def global_request_interceptor(self):
        def wrapper(process_func):
            # wrap the process_func into a class
            # add the class into request interceptors list
            ....
            return wrapper
```

Code Sample

The following example shows a request interceptor class that can print the request received by Alexa service, in AWS CloudWatch logs, before handling it.

AbstractRequestInterceptor Class
from ask_sdk_core.dispatch_components import AbstractRequestInterceptor

class LoggingRequestInterceptor(AbstractRequestInterceptor):
    def process(self, handler_input):
        print("Request received: {}\n\n".format(handler_input.request_envelope.request))

GlobalRequestInterceptor Decorator

from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

@sb.global_request_interceptor()
def request_logger(handler_input):
    print("Request received: {}\n\n".format(handler_input.request_envelope.request))

Registering and Processing the Request Interceptors

Request interceptors are invoked immediately before execution of the request handler for an incoming request. Request attributes in Handler Input’s Attribute Manager provide a way for request interceptors to pass data and entities on to other request interceptors and request handlers.

AbstractRequestInterceptor Class

from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# Implement FooInterceptor, BarInterceptor, BazInterceptor classes

sb.add_global_request_interceptor(FooInterceptor())
sb.add_global_request_interceptor(BarInterceptor())
sb.add_global_request_interceptor(BazInterceptor())

GlobalRequestInterceptor Decorator

from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# decorate foo_interceptor, bar_interceptor, baz_interceptor functions

Note: In the above example, the SDK executes all request interceptors in the following order:

1. FooInterceptor class / foo_interceptor function
2. BarInterceptor class / bar_interceptor function
3. BazInterceptor class / baz_interceptor function

Response Interceptors

The Global Response Interceptor accepts a Handler Input object, a Response and processes them, after executing the supported request handler. Similar to Request Interceptors, custom response interceptors can be implemented in two ways:
• By implementing the AbstractResponseInterceptor class.

• By decorating a custom process function using the Skill Builder global_response_interceptor decorator.

**Warning:** You may use either implementation using classes or decorators to write a skill.

We strongly recommend you to choose one of the options and use it consistently throughout your skill, for better code structure.

### Interface

**AbstractResponseInterceptor Class**

The AbstractResponseInterceptor class usage needs you to implement the `process` method. This method takes a `Handler Input` instance, a Response object that is returned from the previously executed request handler. The method doesn’t return anything.

```python
class AbstractResponseInterceptor(object):
    @abstractmethod
    def process(self, handler_input, response):
        # type: (HandlerInput, Response) -> None
        pass
```

**GlobalResponseInterceptor Decorator**

The global_response_interceptor decorator from SkillBuilder class is a custom wrapper on top of the AbstractResponseInterceptor class and provides the same functionality to any custom decorated function. However, there are couple of things to take into consideration, before using the decorator:

- The decorator should be invoked as a function rather than as a function name, since it requires the skill builder instance, to register the interceptor.
- The decorated function should accept two parameters, which are the `Handler Input` object and `Response` object respectively. The return value from the function is not captured.

```python
class SkillBuilder(object):
    ....
    def global_response_interceptor(self):
        def wrapper(process_func):
            # wrap the process_func into a class
            # add the class into response interceptors list
            ....
            return wrapper
```

### Code Sample

The following example shows a response interceptor class that can print the response received from successfully handling the request, in AWS CloudWatch logs, before returning it to the Alexa Service.

AbstractRequestInterceptor Class

```python
from ask_sdk_core.dispatch_components import AbstractResponseInterceptor

class LoggingResponseInterceptor(AbstractResponseInterceptor):
    (continues on next page)
def process(handler_input, response):
    print("Response generated: {}".format(response))

GlobalRequestInterceptor Decorator

def response_logger(handler_input, response):
    print("Response generated: {}".format(response))

Registering and Processing the Response Interceptors

Response interceptors are invoked immediately after execution of the request handler for an incoming request.

AbstractRequestInterceptor Class

GlobalRequestInterceptor Decorator

Note: Similar to the processing of Request Interceptors, all of the response interceptors are executed in the same order they are registered.

2.2 Response Building

2.2.1 Standard Response

If you are using the AWS lambda as your skill endpoint, you are only responsible for providing the response body in order for Alexa to respond to a customer request. The documentation on the JSON structure of the response body can be found here.

A response body may contain the following properties:

- version
• sessionAttributes
• response

ASK SDK for Python helps filling the version and sessionAttributes so you can focus on building the response instead of writing boilerplate code.

Similar to standard requests, the SDK creates response objects as deserialized model objects (ask-sdk-model package) and internally handles serializing them to response JSON before sending to Alexa Service.

### 2.2.2 Response Factory

The SDK includes a ResponseFactory class, that contains helper functions for constructing responses. A Response may contain multiple elements, and the helper functions aid in generating responses, reducing the need to initialize and set the elements of each response.

#### Interface

```python
class ResponseFactory(object):
    def __init__(self):
        self.response = ....  # Response object

    def speak(self, speech, play_behavior=None):
        # type: (str, ask_sdk_model.ui.play_behavior.PlayBehavior) -> 'ResponseFactory'
        ....

    def ask(self, speech, play_behavior=None):
        # type: (str, ask_sdk_model.ui.play_behavior.PlayBehavior) -> 'ResponseFactory'
        ....

    def set_card(self, card):
        # type: (ask_sdk_model.ui.card.Card) -> 'ResponseFactory'
        ....

    def add_directive(self, directive):
        # type: (ask_sdk_model.directive.Directive) -> 'ResponseFactory'
        ....

    def set_should_end_session(self, end_session):
        # type: (bool) -> 'ResponseFactory'
        ....

    def set_can_fulfill_intent(self, can_fulfill_intent):
        # type: (ask_sdk_model.canfulfill.can_fulfill_intent.CanFulfillIntent) -> 'ResponseFactory'
        ....
```

response_builder, an instance of the ResponseFactory class, is provided to the skill developers through the HandlerInput object, which is the standard argument passed to the skill components.

**Tip:** The contents of the speak and reprompt values get wrapped in SSML tags. Build engaging conversations using supported SSML tags directly in your speech text.
Note:

- For using and adding different directives, look at the Directive model definition.
- For using and setting a card, look at the Card model definition.

Sample Code

The following example shows how to construct a response containing a StandardCard and a BodyTemplate2 display object through handler_input.response_builder.

```python
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_intent_name
from ask_sdk_core.response_helper import get_plain_text_content
from ask_sdk_model.response import Response
from ask_sdk_model import ui

class HelloIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("HelloIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        response_builder = handler_input.response_builder
        speech = "This is a sample response"

        response_builder.set_card(ui.StandardCard(
            title="Card Title",
            text="Hey this is a sample card",
            image=ui.Image(
                small_image_url="<Small Image URL>",
                large_image_url="<Large Image URL>"
            )
        ))

        if supports_display(handler_input):
            img = Image(
                sources=[ImageInstance(url="<Large Image URL>"}],
                title="Template Title"
            primary_text = get_plain_text_content(
                primary_text="some text"
            )

            response_builder.add_directive(
            RenderTemplateDirective(
                BodyTemplate2(
                    back_button=BackButtonBehavior.VISIBLE,
                )))

    (continues on next page)
```
Text Helpers

The following helper functions are provided to skill developers, to help with text content generation:

**get_plain_text_content**

```python
def get_plain_text_content(primary_text, secondary_text, tertiary_text):
    # type: (str, str, str) -> TextContent
    # Create a text content object with text as PlainText type
    ...
```

**get_rich_text_content**

```python
def get_rich_text_content(primary_text, secondary_text, tertiary_text):
    # type: (str, str, str) -> TextContent
    # Create a text content object with text as RichText type
    ...
```

**get_text_content**

```python
def get_text_content(
    primary_text, primary_text_type,
    secondary_text, secondary_text_type,
    tertiary_text, tertiary_text_type):
    # type: (str, str, str, str, str, str) -> TextContent
    # Create a text content object with text as corresponding passed-type
    # Passed-in type is defaulted to PlainText
    ...
```

### 2.3 Configuring Skill Instance

#### 2.3.1 Skill

The **Skill** object is the integration of all your skill logic. It is responsible for initializing SDK utilities such as the **AttributesManager** and **ServiceClientFactory** and also kick off the request handling process.

**Available Methods**

```python
def invoke(self, request_envelope, context):
    # type: (RequestEnvelope, Any) -> ResponseEnvelope
```
2.3.2 Skill Builders

The SDK includes a `SkillBuilder` that provides utility methods, to construct the `Skill` instance, setting custom user agent and creating lambda integration handler. It has the following structure:

```python
class SkillBuilder(object):
    def __init__(self):
        # Initialize empty collections for request components,
        # exception handlers, interceptors.

    def add_request_handler(self, handler):
        # type: (AbstractRequestHandler) -> None
        ....

    def add_exception_handler(self, handler):
        # type: (AbstractExceptionHandler) -> None
        ....

    def add_global_request_interceptor(self, interceptor):
        # type: (AbstractRequestInterceptor) -> None
        ....

    def add_global_response_interceptor(self, interceptor):
        # type: (AbstractResponseInterceptor) -> None
        ....

    @property
    def skill_configuration(self):
        # type: () -> SkillConfiguration
        # Build configuration object using the registered components
        ....

    def create(self):
        # type: () -> Skill
        # Create the skill using the skill configuration
        ....

    def lambda_handler(self):
        # type: () -> LambdaHandler
        # Create a lambda handler function that can be tagged to
        # AWS Lambda handler.
        # Processes the alexa request before invoking the skill,
        # processes the alexa response before providing to the service
        ....

    def request_handler(self, can_handle_func):
        # type: (Callable[[HandlerInput], bool]) -> None
        # Request Handler decorator

    def exception_handler(self, can_handle_func):
        # type: (Callable[[HandlerInput, Exception], bool]) -> None
        # Exception Handler decorator

    def global_request_interceptor(self):
        # type: () -> None
        # Global Request Interceptor decorator

    def global_response_interceptor(self):
```

(continues on next page)
There are two extensions to `SkillBuilder` class, `CustomSkillBuilder` and `StandardSkillBuilder`.

**CustomSkillBuilder Class**

`CustomSkillBuilder` is available in both `ask-sdk-core` and `ask-sdk` package. In addition to the common helper function above, `CustomSkillBuilder` also provides functions that allows you to register custom implementations of `AbstractPersistentAdapter` and `ask_sdk_model.services.ApiClient` classes.

```python
class CustomSkillBuilder(SkillBuilder):
    def __init__(self, persistence_adapter=None, api_client=None):
        # type: (AbstractPersistenceAdapter, ApiClient) -> None
        ....

    @property
def skill_configuration(self):
        # Create skill configuration from skill builder along with
        # registered persistence adapter and api client
        ....
```

**StandardSkillBuilder Class**

`StandardSkillBuilder` is available only in the `ask-sdk` package. It is a wrapper on `CustomSkillBuilder` with `persistence_adapter` as `ask_sdk_dynamo.adapter.DynamoDbPersistenceAdapter` and `api_client` as `ask_sdk_core.api_client.DefaultApiClient` to provide Persistence and Service Client features. It also provides optional parameters for configuring the Dynamo DB table options.

```python
class StandardSkillBuilder(SkillBuilder):
    def __init__(self,
                 table_name=None, auto_create_table=None,
                 partition_keygen=None, dynamodb_client=None):
        # type: (str, bool, Callable[[RequestEnvelope], str], ServiceResource) -> 
        ....

    @property
def skill_configuration(self):
        # Create skill configuration from skill builder along with
        # default api client and dynamodb persistence adapter with
        # the passed in table configuration options.
        ....
```

### 2.4 Skill Attributes

This guide provides information on different scopes of attributes available to the skill developer, and how to use them in the skill.
2.4.1 Attributes

The SDK allows you to store and retrieve attributes at different scopes. For example, attributes can be used to store data that you retrieve on subsequent requests. You can also use attributes in your handler's `can_handle` logic to add conditions during request routing.

An attribute consists of a key and a value. The key is enforced as a `str` type and the value is an unbounded `object`. For session and persistent attributes, you must ensure that value types are serializable so they can be properly stored for subsequent retrieval. This restriction does not apply to request-level attributes because they do not persist outside of the request processing lifecycle.

2.4.2 Attribute Scopes

**Request Attributes**

Request attributes only last within a single request processing lifecycle. Request attributes are initially empty when a request comes in, and are discarded once a response has been produced.

Request attributes are useful with request and response interceptors. For example, you can inject additional data and helper methods into request attributes through a request interceptor so they are retrievable by request handlers.

**Session Attributes**

Session attributes persist throughout the lifespan of the current skill session. Session attributes are available for use with any in-session request. Any attributes set during the request processing lifecycle are sent back to the Alexa service and provided in the next request in the same session.

Session attributes do not require the use of an external storage solution. They are not available for use when handling out-of-session requests. They are discarded once the skill session closes.

**Note:** Since session attributes are stored in the `session` property of Alexa's `ask_sdk_model.request_envelope.RequestEnvelope` and `ask_sdk_model.responseEnvelope.ResponseEnvelope` objects, only serializable types can be stored under them. The `ask_sdk_core.serialize.DefaultSerializer` is used to serialize / deserialize the values.

**Persistent Attributes**

Persistent attributes persist beyond the lifecycle of the current session. How these attributes are stored, including key scope (user ID or device ID), TTL, and storage layer depends on the configuration of the skill.

**Note:** Persistent attributes are only available when you configure the skill instance with a `PersistenceAdapter`. A call to the `AttributesManager` to retrieve or save persistent attributes will raise an exception if the `PersistenceAdapter` has not been configured.

2.4.3 AttributesManager

The `AttributesManager` exposes attributes that you can retrieve and update in your handlers. `AttributesManager` is available to handlers via the `Handler Input` object. The `AttributesManager` takes care of attributes retrieval and saving so that you can interact directly with attributes needed by your skill.
class AttributesManager(object):
    def __init__(self, request_envelope, persistence_adapter=None):
        # type: (RequestEnvelope, AbstractPersistenceAdapter) -> None
        ....

@property
def request_attributes(self):
    # type: () -> Dict[str, Any]
    # Request Attributes getter
    ....

@request_attributes.setter
def request_attributes(self, attributes):
    # type: (Dict[str, Any]) -> None
    # Request Attributes setter
    ....

@property
def session_attributes(self):
    # type: () -> Dict[str, Any]
    # Session Attributes getter
    ....

@session_attributes.setter
def session_attributes(self, attributes):
    # type: (Dict[str, Any]) -> None
    # Session Attributes setter
    ....

@property
def persistent_attributes(self):
    # type: () -> Dict[str, Any]
    # Persistence Attributes getter
    # Uses the Persistence adapter to get the attributes
    ....

@persistent_attributes.setter
def persistent_attributes(self, attributes):
    # type: (Dict[str, Any]) -> None
    # Persistent Attributes setter
    ....

def save_persistent_attributes(self):
    # type: () -> None
    # Save the persistence attributes to the persistence layer
    ....

def delete_persistent_attributes(self):
    # type: () -> None
    # Delete the persistence attributes from the persistence layer
    ....

The following example shows how you can retrieve and save persistent attributes.

class PersistenceAttributesHandler(AbstractRequestHandler):
    (continues on next page)
def can_handle(handler_input):
    persistence_attr = handler_input.attributes_manager.persistent_attributes
    return persistence_attr['foo'] == 'bar'

def handle(handler_input):
    persistence_attr = handler_input.attributes_manager.persistent_attributes
    persistence_attr['foo'] = 'bar'
    handler_input.attributes_manager.save_persistent_attributes()
    return handler_input.response_builder.response

Note: To improve skill performance, AttributesManager caches the persistent attributes locally. persistence_attributes setter will only update the locally cached persistent attributes. You need to call save_persistent_attributes() to save persistent attributes to the persistence layer.

Note: The delete_attributes on the default DynamoDbPersistenceAdapter implementation will delete the persistence attributes from local cache as well as from the persistence layer (DynamoDb table).

2.4.4 PersistenceAdapter

The AbstractPersistenceAdapter is used by AttributesManager when retrieving and saving attributes to persistence layer (i.e. database or local file system). You can register any customized PersistenceAdapter that conforms to the AbstractPersistenceAdapter interface with the SDK.

All implementations of AbstractPersistenceAdapter needs to follow the following interface.

Interface

```python
class AbstractPersistenceAdapter(object):
    def get_attributes(self, request_envelope):
        # type: (RequestEnvelope) -> Dict[str, Any]
        pass

    def save_attributes(self, request_envelope, attributes):
        # type: (RequestEnvelope, Dict[str, Any]) -> None
        pass
```

DynamoDbPersistenceAdapter

The ask-sdk-dynamodb-persistence-adapter package provides an implementation of AbstractPersistenceAdapter using AWS DynamoDB.

Interface

```python
from ask_sdk_dynamodb.adapter import DynamoDBAdapter

adapter = DynamoDBAdapter(table_name, partition_key_name="id")
```

(continues on next page)
Configuration Options

- **table_name** (string) - The name of the DynamoDB table used.
- **partition_key_name** (string) - Optional. The name of the partition key column. Default to "id" if not provided.
- **attributes_name** (string) - Optional. The name of the attributes column. Default to "attributes" if not provided.
- **create_table** (boolean) - Optional. Set to True to have DynamoDbAdapter automatically create the table if it does not exist. Default to False if not provided.
- **partition_keygen** (callable) - Optional. The function used to generate partition key using RequestEnvelope. Default to generate the partition key using the user_id.
- **dynamodb_resource** (AWS.DynamoDB ServiceResource) - Optional. The DynamoDBClient used to query AWS DynamoDB table. You can inject your DynamoDBClient with custom configuration here. Default to use boto3.resource("dynamodb").

2.5 Alexa Service Clients

Alexa Skills Kit provides multiple service APIs that you can use to personalize your skill experience. The SDK includes service clients that you can use to call Alexa APIs from within your skill logic.

Note: SDK also provides support for out-of-session Alexa APIs (Proactive Events, Skill Messaging etc.). For more information on how to call these services through SDK, please check Out Of Session Alexa Service Clients.

2.5.1 ServiceClientFactory

The service_client_factory contained inside the Handler Input allows you to retrieve client instances for every supported Alexa service. It takes care of creating individual service clients and configuring the metadata like api_access_token and api_endpoint.

Since it is available in handler_input through service_client_factory attribute, service clients can be used in any request handler, exception handler, and request, response interceptors.

Available service clients

```python
def get_device_address_service(self):
    # type: () -> ask_sdk_model.services.device_address.DeviceAddressServiceClient

def get_directive_service(self):
    # type: () -> ask_sdk_model.services.directive.DirectiveServiceClient
```

(continues on next page)
def get_list_management_service(self):
    # type: () -> ask_sdk_model.services.list_management.ListManagementServiceClient

def get_monetization_service(self):
    # type: () -> ask_sdk_model.services.monetization.MonetizationServiceClient

def get_ups_service(self):
    # type: () -> ask_sdk_model.services.ups.UpsServiceClient

def get_reminder_management_service(self):
    # type: () -> ask_sdk_model.services.reminder_management.ReminderManagementServiceClient

Note: The service_client_factory is only available for use, when you configure the skill instance with an ApiClient.

2.5.2 ApiClient

The ask_sdk_model.services.api_client.ApiClient is used by the service_client_factory when making API calls to Alexa services. You can register any customized ApiClient that conforms to the following interface with the SDK.

Interface

class ask_sdk_model.services.api_client.ApiClient:
    def invoke(self, request):
        # type: (ApiClientRequest) -> ApiClientResponse

class ask_sdk_model.services.api_client_request.ApiClientRequest(ApiClientMessage):
    def __init__(self, headers=None, body=None, url=None, method=None):
        # type: (List[Tuple[str, str]], str, str, str) -> None

class ask_sdk_model.services.api_client_request.ApiClientResponse(ApiClientMessage):
    def __init__(self, headers=None, body=None, status_code=None):
        # type: (List[Tuple[str, str]], str, int) -> None

class ask_sdk_model.services.api_client_message.ApiClientMessage(object):
    def __init__(self, headers=None, body=None):
        # type: (List[Tuple[str, str]], str) -> None

The CustomSkillBuilder constructor can be used to register the ApiClient.

from ask_sdk_core.skill_builder import CustomSkillBuilder

sb = CustomSkillBuilder(api_client=<YourClassInstance>)

DefaultApiClient

A DefaultApiClient based on the requests library, is made available in the ask_sdk_core.api_client module for skill developers.
This client is registered by default in the StandardSkillBuilder. Alternatively, skill developers can register this client to the CustomSkillBuilder.

```python
from ask_sdk_core.skill_builder import CustomSkillBuilder
from ask_sdk_core.api_client import DefaultApiClient

sb = CustomSkillBuilder(api_client=DefaultApiClient())
```

### 2.5.3 DeviceAddressServiceClient

DeviceAddressServiceClient can be used to query Device Address API for address data associated with the customer’s Alexa device. You can then use this address data to provide key functionality for the skill, or to enhance the customer experience. For example, your skill could provide a list of nearby store locations or provide restaurant recommendations using this address information.

**Interface**

```python
class ask_sdk_model.services.device_address.DeviceAddressServiceClient:
    def get_country_and_postal_code(self, device_id):
        # type: (str) -> Union[ShortAddress, Error]
        ...

    def get_full_address(self, device_id):
        # type: (str) -> Union[Address, Error]
        ...

class ask_sdk_model.services.device_address.ShortAddress:
    def __init__(self, country_code=None, postal_code=None):
        # type: (Optional[str], Optional[str]) -> None
        ...

class ask_sdk_model.services.device_address.Address:
    def __init__(self, address_line1=None, address_line2=None, address_line3=None,
                 country_code=None, state_or_region=None, city=None,
                 district_or_county=None, postal_code=None):
        # type: (Optional[str], Optional[str], Optional[str], Optional[str],
               Optional[str], Optional[str], Optional[str], Optional[str]) -> None
        ...
```

**Note:** The device_id can be retrieved from `handler_input.requestEnvelope.context.system.device.device_id`.

More information on the models can be found [here](#).

### Code Sample

The following example shows how a request handler retrieves customer’s full address.

```python
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_intent_name
from ask_sdk_model.response import Response
from ask_sdk_model.ui import AskForPermissionsConsentCard
from ask_sdk_model.services import ServiceException
```

(continues on next page)
NOTIFY_MISSING_PERMISSIONS = ("Please enable Location permissions in "
   "the Amazon Alexa app.")
NO_ADDRESS = ("It looks like you don't have an address set. "
   "You can set your address from the companion app.")
ADDRESS_AVAILABLE = "Here is your full address: {}, {}, {}"
ERROR = "Uh Oh. Looks like something went wrong."
LOCATION_FAILURE = ("There was an error with the Device Address API. "
   "Please try again.")

permissions = ["read::alexa:device:all:address"]

class GetAddressIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("GetAddressIntent")(handler_input)
    
def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        req_envelope = handler_input.request_envelope
        service_client_fact = handler_input.service_client_factory
        response_builder = handler_input.response_builder

        if not (req_envelope.context.system.user.permissions and
                req_envelope.context.system.user.permissions.consent_token):
            response_builder.speak(NOTIFY_MISSING_PERMISSIONS)
            response_builder.set_card(AssertForPermissionsConsentCard(permissions=permissions))
            return

        try:
            device_id = req_envelope.context.system.device.device_id
            device_addr_client = service_client_fact.get_device_address_service()
            addr = device_addr_client.get_full_address(device_id)

            if addr.address_line1 is None and addr.state_or_region is None:
                response_builder.speak(NO_ADDRESS)
            else:
                response_builder.speak(ADDRESS_AVAILABLE.format(
                    addr.address_line1, addr.state_or_region, addr.postal_code))
                return
        except ServiceException:
            response_builder.speak(ERROR)
            return
        except Exception as e:
            raise e

2.5.4 DirectiveServiceClient

DirectiveServiceClient can be used to send directives to Progressive Response API. Progressive responses can be used to keep the user engaged while your skill prepares a full response to the user’s request.
## Interface

```python
class ask_sdk_model.services.directive.DirectiveServiceClient:
    def enqueue(self, send_directive_request):
        # type: (SendDirectiveRequest) -> Union[Error]

class ask_sdk_model.services.directive.SendDirectiveRequest:
    def __init__(self, header=None, directive=None):
        # type: (Optional[Header], Optional[SpeakDirective]) -> None

class ask_sdk_model.services.directive.SpeakDirective:
    def __init__(self, speech=None):
        # type: (Optional[str]) -> None
```

More information on the models can be found [here](#).

### Code Sample

The following example shows a function that can be used in a `handle` method for sending a progressive response.

```python
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_model.services.directive import (SendDirectiveRequest, Header, SpeakDirective)
import time

def get_progressive_response(handler_input):
    # type: (HandlerInput) -> None
    request_id_holder = handler_input.request_envelope.request.request_id
    directive_header = Header(request_id=request_id_holder)
    speech = SpeakDirective(speech="Ok, give me a minute")
    directive_request = SendDirectiveRequest(
        header=directive_header, directive=speech)

    directive_service_client = handler_input.service_client_factory.get_directive_service()
    directive_service_client.enqueue(directive_request)
    time.sleep(5)
    return
```

### 2.5.5 ListManagementServiceClient

ListManagementServiceClient can be used to access the List Management API in order to read or modify both the Alexa default lists and any custom lists customer may have.

## Interface

```python
class ask_sdk_model.services.list_management.ListManagementServiceClient:
    def get_lists_metadata(self):
        # type: () -> Union[ForbiddenError, Error, AlexaListsMetadata]

    def get_list(self, list_id, status):
        # type: (str, str) -> Union[AlexaList, Error]
```

(continues on next page)
def get_list_item(self, list_id, item_id):
    # type: (str, str) -> Union[AlexaListItem, Error]

def create_list(self, create_list_request):
    # type: (CreateListRequest) -> Union[Error, AlexaListMetadata]

def create_list_item(self, list_id, create_list_item_request):
    # type: (str, CreateListItemRequest) -> Union[AlexaListItem, Error]

def update_list(self, list_id, update_list_request):
    # type: (str, UpdateListRequest) -> Union[Error, AlexaListMetadata]

def update_list_item(self, list_id, item_id, update_list_item_request):
    # type: (str, str, UpdateListItemRequest) -> Union[AlexaListItem, Error]

def delete_list(self, list_id):
    # type: (str) -> Union[Error]

def delete_list_item(self, list_id, item_id):
    # type: (str, str) -> Union[Error]

More information on the models can be found here.

### 2.5.6 MonetizationServiceClient

**In-Skill Purchase Service**

The SDK provides a MonetizationServiceClient that invokes inSkillPurchase API to retrieve all in-skill products associated with the current skill along with indications if each product is purchasable and/or already purchased by the current customer.

**Interface**

```python
class ask_sdk_model.services.monetization.MonetizationServiceClient:
    def get_in_skill_products(
        self, accept_language, purchasable=None, entitled=None,
        product_type=None, next_token=None, max_results=None):
        # type: (str, Optional[PurchasableState], Optional[EntitledState],
        →Optional[ProductType], Optional[str], Optional[float]) → Union[Error,
        →InSkillProductsResponse]

    def get_in_skill_product(self, accept_language, product_id):
        # type: (str, str) → Union[Error, InSkillProduct]

class ask_sdk_model.services.monetization.InSkillProductsResponse:
    def __init__(self, in_skill_products=None, is_truncated=None, next_token=None):
        # type: (Optional[List[InSkillProduct]], Optional[bool], Optional[str]) →
        →None

class ask_sdk_model.services.monetization.InSkillProduct:
    self, product_id=None, reference_name=None, name=None, object_type=None,
    →purchasable=None, entitled=None, active_entitlement_count=None, purchase_mode=None
    def __init__(
```
self, product_id=None, reference_name=None, name=None, 
object_type=None, summary=None, purchasable=None, entitled=None, 
active_entitlement_count=None, purchase_mode=None):
    # type: (Optional[str], Optional[str], Optional[str], Optional[ProductType],
       Optional[str], Optional[PurchasableState], Optional[EntitledState], Optional[int],
       Optional[PurchaseMode]) -> None

class ask_sdk_model.services.monetization.ProductType(Enum):
    SUBSCRIPTION = "SUBSCRIPTION"
    ENTITLEMENT = "ENTITLEMENT"
    CONSUMABLE = "CONSUMABLE"

class ask_sdk_model.services.monetization.PurchasableState(Enum):
    PURCHASABLE = "PURCHASABLE"
    NOT_PURCHASABLE = "NOT_PURCHASABLE"

class ask_sdk_model.services.monetization.EntitledState(Enum):
    ENTITLED = "ENTITLED"
    NOT_ENTITLED = "NOT_ENTITLED"

class ask_sdk_model.services.monetization.PurchaseMode(Enum):
    TEST = "TEST"
    LIVE = "LIVE"

Note: accept_language is the locale of the request and can be retrieved from handler_input.
request_envelope.request.locale.

More information on the models can be found here.

Code Sample

get_in_skill_products

The get_in_skill_products method retrieves all associated in-skill products for the current skill along with
purchasability and entitlement indications for each in-skill product for the current skill and customer.

from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_request_type
from ask_sdk_model.response import Response
from ask_sdk_model.services.monetization import (EntitledState, PurchasableState, InSkillProductsResponse)

class LaunchRequestHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        return is_request_type("LaunchRequest") (handler_input)

    def handle(self, handler_input):
        locale = handler_input.request_envelope.request.locale
        ms = handler_input.service_client_factory.get_monetization_service()
        product_response = ms.get_in_skill_products(locale)
        if isinstance(product_response, InSkillProductsResponse):
            # continue with processing
total_products = len(product_response.in_skill_products)
entitled_products = len([l for l in product_response.in_skill_products
                        if l.entitled == EntitledState.ENTITLED])
purchasable_products = len([l for l in product_response.in_skill_products
                           if l.purchasable == PurchasableState.PURCHASABLE])

speech = (
    "Found total {} products of which {} are purchasable and {} 
    "are entitled.".format(
    total_products, purchasable_products, entitled_products))
else:
speech = "Something went wrong in loading your purchase history."

return handler_input.response_builder.speak(speech).response

The API response contains an array of in-skill product records.

get_in_skill_product:

The get_in_skill_product API retrieves the product record for a single in-skill product identified by a given productId.

from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_request_type
from ask_sdk_model.response import Response
from ask_sdk_model.services.monetization import InSkillProduct

class LaunchRequestHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        return is_request_type("LaunchRequest")(handler_input)

    def handle(self, handler_input):
        locale = handler_input.request_envelope.request.locale
        ms = handler_input.service_client_factory.get_monetization_service()
        product_id = "amzn1.adg.product.<GUID>"
        product_response = ms.get_in_skill_product(locale)

        if isinstance(product_response, InSkillProduct):
            # code to handle InSkillProduct goes here
            speech = ""
            pass
        else:
            speech = "Something went wrong in loading your product."

        return handler_input.response_builder.speak(speech).response

The API response contains a single in-skill product record.

More information on these APIs and their usage for skill implementation is available here: Add In-Skill Purchases to a Custom Skill.
In-Skill Purchase Interface

The SDK provides the `set_directive()` method for skills to initiate in-skill purchase and cancellation requests through Alexa. Amazon systems then manage the voice interaction with customers, handle the purchase transaction and return a status response back to the requesting skill. Three different actions are supported using this interface:

- Upsell
- Buy
- Cancel

More details about these actions and recommended use-cases is available here: Add In-Skill Purchases to a Custom Skill.

Code Sample

Upsell

Skills should initiate the Upsell action to present an in-skill contextually when the user did not explicitly ask for it. E.g. During or after the free content has been served. A productId and upsell message is required to initiate the Upsell action. The upsell message allows developers to specify how Alexa can present the in-skill product to the user before presenting the pricing offer.

```python
from ask_sdk_model.interfaces.connections import SendRequestDirective

# In the skill flow, once a decision is made to offer an in-skill product to a customer without an explicit ask from the customer
return handler_input.response_builder.add_directive(
    SendRequestDirective(
        name="Upsell",
        payload={
            "InSkillProduct": {
                "productId": "<product_id>",
            },
            "upsellMessage": "<introductory upsell description for the in-skill product>",
        },
        token="correlationToken")
    ).response
```

Buy

Skills should initiate the Buy action when a customer asks to buy a specific in-skill product. A product_id is required to initiate the Buy action.

```python
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_intent_name
from ask_sdk_model.response import Response
from ask_sdk_model.interfaces.connections import SendRequestDirective

# Skills would implement a custom intent (BuyProductIntent below) that captures
```
# user's intent to buy an in-skill product and then trigger the Buy request for it.  
# For e.g. 'Alexa, buy <product name>'

class BuyProductIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("BuyProductIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

        # Obtain the corresponding product_id for the requested in-skill 
        # product by invoking InSkillProducts API.  
        # The slot variable product_name used below is only for demonstration.

        locale = handler_input.request_envelope.request.locale
        ms = handler_input.service_client_factory.get_monetization_service()

        product_response = ms.get_in_skill_products(locale)
        slots = handler_input.request_envelope.request.intent.slots
        product_ref_name = slots.get("product_name").value
        product_record = [l
            for l in product_response.in_skill_products
            if l.reference_name == product_ref_name]

        if product_record:
            return handler_input.response_builder.add_directive(
                SendRequestDirective(
                    name="Buy",
                    payload={
                        "InSkillProduct": {
                            "productId": product_record[0].product_id
                        }
                    },
                    token="correlationToken")
            ).response
        else:
            return handler_input.response_builder.speak(
                "I am sorry. That product is not available for purchase"
            ).response

## Cancel

Skills should initiate the Cancel action when a customer asks to cancel an existing Entitlement or Subscription for a supported in-skill product. A product_id is required to initiate the Cancel action.

```python
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_intent_name
from ask_sdk_model.response import Response
from ask_sdk_model.interfaces.connections import SendRequestDirective

# Skills would implement a custom intent (CancelProductIntent below) that captures
# user's intent to cancel an in-skill product and then trigger the Cancel request for
# it.
# For e.g. 'Alexa, cancel <product name>'
```
class CancelProductIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("CancelProductIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        # Obtain the corresponding product_id for the requested in-skill
        # product by invoking InSkillProducts API.
        # The slot variable product_name used below is only for demonstration.

        locale = handler_input.request_envelope.request.locale
        ms = handler_input.service_client_factory.get_monetization_service()

        product_response = ms.get_in_skill_products(locale)
        slots = handler_input.request_envelope.request.intent.slots
        product_ref_name = slots.get("product_name").value
        product_record = [l
                          for l in product_response.in_skill_products
                          if l.reference_name == product_ref_name]

        if product_record:
            return handler_input.response_builder.add_directive(
                SendRequestDirective(
                    name="Cancel",
                    payload={
                        "InSkillProduct": {
                            "productId": product_record[0].product_id
                        }
                    },
                    token="correlationToken")
            ).response
        else:
            return handler_input.response_builder.speak("I am sorry. I don’t know that one").response

2.5.7 UpsServiceClient

UpsServiceClient can be used to query Alexa Customer Profile API for customer contact information and Alexa
Customer Settings API for retrieving customer preferences for the time zone, distance measuring unit and temperature
measurement unit.

Interface

class ask_sdk_model.services.ups.UpsServiceClient:
    def get_profile_email(self):
        # type: () -> Union[str, Error]

    def get_profile_given_name(self):
        # type: () -> Union[str, Error]

    def get_profile_mobile_number(self):
Alexa SKills Kit SDK for Python Documentation, Release 1.10.2

# type: () -> Union[PhoneNumber, Error]
def get_profile_name(self):
    # type: () -> Union[str, Error]

def get_system_distance_units(self, device_id):
    # type: (str) -> Union[Error, DistanceUnits]

def get_system_temperature_unit(self, device_id):
    # type: (str) -> Union[TemperatureUnit, Error]

def get_system_time_zone(self, device_id):
    # type: (str) -> Union[str, Error]

class ask_sdk_model.services.ups.PhoneNumber:
    def __init__(self, country_code=None, phone_number=None):
        # type: (Optional[str], Optional[str]) -> None

class ask_sdk_model.services.DistanceUnits(Enum):
    METRIC = "METRIC"
    IMPERIAL = "IMPERIAL"

class ask_sdk_model.services.TemperatureUnit(Enum):
    CELSIUS = "CELSIUS"
    FAHRENHEIT = "FAHRENHEIT"

Code Sample

Alexa Customer Settings API

The Alexa Customer Settings API provides access to three pieces of information: preferred distance units, preferred temperature units and the device’s current configured time zone. When using the UpsServiceClient, get_system_distance_units and get_system_temperature_unit will return enum objects whose values must be accessed by using the .value attribute. By comparison, get_system_time_zone will simply return a string.

```
device_id = req_envelope.context.system.device.device_id
user_preferences_client = handler_input.service_client_factory.get_ups_service()

# Fetch Preferred Distance Units From Alexa Settings API
preferred_distance_units = user_preferences_client.get_system_distance_units(device_id).value
print (preferred_distance_units) # String of either "IMPERIAL" or "METRIC"

# Fetch Preferred Temperature Units From Alexa Settings API
preferred_temperature_units = user_preferences_client.get_system_temperature_unit(device_id).value
print (preferred_temperature_units) # String of either "FAHRENHEIT" or "CELSIUS"

# Fetch Device's Current Configured Time Zone From Alexa Settings API
time_zone = user_preferences_client.get_system_time_zone(device_id)
print (time_zone) # String representing a time zone for example "America/Los_Angeles"
```

Note: The device_id can be retrieved from handler_input.requestEnvelope.context.system.
device.device_id.

More information on the models can be found here.

## 2.5.8 ReminderManagementServiceClient

ReminderManagementServiceClient can be used to create and manage Reminders for your customers.

**Interface**

### class ask_sdk_model.services.reminder_management.ReminderManagementServiceClient:

```python
def create_reminder(self, reminder_request):
    # type: (ReminderRequest) -> Union[ReminderResponse, Error]

def update_reminder(self, alert_token, reminder_request):
    # type: (str, ReminderRequest) -> Union[ReminderResponse, Error]

def delete_reminder(self, alert_token):
    # type: (str) -> Union[Error]

def get_reminder(self, alert_token):
    # type: (str) -> Union[GetReminderResponse, Error]

def get_reminders(self):
    # type: () -> Union[GetRemindersResponse, Error]
```

**Code Sample**

The following example shows a request handler that creates an instance of the ReminderManagementServiceClient and creates a new reminder.

```python
import logging
import typing
from datetime import datetime
from ask_sdk_core.skill_builder import CustomSkillBuilder
from ask_sdk_model.ui import SimpleCard
from ask_sdk_core.utils import is_intent_name
from ask_sdk_core.api_client import DefaultApiClient
from ask_sdk_model.services.reminder_management import ReminderRequest, Trigger, TriggerType, AlertInfo, PushNotification, PushNotificationStatus, ReminderResponse, SpokenInfo, SpokenText
from ask_sdk_model.services import ServiceException
from ask_sdk_model.ui import AskForPermissionsConsentCard

if typing.TYPE_CHECKING:
    from ask_sdk_core.handler_input import HandlerInput
    from ask_sdk_model import Response

permissions = ["alexa::alerts:reminders:skill:readwrite"]
NOTIFY_MISSING_PERMISSIONS = ("Please enable Reminders permissions in "
                              "the Amazon Alexa app.")
```

(continues on next page)
sb = CustomSkillBuilder(api_client=DefaultApiClient())
logger = logging.getLogger(__name__)
logger.setLevel(logging.INFO)

@sb.request_handler(can_handle_func=is_intent_name("CreateReminder"))
def create_reminder_intent_handler(handler_input):
    req_envelope = handler_input.request_envelope
    response_builder = handler_input.response_builder

    # Check if user gave permissions to create reminders.
    # If not, request to provide permissions to the skill.
    if not (req_envelope.context.system.user.permissions and
             req_envelope.context.system.user.permissions.consent_token):
        response_builder.speak(NOTIFY_MISSING_PERMISSIONS)
        response_builder.set_card(
            AskForPermissionsConsentCard(permissions=permissions))
        return response_builder.response

    reminder_client = handler_input.service_client_factory.get_reminder_management_service()

    try:
        reminder_response = reminder_client.create_reminder(
            request_time=datetime.utcnow(),
            trigger=Trigger(
                object_type=TriggerType.SCHEDULED_RELATIVE,
                offset_in_seconds=60),
            alert_info=AlertInfo(
                spoken_info=SpokenInfo(
                    content=[SpokenText(locale="en-US", text="Test reminder")]),
                push_notification=PushNotification(
                    status=PushNotificationStatus.ENABLED)))
        speech_text = "Great! I've scheduled a reminder for you."
        logger.info("Created reminder : {}".format(reminder_response))
        return handler_input.response_builder.speak(speech_text).set_card(
            SimpleCard(
                "Reminder created with id", reminder_response.alert_token)).response
    except ServiceException as e:
        logger.info("Exception encountered : {}".format(e.body))
        speech_text = "Uh Oh. Looks like something went wrong."
        return handler_input.response_builder.speak(speech_text).set_card(
            SimpleCard(
                "Reminder not created", str(e.body))).response

More information on the models can be found here.
2.6 Alexa Out-Of-Session Service Clients

Some of the Alexa Skills Kit Service APIs can also be used outside your skill logic. For example, you can use Skill Messaging API to send messages to a skill. The skill should be configured to handle the events that are sent through these out-of-service request.

Since these service calls are out-of-session of a customer’s skill context, you need to provide an access token that has proper service-dependant scope. So, to perform this service call without using SDK, you would need to do the following:

- Obtain the required access token from Alexa, by retrieving the ClientId and ClientSecret from skill’s permissions tab in developer console & calling the Alexa endpoint with proper scope.
- Call the service api with appropriate input parameters, along with authorized access token.

However, SDK provides service clients that short-circuits both steps into a single service call. The client takes in your ClientId and ClientSecret, injects the required scope w.r.t the service, retrieves an access token, and uses it to call the Alexa service and provides the end response object. This reduces the boiler plate code you need to set-up, just to get the service call to be running.

**Important:** The ClientId and ClientSecret values on the developer console are only shown for skills with appropriate permissions set. You can retrieve them from the skill -> permissions tab in developer console.

**Note:** Since these service clients are out of context of a skill session, these are not available under service_client_factory in the handler_input object. The In-Session Service Clients document can provide more information on which services can be called in skill session context.

### 2.6.1 Available service clients

- **Proactive Events:** ask_sdk_model.services.proactive_events.proactive_events_service_client.ProactiveEventsServiceClient
- **Skill Messaging:** ask_sdk_model.services.skill_messaging.skill_messaging_service_client.SkillMessagingServiceClient

The service clients needs instances of ask_sdk_model.services.api_configuration.ApiConfiguration and ask_sdk_model.services.authentication_configuration.AuthenticationConfiguration in the constructor.

### 2.6.2 AuthenticationConfiguration

The ask_sdk_model.services.authentication_configuration.AuthenticationConfiguration is the configuration class that accepts the ClientId and ClientSecret for retrieving the access token from Alexa.

### 2.6.3 ApiConfiguration

The ask_sdk_model.services.api_configuration.ApiConfiguration is required for configuring the api_client to be used for making the service calls, the serializer to use for serialization/deserialization of the request/response objects, the api_endpoint to which the calls have to be made.
2.6.4 ProactiveEventsServiceClient

The Proactive Events API enables Alexa Skill Developers to send events to Alexa, which represent factual data that may interest a customer. Upon receiving an event, Alexa proactively delivers the information to customers subscribed to receive these events.

This API currently supports one proactive channel, Alexa Notifications. As more proactive channels are added in the future, developers will be able to take advantage of them without requiring integration with a new API.

Interface

```python
class ask_sdk_model.services.proactive_events.ProactiveEventsServiceClient:
    def __init__(self, api_configuration, authentication_configuration):
        # type: (ApiConfiguration, AuthenticationConfiguration) -> None

    def create_proactive_event(self, create_proactive_event_request, stage):
        # type: (CreateProactiveEventRequest, SkillStage) -> Union[Error]

class ask_sdk_model.services.proactive_events.CreateProactiveEventRequest:
    def __init__(self, timestamp=None, reference_id=None, expiry_time=None, event=None, localized_attributes=None, relevant_audience=None):
        # type: (Optional[datetime], Optional[str], Optional[datetime], Optional[Event], Optional[List[object]], Optional[RelevantAudience]) -> None

class ask_sdk_model.services.proactive_events.SkillStage(Enum):
    DEVELOPMENT = "DEVELOPMENT"
    LIVE = "LIVE"

class ask_sdk_model.services.proactive_events.Event:
    def __init__(self, name=None, payload=None):
        # type: (Optional[str], Optional[object]) -> None

class ask_sdk_model.services.proactive_events.RelevantAudience:
    def __init__(self, object_type=None, payload=None):
        # type: (Optional[RelevantAudienceType], Optional[object]) -> None

class ask_sdk_model.services.proactive_events.RelevantAudienceType(Enum):
    Unicast = "Unicast"
    Multicast = "Multicast"
```
More information on the models can be found [here](#).

### Code-Sample

The following example shows how to send a sample weather proactive event to Alexa, which will multicast it to all users registered on the skill to receive it.

```python
from datetime import datetime, timedelta

from ask_sdk_model.services.proactive_events import (ProactiveEventsServiceClient, CreateProactiveEventRequest, RelevantAudienceType, RelevantAudience, SkillStage, Event)

from ask_sdk_model.services import (ApiConfiguration, AuthenticationConfiguration)

from ask_sdk_core.serialize import DefaultSerializer

from ask_sdk_core.api_client import DefaultApiClient

def create_notification():
    client_id = "XXXX"
    client_secret = "XXXX"
    user_id = "XXXX"

    proactive_client = ProactiveEventsServiceClient(
        api_configuration=ApiConfiguration(
            serializer=DefaultSerializer(),
            api_client=DefaultApiClient(),
            api_endpoint="https://api.amazonalexa.com"),
        authentication_configuration=AuthenticationConfiguration(
            client_id=client_id,
            client_secret=client_secret))

    weather_event = Event(
        name="AMAZON.WeatherAlert.Activated",
        payload={
            "weatherAlert": {
                "alertType": "SNOW_STORM",
                "source": "localizedattribute:source"
            }
        }
    )

    create_event = CreateProactiveEventRequest(
        timestamp=datetime.utcnow(),
        reference_id="1234",
        expiry_time=datetime.utcnow() + timedelta(hours=1),
        event=weather_event,
        localized_attributes=[{"locale": "en-US", "source": "Foo"}],
        relevant_audience=RelevantAudience(
            object_type=RelevantAudienceType.Multicast,
            payload={}
        )
    )

    proactive_client.create_proactive_event(
        create_proactive_event_request=create_event,
        stage=SkillStage.DEVELOPMENT)
```

---

2.6. Alexa Out-Of-Session Service Clients
2.6.5 SkillMessagingServiceClient

The Skill Messaging API can be used to send a message request to a skill for a specified user.

Interface

class ask_sdk_model.services.skill_messaging.SkillMessagingServiceClient:
    def __init__(self, api_configuration, authentication_configuration):
        # type: (ApiConfiguration, AuthenticationConfiguration) -> None
    def send_skill_message(self, user_id, send_skill_messaging_request):
        # type: (str, SendSkillMessagingRequest) -> Union[Error]

class ask_sdk_model.services.skill_messaging.SkillMessagingRequest:
    def __init__(self, data=None, expires_after_seconds=None):
        # type: (Optional[object], Optional[int]) -> None

More information on the models can be found here.

Code-Sample

The following example shows a sample skill message sent to a skill, that handles reminders (by having a handler that can handle requests of type Messaging.MessageReceived).

```python
from ask_sdk_core.api_client import DefaultApiClient
from ask_sdk_model.services import (ApiConfiguration, AuthenticationConfiguration)
from ask_sdk_core.serialize import DefaultSerializer
from ask_sdk_model.services.skill_messaging import (SkillMessagingServiceClient, SendSkillMessagingRequest)

def send_skill_messaging():
    reminder_id = "XXXX"
    client_id = "XXXX"
    client_secret = "XXXX"
    user_id = "XXXX"

    skill_messaging_client = SkillMessagingServiceClient(
        api_configuration=ApiConfiguration(
            serializer=DefaultSerializer(),
            api_client=DefaultApiClient(),
            api_endpoint="https://api.amazonalexa.com"),
        authentication_configuration=AuthenticationConfiguration(
            client_id=client_id,
            client_secret=client_secret)
    )

    message = SendSkillMessagingRequest(
        data={"reminder_id": reminder_id})

    skill_messaging_client.send_skill_message(
        user_id=user_id, send_skill_messaging_request=message)
```

Chapter 2. SDK Features
2.6.6 LwaClient

The LwaClient is used by other out-of-session service clients, to obtain the access token from Alexa, with the required scope specific to the service. However, provided a specific scope, it can also be used natively by the skill developers, to obtain access tokens.

Interface

class ask_sdk_model.services.lwa.LwaClient:
    def __init__(self, api_configuration, authentication_configuration):
        # type: (ApiConfiguration, AuthenticationConfiguration) -> None
    
def get_access_token_for_scope(self, scope):
        # type: (str) -> str

More information on the models can be found here.

Code-Sample

The following example shows how to obtain an access-token for a scope alexa:abc.

```python
from ask_sdk_core.api_client import DefaultApiClient
from ask_sdk_model.services import (ApiConfiguration, AuthenticationConfiguration)
from ask_sdk_core.serialize import DefaultSerializer
from ask_sdk_model.services.lwa import LwaClient

def out_of_session_reminder_update():
    client_id = "XXXX"
    client_secret = "XXXX"
    scope = "alexa:abc"

    api_configuration = ApiConfiguration(
        serializer=DefaultSerializer(),
        api_client=DefaultApiClient(),
        api_endpoint="https://api.amazonalexa.com")

    lwa_client = LwaClient(
        api_configuration=api_configuration,
        authentication_configuration=AuthenticationConfiguration(
            client_id=client_id,
            client_secret=client_secret))

    access_token = lwa_client.get_access_token_for_scope(scope=scope)
```

2.7 Host a Custom Skill as a Web Service

You can build a custom skill for Alexa by implementing a web service that accepts requests from and sends responses to the Alexa service in the cloud.

The web service must meet certain requirements to handle requests sent by Alexa and adhere to the Alexa Skills Kit interface standards. For more information, see Host a Custom Skill as a Web Service in the Alexa Skills Kit technical documentation.
Warning: These features are currently in beta. You can view the source code in the Ask Python Sdk repo on GitHub. The interface might change when the features are released as stable.

2.7.1 ASD SDK Web Service Support

The Alexa Skills Kit SDK (ASK SDK) for Python provides boilerplate code for request and timestamp verification through the `ask-sdk-webservice-support` package, which integrates with the Skill Builder object. This package only provides the verification components and a base handler to call the skill invocation, and is independent of the underlying framework used for the web application development.

Installation

You can install the `ask-sdk-webservice-support` package through `pip`.

Important: The package has the `cryptography` package as a dependency for request verification. The `cryptography` package might have additional prerequisites depending on the operating system. For more information, see the installation instructions in the `cryptography` documentation.

Generic Web Service Handler

The `WebserviceSkillHandler` class registers the skill instance from the `SkillBuilder` object, and provides a `verify_request_and_dispatch` method that verifies the input request before invoking the skill handlers.

You can enable or disable request or timestamp verification for testing purposes by setting the boolean parameters `verify_signature` and `verify_timestamp` on the `WebserviceSkillHandler` instance. You can also provide additional custom verifiers that need to be applied on the input request before skill invocation.

The `verify_request_and_dispatch` method takes the `http_headers` and `http_body` from the web service, and returns the response in string format, on successful skill invocation. You have to convert the input and output into the web service-specific request and response structures.

Usage

```python
from ask_sdk_core.skill_builder import SkillBuilder
from ask_sdk_webservice_support.webservice_handler import WebserviceSkillHandler

skill_builder = SkillBuilder()

# Implement request handlers, exception handlers, etc.
# Register the handlers to the skill builder instance.

webservice_handler = WebserviceSkillHandler(
    skill=skill_builder.create())

# Convert the HTTP request headers and body into native format
# of dict and str respectively, and call the dispatch method.
response = webservice_handler.verify_request_and_dispatch(
    headers, body)

# Convert the response str into web service format and return.
```
2.7.2 Framework-Specific Adapters

Flask and Django are two web service frameworks that are commonly used to build web services in Python. The ASK SDK provides framework-specific extensions to the ask-sdk-webservice-support package for both Flask and Django that handle the request and response conversion internally. This provides an easy way to integrate SDK skills that you’ve already developed to make them work with your web service.

flask-ask-sdk Extension Package

The flask-ask-sdk package provides a Flask extension that can register a Flask application along with a custom skill. It also provides helper methods to register the skill invocation as a URL endpoint to the Flask application.

The flask-ask-sdk package follows the Flask extension structure. The SkillAdapter class constructor takes the following:

• A skill instance.
• A skill id to register the skill instance in the extension directory.
• An optional flask application, to register the extension in the application.

The class also provides an init_app method, to pass in the Flask app instance later, to instantiate and configure the extension.

The request and timestamp verifications are enabled by default. You can use the app configurations VERIFY_SIGNATURE_APP_CONFIG and VERIFY_TIMESTAMP_APP_CONFIG to disable or enable the respective verifications by setting boolean values to them.

You can use the SkillAdapter’s dispatch_request method to register the skill as an endpoint url-rule. It handles the request and response conversion, request and timestamp verification, and skill invocation.

Installation

You can install the flask-ask-sdk package through pip.

Important: The package has the cryptography package as a dependency for request verification. The cryptography package might have additional prerequisites depending on the operating system. For more information, see the installation instructions in the cryptography documentation.

Usage

```python
from flask import Flask
from ask_sdk_core.skill_builder import SkillBuilder
from ask_sdk_sdk.skill_adapter import SkillAdapter

app = Flask(__name__)
skill_builder = SkillBuilder()
# Register your intent handlers to the skill_builder object

skill_adapter = SkillAdapter(
    skill=skill_builder.create(), skill_id=<SKILL_ID>, app=app)

@app.route("/"):  
(continues on next page)"
def invoke_skill:
    return skill_adapter.dispatch_request()

Note: An instance of the extension is added to the application extensions mapping, using the key `ASK_SDK_SKILL_ADAPTER`. Since multiple skills can be configured on different routes in the same application, through multiple extension instances, each extension is added as a skill ID mapping in the app extension's `ASK_SDK_SKILL_ADAPTER` dictionary.

django-ask-sdk Extension Package

The `django-ask-sdk` extension package provides a Django extension that you can use to register a custom skill as an endpoint in the Django application.

The extension provides a `SkillAdapter` view class. You can instantiate the view class with a custom skill instance, built through the ASK SDK Skill Builder object, and register it in the `urls.py` file of the Django app so that the skill is invoked at the corresponding endpoint.

The request and timestamp verifications are enabled by default. You can use the constructor arguments `verify_request` and `verify_timestamp` to disable or enable the respective verifications by setting boolean values to them.

Installation

You can install the `django-ask-sdk` extension through `pip`.

Important: The package has the `cryptography` package as a dependency for request verification. The `cryptography` package might have additional prerequisites depending on the operating system. For more information, see the installation instructions in the `cryptography` documentation.

Note: The `django-ask-sdk` package is compatible with Python 3.0 or higher because it depends on Django 2.0 which only supports Python 3.

Usage

If you develop a skill using the `SkillBuilder` instance, then you can use the following in `example.urls.py` to register it as an endpoint in a Django app called `example`:

```python
import skill
from django_ask_sdk.skill_response import SkillAdapter

view = SkillAdapter.as_view(skill=skill.sb.create())

urlpatterns = [
    path("/myskill", view, name='index')
]```
2.8 Request Processing

Covers how to build request handlers, exception handlers, and request and response interceptors.

2.9 Response Building

Covers how to use the ResponseBuilder to compose multiple elements like text, cards, and audio into a single response.

2.10 Skill Attributes

Covers how to use skill attributes to store and retrieve skill data.

2.11 Skill Builders

Covers how to configure and construct a skill instance.

2.12 Alexa Service Clients

Covers how to use service clients in your skill to access Alexa APIs.

2.13 Alexa Out-Of-Session Service Clients

Covers how to call Alexa APIs that work out of skill’s session context (For example, to send notifications to skill users outside the normal skill flow etc.)

2.14 Hosting Skills as Webservice

Covers how to host a skill as a web service.

2.14.1 Runtime

Request Dispatch Components

Abstract Classes

```
class ask_sdk_runtime.dispatch.AbstractRequestDispatcher
    Bases: object

    Dispatcher which handles dispatching input request to the corresponding handler.
    User needs to implement the dispatch method, to handle the processing of the incoming request in the handler input. A response may be expected out of the dispatch method.
```
dispatch (handler_input)
Dispatches an incoming request to the appropriate request handler and returns the output.

Parameters
handler_input (Input) – generic input to the dispatcher

Returns
generic output returned by handler in the dispatcher

Return type
Union[None, Output]

class ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler
Bases: object

Request Handlers are responsible for processing dispatch inputs and generating output.

Custom request handlers need to implement can_handle and handle methods. can_handle returns True if the handler can handle the current input. handle processes the input and may return a output.

can_handle (handler_input)
Returns true if Request Handler can handle the dispatch input.

Parameters
handler_input (Input) – Generic input passed to the dispatcher.

Returns
Boolean value that tells the dispatcher if the current input can be handled by this handler.

Return type
bool

handle (handler_input)
Handles the dispatch input and provides an output for dispatcher to return.

Parameters
handler_input (Input) – Generic input passed to the dispatcher.

Returns
Generic Output for the dispatcher to return or None

Return type
Union[Output, None]

class ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor
Bases: object

Interceptor that runs before the handler is called.

The process method has to be implemented, to run custom logic on the input, before it is handled by the Handler.

process (handler_input)
Process the input before the Handler is run.

Parameters
handler_input (Input) – Generic input passed to the dispatcher.

Return type
None

class ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor
Bases: object

Interceptor that runs after the handler is called.

The process method has to be implemented, to run custom logic on the input and the dispatch output generated after the handler is executed on the input.

process (handler_input, response)
Process the input and the output after the Handler is run.

Parameters
• handler_input (Input) – Generic input passed to the dispatcher.
• **response** ([`Union[None, Output]`](#)) – Execution result of the Handler on dispatch input.

  **Return type** `None`

```python
class ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandlerChain
    Bases: object

    Abstract class containing Request Handler and corresponding Interceptors.

    request_handler()

        Returns  Registered Request Handler instance.

        **Return type** `object`

    request_interceptors()

        Returns  List of registered Request Interceptors.

        **Return type** `list(ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor)`

    response_interceptors()

        Returns  List of registered Response Interceptors.

        **Return type** `list(ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor)`
```

```python
class ask_sdk_runtime.dispatch_components.request_components.AbstractRequestMapper
    Bases: object

    Class for request routing to the appropriate handler chain.

    User needs to implement get_request_handler_chain method, to provide a routing mechanism of the input to the appropriate request handler chain containing the handler and the interceptors.

    get_request_handler_chain(handler_input)

        Get the handler chain that can process the handler input.

        **Parameters**

        `handler_input` (*Input*) – Generic input passed to the dispatcher.

        **Returns**  Handler Chain that can handle the request under dispatch input.

        **Return type** `AbstractRequestHandlerChain`
```

```python
class ask_sdk_runtime.dispatch_components.request_components.AbstractHandlerAdapter
    Bases: object

    Abstracts handling of a request for specific handler types.

    supports(handler)

        Returns true if adapter supports the handler.

        This method checks if the adapter supports the handler execution. This is usually checked by the type of the handler.

        **Parameters**

        `handler` (*object*) – Request Handler instance.

        **Returns**  Boolean denoting whether the adapter supports the handler.

        **Return type** `bool`

    execute(handler_input, handler)

        Executes the handler with the provided dispatch input.

        **Parameters**

        `handler_input` (*Input*) – Generic input passed to the dispatcher.
• **handler** *(object)* – Request Handler instance.

**Returns**  Result executed by passing handler_input to handler.

**Return type**  Union[None, Output]

class ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler
Bases: object

Handles exception types and optionally produce an output.

The abstract class is similar to Request Handler, with methods can_handle and handle. The can_handle method checks if the handler can support the input and the exception. The handle method processes the input and exception, to optionally produce an output.

can_handle *(handler_input, exception)*

Checks if the handler can support the exception raised during dispatch.

**Parameters**

• **handler_input** *(Input)* – Generic input passed to the dispatcher.

• **exception** *(Exception)* – Exception raised during dispatch.

**Returns**  Boolean whether handler can handle exception or not.

**Return type**  bool

handle *(handler_input, exception)*

Process the dispatch input and exception.

**Parameters**

• **handler_input** *(Input)* – Generic input passed to the dispatcher.

• **exception** *(Exception)* – Exception raised during dispatch.

**Returns**  Optional output object to serve as dispatch return.

**Return type**  Union[None, Output]

class ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionMapper
Bases: object

Mapper to register custom Exception Handler instances.

The exception mapper is used by ask_sdk_runtime.dispatch.GenericRequestDispatcher dispatch method, to handle exceptions. The mapper can contain one or more exception handlers. Handlers are accessed through the mapper to attempt to find a handler that is compatible with the current exception.

get_handler *(handler_input, exception)*

Returns a suitable exception handler to dispatch the specified exception, if one exists.

**Parameters**

• **handler_input** *(Input)* – Generic input passed to the dispatcher.

• **exception** *(Exception)* – Exception thrown by ask_sdk_runtime.dispatch.GenericRequestDispatcher dispatch method.

**Returns**  Exception Handler that can handle the input or None.

**Return type**  Union[None, AbstractExceptionHandler]
Implementations

class ask_sdk_runtime.dispatch.GenericRequestDispatcher(options)
   Bases: ask_sdk_runtime.dispatch.AbstractRequestDispatcher

   Generic implementation of AbstractRequestDispatcher.

   The runtime configuration contains the components required for the dispatcher, which is passed during initialization.

   When the dispatch method is invoked, using a list of ask_sdk_runtime.dispatchComponents.requestComponents.RequestMapper, the Dispatcher finds a handler for the request and delegates the invocation to the supported ask_sdk_runtime.dispatchComponents.requestComponents.HandlerAdapter. If the handler raises any exception, it is delegated to ask_sdk_runtime.dispatchComponents.exceptionComponents.ExceptionMapper to handle or raise it to the upper stack.

dispatch(handler_input)
   Dispatches an incoming request to the appropriate request handler and returns the output.

   Before running the request on the appropriate request handler, dispatcher runs any predefined global request interceptors. On successful response returned from request handler, dispatcher runs predefined global response interceptors, before returning the response.

   Parameters handler_input (Input) – generic input to the dispatcher
   Returns  generic output handled by the handler, optionally containing a response
   Return type Union[None, Output]
   Raises ask_sdk_runtime.exceptions.DispatchException

class ask_sdk_runtime.dispatchComponents.requestComponents.GenericRequestHandlerChain(request_handler, request_interceptors=None, response_interceptors=None)
   Bases: ask_sdk_runtime.dispatchComponents.requestComponents.AbstractRequestHandlerChain

   Generic implementation of AbstractRequestHandlerChain.

   Generic Request Handler Chain accepts request handler of any type.

   Parameters

   • request_handler (ask_sdk_runtime.dispatchComponents.requestComponents.AbstractRequestHandler) – Registered Request Handler instance of generic type.
   • request_interceptors (list(ask_sdk_runtime.dispatchComponents.requestComponents.AbstractRequestInterceptor)) – List of registered Request Interceptors.
   • response_interceptors (list(ask_sdk_runtime.dispatchComponents.requestComponents.AbstractResponseInterceptor)) – List of registered Response Interceptors.

   request_handler
   Returns  Registered Request Handler instance.
   Return type  object
request_interceptors

Returns List of registered Request Interceptors.

Return type list(ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor)

response_interceptors

Returns List of registered Response Interceptors.

Return type list(ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor)

add_request_interceptor(interceptor)

Add interceptor to Request Interceptors list.

Parameters

interceptor (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor) – Request Interceptor instance.

add_response_interceptor(interceptor)

Add interceptor to Response Interceptors list.

Parameters


class ask_sdk_runtime.dispatch_components.request_components.GenericRequestMapper(request_handler_chains)

Bases: ask_sdk_runtime.dispatch_components.request_components.AbstractRequestMapper

Implementation of AbstractRequestMapper that registers RequestHandlerChain.

The class accepts request handler chains of type GenericRequestHandlerChain only. The get_request_handler_chain method returns the GenericRequestHandlerChain instance that can handle the request in the handler input.

Parameters request_handler_chains (list(GenericRequestHandlerChain)) – List of GenericRequestHandlerChain instances.

request_handler_chains

Returns List of GenericRequestHandlerChain instances.

Return type list(GenericRequestHandlerChain)

add_request_handler_chain(request_handler_chain)

Checks the type before adding it to the request_handler_chains instance variable.

Parameters

request_handler_chain (RequestHandlerChain) – Request Handler Chain instance.

Raises ask_sdk_runtime.exceptions.DispatchException if a null input is provided or if the input is of invalid type

get_request_handler_chain(handler_input)

Get the request handler chain that can handle the dispatch input.

Parameters

handler_input (Input) – Generic input passed to the dispatcher.

Returns Handler Chain that can handle the input.

Return type Union[None, GenericRequestHandlerChain]

class ask_sdk_runtime.dispatch_components.request_components.GenericHandlerAdapter

Bases: ask_sdk_runtime.dispatch_components.request_components.AbstractHandlerAdapter
GenericHandler Adapter for handlers of type `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler`.

**supports** *(handler)*

Returns true if handler is `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler` instance.

**Parameters**

- **handler** *(ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler)* – Request Handler instance

**Returns** Boolean denoting whether the adapter supports the handler.

**Return type** `bool`

**execute** *(handler_input, handler)*

Executes the handler with the provided handler input.

**Parameters**

- **handler_input** *(Input)* – Generic input passed to the dispatcher.
- **handler** *(object)* – Request Handler instance.

**Returns** Result executed by passing handler_input to handler.

**Return type** `Union[None, Output]`

**class** `ask_sdk_runtime.dispatch_components.exception_components.GenericExceptionMapper` *(exception_handlers)*

**Bases:** `ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionMapper`

Generic Implementation of exception mapper, to register `AbstractExceptionHandler` instances.

The class accepts exception handlers of type `AbstractExceptionHandler` only. The `get_handler` method returns the `AbstractExceptionHandler` instance that can handle the dispatch input and the exception raised from the dispatch method.

**Parameters**


**exception_handlers**

**Returns** List of `ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler` instances.

**Return type** `list(ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler)`

**add_exception_handler** *(exception_handler)*

Checks the type before adding it to the exception_handlers instance variable.

**Parameters**

- **exception_handler** *(ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler)* – Exception Handler instance.

**Raises** `ask_sdk_runtime.exceptions.DispatchException` if a null input is provided or if the input is of invalid type

**get_handler** *(handler_input, exception)*

Get the exception handler that can handle the input and exception.

**Parameters**

- **handler_input** *(Input)* – Generic input passed to the dispatcher.
• **exception** *(Exception)* – Exception thrown by `ask_sdk_runtime.dispatch.GenericRequestDispatcher` dispatch method.

**Returns** Exception Handler that can handle the input or None.

**Return type** `Union[None, ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler]`

### Skill Components

```python
class ask_sdk_runtime.skill.RuntimeConfiguration(request_mappers, handler_adapters, request_interceptors=None, response_interceptors=None, exception_mapper=None)
```

Bases: `object`

Configuration Object that represents standard components needed to build the dispatcher in the `AbstractSkill`.

**Parameters**

- **request_mappers** *(list(GenericRequestMapper))* – List of request mapper instances.
- **handler_adapters** *(list(GenericHandlerAdapter))* – List of handler adapter instances.
- **request_interceptors** *(list(AbstractRequestInterceptor))* – List of request interceptor instances.
- **response_interceptors** *(list(AbstractResponseInterceptor))* – List of response interceptor instances.
- **exception_mapper** *(GenericExceptionMapper)* – Exception mapper instance.

```python
class ask_sdk_runtime.skill.RuntimeConfigurationBuilder
```

Bases: `object`

Builder class for creating a runtime configuration object, from base dispatch components.

```python
add_request_handler(request_handler)
```

Register input to the request handlers list.

**Parameters** `request_handler` *(AbstractRequestHandler)* – Request Handler instance to be registered.

**Returns** None

```python
add_request_handlers(request_handlers)
```

Register input to the request handlers list.

**Parameters** `request_handlers` *(list(AbstractRequestHandler))* – List of Request Handler instances to be registered.

**Returns** None

```python
add_exception_handler(exception_handler)
```

Register input to the exception handlers list.

**Parameters** `exception_handler` *(AbstractExceptionHandler)* – Exception Handler instance to be registered.

**Returns** None
**add_global_request_interceptor** *(request_interceptor)*
Register input to the global request interceptors list.

Parameters *request_interceptor* *(AbstractRequestInterceptor)* – Request Interceptor instance to be registered.

Returns None

**add_global_response_interceptor** *(response_interceptor)*
Register input to the global response interceptors list.

Parameters *response_interceptor* *(AbstractResponseInterceptor)* – Response Interceptor instance to be registered.

Returns None

**get_runtime_configuration** *
Build the runtime configuration object from the registered components.

Returns Runtime Configuration Object

Return type *RuntimeConfiguration*

**class** *ask_sdk_runtime.skill.AbstractSkill*

Bases: object

Abstract class that acts as entry level container for skill invocation.

Domain SDKs should implement the *supports* and *invoke* methods.

**supports** *(event, context)*
Check if the skill supports the corresponding input.

Parameters

• *event* *(SkillInput)* – input instance containing request information.

• *context* *(Any)* – Context passed during invocation

Returns boolean if this type of request can be handled by this skill.

Return type *bool*

**invoke** *(event, context)*
Invokes the dispatcher, to handle the skill input and return a skill output.

Parameters

• *event* *(SkillInput)* – input instance containing request information.

• *context* *(Any)* – Context passed during invocation

Returns output generated by handling the request.

Return type *SkillOutput*

**class** *ask_sdk_runtime.skill_builder.AbstractSkillBuilder*

Bases: object

Abstract Skill Builder with helper functions for building *ask_sdk_runtime.skill.AbstractSkill* object.

Domain SDKs has to implement the *create* method that returns an instance of the skill implementation for the domain type.

**add_request_handler** *(request_handler)*
Register input to the request handlers list.
Parameters `request_handler` (ask_sdk_runtime.dispatch_components.
request_components.AbstractRequestHandler) – Request Handler instance
to be registered.

Returns None

**add_exception_handler** (exception_handler)
Register input to the exception handlers list.

Parameters `exception_handler` (ask_sdk_runtime.dispatch_components.
request_components.AbstractExceptionHandler) – Exception Handler
instance to be registered.

Returns None

**add_global_request_interceptor** (request_interceptor)
Register input to the global request interceptors list.

Parameters `request_interceptor` (ask_sdk_runtime.
dispatch_components.request_components.AbstractRequestInterceptor)
– Request Interceptor instance to be registered.

Returns None

**add_global_response_interceptor** (response_interceptor)
Register input to the global response interceptors list.

Parameters `response_interceptor` (ask_sdk_runtime.
dispatch_components.request_components.AbstractResponseInterceptor)
– Response Interceptor instance to be registered.

Returns None

**request_handler** (can_handle_func)
Decorator that can be used to add request handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes a single parameter and no varargs or
kwargs. This is because of the RequestHandler class signature restrictions. The returned wrapper function
can be applied as a decorator on any function that returns a response object by the skill. The function should
follow the signature of the handle function in ask_sdk_runtime.dispatch_components.
request_components.AbstractRequestHandler.

Parameters `can_handle_func` (Callable[[Input], bool]) – The function that val-
idates if the request can be handled.

Returns Wrapper function that can be decorated on a handle function.

**exception_handler** (can_handle_func)
Decorator that can be used to add exception handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes two parameters and no varargs or
kwargs. This is because of the ExceptionHandler class signature restrictions. The returned wrapper function
can be applied as a decorator on any function that processes the exception raised during dis-
patcher and returns a response object by the skill. The function should follow the signature of the handle function in ask_sdk_runtime.dispatch_components.
exception_components.AbstractExceptionHandler.

Parameters `can_handle_func` (Callable[[Input, Exception], bool]) – The function that validates if the exception can be handled.

Returns Wrapper function that can be decorated on a handle function.
global_request_interceptor()
Decorator that can be used to add global request interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor` class.

Returns Wrapper function that can be decorated on a interceptor process function.

global_response_interceptor()
Decorator that can be used to add global response interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input and the response generated by the request handler. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor` class.

Returns Wrapper function that can be decorated on a interceptor process function.

create()
Create a skill object using the registered components.

Returns a skill object that can be used for invocation.

Return type AbstractSkill

SDK Exceptions

exception ask_sdk_runtime.exceptions.AskSdkException
    Bases: Exception
    Base class for exceptions raised by the SDK.

exception ask_sdk_runtime.exceptions.DispatchException
    Bases: ask_sdk_runtime.exceptions.AskSdkException
    Class for exceptions raised during dispatch logic.

exception ask_sdk_runtime.exceptions.SerializationException
    Bases: ask_sdk_runtime.exceptions.AskSdkException
    Class for exceptions raised during serialization/deserialization.

exception ask_sdk_runtime.exceptions.SkillBuilderException
    Bases: ask_sdk_runtime.exceptions.AskSdkException
    Base exception class for Skill Builder exceptions.

exception ask_sdk_runtime.exceptions.RuntimeConfigException
    Bases: ask_sdk_runtime.exceptions.AskSdkException
    Base exception class for Runtime Configuration Builder exceptions.

2.14.2 Core

Handler Input

class ask_sdk_core.handler_input.HandlerInput(requestEnvelope, attributes_manager=None, context=None, service_client_factory=None)

Bases: object
Input to Request Handler, Exception Handler and Interceptors.

Handler Input instantiations are passed to the registered instances of AbstractRequestHandler and AbstractExceptionHandler, during skill invocation. The class provides a AttributesManager and a ResponseFactory instance, apart from RequestEnvelope, Context and ServiceClientFactory instances, to utilize during the lifecycle of skill.

**Parameters**

- **request_envelope** *(ask_sdk_model.request_envelope.RequestEnvelope)* – Request Envelope passed from Alexa Service
- **attributes_manager** *(ask_sdk_core.attributes_manager.AttributesManager)* – Attribute Manager instance for managing attributes across skill lifecycle
- **context** *(object)* – Context object passed from Lambda service
- **service_client_factory** *(ask_sdk_model.services.service_client_factory.ServiceClientFactory)* – Service Client Factory instance for calling Alexa services

**service_client_factory**

Service Client Factory instance for calling Alexa services.

To use the Alexa services, one need to configure the API Client in the skill builder object, before creating the skill.

**Request Dispatch Components**

**Abstract Classes**

```python
class ask_sdk_core.dispatch_components.request_components.AbstractRequestHandler
    Bases: ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler

    Request Handlers are responsible for processing Request inside the Handler Input and generating Response.

    Custom request handlers needs to implement can_handle and handle methods. can_handle returns True if the handler can handle the current request. handle processes the Request and may return a Response.

can_handle(handler_input)
    Returns true if Request Handler can handle the Request inside Handler Input.

    Parameters handler_input (HandlerInput) – Handler Input instance with Request Envelope containing Request.

    Returns Boolean value that tells the dispatcher if the current request can be handled by this handler.

    Return type bool

handle(handler_input)
    Handles the Request inside handler input and provides a Response for dispatcher to return.

    Parameters handler_input (HandlerInput) – Handler Input instance with Request Envelope containing Request.

    Returns Response for the dispatcher to return or None

    Return type Union[Response, None]
```
class ask_sdk_core.dispatch_components.request_components.AbstractRequestInterceptor

    Bases: ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor

Interceptor that runs before the handler is called.

The `process` method has to be implemented, to run custom logic on the input, before it is handled by the Handler.

```python
process(handler_input)
```

Process the input before the Handler is run.

    Parameters
        handler_input (HandlerInput) – Handler Input instance.

    Return type None

class ask_sdk_core.dispatch_components.request_components.AbstractResponseInterceptor

    Bases: ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor

Interceptor that runs after the handler is called.

The `process` method has to be implemented, to run custom logic on the input and the response generated after the handler is executed on the input.

```python
process(handler_input, response)
```

Process the input and the response after the Handler is run.

    Parameters
        • handler_input (HandlerInput) – Handler Input instance.
        • response (Union[None, ask_sdk_model.Response]) – Execution result of the Handler on handler input.

    Return type None

class ask_sdk_core.dispatch_components.exception_components.AbstractExceptionHandler

    Bases: ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler

Handles exception types and optionally produce a response.

The abstract class is similar to Request Handler, with methods `can_handle` and `handle`. The `can_handle` method checks if the handler can support the input and the exception. The `handle` method processes the input and exception, to optionally produce a response.

```python
can_handle(handler_input, exception)
```

Checks if the handler can support the exception raised during dispatch.

    Parameters
        • handler_input (HandlerInput) – Handler Input instance.
        • exception (Exception) – Exception raised during dispatch.

    Returns Boolean whether handler can handle exception or not.

    Return type bool

class ask_sdk_core.dispatch_components.exception_components.AbstractExceptionHandler

    Bases: ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler

Handles exception types and optionally produce a response.

The abstract class is similar to Request Handler, with methods `can_handle` and `handle`. The `can_handle` method checks if the handler can support the input and the exception. The `handle` method processes the input and exception, to optionally produce a response.

```python
can_handle(handler_input, exception)
```

Checks if the handler can support the exception raised during dispatch.

    Parameters
        • handler_input (HandlerInput) – Handler Input instance.
        • exception (Exception) – Exception raised during dispatch.

    Returns Boolean whether handler can handle exception or not.

    Return type bool

class ask_sdk_core.dispatch_components.exception_components.AbstractExceptionHandler

    Bases: ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler

Handles exception types and optionally produce a response.

The abstract class is similar to Request Handler, with methods `can_handle` and `handle`. The `can_handle` method checks if the handler can support the input and the exception. The `handle` method processes the input and exception, to optionally produce a response.

```python
can_handle(handler_input, exception)
```

Checks if the handler can support the exception raised during dispatch.

    Parameters
        • handler_input (HandlerInput) – Handler Input instance.
        • exception (Exception) – Exception raised during dispatch.

    Returns Boolean whether handler can handle exception or not.

    Return type bool

class ask_sdk_core.dispatch_components.exception_components.AbstractExceptionHandler

    Bases: ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler

Handles exception types and optionally produce a response.

The abstract class is similar to Request Handler, with methods `can_handle` and `handle`. The `can_handle` method checks if the handler can support the input and the exception. The `handle` method processes the input and exception, to optionally produce a response.

```python
can_handle(handler_input, exception)
```

Checks if the handler can support the exception raised during dispatch.

    Parameters
        • handler_input (HandlerInput) – Handler Input instance.
        • exception (Exception) – Exception raised during dispatch.

    Returns Boolean whether handler can handle exception or not.

    Return type bool

• `exception(Exception)` – Exception raised during dispatch.

Returns Optional response object to serve as dispatch return.

Return type Union[None, `Response`]

Response Builder Components

```python
ask_sdk_core.response_helper.PLAIN_TEXT_TYPE = 'PlainText'
str: Helper variable for plain text type.
ask_sdk_core.response_helper.RICH_TEXT_TYPE = 'RichText'
str: Helper variable for rich text type.

class ask_sdk_core.response_helper.ResponseFactory

ResponseFactory is class which provides helper functions to help building a response.

`speak(speech, play_behavior=None)`
Say the provided speech to the user.

Parameters

• `speech (str)` – the output speech sent back to the user.

• `play_behavior (ask_sdk_model.ui.play_behavior.PlayBehavior)` – attribute to control alexa’s speech interruption

Returns response factory with partial response being built and access from self.response.

Return type `ResponseFactory`

`ask(reprompt, play_behavior=None)`
Provide reprompt speech to the user, if no response for 8 seconds.

The should_end_session value will be set to false except when the video app launch directive is present in directives.

Parameters

• `reprompt (str)` – the output speech to reprompt.

• `play_behavior (ask_sdk_model.ui.play_behavior.PlayBehavior)` – attribute to control alexa’s speech interruption

Returns response factory with partial response being built and access from self.response.

Return type `ResponseFactory`

`set_card(card)`
Renders a card within the response.


Parameters `card (ask_sdk_model.ui.card.Card)` – card object in response sent back to user.

Returns response factory with partial response being built and access from self.response.

Return type `ResponseFactory`

`add_directive(directive)`
Adds directive to response.
Parameters **directive** *(ask_sdk_model.directive.Directive)* – the directive sent back to Alexa device.

**Returns** response factory with partial response being built and access from self.response.

**Return type** *ResponseFactory*

**set_should_end_session** *(should_end_session)*
Sets shouldEndSession value to null/false/true.

Parameters **should_end_session** *(bool)* – value to show if the session should be ended or not.

**Returns** response factory with partial response being built and access from self.response.

**Return type** *ResponseFactory*

**set_can_fulfill_intent** *(can_fulfill_intent)*
Sets CanFulfill intent to the response.

For more information on CanFulfillIntent, check the name-free interaction doc here: https://developer.amazon.com/docs/custom-skills/understand-name-free-interaction-for-custom-skills.html

Parameters **can_fulfill_intent** *(CanFulfillIntent)* – CanFulfill Intent sent back in response.

**Returns** response factory with partial response being built and access from self.response.

**Return type** *ResponseFactory*

```
ask_sdk_core.response_helper.get_plain_text_content
```


**Parameters**

- **primary_text** *(optional) str* – Text for primary_text field
- **secondary_text** *(optional) str* – Text for secondary_text field
- **tertiary_text** *(optional) str* – Text for tertiary_text field

**Returns** Text Content instance with primary, secondary and tertiary text set as Plain Text objects.

**Return type** *TextContent*

**Raises** ValueError

```
ask_sdk_core.response_helper.get_rich_text_content
```


**Parameters**

- **primary_text** *(optional) str* – Text for primary_text field
- **secondary_text** *(optional) str* – Text for secondary_text field
- **tertiary_text** *(optional) str* – Text for tertiary_text field

**Returns** Text Content instance with primary, secondary and tertiary text set as Rich Text objects.
Return type  

**TextContent**

Raises  

**ValueError**

```python
ask_sdk_core.response_helper.get_text_content(prim
ary_text=None, primary_text_type='PlainText', sec
ondary_text=None, secondary_text_type='PlainText', ter
tiary_text=None, tertiary_text_type='PlainText')
```

Responsible for building text content object using ask-sdk-model in Alexa skills kit display interface.  

Parameters

- **primary_text** *(optional)*  
  - Text for primary_text field
- **primary_text_type** *(optional)*  
  - Type of the primary text field. Allowed values are *PlainText* and *RichText*. Defaulted to *PlainText*.
- **secondary_text** *(optional)*  
  - Text for secondary_text field
- **secondary_text_type** –  
  - Type of the secondary text field. Allowed values are *PlainText* and *RichText*. Defaulted to *PlainText*.
- **tertiary_text** *(optional)*  
  - Text for tertiary_text field
- **tertiary_text_type** –  
  - Type of the tertiary text field. Allowed values are *PlainText* and *RichText*. Defaulted to *PlainText*.

Returns  

Text Content instance with primary, secondary and tertiary text set.

Return type  

**TextContent**

Raises  

**ValueError**

### Skill Components

**class ask_sdk_core.skill.SkillConfiguration** *(request_mappers, handler_adapters, request_interceptors=None, response_interceptors=None, exception_mapper=None, persistence_adapter=None, api_client=None, custom_user_agent=None, skill_id=None)*

Bases: *ask_sdk_runtime.skill.RuntimeConfiguration*

Configuration Object that represents standard components needed to build Skill.

Parameters

- **request_mappers** *(list (GenericRequestMapper))*  
  - List of request mapper instances.
- **handler_adapters** *(list (GenericHandlerAdapter))*  
  - List of handler adapter instances.
- **request_interceptors** *(list (ask_sdk_core.dispatch_components.request_components.AbstractRequestInterceptor))*  
  - List of request interceptor instances.
- **response_interceptors** *(list (ask_sdk_core.dispatch_components.request_components.AbstractResponseInterceptor))*  
  - List of response interceptor instances.
• **exception.mapper** *(GenericExceptionMapper)* – Exception mapper instance.
• **persistence_adapter** *(AbstractPersistenceAdapter)* – Persistence adapter instance.
• **api_client** *(ask_sdk_model.services.api_client.ApiClient)* – Api Client instance.
• **custom_user_agent** *(str)* – Custom User Agent string
• **skill_id** *(str)* – ID of the skill.

class ask_sdk_core.skill.CustomSkill(skill_configuration)
   Bases: ask_sdk_runtime.skill.AbstractSkill

Top level container for Request Dispatcher, Persistence Adapter and Api Client.

Parameters **skill_configuration** *(SkillConfiguration)* – Configuration object that holds information about different components needed to build the skill object.

**supports** *(request_envelope, context)*
Check if request envelope is of the expected skill format.

Parameters
   • **request_envelope** *(Dict[str, Any]*) – input instance containing request information.
   • **context** *(Any)* – Context passed during invocation

Returns boolean if this type of request can be handled by this skill.

Return type **bool**

**invoke** *(request_envelope, context)*
Invokes the dispatcher, to handle the request envelope and return a response envelope.

Parameters
   • **request_envelope** *(RequestEnvelope)* – Request Envelope instance containing request information
   • **context** *(Any)* – Context passed during invocation

Returns Response Envelope generated by handling the request

Return type **ResponseEnvelope**

class ask_sdk_core.skill_builder.SkillBuilder
   Bases: ask_sdk_runtime.skill_builder.AbstractSkillBuilder

Skill Builder with helper functions for building ask_sdk_core.skill.Skill object.

**skill_configuration**
Create the skill configuration object using the registered components.

create()
Create a skill object using the registered components.

Returns a skill object that can be used for invocation.

Return type **Skill**

lambda_handler()
Create a handler function that can be used as handler in AWS Lambda console.
The lambda handler provides a handler function, that acts as an entry point to the AWS Lambda console. Users can set the lambda_handler output to a variable and set the variable as AWS Lambda Handler on the console.

As mentioned in the AWS Lambda Handler docs, the handler function receives the event attribute as a str representing the input request envelope JSON from Alexa service, which is deserialized to `ask_sdk_model.request_envelope.RequestEnvelope`, before invoking the skill. The output from the handler function would be the serialized `ask_sdk_model.responseEnvelope` class from the appropriate skill handler.

**Returns** Handler function to tag on AWS Lambda console.

```python
add_exception_handler(exception_handler)
```

Register input to the exception handlers list.

**Parameters**

`exception_handler (ask_sdk_runtime.dispatch_components.request_components.AbstractExceptionHandler)` – Exception Handler instance to be registered.

**Returns** None

```python
add_global_request_interceptor(request_interceptor)
```

Register input to the global request interceptors list.

**Parameters**

`request_interceptor (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor)` – Request Interceptor instance to be registered.

**Returns** None

```python
add_global_response_interceptor(response_interceptor)
```

Register input to the global response interceptors list.

**Parameters**


**Returns** None

```python
add_request_handler(request_handler)
```

Register input to the request handlers list.

**Parameters**

`request_handler (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler)` – Request Handler instance to be registered.

**Returns** None

```python
exception_handler(can_handle_func)
```

Decorator that can be used to add exception handlers easily to the builder.

The `can_handle_func` has to be a Callable instance, which takes two parameters and no varargs or kwargs. This is because of the ExceptionHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that processes the exception raised during dispatcher and returns a response object by the skill. The function should follow the signature of the handle function in `ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler` class.

**Parameters**

`can_handle_func (Callable[[Input, Exception], bool])` – The function that validates if the exception can be handled.

**Returns** Wrapper function that can be decorated on a handle function.
**global_request_interceptor()**

Decorator that can be used to add global request interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor` class.

**Returns**  Wrapper function that can be decorated on a interceptor process function.

**global_response_interceptor()**

Decorator that can be used to add global response interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input and the response generated by the request handler. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor` class.

**Returns**  Wrapper function that can be decorated on a interceptor process function.

**request_handler(can_handle_func)**

Decorator that can be used to add request handlers easily to the builder.

The `can_handle_func` has to be a Callable instance, which takes a single parameter and no varargs or kwargs. This is because of the RequestHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that returns a response object by the skill. The function should follow the signature of the handle function in `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler` class.

**Parameters**

- `can_handle_func` (*Callable[[Input], bool]*) – The function that validates if the request can be handled.

**Returns**  Wrapper function that can be decorated on a handle function.

**class ask_sdk_core.skill_builder.CustomSkillBuilder(persistence_adapter=None, api_client=None)**

**Bases:** `ask_sdk_core.skill_builder.SkillBuilder`

Skill Builder with api client and persistence adapter setter functions.

**skill_configuration**

Create the skill configuration object using the registered components.

**add_exception_handler(exception_handler)**

Register input to the exception handlers list.

**Parameters**

- `exception_handler` (*ask_sdk_runtime.dispatch_components.request_components.AbstractExceptionHandler*) – Exception Handler instance to be registered.

**Returns** None

**add_global_request_interceptor(request_interceptor)**

Register input to the global request interceptors list.

**Parameters**

- `request_interceptor` (*ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor*) – Request Interceptor instance to be registered.

**Returns** None

**add_global_response_interceptor(response_interceptor)**

Register input to the global response interceptors list.
Parameters `response_interceptor` (ask_sdk_runtime.
  dispatch_components.request_components.AbstractResponseInterceptor)
  – Response Interceptor instance to be registered.

Returns None

`add_request_handler` *(request_handler)*

Register input to the request handlers list.

Parameters `request_handler` (ask_sdk_runtime.dispatch_components.
  request_components.AbstractRequestHandler) – Request Handler instance
to be registered.

Returns None

`create` ()

Create a skill object using the registered components.

Returns a skill object that can be used for invocation.

Return type Skill

`exception_handler` *(can_handle_func)*

Decorator that can be used to add exception handlers easily to the builder.

The `can_handle_func` has to be a Callable instance, which takes two parameters and no varargs or
kwargs. This is because of the ExceptionHandler class signature restrictions. The returned wrapper
function can be applied as a decorator on any function that processes the exception raised during dis-
patcher and returns a response object by the skill. The function should follow the signature of the
handle function in ask_sdk_runtime.dispatch_components.exception_components.
AbstractExceptionHandler class.

Parameters `can_handle_func` (Callable[[Input, Exception], bool]) – The
  function that validates if the exception can be handled.

Returns Wrapper function that can be decorated on a handle function.

`global_request_interceptor` ()

Decorator that can be used to add global request interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the
input. The function should follow the signature of the process function in ask_sdk_runtime.
dispatch_components.request_components.AbstractRequestInterceptor class.

Returns Wrapper function that can be decorated on a interceptor process function.

`global_response_interceptor` ()

Decorator that can be used to add global response interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the
input and the response generated by the request handler. The function should follow the signature of
the process function in ask_sdk_runtime.dispatch_components.request_components.
AbstractResponseInterceptor class.

Returns Wrapper function that can be decorated on a interceptor process function.

`lambda_handler` ()

Create a handler function that can be used as handler in AWS Lambda console.

The lambda handler provides a handler function, that acts as an entry point to the AWS Lambda console.
Users can set the lambda_handler output to a variable and set the variable as AWS Lambda Handler on the
console.
As mentioned in the AWS Lambda Handler docs, the handler function receives the event attribute as a str representing the input request envelope JSON from Alexa service, which is deserialized to ask_sdk_model.request_envelope.RequestEnvelope, before invoking the skill. The output from the handler function would be the serialized ask_sdk_model.response_envelope.ResponseEnvelope class from the appropriate skill handler.

**Returns** Handler function to tag on AWS Lambda console.

`request_handler(can_handle_func)`

Decorator that can be used to add request handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes a single parameter and no varargs or kwargs. This is because of the RequestHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that returns a response object by the skill. The function should follow the signature of the handle function in ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler class.

**Parameters**

`can_handle_func(Callable[[Input], bool])` — The function that validates if the request can be handled.

**Returns** Wrapper function that can be decorated on a handle function.

### Skill Attribute Components

**class** ask_sdk_core.attributes_manager.AttributesManager(request_envelope, persistence_adapter=None)

AttributesManager is a class that handles three level attributes: request, session and persistence.

**Parameters**

- `request_envelope (RequestEnvelope)` — request envelope.
- `persistence_adapter (AbstractPersistenceAdapter)` — class used for storing and retrieving persistent attributes from persistence tier

**request_attributes**

Attributes stored at the Request level of the skill lifecycle.

**Returns** request attributes for the request life cycle

**Return type** Dict[str, object]

**session_attributes**

Attributes stored at the Session level of the skill lifecycle.

**Returns** session attributes extracted from request envelope

**Return type** Dict[str, object]

**persistent_attributes**

Attributes stored at the Persistence level of the skill lifecycle.

**Returns** persistent_attributes retrieved from persistence adapter

**Return type** Dict[str, object]

**Raises** ask_sdk_core.exceptions.AttributesManagerException if trying to get persistent attributes without persistence adapter

**save_persistent_attributes()**

Save persistent attributes to the persistence layer if a persistence adapter is provided.

---

2.14. Hosting Skills as Webservice 79
Return type  None

Raises  ask_sdk_core.exceptions.AttributesManagerException if trying to save persistence attributes without persistence adapter

def delete_persistent_attributes() -> None
    Deletes the persistent attributes from the persistence layer.

Return type  None

Raises  ask_sdk_core.exceptions.AttributesManagerException if trying to delete persistence attributes without persistence adapter

Abstract Classes

class ask_sdk_core.attributes_manager.AbstractPersistenceAdapter
    Bases: object

    Abstract class for storing and retrieving persistent attributes from persistence tier given request envelope.

    User needs to implement get_attributes method to get attributes from persistent tier and save_attributes method to save attributes to persistent tier.

    get_attributes(request_envelope) -> Dict[str, object]
        Get attributes from persistent tier.

        Parameters
            request_envelope (RequestEnvelope) – Request Envelope from Alexa service

        Returns  A dictionary of attributes retrieved from persistent tier

        Return type  Dict[str, object]

    save_attributes(request_envelope, attributes) -> None
        Save attributes to persistent tier.

        Parameters
            request_envelope (RequestEnvelope) – request envelope.
            attributes (Dict[str, object]) – attributes to be saved to persistent tier

        Return type  None

    delete_attributes(request_envelope) -> None
        Delete attributes from persistent tier.

        Parameters
            request_envelope (RequestEnvelope) – request envelope.

        Return type  None

API Client

class ask_sdk_core.api_client.DefaultApiClient
    Bases: ask_sdk_model.services.api_client.ApiClient

    Default ApiClient implementation of ask_sdk_model.services.api_client.ApiClient using the requests library.

    invoke(request) -> None
        Dispatches a request to an API endpoint described in the request.
Resolves the method from input request object, converts the list of header tuples to the required format (dict) for the requests lib call and invokes the method with corresponding parameters on requests library. The response from the call is wrapped under the ApiClientResponse object and the responsibility of translating a response code and response/error lies with the caller.

Parameters **request** (ApiClientRequest) – Request to dispatch to the ApiClient

Returns Response from the client call

Return type ApiClientResponse

Raises ask_sdk_core.exceptions.ApiClientException

### SDK Exceptions

**exception** ask_sdk_core.exceptions.AttributesManagerException

Bases: ask_sdk_runtime.exceptions.AskSdkException

Class for exceptions raised during handling attributes logic

**exception** ask_sdk_core.exceptions.SerializationException

Bases: ask_sdk_runtime.exceptions.AskSdkException

Class for exceptions raised during serialization/deserialization.

**exception** ask_sdk_core.exceptions.PersistenceException

Bases: ask_sdk_runtime.exceptions.AskSdkException

Exception class for Persistence Adapter processing.

**exception** ask_sdk_core.exceptions.ApiClientException

Bases: ask_sdk_runtime.exceptions.AskSdkException

Exception class for ApiClient Adapter processing.

### Default Serializer

**class** ask_sdk_core.serialize.DefaultSerializer

Bases: ask_sdk_model.services.serializer.Serializer

**serialize**(obj)

Builds a serialized object.

- If obj is None, return None.
- If obj is str, int, long, float, bool, return directly.
- If obj is datetime.datetime, datetime.date convert to string in iso8601 format.
- If obj is list, serialize each element in the list.
- If obj is dict, return the dict with serialized values.
- If obj is ask sdk model, return the dict with keys resolved from the union of model’s attribute_map and deserialized_types and values serialized based on deserialized_types.
- If obj is a generic class instance, return the dict with keys from instance’s deserialized_types and values serialized based on deserialized_types.

Parameters **obj**(object) – The data to serialize.

Returns The serialized form of data.
**Return type**  Union[Dict[str, Any], List, Tuple, str, int, float, None]

**deserialize**(payload, obj_type)
Deserializes payload into an instance of provided obj_type.

The obj_type parameter can be a primitive type, a generic model object or a list/dict of model objects.

The list or dict object type has to be provided as a string format. For eg:

- 'list[a.b.C]' if the payload is a list of instances of class a.b.C.
- 'dict(str, a.b.C)' if the payload is a dict containing mappings of str : a.b.C class instance types.

The method looks for a deserialized_types dict in the model class, that mentions which payload values has to be deserialized. In case the payload key names are different than the model attribute names, the corresponding mapping can be provided in another special dict attribute_map. The model class should also have the __init__ method with default values for arguments. Check ask_sdk_model.request_envelope.RequestEnvelope source code for an example implementation.

**Parameters**
- payload (str) – data to be deserialized.
- obj_type (Union[object, str]) – resolved class name for deserialized object

**Returns** deserialized object
**Return type** object
**Raises** ask_sdk_core.exceptions.SerializationException

**General Utilities**

ask_sdk_core.utils.predicate.is_canfulfill_intent_name(name)
A predicate function returning a boolean, when name matches the intent name in a CanFulfill Intent Request.

The function can be applied on a ask_sdk_core.handler_input.HandlerInput, to check if the input is of ask_sdk_model.intent_request.CanFulfillIntentRequest type and if the name of the request matches with the passed name.

**Parameters** name (str) – Name to be matched with the CanFulfill Intent Request Name
**Returns** Predicate function that can be used to check name of the request
**Return type** Callable[[HandlerInput], bool]

ask_sdk_core.utils.predicate.is_intent_name(name)
A predicate function returning a boolean, when name matches the name in Intent Request.

The function can be applied on a ask_sdk_core.handler_input.HandlerInput, to check if the input is of ask_sdk_model.intent_request.IntentRequest type and if the name of the request matches with the passed name.

**Parameters** name (str) – Name to be matched with the Intent Request Name
**Returns** Predicate function that can be used to check name of the request
**Return type** Callable[[HandlerInput], bool]

ask_sdk_core.utils.predicate.is_request_type(request_type)
A predicate function returning a boolean, when request type is the passed-in type.
The function can be applied on a `ask_sdk_core.handler_input.HandlerInput`, to check if the input request type is the passed in request type.

**Parameters** `request_type` *(str)* – request type to be matched with the input’s request

**Returns** Predicate function that can be used to check the type of the request

**Return type** `Callable[[HandlerInput], bool]`

class ask_sdk_core.utils.viewport.OrderedEnum
   Bases: enum.Enum

   An enumeration.

   XLOW = 0
   LOW = 1
   MEDIUM = 2
   HIGH = 3
   XHIGH = 4
   XXHIGH = 5

class ask_sdk_core.utils.viewport.Density
   Bases: ask_sdk_core.utils.viewport.OrderedEnum

   An enumeration.

   XLOW = 0
   LOW = 1
   MEDIUM = 2
   HIGH = 3
   XHIGH = 4
   XXHIGH = 5

class ask_sdk_core.utils.viewport.Orientation
   Bases: ask_sdk_core.utils.viewport.OrderedEnum

   An enumeration.

   LANDSCAPE = 0
   EQUAL = 1
   PORTRAIT = 2

class ask_sdk_core.utils.viewport.Size
   Bases: ask_sdk_core.utils.viewport.OrderedEnum

   An enumeration.

   XSMALL = 0
   SMALL = 1
   MEDIUM = 2
   LARGE = 3
   XLARGE = 4

class ask_sdk_core.utils.viewport.ViewportProfile
   Bases: enum.Enum

   An enumeration.

   HUB_ROUND_SMALL = 'HUB_ROUND_SMALL'
   HUB_LANDSCAPE_SMALL = 'HUB_LANDSCAPE_SMALL'
   HUB_LANDSCAPE_MEDIUM = 'HUB_LANDSCAPE_MEDIUM'
   HUB_LANDSCAPE_LARGE = 'HUB_LANDSCAPE_LARGE'
MOBILE_LANDSCAPE_SMALL = 'MOBILE_LANDSCAPE_SMALL'
MOBILE_PORTRAIT_SMALL = 'MOBILE_PORTRAIT_SMALL'
MOBILE_LANDSCAPE_MEDIUM = 'MOBILE_LANDSCAPE_MEDIUM'
MOBILE_PORTRAIT_MEDIUM = 'MOBILE_PORTRAIT_MEDIUM'
TV_LANDSCAPE_XLARGE = 'TV_LANDSCAPE_XLARGE'
TV_PORTRAIT_MEDIUM = 'TV_PORTRAIT_MEDIUM'
TV_LANDSCAPE_MEDIUM = 'TV_LANDSCAPE_MEDIUM'
UNKNOWN_VIEWPORT_PROFILE = 'UNKNOWN_VIEWPORT_PROFILE'

ask_sdk_core.utils.viewport.get_orientation(width, height)

Get viewport orientation from given width and height.

Returns viewport orientation enum

Return type Orientation

ask_sdk_core.utils.viewport.get_size(size)

Get viewport size from given size.

Returns viewport size enum

Return type Size

ask_sdk_core.utils.viewport.get_dpi_group(dpi)

Get viewport density group from given dpi.

Returns viewport density group enum

Return type Density

ask_sdk_core.utils.viewport.get_viewport_profile(request_envelope)

Utility method, to get viewport profile.

The viewport profile is calculated using the shape, current pixel width and height, along with the dpi.

If there is no viewport value in request_envelope.context, then an ViewportProfile.UNKNOWN_VIEWPORT_PROFILE is returned.

Parameters request_envelope (ask_sdk_model.request_envelope.RequestEnvelope) – The alexa request envelope object

Returns Calculated Viewport Profile enum

Return type ViewportProfile

ask_sdk_core.utils.request_util.get_locale(handler_input)

Return locale value from input request.

The method returns the locale value present in the request. More information about the locale can be found here: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html#request-locale

Parameters handler_input (ask_sdk_core.handler_input.HandlerInput) – The handler input instance that is generally passed in the sdk’s request and exception components

Returns Locale value from the request

Return type str
ask_sdk_core.utils.request_util.get_request_type(handler_input)
Return the type of the input request.

The method retrieves the request type of the input request. More information about the different request types are mentioned here: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html#request-body-parameters

**Parameters**

**handler_input** (ask_sdk_core.handler_input.HandlerInput) – The handler input instance that is generally passed in the sdk’s request and exception components

**Returns**

Type value of the input request

**Return type** str

ask_sdk_core.utils.request_util.get_intent_name(handler_input)
Return the name of the intent request.

The method retrieves the intent name from the input request, only if the input request is an ask_sdk_model.intent_request.IntentRequest. If the input is not an IntentRequest, a TypeError is raised.

**Parameters**

**handler_input** (ask_sdk_core.handler_input.HandlerInput) – The handler input instance that is generally passed in the sdk’s request and exception components

**Returns**

Name of the intent request

**Return type** str

**Raises**

TypeError

ask_sdk_core.utils.request_util.get_account_linking_access_token(handler_input)
Return the access token in the request.

The method retrieves the user’s accessToken from the input request. Once a user successfully enables a skill and links their Alexa account to the skill, the input request will have the user’s access token. A None value is returned if there is no access token in the input request. More information on this can be found here: https://developer.amazon.com/docs/account-linking/add-account-linking-logic-custom-skill.html

**Parameters**

**handler_input** (ask_sdk_core.handler_input.HandlerInput) – The handler input instance that is generally passed in the sdk’s request and exception components

**Returns**

User account linked access token if available. None if not available

**Return type** Optional[str]

ask_sdk_core.utils.request_util.get_api_access_token(handler_input)
Return the api access token in the request.

The method retrieves the apiAccessToken from the input request, which has the encapsulated information of permissions granted by the user. This token can be used to call Alexa-specific APIs. More information about this can be found here: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html#system-object

The SDK already includes this token in the API calls done through the service_client_factory in ask_sdk_core.handler_input.HandlerInput.

**Parameters**

**handler_input** (ask_sdk_core.handler_input.HandlerInput) – The handler input instance that is generally passed in the sdk’s request and exception components

**Returns**

Api access token from the input request, which encapsulates any permissions consented by the user

**Return type** str
ask_sdk_core.utils.request_util.get_device_id(handler_input)
Return the device id from the input request.

The method retrieves the `deviceId` property from the input request. This value uniquely identifies the device and is generally used as input for some Alexa-specific API calls. More information about this can be found here: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html#system-object

If there is no device information in the input request, then a `None` is returned.

**Parameters**

- **handler_input** (ask_sdk_core.handler_input.HandlerInput) – The handler input instance that is generally passed in the sdk’s request and exception components.

**Returns**

- Unique device id of the device used to send the alexa request or `None` if device information is not present.

**Return type**

Optional[str]

ask_sdk_core.utils.request_util.get_dialog_state(handler_input)
Return the dialog state enum from the intent request.

The method retrieves the `dialogState` from the intent request, if the skill’s interaction model includes a dialog model. This can be used to determine the current status of user conversation and return the appropriate dialog directives if the conversation is not yet complete. More information on dialog management can be found here: https://developer.amazon.com/docs/custom-skills/define-the-dialog-to-collect-and-confirm-required-information.html

The method returns a `None` if there is no dialog model added or if the intent doesn’t have dialog management.

The method raises a `TypeError` if the input is not an `IntentRequest`.

**Parameters**

- **handler_input** (ask_sdk_core.handler_input.HandlerInput) – The handler input instance that is generally passed in the sdk’s request and exception components.

**Returns**

- State of the dialog model from the intent request.

**Return type**

Optional[ask_sdk_model.dialog_state.DialogState]

**Raises**

- TypeError if the input is not an IntentRequest

ask_sdk_core.utils.request_util.get_slot(handler_input, slot_name)
Return the slot information from intent request.

The method retrieves the slot information `ask_sdk_model.slot.Slot` from the input intent request for the given `slot_name`. More information on the slots can be found here: https://developer.amazon.com/docs/custom-skills/request-types-reference.html#slot-object

If there is no such slot, then a `None` is returned. If the input request is not an `ask_sdk_model.intent_request.IntentRequest`, a `TypeError` is raised.

**Parameters**

- **handler_input** (ask_sdk_core.handler_input.HandlerInput) – The handler input instance that is generally passed in the sdk’s request and exception components.

- **slot_name** (str) – Name of the slot that needs to be retrieved

**Returns**

- Slot information for the provided slot name if it exists, or a `None` value

**Return type**

Optional[ask_sdk_model.slot.Slot]

**Raises**

- TypeError if the input is not an IntentRequest

ask_sdk_core.utils.request_util.get_slot_value(handler_input, slot_name)
Return the slot value from intent request.
The method retrieves the slot value from the input intent request for the given `slot_name`. More information on the slots can be found here: https://developer.amazon.com/docs/custom-skills/request-types-reference.html#slot-object

If there is no such slot, then a `ValueError` is raised. If the input request is not an `ask_sdk_model.intent_request.IntentRequest`, a `TypeError` is raised.

**Parameters**

- `handler_input (ask_sdk_core.handler_input.HandlerInput)`—The handler input instance that is generally passed in the sdk’s request and exception components
- `slot_name (str)`—Name of the slot for which the value has to be retrieved

**Returns** Slot value for the provided slot if it exists

**Return type** `str`

**Raises** `TypeError` if the input is not an IntentRequest. `ValueError` is slot doesn’t exist

`ask_sdk_core.utils.request_util.get_supported_interfaces(handler_input)`

Retrieves the supported interfaces from input request.

The method returns an `ask_sdk_model.supported_interfaces.SupportedInterfaces` object instance listing each interface that the device supports. For example, if `supported_interfaces` includes `audio_player`, then you know that the device supports streaming audio using the AudioPlayer interface. More information on `supportedInterfaces` can be found here: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html#system-object

**Parameters** `handler_input (ask_sdk_core.handler_input.HandlerInput)`—The handler input instance that is generally passed in the sdk’s request and exception components

**Returns** Instance of `ask_sdk_model.supported_interfaces.SupportedInterfaces` mentioning which all interfaces the device supports

**Return type** `ask_sdk_model.supported_interfaces.SupportedInterfaces`

`ask_sdk_core.utils.request_util.is_new_session(handler_input)`

Return if the session is new for the input request.

The method retrieves the `new` value from the input request’s session, which indicates if it’s a new session or not. The `ask_sdk_model.session.Session` is only included on all standard requests except AudioPlayer, VideoApp and PlaybackController requests. More information can be found here: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html#session-object

A `TypeError` is raised if the input request doesn’t have the session information.

**Parameters** `handler_input (ask_sdk_core.handler_input.HandlerInput)`—The handler input instance that is generally passed in the sdk’s request and exception components

**Returns** Boolean if the session is new for the input request

**Return type** `bool`

**Raises** `TypeError` if the input request doesn’t have a session

`ask_sdk_core.utils.request_util.get_user_id(handler_input)`

Return the userId in the request.

The method retrieves the `userId` from the input request. This value uniquely identifies the user and is generally used as input for some Alexa-specific API calls. More information about this can be found here: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html#system-object

:param handler_input: The handler input instance that is generally passed in the sdk’s request and exception components

2.14. Hosting Skills as Webservice 87
Returns
Users userId or None if not available

Return type
Optional[str]

2.14.3 DynamoDB Persistence Adapter

DynamoDB Persistence Adapter

class ask_sdk_dynamodb.adapter.DynamoDbAdapter (table_name, partition_key_name='id',
attribute_name='attributes',
create_table=False, partition_keygen=<function user_id_partition_keygen>,
dynamodb_resource=dynamodb.ServiceResource())

Bases: ask_sdk_core.attributes_manager.AbstractPersistenceAdapter

Persistence Adapter implementation using Amazon DynamoDb.

Amazon DynamoDb based persistence adapter implementation. This internally uses the AWS Python SDK (boto3) to process the dynamodb operations. The adapter tries to create the table if create_table is set, during initialization.

Parameters

- **table_name**(str) – Name of the table to be created or used
- **partition_key_name**(str) – Partition key name to be used. Defaulted to ‘id’
- **attribute_name**(str) – Attribute name for storing and retrieving attributes from dynamodb. Defaulted to ‘attributes’
- **create_table**(bool) – Should the adapter try to create the table if it doesn’t exist. Defaulted to False
- **partition_keygen**(Callable[[RequestEnvelope], str]) – Callable function that takes a request envelope and provides a unique partition key value. Defaulted to user id keygen function
- **dynamodb_resource**(boto3.resources.base.ServiceResource) – Resource to be used, to perform dynamo operations. Defaulted to resource generated from boto3

**delete_attributes**(request_envelope)

Deletes attributes from table in DynamoDB resource.

Deletes the attributes from Dynamodb table. Raises PersistenceException if table doesn’t exist or delete_item fails on the table.

Parameters **request_envelope**(ask_sdk_model.RequestEnvelope) – RequestEnvelope passed during skill invocation

Return type
None

Raises ask_sdk_core.exceptions.PersistenceException

**get_attributes**(request_envelope)

Get attributes from table in DynamoDB resource.

Retrieves the attributes from Dynamodb table. If the table doesn’t exist, returns an empty dict if the create_table variable is set as True, else it raises PersistenceException. Raises PersistenceException if get_item fails on the table.
**Parameters**

request_envelope *(ask_sdk_model.RequestEnvelope)* – Request Envelope passed during skill invocation

**Returns**

Attributes stored under the partition keygen mapping in the table

**Return type**

Dict[str, object]

**Raises**

`ask_sdk_core.exceptions.PersistenceException`

**save_attributes**(request_envelope, attributes)

Saves attributes to table in Dynamodb resource.

Saves the attributes into Dynamodb table. Raises PersistenceException if table doesn’t exist or put_item fails on the table.

**Parameters**

- request_envelope *(ask_sdk_model.RequestEnvelope)* – Request Envelope passed during skill invocation
- attributes *(Dict[str, object])* – Attributes stored under the partition keygen mapping in the table

**Return type**

None

**Raises**

`ask_sdk_core.exceptions.PersistenceException`

### Partition Key Generator Functions

**ask_sdk_dynamodb.partition_keygen.device_id_partition_keygen**(request_envelope)

Retrieve device id from request envelope, to use as partition key.

**Parameters**

request_envelope *(ask_sdk_model.RequestEnvelope)* – Request Envelope passed during skill invocation

**Returns**

Device Id retrieved from request envelope

**Return type**

str

**Raises**

`ask_sdk_core.exceptions.PersistenceException`

**ask_sdk_dynamodb.partition_keygen.user_id_partition_keygen**(request_envelope)

Retrieve user id from request envelope, to use as partition key.

**Parameters**

request_envelope *(ask_sdk_model.RequestEnvelope)* – Request Envelope passed during skill invocation

**Returns**

User Id retrieved from request envelope

**Return type**

str

**Raises**

`ask_sdk_core.exceptions.PersistenceException`

### 2.14.4 Standard

**Standard Skill Builder**

**class** ask_sdk.standard.StandardSkillBuilder *(table_name=None, auto_create_table=None, partition_keygen=None, dynamodb_client=None)*

**Bases:** ask_sdk_core.skill_builder.SkillBuilder
Skill Builder with api client and db adapter coupling to Skill.

Standard Skill Builder is an implementation of `ask_sdk_core.skill_builder.SkillBuilder` with coupling of DynamoDb Persistence Adapter settings and a Default Api Client added to the `ask_sdk_core.skill.Skill`.

Parameters

- **table_name** *(str)* – Name of the table to be created or used
- **auto_create_table** *(bool)* – Should the adapter try to create the table if it doesn’t exist.
- **partition_keygen** *(Callable[[RequestEnvelope], str]*) – Callable function that takes a request envelope and provides a unique partition key value.
- **dynamodb_client** *(boto3.resources.base.ServiceResource)* – Resource to be used, to perform dynamo operations.

add_exception_handler *(exception_handler)*

Register input to the exception handlers list.

Parameters **exception_handler** *(ask_sdk_runtime.dispatch_components.request_components.AbstractExceptionHandler)* – Exception Handler instance to be registered.

Returns None

add_global_request_interceptor *(request_interceptor)*

Register input to the global request interceptors list.

Parameters **request_interceptor** *(ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor)* – Request Interceptor instance to be registered.

Returns None

add_global_response_interceptor *(response_interceptor)*

Register input to the global response interceptors list.

Parameters **response_interceptor** *(ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor)* – Response Interceptor instance to be registered.

Returns None

add_request_handler *(request_handler)*

Register input to the request handlers list.

Parameters **request_handler** *(ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler)* – Request Handler instance to be registered.

Returns None

create ()

Create a skill object using the registered components.

Returns a skill object that can be used for invocation.

Return type Skill

exception_handler *(can_handle_func)*

Decorator that can be used to add exception handlers easily to the builder.
The `can_handle_func` has to be a Callable instance, which takes two parameters and no varargs or kwargs. This is because of the ExceptionHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that processes the exception raised during dispatcher and returns a response object by the skill. The function should follow the signature of the handle function in `ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler` class.

**Parameters**

`can_handle_func (Callable[[Input, Exception], bool])` – The function that validates if the exception can be handled.

**Returns**

Wrapper function that can be decorated on a handle function.

`global_request_interceptor()`

Decorator that can be used to add global request interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor` class.

**Returns**

Wrapper function that can be decorated on a interceptor process function.

`global_response_interceptor()`

Decorator that can be used to add global response interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input and the response generated by the request handler. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor` class.

**Returns**

Wrapper function that can be decorated on a interceptor process function.

`lambda_handler()`

Create a handler function that can be used as handler in AWS Lambda console.

The lambda handler provides a handler function, that acts as an entry point to the AWS Lambda console. Users can set the lambda_handler output to a variable and set the variable as AWS Lambda Handler on the console.

As mentioned in the AWS Lambda Handler docs, the handler function receives the event attribute as a str representing the input request envelope JSON from Alexa service, which is deserialized to `ask_sdk_model.request_envelope.RequestEnvelope`, before invoking the skill. The output from the handler function would be the serialized `ask_sdk_model.response_envelope.ResponseEnvelope` class from the appropriate skill handler.

**Returns**

Handler function to tag on AWS Lambda console.

`request_handler (can_handle_func)`

Decorator that can be used to add request handlers easily to the builder.

The `can_handle_func` has to be a Callable instance, which takes a single parameter and no varargs or kwargs. This is because of the RequestHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that returns a response object by the skill. The function should follow the signature of the handle function in `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler` class.

**Parameters**

`can_handle_func (Callable[[Input], bool])` – The function that validates if the request can be handled.

**Returns**

Wrapper function that can be decorated on a handle function.

`skill_configuration`

Create the skill configuration object using the registered components.
2.14.5 Webservice Support and Framework Adapters

ask-sdk-webservice-support package

ask_sdk_webservice_support.verifier_constants module

```
ask_sdk_webservice_support.verifier_constants.SIGNATURE_CERT_CHAIN_URL_HEADER = 'SignatureCertChainUrl'
# Header key to be used, to retrieve request header that contains the URL for the certificate chain needed to verify
# the request signature. For more info, check link.

ask_sdk_webservice_support.verifier_constants.SIGNATURE_HEADER = 'Signature'
# Header key to be used, to retrieve request header that contains the request signature. For more info, check link.

ask_sdk_webservice_support.verifier_constants.CERT_CHAIN_URL_PROTOCOL = 'https'
# Case insensitive protocol to be checked on signature certificate url. For more info, check link.

ask_sdk_webservice_support.verifier_constants.CERT_CHAIN_URL_HOSTNAME = 's3.amazonaws.com'
# Case insensitive hostname to be checked on signature certificate url. For more info, check link.

ask_sdk_webservice_support.verifier_constants.CERT_CHAIN_URL_STARTPATH = '/echo.api/'
# Path presence to be checked on signature certificate url. For more info, check link.

ask_sdk_webservice_support.verifier_constants.CERT_CHAIN_URL_PORT = 443
# Port to be checked on signature certificate url. For more info, check link.

ask_sdk_webservice_support.verifier_constants.CERT_CHAIN_DOMAIN = 'echo-api.amazon.com'
# Domain presence check in Subject Alternative Names (SANs) of signing certificate. For more info, check link.

ask_sdk_webservice_support.verifier_constants.CHARACTER_ENCODING = 'utf-8'
# Character encoding used in the request.

ask_sdk_webservice_support.verifier_constants.DEFAULT_TIMESTAMP_TOLERANCE_IN_MILLIS = 30000
# Default allowable tolerance in request timestamp. For more info, check link.

ask_sdk_webservice_support.verifier_constants.MAX_TIMESTAMP_TOLERANCE_IN_MILLIS = 3600000
# Maximum allowable tolerance in request timestamp. For more info, check link.
```

ask_sdk_webservice_support.verifier module

```
exception ask_sdk_webservice_support.verifier.VerificationException
    Bases: ask_sdk_runtime.exceptions.AskSdkException

    Class for exceptions raised during Request verification.

    args
    with_traceback()
        Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

class ask_sdk_webservice_support.verifier.AbstractVerifier
    Bases: object

    Abstract verifier class for implementing custom verifiers.

    verify(headers, serialized_request_env, deserialized_request_env)
        Abstract verify method that verifies and validates inputs.

        Custom verifiers should implement this method, to validate the headers and body of the input POST re-
        quest. The method returns a VerificationException if the validation fails, or succeeds silently.

        Parameters
```
• **headers** (*Dict[str, Any]*) – headers of the input POST request
• **serialized_request_env** (*str*) – raw request envelope in the input POST request
• **deserialized_request_env** (*ask_sdk_model.requestEnvelope.RequestEnvelope*) – deserialized request envelope instance of the input POST request

**Raises** [VerificationException](#) if verification fails

```python
class ask_sdk_webservice_support.verifier.RequestVerifier:
    signature_cert_chain_url_key='SignatureCertChainUrl',
    signature_key='Signature',
    padding=<cryptography.hazmat.primitives.asymmetric.padding.PKCS1v15 object>,
    hash_algorithm=<cryptography.hazmat.primitives.hashes.SHA1 object>)
```

Bases: `ask_sdk_webservice_support.verifier.AbstractVerifier`

Verifier that performs request signature verification.

This is a concrete implementation of `AbstractVerifier` class, handling the request signature verification of the input request. This verifier uses the Cryptography module x509 functions to validate the signature chain in the input request. The verification follows the mechanism explained here: [https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html#checking-the-signature-of-the-request](https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html#checking-the-signature-of-the-request)

The constructor takes the header key names for retrieving Signature Certificate Chain and Signature. They are defaulted to the header names present in the `ask_sdk_webservice_support.verifier_constants`. Additionally, one can also provide the Padding and the Hash Algorithm function that is used to verify the input body.

The verify method retrieves the Signature Certificate Chain URL, validates the URL, retrieves the chain from the URL, validates the signing certificate, extract the public key, base64 decode the Signature and verifies if the hash value of the request body matches with the decrypted signature.

```python
verify(headers, serialized_request_env, deserialized_request_env)
```

Verify if the input request signature and the body matches.

The verify method retrieves the Signature Certificate Chain URL, validates the URL, retrieves the chain from the URL, validates the signing certificate, extract the public key, base64 decode the Signature and verifies if the hash value of the request body matches with the decrypted signature.

**Parameters**

• **headers** (*Dict[str, Any]*) – headers of the input POST request
• **serialized_request_env** (*str*) – raw request envelope in the input POST request
• **deserialized_request_env** (*ask_sdk_model.requestEnvelope.RequestEnvelope*) – deserialized request envelope instance of the input POST request

**Raises** [VerificationException](#) if headers doesn’t exist or verification fails

```python
class ask_sdk_webservice_support.verifier.TimestampVerifier (tolerance_in_millis=30000)
```

Bases: `ask_sdk_webservice_support.verifier.AbstractVerifier`

Verifier that performs request timestamp verification.

This is a concrete implementation of `AbstractVerifier` class, handling the request timestamp verification of the input request. The verification follows the mechanism explained here: [https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html#timestamp](https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html#timestamp)
The constructor takes the tolerance value in milliseconds, that is the maximum tolerance limit the input request can have, with the current timestamp.

The verify method retrieves the request timestamp and check if it falls in the limit set by the tolerance.

```python
verify(headers, serialized_request_env, deserialized_request_env)
```

Verify if the input request timestamp is in tolerated limits.

The verify method retrieves the request timestamp and check if it falls in the limit set by the tolerance, by checking with the current timestamp in UTC.

**Parameters**

- **headers** (`Dict[str, Any]`) – headers of the input POST request
- **serialized_request_env** (`str`) – raw request envelope in the input POST request
- **deserialized_request_env** (`ask_sdk_model.request_envelope.RequestEnvelope`) – deserialized request envelope instance of the input POST request

**Raises** `VerificationException` if difference between local timestamp and input request timestamp is more than specific tolerance limit

---

**ask_sdk_webservice_support.webservice_handler module**

```python
class ask_sdk_webservice_support.webservice_handler.WebserviceSkillHandler(skill, verify_signature=True, verify_timestamp=True, verifiers=None)
```

Skill Handler for skill as webservice.

This class can be used by skill developers when they want their skills to be deployed as a web service, rather than using AWS Lambda.

The class constructor takes in a custom skill instance that is used for routing the input request. The boolean `verify_signature` variable configures if the request signature is verified for each input request. The boolean `verify_timestamp` configures if the request timestamp is verified for each input request. Additionally, an optional list of verifiers can also be provided, to be applied on the input request.

The `verify_request_and_dispatch` method provides the dispatch functionality that can be used as an entry point for skill invocation as web service.

```python
verify_request_and_dispatch(http_request_headers, http_request_body)
```

Entry point for webservice skill invocation.

This method takes in the input request headers and request body, handles the deserialization of the input request to the `ask_sdk_model.request_envelope.RequestEnvelope` object, run the input through registered verifiers, invoke the skill and return the serialized response from the skill invocation.

**Parameters**

- **http_request_headers** (`Dict[str, Any]`) – Request headers of the input request to the webservice
- **http_request_body** (`str`) – Raw request body of the input request to the webservice
**Returns**  Serialized response object returned by the skill instance, when invoked with the input request

**Return type**  str

** Raises**  ask_sdk_core.exceptions.AskSdkException when skill deserialization, verification, invocation or serialization fails

---

**flask-ask-sdk package**

**flask_ask_sdk.skill_adapter module**

```python
class flask_ask_sdk.skill_adapter.SkillAdapter(skill, skill_id, verifiers=None, app=None):

Bases: object

Provides a base interface to register skill and dispatch the request.

The class constructor takes a ask_sdk_core.skill.CustomSkill instance, the skill id, an optional list of ask_sdk_webservice_support.verifier.AbstractVerifier instances and an optional flask application. One can also use the init_app() method, to pass in a flask.Flask application instance, to instantiate the config values and the webservice handler.

The dispatch_request() function can be used to map the input request to the skill invocation. The register() function can also be used alternatively, to register the dispatch_request() function to the provided route.

By default, the ask_sdk_webservice_support.verifier.RequestVerifier and ask_sdk_webservice_support.verifier.TimestampVerifier instances are added to the skill verifier list. To disable this, set the VERIFY_SIGNATURE_APP_CONFIG and VERIFY_TIMESTAMP_APP_CONFIG app configuration values to False.

An instance of the extension is added to the application extensions mapping, under the key EXTENSION_NAME. Since multiple skills can be configured on different routes in the same application, through multiple extension instances, each extension is added as a skill id mapping under the app extensions EXTENSION_NAME dictionary.

For example, to use this class with a skill created using ask-sdk-core:

```python
from flask import Flask
from ask_sdk_core.skill_builder import SkillBuilder
from flask_ask_sdk.skill_adapter import SkillAdapter

app = Flask(__name__)
skill_builder = SkillBuilder()
# Register your intent handlers to skill_builder object

skill_adapter = SkillAdapter(
    skill=skill_builder.create(), skill_id=<SKILL_ID>, app=app)

@app.route("/"):  
def invoke_skill:
    return skill_adapter.dispatch_request()

Alternatively, you can also use the register method:
```
```python
def init_app(app):
    # Register the extension on the given Flask application.
    # Use this function only when no Flask application was provided in the app keyword argument to the
    # constructor of this class.
    # The function sets True defaults for VERIFY_SIGNATURE_APP_CONFIG and
    # VERIFY_TIMESTAMP_APP_CONFIG configurations. It adds the skill id: self instance mapping to the
    # application extensions, and creates a ask_sdk_webservice_support.webservice_handler.
    # WebserviceHandler instance, for request verification and dispatch.
    # Parameters app (flask.Flask) – A flask.Flask application instance
    # Return type None

def dispatch_request():
    # Method that handles request verification and routing.
    # This method can be used as a function to register on the URL rule. The request is verified through the
    # registered list of verifiers, before invoking the request handlers. The method returns a JSON response for
    # the Alexa service to respond to the request.
    # Returns The skill response for the input request
    # Return type flask.Response
    # Raises werkzeug.exceptions.MethodNotAllowed if the method is invoked for other
    # than HTTP POST request. werkzeug.exceptions.BadRequest if the verification
    # fails. werkzeug.exceptions.InternalServerError for any internal exception.

def register(app, route, endpoint=None):
    # Method to register the routing on the app at provided route.
    # This is a utility method, that can be used for registering the dispatch_request on the provided
    # flask.Flask application at the provided URL route.
    # Parameters
    # • app (flask.Flask) – A flask.Flask application instance
    # • route (str) – The URL rule where the skill dispatch has to be registered
    # • endpoint (str) – The endpoint for the registered URL rule. This can be used to set
    #   multiple skill endpoints on same app.
    # Return type None
    # Raises TypeError if app or route is not provided or is of an invalid type
```

The `init_app` function registers the extension on the given Flask application. It is used when no Flask application is provided as a keyword argument to the class constructor.

The `dispatch_request` method handles request verification and routing. It verifies the request through registered verifiers before invoking the request handlers. It returns a JSON response for the Alexa service to respond to the request.

The `register` function registers the routing on the Flask application at the provided route. It is a utility method used for registering the dispatch_request on the Flask application at the provided URL route.

### Parameters
- **app** (flask.Flask): A flask.Flask application instance
- **route** (str): The URL rule where the skill dispatch has to be registered
- **endpoint** (str): The endpoint for the registered URL rule. This can be used to set multiple skill endpoints on the same app.

### Returns
- None

### Raises
- TypeError: If `app` or `route` is not provided or is of an invalid type.
**flask_ask_sdk.skill_adapter.**

**EXTENSION_NAME = 'ASK_SDK_SKILL_ADAPTER'**

Extension name used for saving extension instance in app.extensions

---

**django-ask-sdk package**

**django_ask_sdk.skill_adapter module**

**class** `django_ask_sdk.skill_adapter.SkillAdapter` (skill, verify_signature=True, verify_timestamp=True, verifiers=None)

**Bases:** `django.views.generic.base.View`

Provides a base interface to register skill and dispatch the request.

The class constructor takes a `ask_sdk_core.skill.CustomSkill` instance, an optional `verify_request` boolean, an optional `verify_timestamp` boolean and an optional list of `ask_sdk_webservice_support.verifier.AbstractVerifier` instances.

The `post()` function is the only available method on the view, that intakes the input POST request from Alexa, verifies the request and dispatch it to the skill.

By default, the `ask_sdk_webservice_support.verifier.RequestVerifier` and `ask_sdk_webservice_support.verifier.TimestampVerifier` instances are added to the skill verifier list. To disable this, set the `verify_request` and `verify_timestamp` input arguments to `False` respectively.

For example, if you developed a skill using an instance of `ask_sdk_core.skill_builder.SkillBuilder` or its subclasses, then to register it as an endpoint in your django app example, you can add the following in `example.urls.py`:

```python
import skill
from django_ask_sdk.skill_response import SkillAdapter

view = SkillAdapter.as_view(skill=skill.sb.create())

urlpatterns = [
    path("/myskill", view, name='index')
]
```

---

**skill = None**

**verify_signature = None**

**verify_timestamp = None**

**verifiers = None**

**dispatch** *(request, *args, **kwargs)*

Inspect the HTTP method and delegate to the view method.

This is the default implementation of the `django.views.View` method, which will inspect the HTTP method in the input request and delegate it to the corresponding method in the view. The only allowed method on this view is `post`.

**Parameters** `request` *(django.http.HttpRequest)* – The input request sent to the view

**Returns** The response from the view

**Return type** `django.http.HttpResponse`

---

2.14. Hosting Skills as Webservice

```python
post(request, *args, **kwargs)
```
The method that handles HTTP POST request on the view.

This method is called when the view receives a HTTP POST request, which is generally the request sent from Alexa during skill invocation. The request is verified through the registered list of verifiers, before invoking the request handlers. The method returns a `django.http.JsonResponse` in case of successful skill invocation.

Parameters

- `request` (`django.http.HttpRequest`) – The input request sent by Alexa to the skill

Returns

The response from the skill to Alexa

Return type

`django.http.JsonResponse`

Raises

- `django.http.HttpResponseBadRequest` if the request verification fails.

```python
classmethod as_view(**initkwargs)
```
Main entry point for a request-response process.

```python
http_method_names = ['get', 'post', 'put', 'patch', 'delete', 'head', 'options', 'trace']
```

```python
http_method_not_allowed(request, *args, **kwargs)
```
Handle responding to requests for the OPTIONS HTTP verb.

```python
options(request, *args, **kwargs)
```

```python
setup(request, *args, **kwargs)
```
Initialize attributes shared by all view methods.

**2.14.6 Models**

The SDK works on model classes rather than native Alexa JSON requests and responses. These model classes are generated using the Request, Response JSON schemas from the developer docs. The source code for the model classes can be found here.

**Subpackages**

`ask_sdk_model.canfulfill` package

**Submodules**

`django_ask_sdk.skill_adapter`:

- `SIGNATURE_CERT_CHAIN_URL_KEY = 'HTTP_SIGNATURECERTCHAINURL'`
  
  Signature Certificate Chain URL header key in Django HTTP Headers. This is different from the header key provided by Alexa, because of Django’s HTTP Meta headers.

- `SIGNATURE_KEY = 'HTTP_SIGNATURE'`
  
  Signature header key in Django HTTP Headers. This is different from the header key provided by Alexa, because of Django’s HTTP Meta headers.
Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.
For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.canfulfill.can_fulfill_intent module

class ask_sdk_model.canfulfill.can_fulfill_intent.CanFulfillIntent(can_fulfill=None, slots=None)
Bases: object
CanFulfillIntent represents the response to canFulfillIntentRequest includes the details about whether the skill can understand and fulfill the intent request with detected slots.

Parameters

- **can_fulfill**
  ((optional) ask_sdk_model.canfulfill.can_fulfill_intent_values.CanFulfillIntentValues) -

- **slots**
  ((optional) dict(str, ask_sdk_model.canfulfill.can_fulfill_slot.CanFulfillSlot)) – A map that represents skill’s detailed response to each detected slot within the intent such as if skill can understand and fulfill the detected slot. This supplements the overall canFulfillIntent response and help Alexa make better ranking and arbitration decisions. The key is the name of the slot. The value is an object of type CanFulfillSlot.

to_dict()
  Returns the model properties as a dict

to_str()
  Returns the string representation of the model

ask_sdk_model.canfulfill.can_fulfill_intent_request module

class ask_sdk_model.canfulfill.can_fulfill_intent_request.CanFulfillIntentRequest(request_id=None, times-tamp=None, locale=None, dia-log_state=None, in-tent=None)
Bases: ask_sdk_model.request.Request
An object that represents a request made to skill to query whether the skill can understand and fulfill the intent request with detected slots, before actually asking the skill to take action. Skill should be aware this is not to actually take action, skill should handle this request without causing side-effect, skill should not modify some state outside its scope or has an observable interaction with its calling functions or the outside world besides returning a value, such as playing sound, turning on/off lights, committing a transaction or a charge.

Parameters
• request_id ((optional) str) – Represents the unique identifier for the specific request.

• timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• dialog_state ((optional) ask_sdk_model.dialog_state.DialogState) –

• intent ((optional) ask_sdk_model.intent.Intent) –

    to_dict()  
    Returns the model properties as a dict

    to_str()  
    Returns the string representation of the model

ask_sdk_model.canfulfill.can_fulfill_intent_values module

class ask_sdk_model.canfulfill.can_fulfill_intent_values.CanFulfillIntentValues  
Bases: enum.Enum

Overall if skill can understand and fulfill the intent with detected slots. Respond YES when skill understands all slots, can fulfill all slots, and can fulfill the request in its entirety. Respond NO when skill either cannot understand the intent, cannot understand all the slots, or cannot fulfill all the slots. Respond MAYBE when skill can understand the intent, can partially or fully understand the slots, and can partially or fully fulfill the slots. The only cases where should respond MAYBE is when skill partially understand the request and can potentially complete the request if skill get more data, either through callbacks or through a multi-turn conversation with the user.

Allowed enum values: [YES, NO, MAYBE]

    to_dict()  
    Returns the model properties as a dict

    to_str()  
    Returns the string representation of the model

ask_sdk_model.canfulfill.can_fulfill_slot module

class ask_sdk_model.canfulfill.can_fulfill_slot.CanFulfillSlot (can_understand=None,  
                 can_fulfill=None)  
Bases: object

This represents skill’s capability to understand and fulfill each detected slot.

    Parameters

    • can_understand ((optional) ask_sdk_model.canfulfill.
        can_understand_slot_values.CanUnderstandSlotValues) –

    • can_fulfill ((optional) ask_sdk_model.canfulfill.
        can_fulfill_slot_values.CanFulfillSlotValues) –

    to_dict()  
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.canfulfill.can_fulfill_slot_values module

class ask_sdk_model.canfulfill.can_fulfill_slot_values.CanFulfillSlotValues
    Bases: enum.Enum
    This field indicates whether skill can fulfill relevant action for the slot, that has been partially or fully understood. The definition of fulfilling the slot is dependent on skill and skill is required to have logic in place to determine whether a slot value can be fulfilled in the context of skill or not. Return YES if Skill can certainly fulfill the relevant action for this slot value. Return NO if skill cannot fulfill the relevant action for this slot value. For specific recommendations to set the value refer to the developer docs for more details.
    Allowed enum values: [YES, NO]
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.canfulfill.can_understand_slot_values module

class ask_sdk_model.canfulfill.can_understand_slot_values.CanUnderstandSlotValues
    Bases: enum.Enum
    This field indicates whether skill has understood the slot value. In most typical cases, skills will do some form of entity resolution by looking up a catalog or list to determine whether they recognize the slot or not. Return YES if skill have a perfect match or high confidence match (for eg. synonyms) with catalog or list maintained by skill. Return NO if skill cannot understand or recognize the slot value. Return MAYBE if skill have partial confidence or partial match. This will be true when the slot value doesn’t exist as is, in the catalog, but a variation or a fuzzy match may exist. For specific recommendations to set the value refer to the developer docs for more details.
    Allowed enum values: [YES, NO, MAYBE]
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.dialog package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.
ask_sdk_model.dialog.confirm_intent_directive module

class ask_sdk_model.dialog.confirm_intent_directive.ConfirmIntentDirective(optimized_intent=None)
    Bases: ask_sdk_model.directive.Directive
    Parameters
    * updated_intent (optional) ask_sdk_model.intent.Intent -
    `to_dict()`
        Returns the model properties as a dict
    `to_str()`
        Returns the string representation of the model

ask_sdk_model.dialog.confirm_slot_directive module

class ask_sdk_model.dialog.confirm_slot_directive.ConfirmSlotDirective(optimized_intent=None, slot_to_confirm=None)
    Bases: ask_sdk_model.directive.Directive
    Parameters
    * updated_intent (optional) ask_sdk_model.intent.Intent -
    * slot_to_confirm (optional) str -
    `to_dict()`
        Returns the model properties as a dict
    `to_str()`
        Returns the string representation of the model

ask_sdk_model.dialog.delegate_directive module

class ask_sdk_model.dialog.delegate_directive.DelegateDirective(optimized_intent=None)
    Bases: ask_sdk_model.directive.Directive
    Parameters
    * updated_intent (optional) ask_sdk_model.intent.Intent -
    `to_dict()`
        Returns the model properties as a dict
    `to_str()`
        Returns the string representation of the model

ask_sdk_model.dialog.elicit_slot_directive module

class ask_sdk_model.dialog.elicit_slot_directive.ElicitSlotDirective(optimized_intent=None, slot_to_elicit=None)
    Bases: ask_sdk_model.directive.Directive
    Parameters
    * updated_intent (optional) ask_sdk_model.intent.Intent -
    * slot_to_elicit (optional) str -
    `to_dict()`
        Returns the model properties as a dict
to_str()
   Returns the string representation of the model

ask_sdk_model.events package

Subpackages

ask_sdk_model.events.skillevents package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.
For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.events.skillevents.account_linked_body module

class ask_sdk_model.events.skillevents.account_linked_body.AccountLinkedBody (access_token=None)
   Bases: object

   Parameters access_token ((optional) str)-

   to_dict()
      Returns the model properties as a dict

   to_str()
      Returns the string representation of the model

ask_sdk_model.events.skillevents.account_linked_request module

class ask_sdk_model.events.skillevents.account_linked_request.AccountLinkedRequest (request_id=None,
timestamp=None,
locale=None,
body=None,
event_creation_time=None,
event_publishing_time=None)
   Bases: ask_sdk_model.request.Request

This event indicates that a customer has linked an account in a third-party application with the Alexa app. This event is useful for an application that support out-of-session (non-voice) user interactions so that this application can be notified when the internal customer can be associated with the Alexa customer. This event is required for many applications that synchronize customer Alexa lists with application lists. During the account linking process, the Alexa app directs the user to the skill website where the customer logs in. When the customer logs in, the skill then provides an access token and a consent token to Alexa. The event includes the same access token and consent token.

Parameters
• **request_id** *(optional)* `str* – Represents the unique identifier for the specific request.

• **timestamp** *(optional)* `datetime* – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• **locale** *(optional)* `str* – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• **body** *(optional)* `ask_sdk_model.events.skillevents.account_linked_body.AccountLinkedBody* –

• **event_creation_time** *(optional)* `datetime* –

• **event_publishing_time** *(optional)* `datetime* –

    `to_dict()`
    Returns the model properties as a dict

    `to_str()`
    Returns the string representation of the model

**ask_sdk_model.events.skillevents.permission module**

class `ask_sdk_model.events.skillevents.permission.Permission`(scope=None)
    Bases: `object`

    Parameters `scope` *(optional)* `str* –

    `to_dict()`
    Returns the model properties as a dict

    `to_str()`
    Returns the string representation of the model

**ask_sdk_model.events.skillevents.permission_accepted_request module**

class `ask_sdk_model.events.skillevents.permission_accepted_request.PermissionAcceptedRequest`(request_id=None, timestamp=None, locale=None, body=None, event_creation_time=None, event_publishing_time=None)
    Bases: `ask_sdk_model.request.Request`

    Parameters

    • **request_id** *(optional)* `str* – Represents the unique identifier for the specific request.

    • **timestamp** *(optional)* `datetime* – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

    • **locale** *(optional)* `str* – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
• body 
  ((optional) ask_sdk_model.events.skillevents.
  permission_body.PermissionBody)

- event_creation_time ((optional) datetime)
- event_publishing_time ((optional) datetime)

to_dict()
  Returns the model properties as a dict

to_str()
  Returns the string representation of the model

ask_sdk_model.events.skillevents.permission_body module

class ask_sdk_model.events.skillevents.permission_body.PermissionBody (accepted_permissions=None)
  Bases: object

Parameters
  accepted_permissions ((optional) list[ask_sdk_model.
  events.skillevents.permission.Permission])

  to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model

ask_sdk_model.events.skillevents.permission_changed_request module

class ask_sdk_model.events.skillevents.permission_changed_request.PermissionChangedRequest
  Bases: ask_sdk_model.request.Request

Parameters

- request_id ((optional) str) – Represents the unique identifier for the specific
  request.

- timestamp ((optional) datetime) – Provides the date and time when Alexa sent
  the request as an ISO 8601 formatted string. Used to verify the request when hosting
  your skill as a web service.

- locale ((optional) str) – A string indicating the user’s locale. For example: en-
  US. This value is only provided with certain request types.

- body 
  ((optional) ask_sdk_model.events.skillevents.
  permission_body.PermissionBody)

- event_creation_time ((optional) datetime)
- event_publishing_time ((optional) datetime)

  to_dict()
  Returns the model properties as a dict
def to_str() -> str:
    """Returns the string representation of the model"""

ask_sdk_model.events.skillevents.skill_disabled_request module

class ask_sdk_model.events.skillevents.skill_disabled_request.SkillDisabledRequest(request_id=None, timestamp=None, locale=None, event_creation_time=None, event_publishing_time=None):
    Bases: ask_sdk_model.request.Request

    Parameters
    - request_id (optional, str) – Represents the unique identifier for the specific request.
    - timestamp (optional, datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
    - locale (optional, str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
    - event_creation_time (optional, datetime) –
    - event_publishing_time (optional, datetime) –

def to_dict() -> dict:
    """Returns the model properties as a dict"""

def to_str() -> str:
    """Returns the string representation of the model"""

ask_sdk_model.events.skillevents.skill_enabled_request module

class ask_sdk_model.events.skillevents.skill_enabled_request.SkillEnabledRequest(request_id=None, timestamp=None, locale=None, event_creation_time=None, event_publishing_time=None):
    Bases: ask_sdk_model.request.Request

    Parameters
    - request_id (optional, str) – Represents the unique identifier for the specific request.
    - timestamp (optional, datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
    - locale (optional, str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
• `event_creation_time` *(optional) datetime* –
• `event_publishing_time` *(optional) datetime* –

`to_dict()`
Returns the model properties as a dict

`to_str()`
Returns the string representation of the model

`ask_sdk_model.interfaces` package

Subpackages

`ask_sdk_model.interfaces.alexa` package

Subpackages

`ask_sdk_model.interfaces.alexa.presentation` package

Subpackages

`ask_sdk_model.interfaces.alexa.presentation.apl` package

Submodules

**Note**: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with class `C`, you can do `from a import C` instead of `from a.b import C`.

`ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface` module

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface.AlexaPresentationAplInterface
    Bases: object

    Parameters
    runtime ((optional) ask_sdk_model.interfaces.alexa.presentation.apl.runtime.Runtime) –

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model
```

`ask_sdk_model.interfaces.alexa.presentation.apl.align` module

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.align.Align
    Bases: enum.Enum
```

2.14. Hosting Skills as Webservice
The alignment of the item after scrolling. Defaults to visible.
Allowed enum values: [center, first, last, visible]

to_dict()
   Returns the model properties as a dict

to_str()
   Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.auto_page_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.auto_page_command.AutoPageCommand
delay=None,
description=None,
when=None,
component_id=None,
count=None,
duration=None
)


Automatically progress through a series of pages displayed in a Pager component. The AutoPage command finishes after the last page has been displayed for the requested time period.

Parameters

- delay (optional int) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- description (optional str) – A user-provided description of this command.
- when (optional bool) – If false, the execution of the command is skipped. Defaults to true.
- component_id (optional str) – The id of the Pager component.
- count (optional int) – Number of pages to display. Defaults to all of them.
- duration (optional int) – Time to wait between pages (in milliseconds). Defaults to 0.

to_dict()
   Returns the model properties as a dict

to_str()
   Returns the string representation of the model
ask_sdk_model.interfaces.alexa.presentation.apl.command module

class ask_sdk_model.interfaces.alexa.presentation.apl.command.Command(object_type=None, delay=None, description=None, when=None)

Bases: object

A message that can change the visual or audio presentation of the content on the screen.

Parameters

- **object_type** *(optional) str* – Defines the command type and dictates which properties must/can be included.
- **delay** *(optional) int* – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- **description** *(optional) str* – A user-provided description of this command.
- **when** *(optional) bool* – If false, the execution of the command is skipped. Defaults to true.

**Note:** This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

**SetPage:** ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command.SetPageCommand,

**ControlMedia:** ask_sdk_model.interfaces.alexa.presentation.apl.control_media_command.ControlMediaCommand,

**Sequential:** ask_sdk_model.interfaces.alexa.presentation.apl.sequential_command.SequentialCommand,

**setState:** ask_sdk_model.interfaces.alexa.presentation.apl.set_state_command.SetStateCommand,

**SpeakItem:** ask_sdk_model.interfaces.alexa.presentation.apl.speak_item_command.SpeakItemCommand,

**AutoPage:** ask_sdk_model.interfaces.alexa.presentation.apl.auto_page_command.AutoPageCommand,

**Parallel:** ask_sdk_model.interfaces.alexa.presentation.apl.parallel_command.ParallelCommand,

**OpenURL:** ask_sdk_model.interfaces.alexa.presentation.apl.open_url_command.OpenUrlCommand,
PlayMedia: `ask_sdk_model.interfaces.alexa.presentation.apl.play_media_command.PlayMediaCommand`,

ClearFocus: `ask_sdk_model.interfaces.alexa.presentation.apl.clear_focus_command.ClearFocusCommand`,

ScrollToIndex: `ask_sdk_model.interfaces.alexa.presentation.apl.scroll_to_index_command.ScrollToIndexCommand`,

Scroll: `ask_sdk_model.interfaces.alexa.presentation.apl.scroll_command.ScrollCommand`,

Idle: `ask_sdk_model.interfaces.alexa.presentation.apl.idle_command.IdleCommand`,

AnimateItem: `ask_sdk_model.interfaces.alexa.presentation.apl.animate_item_command.AnimateItemCommand`,

SetValue: `ask_sdk_model.interfaces.alexa.presentation.apl.set_value_command.SetValueCommand`,

SetFocus: `ask_sdk_model.interfaces.alexa.presentation.apl.set_focus_command.SetFocusCommand`,

SendEvent: `ask_sdk_model.interfaces.alexa.presentation.apl.send_event_command.SendEventCommand`,

SpeakList: `ask_sdk_model.interfaces.alexa.presentation.apl.speak_list_command.SpeakListCommand`

```
classmethod get_real_child_model(data)
    Returns the real base class specified by the discriminator

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

`ask_sdk_model.interfaces.alexa.presentation.apl.execute_commands_directive` module


    Alexa.Presentation.APL.ExecuteCommands directive used to send APL commands to a device.

    Parameters
• **commands**  
  `((optional) list[ask_sdk_model.interfaces.alexa.presentation.apl.command.Command])` – List of Command instances

• **token**  
  `((optional) str)` – A skill defined token, unique for each presentation. Must match the token provided by the skill in the RenderDocument directive used to render the original APL document.

```python

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.alexa.presentation.apl.highlight_mode module**

```python

class ask_sdk_model.interfaces.alexa.presentation.apl.highlight_mode.HighlightMode
    Bases: enum.Enum

    How highlighting is applied: on a line-by-line basis, or to the entire block. Defaults to block.
    Allowed enum values: [block, line]

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.alexa.presentation.apl.position module**

```python

class ask_sdk_model.interfaces.alexa.presentation.apl.position.Position
    Bases: enum.Enum

    Whether the value is a relative or absolute offset. Defaults to absolute.
    Allowed enum values: [absolute, relative]

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.alexa.presentation.apl.render_document_directive module**

```python

    Bases: ask_sdk_model.directive.Directive

    Parameters
    
    • **token**  
      `((optional) str)` – A unique identifier for the presentation.
```
- `document ((optional) dict(str, object))` – The APL document that the devices need to render a presentation.
- `datasources ((optional) dict(str, object))` – Data sources to bind to the document when rendering.
- `packages ((optional) list[object])` – A list of packages including layouts, styles, and images etc.

```python
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model
```

### ask_sdk_model.interfaces.alexa.presentation.apl.runtime module

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.runtime.Runtime (max_version=None)
    Bases: object

    Contains the runtime information for the interface.

    Parameters max_version ((optional) str) – Maximum APL version supported by the runtime.

to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model
```

### ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command module

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command.SetPageCommand (delay=None,
    description=None, when=None, component_id=None, position=None, value=None)

    Change the page displayed in a Pager component. The SetPage command finishes when the item is fully in view.

    Parameters

    - delay ((optional) int) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
    - description ((optional) str) – A user-provided description of this command.
    - when ((optional) bool) – If false, the execution of the command is skipped. Defaults to true.
```

112 Chapter 2. SDK Features
• **component_id** ((optional) *str*) – The id of the Pager component.

• **position** ((optional) `ask_sdk_model.interfaces.alexa.presentation.apl.position.Position`) –

• **value** ((optional) *int*) – The distance to move. May be an absolute value or a relative value.

`to_dict()`

Returns the model properties as a dict

`to_str()`

Returns the string representation of the model

### ask_sdk_model.interfaces.alexa.presentation.apl.speak_item_command module

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.speak_item_command.SpeakItemCommand(
    delay=None,
    description=None,
    when=None,
    align=None,
    component_id=None,
    highlight_mode=None,
    minimum_dwell_time=None
)
```

**Bases:** `ask_sdk_model.interfaces.alexa.presentation.apl.command.Command`

Reads the contents of a single item on the screen. By default the item will be scrolled into view if it is not currently visible.

**Parameters**

- **delay** ((optional) *int*) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.

- **description** ((optional) *str*) – A user-provided description of this command.

- **when** ((optional) *bool*) – If false, the execution of the command is skipped. Defaults to true.

- **align** ((optional) `ask_sdk_model.interfaces.alexa.presentation.apl.align.Align`) –

- **component_id** ((optional) *str*) – The id of the component to speak.

- **highlight_mode** ((optional) `ask_sdk_model.interfaces.alexa.presentation.apl.highlight_mode.HighlightMode`) –

- **minimum_dwell_time** ((optional) *int*) – The minimum number of milliseconds that an item should be highlighted for. Defaults to 0.

`to_dict()`

Returns the model properties as a dict

`to_str()`

Returns the string representation of the model

### 2.14. Hosting Skills as Webservice

113
ask_sdk_model.interfaces.alexa.presentation.apl.user_event module

class ask_sdk_model.interfaces.alexa.presentation.apl.user_event.UserEvent

Bases: ask_sdk_model.request.Request

Parameters

- **request_id** *(optional) str* – Represents the unique identifier for the specific request.

- **timestamp** *(optional) datetime* – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

- **locale** *(optional) str* – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

- **token** *(optional) str* – A unique token for the active presentation.

- **arguments** *(optional) list[object]* – The array of argument data to pass to Alexa.

- **source** *(optional) object* – Meta-information about what caused the event to be generated.

- **components** *(optional) object* – Components associated with the request.

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.audio_track module

class ask_sdk_model.interfaces.alexa.presentation.apl.audio_track.AudioTrack

Bases: enum.Enum

The audio track to play on. Defaults to “foreground”

Allowed enum values: [foreground]

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model
ask_sdk_model.interfaces.alexa.presentation.apl.component_state module

class ask_sdk_model.interfaces.alexa.presentation.apl.component_state.ComponentState
    Bases: enum.Enum
    Component state.
    Allowed enum values: [checked, disabled, focused]
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.control_media_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.control_media_command.ControlMediaCommand
    delay=None,
    description=None,
    when=None,
    command=None,
    component_id=None,
    value=None
    Control a media player to play, pause, change tracks, or perform some other common action.
    Parameters
        • delay ((optional) int) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
        • description ((optional) str) – A user-provided description of this command.
        • when ((optional) bool) – If false, the execution of the command is skipped. Defaults to true.
        • command ((optional) ask_sdk_model.interfaces.alexa.presentation.apl.media_command_type.MediaCommandType) – The command to issue on the media player
        • component_id ((optional) str) – The name of the media playing component
        • value ((optional) int) – Optional data value
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model
ask_sdk_model.interfaces.alexa.presentation.apl.idle_command module

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.idle_command.IdleCommand:

    The idle command does nothing. It may be a placeholder or used to insert a calculated delay in a longer series of commands.

    Parameters
    • delay ((optional) int) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
    • description ((optional) str) – A user-provided description of this command.
    • when ((optional) bool) – If false, the execution of the command is skipped. Defaults to true.

    to_dict() -> Returns the model properties as a dict
    to_str() -> Returns the string representation of the model
```

ask_sdk_model.interfaces.alexa.presentation.apl.media_command_type module

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.media_command_type.MediaCommandType:
    Bases: enum.Enum

    The command enumerated value is the operation that should be performed on the media player.

    Allowed enum values: [play, pause, next, previous, rewind, seek, setTrack]

    to_dict() -> Returns the model properties as a dict
    to_str() -> Returns the string representation of the model
```

ask_sdk_model.interfaces.alexa.presentation.apl.parallel_command module

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.parallel_command.ParallelCommand:

    Execute a series of commands in parallel. The parallel command starts executing all child command simultaneously. The parallel command is considered finished when all of its child commands have finished. When the parallel command is terminated early, all currently executing commands are terminated.

    Parameters
```
• **delay** *(optional) int* – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.

• **description** *(optional) str* – A user-provided description of this command.

• **when** *(optional) bool* – If false, the execution of the command is skipped. Defaults to true.

• **commands** *(optional) list[ask_sdk_model.interfaces.alexa.presentation.apl.command.Command]* – An un-ordered array of commands to execute in parallel. Once all commands have finished executing the parallel command finishes. Please note that the delay of parallel command and the delay of each command are additive.

    **to_dict()**
    Returns the model properties as a dict

    **to_str()**
    Returns the string representation of the model

**ask_sdk_model.interfaces.alexa.presentation.apl.play_media_command module**

**class ask_sdk_model.interfaces.alexa.presentation.apl.play_media_command.PlayMediaCommand**(delay=None, description=None, when=None, audio_track=None, component_id=None, source=None)

Bases: *ask_sdk_model.interfaces.alexa.presentation.apl.command.Command*

Plays media on a media player (currently only a Video player; audio may be added in the future). The media may be on the background audio track or may be sequenced with speak directives.

**Parameters**

• **delay** *(optional) int* – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.

• **description** *(optional) str* – A user-provided description of this command.

• **when** *(optional) bool* – If false, the execution of the command is skipped. Defaults to true.

• **audio_track** *(optional)*

  ask_sdk_model.interfaces.alexa.presentation.apl.audio_track.AudioTrack – The command to issue on the media player

• **component_id** *(optional) str* – The name of the media playing component

• **source** *(optional) list*[ask_sdk_model.interfaces.alexa.presentation.apl.video_source.VideoSource]* – The media source

    **to_dict()**
    Returns the model properties as a dict

2.14. Hosting Skills as Webservice
to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.scroll_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.scroll_command.ScrollCommand(delay=None, description=None, when=None, distance=None, component_id=None)


Scroll a ScrollView or Sequence forward or backward by a number of pages. The Scroll command has the following properties in addition to the regular command properties.

Parameters

- **delay** *(optional) int* – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- **description** *(optional) str* – A user-provided description of this command.
- **when** *(optional) bool* – If false, the execution of the command is skipped. Defaults to true.
- **distance** *(optional) int* – The number of pages to scroll. Defaults to 1.
- **component_id** *(optional) str* – The id of the component.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.scroll_to_index_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.scroll_to_index_command.ScrollToIndexCommand(delay=None, description=None, when=None, align=None, component_id=None, index=None)


Scroll forward or backward through a ScrollView or Sequence to ensure that a particular child component is in view.
Parameters

- **delay**: (optional) int – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.

- **description**: (optional) str – A user-provided description of this command.

- **when**: (optional) bool – If false, the execution of the command is skipped. Defaults to true.

- **align**: (optional) ask_sdk_model.interfaces.alexa.presentation.apl.align.Align

- **component_id**: (optional) str – The id of the component.

- **index**: (optional) int – The 0-based index of the child to display.

### to_dict()

Returns the model properties as a dict

### to_str()

Returns the string representation of the model

**ask_sdk_model.interfaces.alexa.presentation.apl.send_event_command module**

```python
class ask_sdk_model.interfaces.alexa.presentation.apl.send_event_command.SendEventCommand(delay=None, description=None, when=None, arguments=None, components=None)
```


The SendEvent command allows the APL author to generate and send an event to Alexa.

#### Parameters

- **delay**: (optional) int – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.

- **description**: (optional) str – A user-provided description of this command.

- **when**: (optional) bool – If false, the execution of the command is skipped. Defaults to true.

- **arguments**: (optional) list[str] – An array of argument data to pass to Alexa.

- **components**: (optional) list[str] – An array of components to extract value data from and provide to Alexa.

### to_dict()

Returns the model properties as a dict

### to_str()

Returns the string representation of the model
A sequential command executes a series of commands in order. The sequential command executes the command list in order, waiting for the previous command to finish before executing the next. The sequential command is finished when all of its child commands have finished. When the Sequential command is terminated early, the currently executing command is terminated and no further commands are executed.

**Parameters**

- `delay (optional int)` – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- `description (optional str)` – A user-provided description of this command.
- `when (optional bool)` – If false, the execution of the command is skipped. Defaults to true.
- `catch (optional list[ask_sdk_model.interfaces.alexa.presentation.apl.command.Command])` – An ordered list of commands to execute if this sequence is prematurely terminated.
- `commands (optional list[ask_sdk_model.interfaces.alexa.presentation.apl.command.Command])` – An array of commands to execute. The commands execute in order; each command must finish before the next can begin. Please note that the delay of sequential command and the delay of the first command in the sequence are additive.
- `object_finally (optional list[ask_sdk_model.interfaces.alexa.presentation.apl.command.Command])` – An ordered list of commands to execute after the normal commands and the catch commands.
- `repeat_count (optional int)` – The number of times to repeat this series of commands. Defaults to 0. Negative values will be ignored. Note that the delay assigned to overall sequential command only applies the first time. For example, in the sample sequential command below the first SendEvent fires at 3000 milliseconds, the second at 5000, the first SendEvent fires again at 7000 milliseconds, and so forth. ```json{"type": "Sequential","delay": 1000,"repeatCount": 2,"commands": [{"type": "SendEvent","delay": 2000},{"type": "SendEvent","delay": 2000}]}```
to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.set_state_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.set_state_command.SetStateCommand(
    delay=None,
    description=None,
    when=None,
    component_id=None,
    state=None,
    value=None
)


The SetState command changes one of the component’s state settings. The SetState command can be used to change the checked, disabled, and focused states. The karaoke and pressed states may not be directly set; use the Select command or SpeakItem commands to change those states. Also, note that the focused state may only be set - it can’t be cleared.

Parameters

- delay (optional int) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- description (optional str) – A user-provided description of this command.
- when (optional bool) – If false, the execution of the command is skipped. Defaults to true.
- component_id (optional str) – The id of the component whose value should be set.
- state (optional ask_sdk_model.interfaces.alexa.presentation.apl.component_state.ComponentState) – The name of the state to set. Must be one of “checked”, “disabled”, and “focused”.
- value (optional bool) – The value to set on the property

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model
class ask_sdk_model.interfaces.alexa.presentation.apl.speak_list_command.SpeakListCommand(
    delay=None,
    description=None,
    when=None,
    align=None,
    component_id=None,
    count=None,
    minimum_dwell_time=None,
    start=None
)


Read the contents of a range of items inside a common container. Each item will scroll into view before speech. Each item should have a speech property, but it is not required.

Parameters

- **delay** ((optional) int) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- **description** ((optional) str) – A user-provided description of this command.
- **when** ((optional) bool) – If false, the execution of the command is skipped. Defaults to true.
- **align** ((optional) ask_sdk_model.interfaces.alexa.presentation.apl.align.Align) –
- **component_id** ((optional) str) – The id of the component to read.
- **count** ((optional) int) – The number of items to speak
- **minimum_dwell_time** ((optional) int) – The minimum number of milliseconds that an item will be highlighted for. Defaults to 0.
- **start** ((optional) int) – The 0-based index of the first item to speak

to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model
ask_sdk_model.interfaces.alexa.presentation.apl.set_value_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.set_value_command.SetValueCommand(
    delay=None,
    description=None,
    when=None,
    component_id=None,
    object_property=None,
    value=None
)


Change a dynamic property of a component without redrawing the screen.

Parameters

• delay ((optional) int) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
• description ((optional) str) – A user-provided description of this command.
• when ((optional) bool) – If false, the execution of the command is skipped. Defaults to true.
• component_id ((optional) str) – The id of the component whose value to set.
• object_property ((optional) str) – The name of the property to set.
• value ((optional) str) – The property value to set.

to_dict()  
Returns the model properties as a dict
to_str()  
Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.video_source module

class ask_sdk_model.interfaces.alexa.presentation.apl.video_source.VideoSource(
    description=None,
    duration=None,
    url=None,
    repeat_count=None,
    offset=None
)

Bases: object

The source property holds the video clip or sequence of video clips to play.

Parameters

• description ((optional) str) – Optional description of this source material
• duration ((optional) int) – Duration of time to play. If not set, defaults to the entire stream. Expressed in milliseconds.
• `url ((optional) str)`  – Media source material

• `repeat_count ((optional) int)`  – Number of times to loop the video. Defaults to 0.

• `offset ((optional) int)`  – Offset to start playing at in the stream (defaults to 0).

to_dict()
  Returns the model properties as a dict

to_str()
  Returns the string representation of the model

Module contents

Module contents

ask_sdk_model.interfaces.amazonpay package

Subpackages

ask_sdk_model.interfaces.amazonpay.model package

Subpackages

ask_sdk_model.interfaces.amazonpay.model.v1 package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package `a` has module `b` with class `C`, you can do `from a import C` instead of `from a.b import C`.
**ask_sdk_model.interfaces.amazonpay.model.v1.authorization_details module**

```python
class ask_sdk_model.interfaces.amazonpay.model.v1.authorization_details.AuthorizationDetails(
    amazon_authorization_id=None,
    authorization_reference_id=None,
    seller_authorization_note=None,
    authorization_amount=None,
    captured_amount=None,
    authorization_fee=None,
    id_list=None,
    creation_timestamp=None,
    expiration_timestamp=None,
    authorization_status=None,
    soft_decline=None,
    capture_now=None,
    soft_descriptor=None
)
```

**Bases:** object

This object encapsulates details about an Authorization object including the status, amount captured and fee charged.

**Parameters**

- `amazon_authorization_id ((optional) str)` – This is AmazonPay generated identifier for this authorization transaction.
- `authorization_reference_id ((optional) str)` – This is 3P seller’s identifier for this authorization transaction. This identifier must be unique for all of your authorization transactions.
- `seller_authorization_note ((optional) str)` – A description for the transaction that is included in emails to the user. Appears only when AuthorizeAndCapture is chosen.
- `authorization_amount ((optional) ask_sdk_model.interfaces.amazonpay.model.v1.price.Price)` –
- `captured_amount ((optional) ask_sdk_model.interfaces.amazonpay.model.v1.price.Price)` –

---

2.14. Hosting Skills as Webservice
• **authorization_fee** *(optional) ask_sdk_model.interfaces.amazonpay.model.v1.price.Price* –

• **id_list** *(optional) list[str]*) – list of AmazonCaptureId identifiers that have been requested on this Authorization object.

• **creation_timestamp** *(optional) datetime* – This is the time at which the authorization was created.

• **expiration_timestamp** *(optional) datetime* – This is the time at which the authorization expires.

• **authorization_status** *(optional) ask_sdk_model.interfaces.amazonpay.model.v1.authorization_status.AuthorizationStatus* –

  • **soft_decline** *(optional) bool* – This indicates whether an authorization resulted in a soft decline.

  • **capture_now** *(optional) bool* – This indicates whether a direct capture against the payment contract was specified.

  • **soft_descriptor** *(optional) str* – This is the description to be shown on the buyer’s payment instrument statement if AuthorizeAndCapture was chosen.

*to_dict()*
Returns the model properties as a dict

*to_str()*
Returns the string representation of the model

### ask_sdk_model.interfaces.amazonpay.model.v1.authorization_status module

#### class ask_sdk_model.interfaces.amazonpay.model.v1.authorization_status.AuthorizationStatus

**Bases:** object

Indicates the current status of an Authorization object, a Capture object, or a Refund object.

**Parameters**

• **state** *(optional) ask_sdk_model.interfaces.amazonpay.model.v1.state.State* –

• **reason_code** *(optional) str* – The reason that the Authorization object, Capture object, or Refund object is in the current state. For more information, see - https://pay.amazon.com/us/developer/documentation/apireference/201752950

• **reason_description** *(optional) str* – Reason description corresponding to the reason code

• **last_update_timestamp** *(optional) datetime* – A timestamp that indicates the time when the authorization, capture, or refund state was last updated. In ISO 8601 format

*to_dict()*
Returns the model properties as a dict
to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.authorize_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.authorize_attributes.AuthorizeAttributes

Bases: object

This is an object to set the attributes specified in the AuthorizeAttributes table. See the “AuthorizationDetails” section of the Amazon Pay API reference guide for details about this object.

Parameters

• authorization_reference_id ((optional) str) – This is 3P seller’s identifier for this authorization transaction. This identifier must be unique for all of your authorization transactions.

• authorization_amount ((optional) ask_sdk_model.interfaces.amazonpay.model.v1.price.Price) –

• transaction_timeout ((optional) int) – The maximum number of minutes allocated for the Authorize operation call to be processed. After this the authorization is automatically declined and you cannot capture funds against the authorization. The default value for Alexa transactions is 0. In order to speed up checkout time for voice users we recommend to not change this value.

• seller_authorization_note ((optional) str) – A description for the transaction that is included in emails to the user. Appears only when AuthorizeAndCapture is chosen.

• soft_descriptor ((optional) str) – The description to be shown on the user’s payment instrument statement if AuthorizeAndCapture is chosen. Format of soft descriptor sent to the payment processor is &quot;AMZ* &lt;soft descriptor specified here&gt;&quot;. Default is &quot;AMZ*&lt;SELLER_NAME&gt; amzn.com/pmts WA&quot;. Maximum length can be 16 characters.

to_dict()
Returns the model properties as a dict
to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_attributes.BillingAttributes

Bases: object

2.14. Hosting Skills as Webservice
The merchant can choose to set the attributes specified in the BillingAgreementAttributes.

Parameters

- **platform_id** *(optional)*  
  str  
  Represents the SellerId of the Solution Provider that developed the eCommerce platform. This value is only used by Solution Providers, for whom it is required. It should not be provided by merchants creating their own custom integration. Do not specify the SellerId of the merchant for this request parameter. If you are a merchant, do not enter a PlatformId.

- **seller_note** *(optional)*  
  str  
  Represents a description of the billing agreement that is displayed in emails to the buyer.

- **seller_billing_agreement_attributes** *(optional)*  
  ask_sdk_model.interfaces.amazonpay.model.v1.seller_billing_agreement_attributes.SellerBillingAgreementAttributes
  
  to_dict()  
  Returns the model properties as a dict

  to_str()  
  Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details module

class ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details.BillingAgreementDetails

Bases: object

The result attributes from successful SetupAmazonPay call.

Parameters

- **billing_agreement_id** *(optional)*  
  str  
  Billing agreement id which can be used for one time and recurring purchases

- **creation_timestamp** *(optional)*  
  datetime  
  Time at which billing agreement details created.

- **destination** *(optional)*  
  ask_sdk_model.interfaces.amazonpay.model.v1.destination.Destination  
  The default shipping address of the buyer. Returned if needAmazonShippingAddress is set to true.

- **checkout_language** *(optional)*  
  str  
  Merchant’s preferred language of checkout.

- **release_environment** *(optional)*  
  ask_sdk_model.interfaces.amazonpay.model.v1.release_environment.ReleaseEnvironment
  
  -
• billing_agreement_status ((optional) ask_sdk_model.
  interfaces.amazonpay.model.v1.billing_agreement_status.
  BillingAgreementStatus)–

  to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status module

class ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status.BillingAgreementStatus
  Bases: enum.Enum

  Indicates the current status of the billing agreement. For more information about the State and Reason-
  Code response elements, see Billing agreement states and reason codes - https://pay.amazon.com/us/developer/
  documentation/apireference/201752870

  Allowed enum values: [CANCELLED, CLOSED, DRAFT, OPEN, SUSPENDED]

  to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.destination module

class ask_sdk_model.interfaces.amazonpay.model.v1.destination.Destination
  name=Name, com-
  pany_name=None, ad-
  dress_line1=None, ad-
  dress_line2=None, ad-
  dress_line3=None, city=None,
  district_or_county=None, state_or_region=None, postal_code=None, coun-
  try_code=None, phone=None)

  Bases: object

  Destination object containing the details of an Address.

  Parameters

  • name ((optional) str) – The name or business name

  • company_name ((optional) str) – The company name

  • address_line1 ((optional) str) – The first line of the address. At least one Ad-
    dressLine (AddressLine1, AddressLine2, or AddressLine3) is required.
• **address_line2** (**optional** str) – The second line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.

• **address_line3** (**optional** str) – The third line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.

• **city** (**optional** str) – The city

• **district_or_county** (**optional** str) – The district or County

• **state_or_region** (**optional** str) – The state or region. This element is free text and can be either a 2-character code, fully spelled out, or abbreviated. Required. Note :- This response element is returned only in the U.S.

• **postal_code** (**optional** str) – The postal code.

• **country_code** (**optional** str) – The country code, in ISO 3166 format

• **phone** (**optional** str) – The phone number


\[\text{to_dict()}\]

Returns the model properties as a dict

\[\text{to_str()}\]

Returns the string representation of the model

**ask_sdk_model.interfaces.amazonpay.model.v1.payment_action module**

class ask_sdk_model.interfaces.amazonpay.model.v1.payment_action.PaymentAction

Bases: enum.Enum

• This is used to specify applicable payment action. * Authorize – you want to confirm the order and authorize a certain amount, but you do not want to capture at this time. * AuthorizeAndCapture – you want to confirm the order, authorize for the given amount, and capture the funds.

Allowed enum values: [Authorize, AuthorizeAndCapture]

\[\text{to_dict()}\]

Returns the model properties as a dict

\[\text{to_str()}\]

Returns the string representation of the model

**ask_sdk_model.interfaces.amazonpay.model.v1.price module**

class ask_sdk_model.interfaces.amazonpay.model.v1.price.Price (**amount=** None, **currency_code=** None)

Bases: object

This object specifies amount and currency authorized/captured.

Parameters

• **amount** (**optional** str) – Amount authorized/captured.

• **currency_code** (**optional** str) – Currency code for the amount.

\[\text{to_dict()}\]

Returns the model properties as a dict
**ask_sdk_model.interfaces.amazonpay.model.v1.provider_attributes module**

**class** ask_sdk_model.interfaces.amazonpay.model.v1.provider_attributes.ProviderAttributes

Bases: object

This is required only for Ecommerce provider (Solution provider) use cases.

**Parameters**

- **provider_id** *(optional) str* – Solution provider ID.
- **provider_credit_list** *(optional) list[ask_sdk_model.interfaces.amazonpay.model.v1.provider_credit.ProviderCredit]* – List of provider credit.

**to_dict**

Returns the model properties as a dict

**to_str**

Returns the string representation of the model

**ask_sdk_model.interfaces.amazonpay.model.v1.provider_credit module**

**class** ask_sdk_model.interfaces.amazonpay.model.v1.provider_credit.ProviderCredit

Bases: object

**Parameters**

- **provider_id** *(optional) str* – This is required only for Ecommerce provider (Solution provider) use cases.
- **credit** *(optional) ask_sdk_model.interfaces.amazonpay.model.v1.price.Price* –

**to_dict**

Returns the model properties as a dict

**to_str**

Returns the string representation of the model

**ask_sdk_model.interfaces.amazonpay.model.v1.release_environment module**

**class** ask_sdk_model.interfaces.amazonpay.model.v1.release_environment.ReleaseEnvironment

Bases: enum.Enum

Indicates if the order is for a Live (Production) or Sandbox environment.

Allowed enum values: [LIVE, SANDBOX]

**to_dict**

Returns the model properties as a dict

**to_str**

Returns the string representation of the model
ask_sdk_model.interfaces.amazonpay.model.v1.seller_billing_agreement_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.seller_billing_agreement_attributes.SellerBillingAgreementAttributes

Bases: object

Provides more context about the billing agreement that is represented by this Billing Agreement object.

Parameters

- **seller_billing_agreement_id** *(optional) str* – The merchant-specified identifier of this billing agreement. At least one request parameter must be specified. Amazon recommends that you use only the following characters: lowercase a-z, uppercase A-Z, numbers 0-9, dash (-), underscore (_).
- **store_name** *(optional) str* – The identifier of the store from which the order was placed. This overrides the default value in Seller Central under Settings &gt; Account Settings. It is displayed to the buyer in their emails and transaction history on the Amazon Payments website.
- **custom_information** *(optional) str* – Any additional information that you wish to include with this billing agreement. At least one request parameter must be specified.

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.seller_order_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.seller_order_attributes.SellerOrderAttributes

Bases: object

This object includes elements shown to buyers in emails and in their transaction history. See the “SellerOrderAttributes” section of the Amazon Pay API reference guide for details about this object.

Parameters

- **seller_order_id** *(optional) str* – The merchant-specified identifier of this order. This is shown to the buyer in their emails and transaction history on the Amazon Pay website.
- **store_name** *(optional) str* – The identifier of the store from which the order was placed. This overrides the default value in Seller Central under Settings &gt; Account Settings. It is displayed to the buyer in their emails and transaction history on the Amazon Payments website.
- **custom_information** *(optional) str* – Any additional information that you want to include with this order reference.
- **seller_note** *(optional) str* – This represents a description of the order that is displayed in emails to the buyer.
**to_dict()**  
Returns the model properties as a dict

**to_str()**  
Returns the string representation of the model

### ask_sdk_model.interfaces.amazonpay.model.v1.state module

#### class ask_sdk_model.interfaces.amazonpay.model.v1.state.State

**Bases:** enum.Enum

Indicates the state that the Authorization object, Capture object, or Refund object is in. For more information see - https://pay.amazon.com/us/developer/documentation/apireference/201752950

Allowed enum values: [Pending, Open, Declined, Closed, Completed]

**to_dict()**  
Returns the model properties as a dict

**to_str()**  
Returns the string representation of the model

### ask_sdk_model.interfaces.amazonpay.model.request package

#### Submodules

**Note:** Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

*For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.*

### ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes module

#### class ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes.AuthorizeAttributes

**Bases:** ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity

This is an object to set the attributes specified in the AuthorizeAttributes table. See the “AuthorizationDetails” section of the Amazon Pay API reference guide for details about this object.

**Parameters**
• `authorization_reference_id` *(optional)* `str` – This is 3P seller’s identifier for this authorization transaction. This identifier must be unique for all of your authorization transactions.

• `authorization_amount` *(optional)* `ask_sdk_model.interfaces.amazonpay.model.request.price.Price` –

• `transaction_timeout` *(optional)* `int` – The maximum number of minutes allocated for the Authorize operation call to be processed. After this the authorization is automatically declined and you cannot capture funds against the authorization. The default value for Alexa transactions is 0. In order to speed up checkout time for voice users we recommend to not change this value.

• `seller_authorization_note` *(optional)* `str` – A description for the transaction that is included in emails to the user. Appears only when AuthorizeAndCapture is chosen.

• `soft_descriptor` *(optional)* `str` – The description to be shown on the user’s payment instrument statement if AuthorizeAndCapture is chosen. Format of soft descriptor sent to the payment processor is "AMZ*<soft descriptor specified here>". Default is "AMZ*<SELLER_NAME> amzn.com/ pmts WA". Maximum length can be 16 characters.

• `version` *(optional)* `str` – Version of the Amazon Pay Entity. Can be 1 or greater.

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

`ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity` module

class `ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity`

Bases: `object`

Parameters

• `object_type` *(optional)* `str` –

• `version` *(optional)* `str` – Version of the Amazon Pay Entity. Can be 1 or greater.

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `@type` variable.

AuthorizeAttributes: `ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes.AuthorizeAttributes`,

SellerBillingAgreementAttributes: `ask_sdk_model.interfaces.amazonpay.model.request.seller_billing_agreement_attributes.SellerBillingAgreementAttributes`,

SetupAmazonPayRequest: `ask_sdk_model.interfaces.amazonpay.request.setup_amazon_pay_request.SetupAmazonPayRequest`,

134 Chapter 2. SDK Features
ProviderCredit: `ask_sdk_model.interfaces.amazonpay.model.request.provider_credit.ProviderCredit`,

Price: `ask_sdk_model.interfaces.amazonpay.model.request.price.Price`,

ChargeAmazonPayRequest: `ask_sdk_model.interfaces.amazonpay.request.charge_amazon_pay_request.ChargeAmazonPayRequest`,

BillingAgreementAttributes: `ask_sdk_model.interfaces.amazonpay.model.request.billing_agreement_attributes.BillingAgreementAttributes`,

SellerOrderAttributes: `ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes.SellerOrderAttributes`,

ProviderAttributes: `ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes.ProviderAttributes`
• **version** *(optional) str* – Version of the Amazon Pay Entity. Can be 1 or greater.

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

### ask_sdk_model.interfaces.amazonpay.model.request.payment_action module

class ask_sdk_model.interfaces.amazonpay.model.request.payment_action.PaymentAction
    Bases: enum.Enum

    • This is used to specify applicable payment action. * Authorize – you want to confirm the order and authorize a certain amount, but you do not want to capture at this time. * AuthorizeAndCapture – you want to confirm the order, authorize for the given amount, and capture the funds.

    Allowed enum values: [Authorize, AuthorizeAndCapture]

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

### ask_sdk_model.interfaces.amazonpay.model.request.price module

class ask_sdk_model.interfaces.amazonpay.model.request.price.Price
    Bases: ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity

This request object specifies amount and currency authorized/captured.

**Parameters**

- **amount** *(optional) str* – Amount authorized/captured.
- **currency_code** *(optional) str* – Currency code for the amount.
- **version** *(optional) str* – Version of the Amazon Pay Entity. Can be 1 or greater.

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

### ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes module

class ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes.ProviderAttributes
    Bases: ask_sdk_model.interfaces.amazonpay.model.request.
```
This is required only for Ecommerce provider (Solution provider) use cases.

Parameters

- **provider_id** *(optional) str* – Solution provider ID.
- **provider_credit_list** *(optional) list[ask_sdk_model.interfaces.amazonpay.model.request.provider_credit.ProviderCredit]* – List of provider credit.
- **version** *(optional) str* – Version of the Amazon Pay Entity. Can be 1 or greater.

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

Provides more context about the billing agreement that is represented by this Billing Agreement object.

Parameters
• **seller_billing_agreement_id** (*optional str*) – The merchant-specified identifier of this billing agreement. At least one request parameter must be specified. Amazon recommends that you use only the following characters: lowercase a-z, uppercase A-Z, numbers 0-9, dash (-), underscore (_).

• **store_name** (*optional str*) – The identifier of the store from which the order was placed. This overrides the default value in Seller Central under Settings &gt; Account Settings. It is displayed to the buyer in their emails and transaction history on the Amazon Payments website.

• **custom_information** (*optional str*) – Any additional information that you wish to include with this billing agreement. At least one request parameter must be specified.

• **version** (*optional str*) – Version of the Amazon Pay Entity. Can be 1 or greater.

```
to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model
```

### ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes module

```
class ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes.SellerOrderAttributes

Bases: 
ask_sdk_model.interfaces.amazonpay.model.request.
base_amazon_pay_entity.BaseAmazonPayEntity

This object includes elements shown to buyers in emails and in their transaction history. See the “SellerOrderAttributes” section of the Amazon Pay API reference guide for details about this object.

Parameters

• **seller_order_id** (*optional str*) – The merchant-specified identifier of this order. This is shown to the buyer in their emails and transaction history on the Amazon Pay website.

• **store_name** (*optional str*) – The identifier of the store from which the order was placed. This overrides the default value in Seller Central under Settings &gt; Account Settings. It is displayed to the buyer in their emails and transaction history on the Amazon Payments website.

• **custom_information** (*optional str*) – Any additional information that you want to include with this order reference.

• **seller_note** (*optional str*) – This represents a description of the order that is displayed in emails to the buyer.

• **version** (*optional str*) – Version of the Amazon Pay Entity. Can be 1 or greater.

```
to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model
```
ask_sdk_model.interfaces.amazonpay.model.response package

Submodules

**Note:** Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package `a` has module `b` with class `C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.amazonpay.model.response.authorization_details module

class ask_sdk_model.interfaces.amazonpay.model.response.authorization_details.Authorization


This object encapsulates details about an Authorization object including the status, amount captured and fee

2.14. Hosting Skills as Webservice
charged.

Parameters

- **amazon_authorization_id** *(optional)* *str* – This is AmazonPay generated identifier for this authorization transaction.

- **authorization_reference_id** *(optional)* *str* – This is 3P seller's identifier for this authorization transaction. This identifier must be unique for all of your authorization transactions.

- **seller_authorization_note** *(optional)* *str* – A description for the transaction that is included in emails to the user. Appears only when AuthorizeAndCapture is chosen.

- **authorization_amount** *(optional)* *ask_sdk_model.interfaces.amazonpay.model.response.price.Price* –

- **captured_amount** *(optional)* *ask_sdk_model.interfaces.amazonpay.model.response.price.Price* –

- **authorization_fee** *(optional)* *ask_sdk_model.interfaces.amazonpay.model.response.price.Price* –

- **id_list** *(optional)* *list*[str] – list of AmazonCaptureId identifiers that have been requested on this Authorization object.

- **creation_timestamp** *(optional)* *datetime* – This is the time at which the authorization was created.

- **expiration_timestamp** *(optional)* *datetime* – This is the time at which the authorization expires.

- **authorization_status** *(optional)* *ask_sdk_model.interfaces.amazonpay.model.response.authorization_status.AuthorizationStatus* –

- **soft_decline** *(optional)* *bool* – This indicates whether an authorization resulted in a soft decline.

- **capture_now** *(optional)* *bool* – This indicates whether a direct capture against the payment contract was specified.

- **soft_descriptor** *(optional)* *str* – This is the description to be shown on the buyer's payment instrument statement if AuthorizeAndCapture was chosen.

- **authorization_billing_address** *(optional)* *ask_sdk_model.interfaces.amazonpay.model.response.destination.Destination* –

  to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model
class ask_sdk_model.interfaces.amazonpay.model.response.authorization_status.AuthorizationStatus


Indicates the current status of an Authorization object, a Capture object, or a Refund object.

Parameters

- **state** *(optional)* ask_sdk_model.interfaces.amazonpay.model.response.state.State
- **reason_code** *(optional)* str
- **reason_description** *(optional)* str
- **last_update_timestamp** *(optional)* datetime

*to_dict()*

Returns the model properties as a dict

*to_str()*

Returns the string representation of the model

class ask_sdk_model.interfaces.amazonpay.model.response.billing_agreement_details.BillingAgreementDetails


The result attributes from successful SetupAmazonPay call.

Parameters

- **billing_agreement_id** *(optional)* str
• **creation_timestamp** *(optional) datetime* – Time at which billing agreement details created.

• **destination** *(optional) ask_sdk_model.interfaces.amazonpay.model.v1.destination.Destination* – The default shipping address of the buyer. Returned if `needAmazonShippingAddress` is set to true.

• **checkout_language** *(optional) str* – Merchant’s preferred language of checkout.

• **release_environment** *(optional) ask_sdk_model.interfaces.amazonpay.model.response.release_environment.ReleaseEnvironment* –

• **billing_agreement_status** *(optional) ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status.BillingAgreementStatus* –

• **billing_address** *(optional) ask_sdk_model.interfaces.amazonpay.model.response.destination.Destination* – The Billing Address of the payment instrument associated with Billing Agreement.

```python
to_dict()
Returns the model properties as a dict
to_str()
Returns the string representation of the model
```

**ask_sdk_model.interfaces.amazonpay.model.response.destination module**

```python
class ask_sdk_model.interfaces.amazonpay.model.response.destination.Destination(name=None, company_name=None, address_line1=None, address_line2=None, address_line3=None, city=None, district_or_county=None, state_or_region=None, postal_code=None, country_code=None, phone=None)
```

Bases: `ask_sdk_model.interfaces.amazonpay.model.v1.destination.Destination`

**Parameters**

• **name** *(optional) str* – The name or business name

• **company_name** *(optional) str* – The company name

• **address_line1** *(optional) str* – The first line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.
• **address_line2** *(optional)* *(str)* – The second line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.

• **address_line3** *(optional)* *(str)* – The third line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.

• **city** *(optional)* *(str)* – The city

• **district_or_county** *(optional)* *(str)* – The district or County

• **state_or_region** *(optional)* *(str)* – The state or region. This element is free text and can be either a 2-character code, fully spelled out, or abbreviated. Required. Note :- This response element is returned only in the U.S.

• **postal_code** *(optional)* *(str)* – The postal code.

• **country_code** *(optional)* *(str)* – The country code, in ISO 3166 format

• **phone** *(optional)* *(str)* – The phone number

```python
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.amazonpay.model.response.price module**

```python
class ask_sdk_model.interfaces.amazonpay.model.response.price.Price(amount=None, currency_code=None)
```

Bases: `ask_sdk_model.interfaces.amazonpay.model.v1.price.Price`

This response object specifies amount and currency authorized/captured.

Parameters

• **amount** *(optional)* *(str)* – Amount authorized/captured.

• **currency_code** *(optional)* *(str)* – Currency code for the amount.

```python
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.amazonpay.model.response.release_environment module**

```python
class ask_sdk_model.interfaces.amazonpay.model.response.release_environment.ReleaseEnvironment
```

Bases: `enum.Enum`

Indicates if the contract is for a Live (Production) or Sandbox environment.

Allowed enum values: [LIVE, SANDBOX]

```python
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model
```
ask_sdk_model.interfaces.amazonpay.model.response.state module

class ask_sdk_model.interfaces.amazonpay.model.response.state.State
    Bases: enum.Enum

    Indicates the state that the Authorization object is in. For more information see “Authorization states and reason codes” under “States and reason codes” section in Amazon Pay API Reference Guide.
    Allowed enum values: [Pending, Open, Declined, Closed]

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.v1 package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.interfaces.amazonpay.v1.amazon_pay_error_response module

class ask_sdk_model.interfaces.amazonpay.v1.amazon_pay_error_response.AmazonPayErrorResponse
    Bases: object

    Error response for SetupAmazonPay and ChargeAmazonPay calls.

    Parameters
        • error_code ((optional) str) – Error code indicating the succinct cause of error
        • error_message ((optional) str) – Description of the error.

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay module

class ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay.ChargeAmazonPay(consent_token=None, seller_id=None, billing_agreement_id=None, payment_action=None, authorize_attributes=None, seller_order_attributes=None, provider_attributes=None):

    Bases: object

Charge Amazon Pay Request Object

Parameters

- **consent_token** (*optional* `str*) – Authorization token that contains the permissions consented to by the user.
- **seller_id** (*optional* `str*) – The seller ID (also known as merchant ID). If you are an Ecommerce Provider (Solution Provider), please specify the ID of the merchant, not your provider ID.
- **billing_agreement_id** (*optional* `str*) – The payment contract i.e. billing agreement created for the user.
- **payment_action** (*optional* `ask_sdk_model.interfaces.amazonpay.model.v1.payment_action.PaymentAction*) –
- **authorize_attributes** (*optional* `ask_sdk_model.interfaces.amazonpay.model.v1.authorize_attributes.AuthorizeAttributes*) –
- **seller_order_attributes** (*optional* `ask_sdk_model.interfaces.amazonpay.model.v1.seller_order_attributes.SellerOrderAttributes*) –
- **provider_attributes** (*optional* `ask_sdk_model.interfaces.amazonpay.model.v1.provider_attributes.ProviderAttributes*) –

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay_result module

class ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay_result.ChargeAmazonPayResult(amazon_order_reference_id=None, authorization_details=None):

    Bases: object

Charge Amazon Pay Result Object. It is sent as part of the response to ChargeAmazonPay request.

2.14. Hosting Skills as Webservice 145
Parameters


  - `to_dict()`
    - Returns the model properties as a dict
  
  - `to_str()`
    - Returns the string representation of the model

`ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay module`

```python
class ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay.SetupAmazonPay(consent_token=None, seller_id=None, country_of_establishment=None, ledger_currency=None, checkout_language=None, billing_agreement_attributes=None, need_amazon_shipping_address=False, sandbox_mode=False, sandbox_customer_email_id=None)
```

**Bases:** object

Setup Amazon Pay Request Object

**Parameters**

- `consent_token` *(optional) str* – Authorization token that contains the permissions consented to by the user.

- `seller_id` *(optional) str* – The seller ID (also known as merchant ID). If you are an Ecommerce Provider (Solution Provider), please specify the ID of the merchant, not your provider ID.

- `country_of_establishment` *(optional) str* – The country in which the merchant has registered, as an Amazon Payments legal entity.

- `ledger_currency` *(optional) str* – The currency of the merchant’s ledger account.

- `checkout_language` *(optional) str* – The merchant’s preferred language for checkout.

- `billing_agreement_attributes` *(optional) ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_attributes.BillingAgreementAttributes* –

- `need_amazon_shipping_address` *(bool)* – To receive the default user shipping address in the response, set this parameter to true. Not required if a user shipping address is not required.
• **sandbox_mode** (**bool**) – To test in Sandbox mode, set this parameter to true.

• **sandbox_customer_email_id** (**optional str**) – Use this parameter to create a Sandbox payment object. In order to use this parameter, you first create a Sandbox user account in Seller Central. Then, pass the email address associated with that Sandbox user account.

```python
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay_result module**

```python
class ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay_result.SetupAmazonPayResult(billing_agreement_details=None):
    # Setup Amazon Pay Result Object. It is sent as part of the response to SetupAmazonPay request.
    #
    # Parameters
    #    billing_agreement_details ((optional) ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details.BillingAgreementDetails)
    
    to_dict()
        Returns the model properties as a dict
to_str()
        Returns the string representation of the model
```

**ask_sdk_model.interfaces.amazonpay.request package**

**Submodules**

**Note:** Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do `from a import C` instead of `from a.b import C`.

**ask_sdk_model.interfaces.amazonpay.request.charge_amazon_pay_request module**

```python
class ask_sdk_model.interfaces.amazonpay.request.charge_amazon_pay_request.ChargeAmazonPayRequest:
    #
    # Bases:
    #    ask_sdk_model.interfaces.amazonpay.model.request.
```

2.14. Hosting Skills as Webservice
Charge Amazon Pay Request Object.

Parameters

- **version** *(optional) str* – Version of the Amazon Pay Entity. Can be 1 or greater.
- **seller_id** *(optional) str* – The seller ID (also known as merchant ID). If you are an Ecommerce Provider (Solution Provider), please specify the ID of the merchant, not your provider ID.
- **billing_agreement_id** *(optional) str* – The payment contract i.e. billing agreement created for the user.
- **payment_action** *(optional) ask_sdk_model.interfaces.amazonpay.model.request.payment_action.PaymentAction* –
- **authorize_attributes** *(optional) ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes.AuthorizeAttributes* –
- **seller_order_attributes** *(optional) ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes.SellerOrderAttributes* –
- **provider_attributes** *(optional) ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes.ProviderAttributes* –

```python
to_dict()
Returns the model properties as a dict
to_str()
Returns the string representation of the model
```

Setup Amazon Pay Request Object.

Parameters

- **version** *(optional) str* – Version of the Amazon Pay Entity. Can be 1 or greater.
• **seller_id** *(optional)* *(str)* – The seller ID (also known as merchant ID). If you are an Ecommerce Provider (Solution Provider), please specify the ID of the merchant, not your provider ID.

• **country_of_establishment** *(optional)* *(str)* – The country in which the merchant has registered, as an Amazon Payments legal entity.

• **ledger_currency** *(optional)* *(str)* – The currency of the merchant’s ledger account.

• **checkout_language** *(optional)* *(str)* – The merchant’s preferred language for checkout.

• **billing_agreement_attributes** *(optional)*

  - **need_amazon_shipping_address** *(bool)* – To receive the default user shipping address in the response, set this parameter to true. Not required if a user shipping address is not required.

• **sandbox_mode** *(bool)* – To test in Sandbox mode, set this parameter to true.

• **sandbox_customer_email_id** *(optional)* *(str)* – Use this parameter to create a Sandbox payment object. In order to use this parameter, you first create a Sandbox user account in Seller Central. Then, pass the email address associated with that Sandbox user account.

  
  - **to_dict**
    - Returns the model properties as a dict

  - **to_str**
    - Returns the string representation of the model

---

**ask_sdk_model.interfaces.amazonpay.response package**

**Submodules**

**Note:** Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

---

**ask_sdk_model.interfaces.amazonpay.response.amazon_pay_error_response module**

**class** ask_sdk_model.interfaces.amazonpay.response.amazon_pay_error_response.AmazonPayErrorResponse

  **Bases:** ask_sdk_model.interfaces.amazonpay.v1.amazon_pay_error_response.AmazonPayErrorResponse

  Error response for SetupAmazonPay and ChargeAmazonPay calls.

  **Parameters**
• `error_code ((optional) str)` – Error code indicating the succinct cause of error

• `error_message ((optional) str)` – Description of the error.

`to_dict()`
- Returns the model properties as a dict

`to_str()`
- Returns the string representation of the model

`ask_sdk_model.interfaces.amazonpay.response.charge_amazon_pay_result module`

class `ask_sdk_model.interfaces.amazonpay.response.charge_amazon_pay_result.ChargeAmazonPayResult`

Bases: `ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay_result.ChargeAmazonPayResult`

Charge Amazon Pay Result Object. It is sent as part of the response to ChargeAmazonPayRequest.

Parameters

• `amazon_order_reference_id ((optional) str)` – The order reference identifier.

• `authorization_details ((optional) ask_sdk_model.interfaces.amazonpay.model.response.authorization_details.AuthorizationDetails)`

`to_dict()`
- Returns the model properties as a dict

`to_str()`
- Returns the string representation of the model

`ask_sdk_model.interfaces.amazonpay.response.setup_amazon_pay_result module`

class `ask_sdk_model.interfaces.amazonpay.response.setup_amazon_pay_result.SetupAmazonPayResult`

Bases: `object`

Setup Amazon Pay Result Object. It is sent as part of the response to SetupAmazonPayRequest.

Parameters `billing_agreement_details ((optional) ask_sdk_model.interfaces.amazonpay.model.response.billing_agreement_details.BillingAgreementDetails)`

`to_dict()`
- Returns the model properties as a dict

`to_str()`
- Returns the string representation of the model

`ask_sdk_model.interfaces.audioplayer package`

Submodules
Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.interfaces.audioplayer.audio_item module

class ask_sdk_model.interfaces.audioplayer.audio_item.AudioItem(stream=None, metadata=None):
    Bases: object

    Parameters
    • stream ((optional) ask_sdk_model.interfaces.audioplayer.stream.Stream) -
    • metadata ((optional) ask_sdk_model.interfaces.audioplayer.audio_item_metadata.AudioItemMetadata) -

    to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.audio_item_metadata module

class ask_sdk_model.interfaces.audioplayer.audio_item_metadata.AudioItemMetadata (title=None, subtitle=None, art=None, background_image=None):
    Bases: object

    Encapsulates the metadata about an AudioItem.

    Parameters
    • title ((optional) str) - An optional title of the audio item.
    • subtitle ((optional) str) - An optional subtitle of the audio item.
    • art ((optional) ask_sdk_model.interfaces.display.image.Image) - An optional cover art image for the audio item.
    • background_image ((optional) ask_sdk_model.interfaces.display.image.Image) - An optional background image for the audio item.

    to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
ask_sdk_model.interfaces.audioplayer.audio_player_interface module

class ask_sdk_model.interfaces.audioplayer.audio_player_interface.AudioPlayerInterface
    Bases: object
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.audio_player_state module

class ask_sdk_model.interfaces.audioplayer.audio_player_state.AudioPlayerState
    (offset_inMilliseconds=None, token=None, player_activity=None)
    Bases: object
    Parameters
        • offset_inMilliseconds ((optional) int)-
        • token ((optional) str)-
        • player_activity ((optional) ask_sdk_model.interfaces.audioplayer.player_activity.PlayerActivity)-
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.clear_behavior module

class ask_sdk_model.interfaces.audioplayer.clear_behavior.ClearBehavior
    Bases: enum.Enum
    Allowed enum values: [CLEAR_ALL, CLEAR_ENQUEUED]
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.clear_queue_directive module

class ask_sdk_model.interfaces.audioplayer.clear_queue_directive.ClearQueueDirective
    (clear_behavior=None)
    Bases: ask_sdk_model.directive.Directive
    Parameters clear_behavior ((optional) ask_sdk_model.interfaces.audioplayer.clear_behavior.ClearBehavior)-
    to_dict()
        Returns the model properties as a dict
to_str()
Returns the string representation of the model

ask_sdk_modelinterfacesaudioplayer.current_playback_state module

class ask_sdk_model.interfaces.audioplayer.current_playback_state.CurrentPlaybackState
Bases: object
Parameters
- offset_in_milliseconds (optional int)-
- player_activity ((optional) ask_sdk_model.interfaces.audioplayer.player_activity.PlayerActivity)-
- token (optional str)-
to_dict()
Returns the model properties as a dict
to_str()
Returns the string representation of the model

ask_sdk_modelinterfacesaudioplayer.error module

class ask_sdk_model.interfaces.audioplayer.error.Error
Bases: object
Parameters
- message (optional str)-
- object_type ((optional) ask_sdk_model.interfaces.audioplayer.error_type.ErrorType)-
to_dict()
Returns the model properties as a dict
to_str()
Returns the string representation of the model

ask_sdk_modelinterfacesaudioplayer.error_type module

class ask_sdk_model.interfaces.audioplayer.error_type.ErrorType
Bases: enum.Enum
Allowed enum values: [MEDIA_ERROR_INTERNAL_DEVICE_ERROR, MEDIA_ERROR_INTERNAL_SERVER_ERROR, MEDIA_ERROR_INVALID_REQUEST, MEDIA_ERROR_SERVICE_UNAVAILABLE, MEDIA_ERROR_UNKNOWN]
to_dict()
Returns the model properties as a dict

2.14. Hosting Skills as Webservice
to_str()  
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.play_behavior module

class ask_sdk_model.interfaces.audioplayer.play_behavior.PlayBehavior
    Bases: enum.Enum

    Allowed enum values: [ENQUEUE, REPLACE_ALL, REPLACE_ENQUEUED]

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.play_directive module

class ask_sdk_model.interfaces.audioplayer.play_directive.PlayDirective
    (play_behavior=None, audio_item=None)

    Bases: ask_sdk_model.directive.Directive

    Parameters
    • play_behavior ((optional) ask_sdk_model.interfaces.audioplayer.play_behavior.PlayBehavior)–
    • audio_item ((optional) ask_sdk_model.interfaces.audioplayer.audio_item.AudioItem)–

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.playback_failed_request module

class ask_sdk_model.interfaces.audioplayer.playback_failed_request.PlaybackFailedRequest
    (request_id=None, timestamp=None, location=None, current_playback_state=None, error=None, token=None)

    Bases: ask_sdk_model.request.Request

    Parameters
    • request_id ((optional) str) – Represents the unique identifier for the specific request.
• timestamp (optional datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• locale (optional str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• current_playback_state (optional ask_sdk_model.interfaces.audioplayer.current_playback_state.CurrentPlaybackState)

• error (optional ask_sdk_model.interfaces.audioplayer.error.Error)

• token (optional str)

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.playback_finished_request module

class ask_sdk_model.interfaces.audioplayer.playback_finished_request.PlaybackFinishedRequest

Bases: ask_sdk_model.request.Request

Parameters

• request_id (optional str) – Represents the unique identifier for the specific request.

• timestamp (optional datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• locale (optional str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• offset_in_milliseconds (optional int)

• token (optional str)

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model
ask_sdk_model.interfaces.audioplayer.playback_nearly_finished_request module

class ask_sdk_model.interfaces.audioplayer.playback_nearly_finished_request.PlaybackNearlyFinishedRequest

Bases: ask_sdk_model.request.Request

Parameters

- request_id (optional str) – Represents the unique identifier for the specific request.
- timestamp (optional datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- locale (optional str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
- offset_in_milliseconds (optional int)
- token (optional str)

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.playback_started_request module

class ask_sdk_model.interfaces.audioplayer.playback_started_request.PlaybackStartedRequest

Bases: ask_sdk_model.request.Request

Parameters

- request_id (optional str) – Represents the unique identifier for the specific request.
- timestamp (optional datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- locale (optional str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
• offset_in_milliseconds (optional) int
• token (optional) str

\texttt{to\_dict()}
  Returns the model properties as a dict

\texttt{to\_str()}
  Returns the string representation of the model

\texttt{ask\_sdk\_model.interfaces.audioplayer.playback\_stopped\_request module}

\texttt{class ask\_sdk\_model.interfaces.audioplayer.playback\_stopped\_request.PlaybackStoppedRequest}

\texttt{Bases: ask\_sdk\_model.request.Request}

Parameters

• \texttt{request\_id (optional) str} – Represents the unique identifier for the specific request.
• \texttt{timestamp (optional) datetime} – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
• \texttt{locale (optional) str} – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
• \texttt{offset\_in\_milliseconds (optional) int}
• \texttt{token (optional) str}

\texttt{to\_dict()}
  Returns the model properties as a dict

\texttt{to\_str()}
  Returns the string representation of the model

\texttt{ask\_sdk\_model.interfaces.audioplayer.player\_activity module}

\texttt{class ask\_sdk\_model.interfaces.audioplayer.player\_activity.PlayerActivity}

\texttt{Bases: enum.Enum}

Allowed enum values: [PLAYING, PAUSED, FINISHED, BUFFER\_UNDERRUN, IDLE, STOPPED]

\texttt{to\_dict()}
  Returns the model properties as a dict

\texttt{to\_str()}
  Returns the string representation of the model
ask_sdk_model.interfaces.audioplayer.stop_directive module

```python
class ask_sdk_model.interfaces.audioplayer.stop_directive.StopDirective:
    Bases: ask_sdk_model.directive.Directive

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model
```

ask_sdk_model.interfaces.audioplayer.stream module

```python
class ask_sdk_model.interfaces.audioplayer.stream.Stream(expected_previous_token=none, token=none, url=none, offset_in_milliseconds=none):
    Bases: object

    Parameters
    ----------
    expected_previous_token : (optional) str
    token : (optional) str
    url : (optional) str
    offset_in_milliseconds : (optional) int

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model
```

ask_sdk_model.interfaces.connections package

Subpackages

ask_sdk_model.interfaces.connections.entities package

Submodules

---

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.
ask_sdk_model.interfaces.connections.entities.base_entity module

class ask_sdk_model.interfaces.connections.entities.base_entity.BaseEntity:
Bases: object

Parameters
- object_type (optional str)
- version (optional str) – version of the request

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets @type variable.

Restaurant:
ask_sdk_model.interfaces.connections.entities.restaurant.Restaurant

PostalAddress:
ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress

classmethod get_real_child_model(data)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections.entities.postal_address module

class ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress:
Bases: ask_sdk_model.interfaces.connections.entities.base_entity.BaseEntity

Postal Address

Parameters
- version (optional str) – version of the request
- street_address (optional str) – street address
- locality (optional str) – locality/city
• **region** *(optional)* str – state/region
• **postal_code** *(optional)* str – postal/zip code
• **country** *(optional)* str – country

```python
    to_dict()
    Returns the model properties as a dict
```

```python
    to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.connections.entities.restaurant module**

class ask_sdk_model.interfaces.connections.entities.restaurant.Restaurant *(version=None, name=None, location=None)*

Bases: ask_sdk_model.interfaces.connections.entities.base_entity.BaseEntity

Restaurant entity

Parameters

• **version** *(optional)* str – version of the request
• **name** *(optional)* str – name of the restaurant
• **location** *(optional)* ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress – location

```python
    to_dict()
    Returns the model properties as a dict
```

```python
    to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.connections.requests package**

**Submodules**

**Note:** Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

**ask_sdk_model.interfaces.connections.requests.base_request module**

class ask_sdk_model.interfaces.connections.requests.requests.base_request.BaseRequest *(object_type=None, version=None)*

Bases: object

Parameters
• **object_type** *(optional) str*—
• **version** *(optional) str*— version of the request

**Note:** This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets @type variable.

`ScheduleFoodEstablishmentReservationRequest: ask_sdk_model.interfaces.connections.requests.schedule_food_establishment_reservation_request.ScheduleFoodEstablishmentReservationRequest`,

`PrintPDFRequest: ask_sdk_model.interfaces.connections.requests.print_pdf_request.PrintPDFRequest`,

`PrintImageRequest: ask_sdk_model.interfaces.connections.requests.print_image_request.PrintImageRequest`,

`ScheduleTaxiReservationRequest: ask_sdk_model.interfaces.connections.requests.schedule_taxi_reservation_request.ScheduleTaxiReservationRequest`,

`PrintWebPageRequest: ask_sdk_model.interfaces.connections.requests.print_web_page_request.PrintWebPageRequest`

classmethod **get_real_child_model** *(data)*
   Returns the real base class specified by the discriminator

**to_dict** *
   Returns the model properties as a dict

**to_str** *
   Returns the string representation of the model

*ask_sdk_model.interfaces.connections.requests.print_image_request module*

class *ask_sdk_model.interfaces.connections.requests.print_image_request.PrintImageRequest* *(version=None, title=None, url=None, description=None, image_type=None)*

**Bases:** *ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest*

Payload Request object for PrintImage

**Parameters**

• **version** *(optional) str*— version of the request
class ask_sdk_model.interfaces.connections.requests.print_pdf_request.PrintPDFRequest (version=None, title=None, url=None, description=None)

Bases: ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest

Payload Request object for PrintPDF

Parameters

- **version** (optional str) – version of the request
- **title** (optional str) – title of the image
- **url** (optional str) – url of the image
- **description** (optional str) – description of the image

  to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model

class ask_sdk_model.interfaces.connections.requests.print_web_page_request.PrintWebPageRequest (version=None, title=None, url=None, description=None)

Bases: ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest

Payload Request object for PrintWebPage

Parameters
• **version** *(optional)* `str*– version of the request
  • **title** *(optional)* `str*– title of the image
  • **url** *(optional)* `str*– url of the image
  • **description** *(optional)* `str*– description of the image

  `to_dict()`
  Returns the model properties as a dict

  `to_str()`
  Returns the string representation of the model

  `ask_sdk_model.interfaces.connections.requests.schedule_food_establishment_reservation_request` module

  `class ask_sdk_model.interfaces.connections.requests.schedule_food_establishment_reservation_request.ScheduleFoodEstablishmentReservationRequest`

  Bases: `ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest`

  ScheduleFoodEstablishmentReservationRequest for booking restaurant reservation

  Parameters

  • **version** *(optional)* `str*– version of the request
  • **start_time** *(optional)* `str*– start time of the reservation
  • **party_size** *(optional)* `str*– party size
  • **restaurant** *(optional)* `ask_sdk_model.interfaces.connections.entities.restaurant.Restaurant*– restaurant

  `to_dict()`
  Returns the model properties as a dict

  `to_str()`
  Returns the string representation of the model

  `ask_sdk_model.interfaces.connections.requests.schedule_taxi_reservation_request` module

  `class ask_sdk_model.interfaces.connections.requests.schedule_taxi_reservation_request.ScheduleTaxiReservationRequest`

  Bases: `ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest`

  ScheduleTaxiReservationRequest for booking taxi reservation

  Parameters

  • **version** *(optional)* `str*– version of the request
  • **pickup_time** *(optional)* `str*– pickup time

2.14. Hosting Skills as Webservice
• **party_size** ((optional) *str*) – party size

• **pickup_location** ((optional) *ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress*) – pick up location

• **drop_off_location** ((optional) *ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress*) – drop off location

```python
to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model
```

### Submodules

**Note:** Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package `a` has module `b` with class `C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.connections.connections_request module

```python
class ask_sdk_model.interfaces.connections.connections_request.ConnectionsRequest (request_id=None, timestamp=None, locale=None, name=None, payload=None)
```

**Bases:** `ask_sdk_model.request.Request`

This is the request object that a skill will receive as a result of `Connections.SendRequest` directive from sender skill.

**Parameters**

• **request_id** ((optional) *str*) – Represents the unique identifier for the specific request.

• **timestamp** ((optional) *datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• **locale** ((optional) *str*) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• **name** ((optional) *str*) – Name of the action sent by the referrer skill.

• **payload** ((optional) *dict(str, object]*) – This is an object sent between the two skills for processing a `ConnectionsRequest` or `ConnectionsResponse`. This will always be a valid payload based on Action schema for the requester action.
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.connections.connections_response module

class ask_sdk_model.interfaces.connections.connections_response.ConnectionsResponse(request_id=None, timestamp=None, locale=None, status=None, name=None, payload=None, token=None):
    Bases: ask_sdk_model.request.Request

    This is the request object that a skill will receive as a result of Connections.SendResponse directive from referrer skill.

    Parameters

    • request_id (optional str) – Represents the unique identifier for the specific request.

    • timestamp (optional datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

    • locale (optional str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

    • status (optional ask_sdk_model.interfaces.connections.connections_status.ConnectionsStatus) –

    • name (optional str) – Name of the action for which response is received.

    • payload (optional dict(str, object)) – This is an object sent from referrer skill as is.

    • token (optional str) – This is the token that the skill originally sent with the ConnectionsSendRequest directive.

        to_dict()
            Returns the model properties as a dict

        to_str()
            Returns the string representation of the model
**ask_sdk_model.interfaces.connections.connections_status module**

```python
class ask_sdk_model.interfaces.connections.connections_status.ConnectionsStatus(
    code=None,
    message=None
)
```

**Bases:** `object`

Connection Status indicates a high level understanding of the result of ConnectionsRequest.

**Parameters**

- `code` *(optional)* `str* — This is a code signifying the status of the request sent by the skill. Protocol adheres to HTTP status codes.
- `message` *(optional)* `str* — This is a message that goes along with response code that can provide more information about what occurred.

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

**ask_sdk_model.interfaces.connections.send_request_directive module**

```python
class ask_sdk_model.interfaces.connections.send_request_directive.SendRequestDirective(
    name=None,
    payload=None,
    token=None
)
```

**Bases:** `ask_sdk_model.directive.Directive`

This is the directive that a skill can send as part of their response to a session based request to execute a predefined Connections. This will also return a result to the referring skill. (No Guarantee response will be returned)

**Parameters**

- `name` *(optional)* `str* — This defines the name of the Connection skill is trying to execute. It must be a valid and supported Connection name.
- `payload` *(optional)* `dict(str, object)`* — This is an object sent between the two skills for processing a ConnectionsRequest or ConnectionsResponse. The contract for the object is based on the schema of the Action used in the SendRequestDirective. Invalid payloads will result in errors sent back to the referrer.
- `token` *(optional)* `str* — This is an echo back string that skills send when during Connections.SendRequest directive. They will receive it when they get the ConnectionsResponse. It is never sent to the skill handling the request.

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model
ask_sdk_model.interfaces.connections.send_response_directive module

class ask_sdk_model.interfaces.connections.send_response_directive.SendResponseDirective(status=None, payload=None)

Bases: ask_sdk_model.directive.Directive

This is the directive that a skill can send as part of their response to a session based request to return a response to ConnectionsRequest.

Parameters

• `status` ((optional) ask_sdk_model.interfaces.connections.connections_status.ConnectionsStatus) –
• `payload` ((optional) dict(str, object)) – This is an object sent to referrer skill as is.

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

ask_sdk_model.interfaces.display package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.interfaces.display.back_button_behavior module

class ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior

Bases: enum.Enum

Allowed enum values: [HIDDEN, VISIBLE]

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model
ask_sdk_model.interfaces.display.body_template1 module

class ask_sdk_model.interfaces.display.body_template1.BodyTemplate1(token=None, back_button=None, background_image=None, title=None, text_content=None)

Bases: ask_sdk_model.interfaces.display.template.Template

Parameters

• **token** *(optional) str* -

• **back_button** *(optional) ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior* -

• **background_image** *(optional) ask_sdk_model.interfaces.display.image.Image* -

• **title** *(optional) str* -

• **text_content** *(optional) ask_sdk_model.interfaces.display.text_content.TextContent* -

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.display.body_template2 module

class ask_sdk_model.interfaces.display.body_template2.BodyTemplate2(token=None, back_button=None, background_image=None, image=None, title=None, text_content=None)

Bases: ask_sdk_model.interfaces.display.template.Template

Parameters

• **token** *(optional) str* -

• **back_button** *(optional) ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior* -

• **background_image** *(optional) ask_sdk_model.interfaces.display.image.Image* -

• **image** *(optional) ask_sdk_model.interfaces.display.image.Image* -

• **title** *(optional) str* -
• **text_content**  
  
  Returns the model properties as a dict

  ```python
  to_dict()
  ```
  Returns the model properties as a dict

  ```python
  to_str()
  ```
  Returns the string representation of the model

**ask_sdk_model.interfaces.display.body_template3 module**

```python
class ask_sdk_model.interfaces.display.body_template3.BodyTemplate3(  
    token=None,  
    back_button=None,  
    background_image=None,  
    image=None,  
    title=None,  
    text_content=None)
```

Bases: `ask_sdk_model.interfaces.display.template.Template`

Parameters

- **token**  
  - `(optional) str`

- **back_button**  
  - `(optional) ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior`

- **background_image**  
  - `(optional) ask_sdk_model.interfaces.display.image.Image`

- **image**  
  - `(optional) ask_sdk_model.interfaces.display.image.Image`

- **title**  
  - `(optional) str`

- **text_content**  
  - `(optional) ask_sdk_model.interfaces.display.text_content.TextContent`

```python
def to_dict()  
    Returns the model properties as a dict
```

```python
def to_str()  
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.display.body_template6 module**

```python
class ask_sdk_model.interfaces.display.body_template6.BodyTemplate6(  
    token=None,  
    back_button=None,  
    background_image=None,  
    image=None,  
    title=None,  
    text_content=None)
```

Bases: `ask_sdk_model.interfaces.display.template.Template`

Parameters

```python
def to_dict()  
    Returns the model properties as a dict
```

```python
def to_str()  
    Returns the string representation of the model
```
**ask_sdk_model.interfaces.display.body_template7 module**

class ask_sdk_model.interfaces.display.body_template7.BodyTemplate7(token=None, back_button=None, title=None, image=None, background_image=None):

Bases: ask_sdk_model.interfaces.display.template.Template

Parameters

- **token** ((optional) str)-
- **back_button** ((optional) ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior)-
- **title** ((optional) str)-
- **image** ((optional) ask_sdk_model.interfaces.display.image.Image)-
- **background_image** ((optional) ask_sdk_model.interfaces.display.image.Image)-

**to_dict**()
Returns the model properties as a dict

**to_str**()
Returns the string representation of the model

**ask_sdk_model.interfaces.display.display_interface module**

class ask_sdk_model.interfaces.display.display_interface.DisplayInterface(template_version=None, markup_version=None):

Bases: object

Parameters

170 Chapter 2. SDK Features
• **template_version** *(optional)* *str* –
• **markup_version** *(optional)* *str* –

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model

### ask_sdk_model.interfaces.display.display_state module

class ask_sdk_model.interfaces.display.display_state.DisplayState *(token=None)*  
Bases: object

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model

### ask_sdk_model.interfaces.display.element_selected_request module

class ask_sdk_model.interfaces.display.element_selected_request.ElementSelectedRequest *(request_id=None, timestamp=None, locale=None, token=None)*  
Bases: ask_sdk_model.request.Request

**Parameters**

• **request_id** *(optional)* *str* – Represents the unique identifier for the specific request.

• **timestamp** *(optional)* *datetime* – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• **locale** *(optional)* *str* – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• **token** *(optional)* *str* –

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model

### ask_sdk_model.interfaces.display.hint module

class ask_sdk_model.interfaces.display.hint.Hint *(object_type=None)*  
Bases: object

---

2.14. Hosting Skills as Webservice 171
Parameters **object_type**: *(optional) str*

*Note:* This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

```
PlainText: ask_sdk_model.interfaces.display.plain_text_hint.PlainTextHint
```

```python
classmethod get_real_child_model(data)
    Returns the real base class specified by the discriminator

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.display.hint_directive module**

```python
class ask_sdk_model.interfaces.display.hint_directive.HintDirective(hint=None)
    Bases: ask_sdk_model.directive.Directive

Parameters **hint**: *(optional) ask_sdk_model.interfaces.display.hint.Hint*

    to_dict()
        Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.interfaces.display.image module**

```python
class ask_sdk_model.interfaces.display.image.Image(content_description=None, sources=None)
    Bases: object

Parameters

- **content_description**: *(optional) str*
- **sources**: *(optional) list[ask_sdk_model.interfaces.display.image_instance.ImageInstance]*

    to_dict()
        Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```
ask_sdk_model.interfaces.display.image_instance module

class ask_sdk_model.interfaces.display.image_instance.ImageInstance(url=None, size=None, width_pixels=None, height_pixels=None)

Bases: object

Parameters

  • url (optional) str-
  • size (optional) ask_sdk_model.interfaces.display.image_size.ImageSize-
  • width_pixels (optional) int-
  • height_pixels (optional) int-

to_dict()
  Returns the model properties as a dict

to_str()
  Returns the string representation of the model

ask_sdk_model.interfaces.display.image_size module

class ask_sdk_model.interfaces.display.image_size.ImageSize

Bases: enum.Enum

Allowed enum values: [X_SMALL, SMALL, MEDIUM, LARGE, X_LARGE]

to_dict()
  Returns the model properties as a dict

to_str()
  Returns the string representation of the model

ask_sdk_model.interfaces.display.list_item module

class ask_sdk_model.interfaces.display.list_item.ListItem(token=None, image=None, text_content=None)

Bases: object

Parameters

  • token (optional) str-
  • image ((optional) ask_sdk_model.interfaces.display.image.Image)-
  • text_content ((optional) ask_sdk_model.interfaces.display.text_content.TextContent)-

to_dict()
  Returns the model properties as a dict

to_str()
  Returns the string representation of the model
ask_sdk_model.interfaces.display.list_template1 module

class ask_sdk_model.interfaces.display.list_template1.ListTemplate1(token=None,
back_button=None,
background_image=None,
title=None,
list_items=None)

Bases: ask_sdk_model.interfaces.display.template.Template

Parameters

• **token** *(optional)* str-

• **back_button** *(optional)* ask_sdk_model.interfaces.display.
  back_button_behavior.BackButtonBehavior-

• **background_image** *(optional)* ask_sdk_model.interfaces.display.
  image.Image-

• **title** *(optional)* str-

• **list_items** *(optional)* list[ask_sdk_model.interfaces.display.
  list_item.ListItem]-

to_dict()
  Returns the model properties as a dict

to_str()
  Returns the string representation of the model

ask_sdk_model.interfaces.display.list_template2 module

class ask_sdk_model.interfaces.display.list_template2.ListTemplate2(token=None,
back_button=None,
background_image=None,
title=None,
list_items=None)

Bases: ask_sdk_model.interfaces.display.template.Template

Parameters

• **token** *(optional)* str-

• **back_button** *(optional)* ask_sdk_model.interfaces.display.
  back_button_behavior.BackButtonBehavior-

• **background_image** *(optional)* ask_sdk_model.interfaces.display.
  image.Image-

• **title** *(optional)* str-

• **list_items** *(optional)* list[ask_sdk_model.interfaces.display.
  list_item.ListItem]-

to_dict()
  Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.display.plain_text module

class ask_sdk_model.interfaces.display.plain_text.PlainText (text=None)
    Bases: ask_sdk_model.interfaces.display.text_field.TextField

    Parameters text ((optional) str) -

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.display.plain_text_hint module

class ask_sdk_model.interfaces.display.plain_text_hint.PlainTextHint (text=None)
    Bases: ask_sdk_model.interfaces.display.hint.Hint

    Parameters text ((optional) str) -

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.display.render_template_directive module

class ask_sdk_model.interfaces.display.render_template_directive.RenderTemplateDirective (text=None)
    Bases: ask_sdk_model.directive.Directive

    Parameters template ((optional) ask_sdk_model.interfaces.display.
        template.Template) -

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.display.rich_text module

class ask_sdk_model.interfaces.display.rich_text.RichText (text=None)
    Bases: ask_sdk_model.interfaces.display.text_field.TextField

    Parameters text ((optional) str) -

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
ask_sdk_model.interfaces.display.template module

class ask_sdk_model.interfaces.display.template.Template(object_type=None, token=None, back_button=None)

Bases: object

Parameters

• **object_type** *(optional)* *str*—

• **token** *(optional)* *str*—

• **back_button** *(optional)* *ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior*—

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

ListTemplate2: ask_sdk_model.interfaces.display.list_template2.ListTemplate2,

ListTemplate1: ask_sdk_model.interfaces.display.list_template1.ListTemplate1,

BodyTemplate7: ask_sdk_model.interfaces.display.body_template7.BodyTemplate7,

BodyTemplate6: ask_sdk_model.interfaces.display.body_template6.BodyTemplate6,

BodyTemplate3: ask_sdk_model.interfaces.display.body_template3.BodyTemplate3,

BodyTemplate2: ask_sdk_model.interfaces.display.body_template2.BodyTemplate2,

BodyTemplate1: ask_sdk_model.interfaces.display.body_template1.BodyTemplate1

```
classmethod get_real_child_model(data)
    Returns the real base class specified by the discriminator

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

ask_sdk_model.interfaces.display.text_content module

class ask_sdk_model.interfaces.display.text_content.TextContent(primary_text=None, secondary_text=None, tertiary_text=None)

Bases: object
Parameters

- **primary_text** ((optional) ask_sdk_model.interfaces.display.text_field.TextField)
- **secondary_text** ((optional) ask_sdk_model.interfaces.display.text_field.TextField)
- **tertiary_text** ((optional) ask_sdk_model.interfaces.display.text_field.TextField)

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model

**ask_sdk_model.interfaces.display.text_field module**

**class** ask_sdk_model.interfaces.display.text_field.TextField(object_type=None)
Bases: object

**Parameters**

- **object_type** ((optional) str)

**Note:** This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

RichText: ask_sdk_model.interfaces.display.rich_text.RichText,

PlainText: ask_sdk_model.interfaces.display.plain_text.PlainText

**classmethod get_real_child_model(data)**
Returns the real base class specified by the discriminator

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model

**ask_sdk_model.interfaces.geolocation package**

**Submodules**

**Note:** Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.
ask_sdk_model.interfaces.geolocation.access module

class ask_sdk_model.interfaces.geolocation.access.Access
    Bases: enum.Enum

    A string representing if Alexa has access to location services running on the hostOS of device.
    Allowed enum values: [ENABLED, DISABLED, UNKNOWN]

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.altitude module

class ask_sdk_model.interfaces.geolocation.altitude.Altitude(altitude_in_meters=None,
                                                          accuracy_in_meters=None)

    Bases: object

    An object containing the altitude information of the device.

    Parameters
    • altitude_in_meters ((optional) float) – A double representing the altitude
      of the device in meters.
    • accuracy_in_meters ((optional) float) – A double representing the accuracy
      of the altitude measurement in meters.

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.coordinate module

class ask_sdk_model.interfaces.geolocation.coordinate.Coordinate(latitude_in_degrees=None,
                                                                 longitude_in_degrees=None,
                                                                 accuracy_in_meters=None)

    Bases: object

    An object containing the location information of the device.

    Parameters
    • latitude_in_degrees ((optional) float) – A double representing the latitude
      in degrees of the device.
    • longitude_in_degrees ((optional) float) – A double representing the longi-
      tude in degrees of the device.
    • accuracy_in_meters ((optional) float) – A double representing the accuracy
      of geolocation data in meters.
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.geolocation_interface module

class ask_sdk_model.interfaces.geolocation.geolocation_interface.GeoLocationInterface
    Bases: object

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.geolocation_state module

class ask_sdk_model.interfaces.geolocation.geolocation_state.GeoLocationState (timestamp=None, coordinate=None, altitude=None, heading=None, speed=None, location_services=None)
    Bases: object

    Parameters

    • timestamp (optional str) – Specifies the time when the geolocation data was last collected on the device.
    • coordinate (optional ask_sdk_model.interfaces.geolocation.coordinate.Coordinate)
    • altitude (optional ask_sdk_model.interfaces.geolocation.altitude.Altitude)
    • heading (optional ask_sdk_model.interfaces.geolocation.heading.Heading)
    • speed (optional ask_sdk_model.interfaces.geolocation.speed.Speed)
    • location_services (optional ask_sdk_model.interfaces.geolocation.location_services.LocationServices)

    to_dict()
        Returns the model properties as a dict
to_str()
   Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.heading module

class ask_sdk_model.interfaces.geolocation.heading.Heading(direction_in_degrees=None,
   accuracy_in_degrees=None)
   Bases: object
   An object containing the heading direction information of the device.

   Parameters
   • direction_in_degrees (optional float) – A double representing the direction of the device in degrees.
   • accuracy_in_degrees (optional float) – A double representing the accuracy of the heading measurement in degrees.

to_dict()
   Returns the model properties as a dict

to_str()
   Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.location_services module

class ask_sdk_model.interfaces.geolocation.location_services.LocationServices(status=None,
   access=None)
   Bases: object
   An object containing status and access.

   Parameters
   • status ((optional) ask_sdk_model.interfaces.geolocation.status.Status) – A string representing the status of whether location services is currently running or not on the host OS of device.
   • access ((optional) ask_sdk_model.interfaces.geolocation.access.Access) – A string representing if Alexa has access to location services running on the hostOS of device.

to_dict()
   Returns the model properties as a dict

to_str()
   Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.speed module

class ask_sdk_model.interfaces.geolocation.speed.Speed(speed_in_meters_per_second=None,
   accuracy_in_meters_per_second=None)
   Bases: object
An object containing the speed information of the device.

Parameters

- **speed_in_meters_per_second** *(optional) float* – A double representing the speed of the device in meters.
- **accuracy_in_meters_per_second** *(optional) float* – A double representing the accuracy of the speed measurement in meters.

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.status module

class ask_sdk_model.interfaces.geolocation.status.Status

Bases: enum.Enum

A string representing the status of whether location services is currently running or not on the host OS of device.

Allowed enum values: [RUNNING, STOPPED]

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

ask_sdk_model.interfaces.gadget_controller package

Submodules

**Note:** Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.interfaces.gadget_controller.set_light_directive module

class ask_sdk_model.interfaces.gadget_controller.set_light_directive.SetLightDirective

Bases: ask_sdk_model.directive.Directive

Sends Alexa a command to modify the behavior of connected Echo Buttons.

Parameters
• **version** *(optional) int* – The version of the directive. Must be set to 1.

• **target_gadgets** *(optional) list[str]* – The gadget IDs that will receive the command. An empty array, or leaving this parameter out, signifies that all gadgets will receive the command.

• **parameters** *(optional) ask_sdk_model.services.gadget_controller.set_light_parameters.SetLightParameters* –

  `to_dict()`
  Returns the model properties as a dict

  `to_str()`
  Returns the string representation of the model

**ask_sdk_model.interfaces.game_engine package**

**Submodules**

*Note:* Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do `from a import C` instead of `from a.b import C`.

**ask_sdk_model.interfaces.game_engine.input_handler_event_request module**

**class** `ask_sdk_model.interfaces.game_engine.input_handler_event_request.InputHandlerEventRequest`

Bases: `ask_sdk_model.request.Request`

Sent when the conditions of an Echo Button event that your skill defined were met.

**Parameters**

• **request_id** *(optional) str* – Represents the unique identifier for the specific request.

• **timestamp** *(optional) datetime* – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• **locale** *(optional) str* – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
• **originating_request_id** ((optional) *str*) – The corresponding identifier of the request that started the input handler.

• **events** ((optional) *list*[ask_sdk_model.services.game_engine.
  input_handler_event.InputHandlerEvent]) –

  to_dict()  
  Returns the model properties as a dict

  to_str()  
  Returns the string representation of the model

```
ask_sdk_model.interfaces.game_engine.start_input_handler_directive module

class ask_sdk_model.interfaces.game_engine.start_input_handler_directive.StartInputHandlerDirective

Bases: ask_sdk_model.directive.Directive

Parameters

  • **timeout** ((optional) *int*) – The maximum run time for this Input Handler, in milliseconds. Although this parameter is required, you can specify events with conditions on which to end the Input Handler earlier.

  • **proxies** ((optional) *list*[str]) – Names for unknown gadget IDs to use in recognizers, allocated on a first-come, first-served basis.

  • **recognizers** ((optional) *dict*(str, ask_sdk_model.services.game_engine.recognizer.Recognizer)) – Conditions that, at any moment, are either true or false. You use recognizers when you specify the conditions under which your skill is notified of Echo Button input.

  • **events** ((optional) *dict*(str, ask_sdk_model.services.game_engine.event.Event)) – The logic that determines when your skill is notified of Echo Button input. Events are listed here as object keys, where the keys specify the name of an event.

  to_dict()  
  Returns the model properties as a dict

  to_str()  
  Returns the string representation of the model
```

```
ask_sdk_model.interfaces.game_engine.stop_input_handler_directive module

class ask_sdk_model.interfaces.game_engine.stop_input_handler_directive.StopInputHandlerDirective

Bases: ask_sdk_model.directive.Directive

Parameters originating_request_id ((optional) *str*) – The

  &amp;x60;requestId&amp;x60; of the request that started the input handler.

2.14. Hosting Skills as Webservice

183
**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

---

**ask_sdk_model.interfaces.messaging package**

**Submodules**

---

**Note:** Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

---

**ask_sdk_model.interfaces.messaging.message_received_request module**

**class** ask_sdk_model.interfaces.messaging.message_received_request.MessageReceivedRequest

**Bases:** ask_sdk_model.request.Request

**Parameters**

- **request_id** *(optional) str* – Represents the unique identifier for the specific request.

- **timestamp** *(optional) datetime* – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

- **locale** *(optional) str* – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

- **message** *(optional) dict(str, object)* –

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

---

**ask_sdk_model.interfaces.monetization package**

**Subpackages**

---

**ask_sdk_model.interfaces.monetization.v1 package**

---

Chapter 2. SDK Features
Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.interfaces.monetization.v1.in_skill_product module

```python
class ask_sdk_model.interfaces.monetization.v1.in_skill_product.InSkillProduct(product_id=None):
    Bases: object
    Entity to define In Skill Product over which actions will be performed.
    Parameters product_id (optional str) – The product ID of In Skill Product.
    to_dict()
    Returns the model properties as a dict
    to_str()
    Returns the string representation of the model
```

ask_sdk_model.interfaces.monetization.v1.purchase_result module

```python
class ask_sdk_model.interfaces.monetization.v1.purchase_result.PurchaseResult:
    Bases: enum.Enum
    Response from purchase directives: * ACCEPTED - User have accepted the offer to purchase the product * DECLINED - User have declined the offer to purchase the product * NOT_ENTITLED - User tries to cancel/return a product he/she is not entitled to. * ALREADY_PURCHASED - User has already purchased the product * ERROR - An internal error occurred
    Allowed enum values: [ACCEPTED, DECLINED, NOT_ENTITLED, ERROR, ALREADY_PURCHASED]
    to_dict()
    Returns the model properties as a dict
    to_str()
    Returns the string representation of the model
```

ask_sdk_model.interfaces.playbackcontroller package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

2.14. Hosting Skills as Webservice 185
ask_sdk_model.interfaces.playbackcontroller.next_command_issued_request module

class ask_sdk_model.interfaces.playbackcontroller.next_command_issued_request.NextCommandIssuedRequest(request_id=None, timestamp=None, locale=None):
    Bases: ask_sdk_model.request.Request

    Parameters
    • request_id ((optional) str) – Represents the unique identifier for the specific request.
    • timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
    • locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.playbackcontroller.pause_command_issued_request module

class ask_sdk_model.interfaces.playbackcontroller.pause_command_issued_request.PauseCommandIssuedRequest(request_id=None, timestamp=None, locale=None):
    Bases: ask_sdk_model.request.Request

    Parameters
    • request_id ((optional) str) – Represents the unique identifier for the specific request.
    • timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
    • locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model
ask_sdk_model.interfaces.playbackcontroller.play_command_issued_request module

class ask_sdk_model.interfaces.playbackcontroller.play_command_issued_request.PlayCommandIssuedRequest:

Bases: ask_sdk_model.request.Request

Parameters

- **request_id** ((optional) str) – Represents the unique identifier for the specific request.
- **timestamp** ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

ask_sdk_model.interfaces.playbackcontroller.previous_command_issued_request module

class ask_sdk_model.interfaces.playbackcontroller.previous_command_issued_request.PreviousCommandIssuedRequest:

Bases: ask_sdk_model.request.Request

Parameters

- **request_id** ((optional) str) – Represents the unique identifier for the specific request.
- **timestamp** ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

ask_sdk_model.interfaces.system package

Submodules
Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.interfaces.system.error module

class ask_sdk_model.interfaces.system.error.Error(object_type=None, message=None)

Bases: object

Parameters

• object_type ((optional) ask_sdk_model.interfaces.system.error_type.ErrorType) –

• message ((optional) str) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.system.error_cause module

class ask_sdk_model.interfaces.system.error_cause.ErrorCause(request_id=None)

Bases: object

Parameters request_id ((optional) str) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.system.error_type module

class ask_sdk_model.interfaces.system.error_type.ErrorType

Bases: enum.Enum

Allowed enum values: [INVALID_RESPONSE, DEVICE_COMMUNICATION_ERROR, INTERNAL_SERVICE_ERROR]

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model
ask_sdk_model.interfaces.system.exception_encountered_request module

class ask_sdk_model.interfaces.system.exception_encountered_request.ExceptionEncounteredRequest

    Bases: ask_sdk_model.request.Request

    Parameters

        • request_id ((optional) str) – Represents the unique identifier for the specific request.

        • timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

        • locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

        • error ((optional) ask_sdk_model.interfaces.system.error.Error) –

        • cause ((optional) ask_sdk_model.interfaces.system.error_cause.ErrorCause) –

    to_dict

        Returns the model properties as a dict

    to_str

        Returns the string representation of the model

ask_sdk_model.interfaces.system.system_state module

class ask_sdk_model.interfaces.system.system_state.SystemState

    Bases: object

    Parameters

        • application ((optional) ask_sdk_model.application.Application) –

        • user ((optional) ask_sdk_model.user.User) –

        • device ((optional) ask_sdk_model.device.Device) –

        • api_endpoint ((optional) str) –

        • api_access_token ((optional) str) –

    to_dict

        Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.interfaces.videoapp package

Submodules

**Note:** Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

---

ask_sdk_model.interfaces.videoapp.launch_directive module

class ask_sdk_model.interfaces.videoapp.launch_directive.LaunchDirective(video_item=None)
    Bases: ask_sdk_model.directive.Directive
    Parameters video_item ((optional) ask_sdk_model.interfaces.videoapp.video_item.VideoItem)-
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.videoapp.metadata module

class ask_sdk_model.interfaces.videoapp.metadata.Metadata(title=None, subtitle=None)
    Bases: object
    Parameters
        • title ((optional) str)-
        • subtitle ((optional) str)-
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model

ask_sdk_model.interfaces.videoapp.video_app_interface module

class ask_sdk_model.interfaces.videoapp.video_app_interface.VideoAppInterface
    Bases: object
    to_dict()
        Returns the model properties as a dict
to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.videoapp.video_item module

class ask_sdk_model.interfaces.videoapp.video_item.VideoItem

Bases: object

Parameters

• source ((optional) str) –
• metadata ((optional) ask_sdk_model.interfaces.videoapp.metadata.Metadata)

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.viewport package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.interfaces.viewport.experience module

class ask_sdk_model.interfaces.viewport.experience.Experience

Bases: object

An experience represents a viewing mode used to interact with the device.

Parameters

• arc_minute_width ((optional) float) – The number of horizontal arc minutes the viewport occupies in the user’s visual field when viewed within this experience.

• arc_minute_height ((optional) float) – The number of vertical arc minutes the viewport occupies in the user’s visual field when viewed within this experience.

• can_rotate ((optional) bool) – Indicates if the viewport can be rotated through 90 degrees.

• can_resize ((optional) bool) – Indicates if the viewport can be resized, limiting the area which can be used to render the APL response.
to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.viewport.keyboard module

class ask_sdk_model.interfaces.viewport.keyboard.Keyboard
    Bases: enum.Enum

    Represents a physical button input mechanism which can be used to interact with elements shown on the viewport.

    Allowed enum values: [DIRECTION]

to_dict()

    Returns the model properties as a dict

to_str()

    Returns the string representation of the model

ask_sdk_model.interfaces.viewport.shape module

class ask_sdk_model.interfaces.viewport.shape.Shape
    Bases: enum.Enum

    The shape of the viewport.

    Allowed enum values: [RECTANGLE, ROUND]

to_dict()

    Returns the model properties as a dict

to_str()

    Returns the string representation of the model

ask_sdk_model.interfaces.viewport.touch module

class ask_sdk_model.interfaces.viewport.touch.Touch
    Bases: enum.Enum

    Represents a type of touch input supported by the device.

    Allowed enum values: [SINGLE]

to_dict()

    Returns the model properties as a dict

to_str()

    Returns the string representation of the model
ask_sdk_model.interfaces.viewport.viewport_state module

class ask_sdk_model.interfaces.viewport.viewport_state.ViewportState(experiences=None, shape=None, pixel_width=None, pixel_height=None, dpi=None, current_pixel_width=None, current_pixel_height=None, touch=None, keyboard=None, video=None):

Bases: object

This object contains the characteristics related to the device’s viewport.

Parameters

- **experiences** ((optional) list[ask_sdk_model.interfaces.viewport.experience.Experience]) – The experiences supported by the device, in descending order of arcMinuteWidth and arcMinuteHeight.
- **shape** ((optional) ask_sdk_model.interfaces.viewport.shape.Shape)
- **pixel_width** (optional) float – The number of pixels present in the viewport at its maximum width.
- **pixel_height** (optional) float – The number of pixels present in the viewport at its maximum height.
- **dpi** (optional) float – The pixel density of the viewport.
- **current_pixel_width** (optional) float – The number of horizontal pixels in the viewport that are currently available for Alexa to render an experience.
- **current_pixel_height** (optional) float – The number of vertical pixels in the viewport that are currently available for Alexa to render an experience.
- **touch** (optional) list[ask_sdk_model.interfaces.viewport.touch.Touch] – The types of touch supported by the device. An empty array indicates no touch support.
- **keyboard** (optional) list[ask_sdk_model.interfaces.viewport.keyboard.Keyboard] – The physical button input mechanisms supported by the device. An empty array indicates physical button input is unsupported.
- **video** (optional) ask_sdk_model.interfaces.viewport.viewport_state_video.Video

**to_dict**()

Returns the model properties as a dict

**to_str**()

Returns the string representation of the model
Module contents

ask_sdk_model.services package

Subpackages

ask_sdk_model.services.device_address package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.services.device_address.address module

class ask_sdk_model.services.device_address.address.Address (address_line1=None, address_line2=None, address_line3=None, country_code=None, state_or_region=None, city=None, district_or_county=None, postal_code=None)

Bases: object

Represents the full address response from the service.

Parameters

- address_line1 ((optional) str)
- address_line2 ((optional) str)
- address_line3 ((optional) str)
- country_code ((optional) str)
- state_or_region ((optional) str)
- city ((optional) str)
- district_or_county ((optional) str)
- postal_code ((optional) str)

to_dict ()
Returns the model properties as a dict

to_str ()
Returns the string representation of the model
ask_sdk_model.services.device_address.device_address_service_client module

class ask_sdk_model.services.device_address.device_address_service_client.DeviceAddressServiceClient
    Bases: ask_sdk_model.services.base_service_client.BaseServiceClient
    ServiceClient for calling the DeviceAddressService APIs.

    Parameters
    api_configuration (ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of ask_sdk_model.services.api_configuration.ApiConfiguration

    get_country_and_postal_code(device_id, **kwargs)
    Gets the country and postal code of a device

    Parameters
    device_id (str) – (required) The device Id for which to get the country and postal code

    Return type
    Union[ShortAddress, Error]

    get_full_address(device_id, **kwargs)
    Gets the address of a device

    Parameters
    device_id (str) – (required) The device Id for which to get the address

    Return type
    Union[Address, Error]

ask_sdk_model.services.device_address.error module

class ask_sdk_model.services.device_address.error.Error
    Bases: object

    Parameters

    • object_type ((optional) str) – The corresponding type of the http status code being returned.

    • message ((optional) str) – A human readable description of error.

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.services.device_address.short_address module

class ask_sdk_model.services.device_address.short_address.ShortAddress
    Bases: object

    Parameters

    • country_code ((optional) str) –

    • postal_code ((optional) str) –

    to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.services.directive package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.services.directive.directive module

class ask_sdk_model.services.directive.directive.Directive(object_type=None)
    Bases: object

    Parameters object_type ((optional) str) -

    Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

    VoicePlayer.Speak:
    ask_sdk_model.services.directive.speak_directive.SpeakDirective

    @classmethod get_real_child_model(cls, data)
    Returns the real base class specified by the discriminator

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.services.directive.directive_service_client module

class ask_sdk_model.services.directive.directive_service_client.DirectiveServiceClient(api_configuration)
    Bases: ask_sdk_model.services.base_service_client.BaseServiceClient

    ServiceClient for calling the DirectiveService APIs.

    Parameters api_configuration (ask_sdk_model.services.api_configuration.ApiConfiguration) - Instance of ask_sdk_model.services.api_configuration.ApiConfiguration

    enqueue(send_directive_request, **kwargs)
    Send directives to Alexa.
Parameters send_directive_request (ask_sdk_model.services.directive.send_directive_request.SendDirectiveRequest) – (required) Represents the request object to send in the payload.

Return type None

ask_sdk_model.services.directive.error module

class ask_sdk_model.services.directive.error.Error(code=None, message=None)
Bases: object

Parameters

• code ((optional) int) – error code to find more information in developer.amazon.com.

• message ((optional) str) – Readable description of error.

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.services.directive.header module

class ask_sdk_model.services.directive.header.Header(request_id=None)
Bases: object

Parameters request_id (optional) str – This represents the current requestId for what the skill/speechlet was invoked.


to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.services.directive.send_directive_request module

class ask_sdk_model.services.directive.send_directive_request.SendDirectiveRequest(header=None, directive=None)
Bases: object

Send Directive Request payload.

Parameters

• header ((optional) ask_sdk_model.services.directive.header.Header) – contains the header attributes of the send directive request.


to_dict()
    Returns the model properties as a dict

2.14. Hosting Skills as Webservice 197
to_str()

Returns the string representation of the model

ask_sdk_model.services.directive.speak_directive module

class ask_sdk_model.services.directive.speak_directive.SpeakDirective(speech=None)
    Bases: ask_sdk_model.services.directive.directive.Directive

    Parameters
    speech ((optional) str) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.gadget_controller package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class
directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b
import C.

ask_sdk_model.services.gadget_controller.animation_step module

class ask_sdk_model.services.gadget_controller.animation_step.AnimationStep(duration_ms=None, color=None, blend=None)

    Bases: object

    Parameters

    • duration_ms ((optional) int) – The duration in milliseconds to render this step.
    • color ((optional) str) – The color to render specified in RGB hexadecimal values.
        There are a number of Node.js libraries available for working with color.
    • blend ((optional) bool) – A boolean that indicates whether to interpolate from the
        previous color into this one over the course of this directive’s durationMs.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model
ask_sdk_model.services.gadget_controller.light_animation module

class ask_sdk_model.services.gadget_controller.light_animation.LightAnimation (repeat=None, target_lights=None, sequence=None)

Bases: object

Parameters

- **repeat** *(optional) int* – The number of times to play this animation.
- **target_lights** *(optional) list[str]* – An array of strings that represent the light addresses on the target gadgets that this animation will be applied to. Because the Echo Button has one light only, use ["1"] to signify that this animation should be sent to light one.
- **sequence** *(optional) list[ask_sdk_model.services.gadget_controller.animation_step.AnimationStep]* – The animation steps to render in order. The maximum number of steps that you can define is 38. The minimum is 0. Each step must have the following fields, all of which are required.

```
to_dict()
  Returns the model properties as a dict
to_str()
  Returns the string representation of the model
```

ask_sdk_model.services.gadget_controller.set_light_parameters module

class ask_sdk_model.services.gadget_controller.set_light_parameters.SetLightParameters (trigger_event=None, trigger_event_time_ms=None, animations=None)

Bases: object

Arguments that pertain to animating the buttons.

Parameters

- **trigger_event** *(optional) ask_sdk_model.services.gadget_controller.trigger_event_type.TriggerEventType*
- **trigger_event_time_ms** *(optional) int*
- **animations** *(optional) list[ask_sdk_model.services.gadget_controller.light_animation.LightAnimation]*

```
to_dict()
  Returns the model properties as a dict
to_str()
  Returns the string representation of the model
```

2.14. Hosting Skills as Webservice
**ask_sdk_model.services.gadget_controller.trigger_event_type module**

```python
class ask_sdk_model.services.gadget_controller.trigger_event_type.TriggerEventType
    Bases: enum.Enum

    The action that triggers the animation. Possible values are as follows:
    * `buttonDown` - Play the animation when the button is pressed.
    * `buttonUp` - Play the animation when the button is released.
    * `none` - Play the animation as soon as it arrives.
    Allowed enum values: [buttonDown, buttonUp, none]

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.services.game_engine package**

**Submodules**

*Note:* Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package `a` has module `b` with class `C`, you can do `from a import C` instead of `from a.b import C`.

**ask_sdk_model.services.game_engine.deviation_recognizer module**

```python
class ask_sdk_model.services.game_engine.deviation_recognizer.DeviationRecognizer(recognizer=None)
    Bases: ask_sdk_model.services.game_engine.recognizer.Recognizer

    The deviation recognizer returns true when another specified recognizer reports that the player has deviated from its expected pattern.

    Parameters
    recognizer : (optional) str
        The name of the recognizer that defines a pattern that must not be deviated from.

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.services.game_engine.event module**

```python
class ask_sdk_model.services.game_engine.event.Event(should_end_input_handler=None,
    meets=None, fails=None, reports=None,
    maximum_invocations=None, trigger_time_milliseconds=None)
    Bases: object
```
The events object is where you define the conditions that must be met for your skill to be notified of Echo Button input. You must define at least one event.

**Parameters**

- **should_end_input_handler** *(optional) bool* – Whether the Input Handler should end after this event fires. If true, the Input Handler will stop and no further events will be sent to your skill unless you call StartInputHandler again.

- **meets** *(optional) list[str]* –

- **fails** *(optional) list[str]* –

- **reports** *(optional) ask_sdk_model.services.game_engine.event_reporting_type.EventReportingType* –

- **maximum_invocations** *(optional) int* – Enables you to limit the number of times that the skill is notified about the same event during the course of the Input Handler. The default value is 1. This property is mutually exclusive with triggerTimeMilliseconds.

- **trigger_time_milliseconds** *(optional) int* – Adds a time constraint to the event. Instead of being considered whenever a raw button event occurs, an event that has this parameter will only be considered once at triggerTimeMilliseconds after the Input Handler has started. Because a time-triggered event can only fire once, the maximumInvocations value is ignored. Omit this property entirely if you do not want to time-constrain the event.

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

**ask_sdk_model.services.game_engine.event_reporting_type module**

**class** ask_sdk_model.services.game_engine.event_reporting_type.EventReportingType

**Bases:** enum.Enum

Specifies what raw button presses to put in the inputEvents field of the event. * history - All button presses since this Input Handler was started. * matches - Just the button presses that contributed to this event (that is, were in the recognizers). To receive no raw button presses, leave this array empty or do not specify it at all.

Allowed enum values: [history, matches]

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

**ask_sdk_model.services.game_engine.input_event module**

**class** ask_sdk_model.services.game_engine.input_event.InputEvent

**Bases:** object
Parameters

- **gadget_id** *(optional)* `str` – The identifier of the Echo Button in question. It matches the gadgetId that you will have discovered in roll call.

- **timestamp** *(optional)* `str` – The event’s original moment of occurrence, in ISO format.

- **action** *(optional)* `ask_sdk_model.services.game_engine.input_event_action_type.InputEventActionType` –

- **color** *(optional)* `str` – The hexadecimal RGB values of the button LED at the time of the event.

- **feature** *(optional)* `str` – For gadgets with multiple features, this is the feature that the event represents. Echo Buttons have one feature only, so this is always `press`.

```python
to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model
```

**ask_sdk_model.services.game_engine.input_event_action_type module**

```python
class ask_sdk_model.services.game_engine.input_event_action_type.InputEventActionType
Bases: enum.Enum
    Either "down" for a button pressed or "up" for a button released.
    Allowed enum values: [down, up]

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.services.game_engine.input_handler_event module**

```python
class ask_sdk_model.services.game_engine.input_handler_event.InputHandlerEvent
    (name=None, input_events=None)
Bases: object

Parameters

- **name** *(optional)* `str` – The name of the event as you defined it in your GameEngine.StartInputHandler directive.

- **input_events** *(optional)* `list[ask_sdk_model.services.game_engine.input_event.InputEvent]` – A chronologically ordered report of the raw Button Events that contributed to this Input Handler Event.

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```
**ask_sdk_model.services.game_engine.pattern module**

```python
class ask_sdk_model.services.game_engine.pattern.Pattern(gadget_ids=None, colors=None, action=None, repeat=None):
    Bases: object
```

An object that provides all of the events that need to occur, in a specific order, for this recognizer to be true. Omitting any parameters in this object means "match anything".

**Parameters**

- `gadget_ids` *(optional) list[str]* – A whitelist of gadgetIds that are eligible for this match.
- `colors` *(optional) list[str]* – A whitelist of colors that are eligible for this match.
- `action` *(optional) ask_sdk_model.services.game_engine.input_event_action_type.InputEventActionType* –
- `repeat` *(optional) int* – The number of times that the specified action must occur to be considered complete.

```python
to_dict()  
    Returns the model properties as a dict
to_str()  
    Returns the string representation of the model
```

**ask_sdk_model.services.game_engine.pattern_recognizer module**

```python
class ask_sdk_model.services.game_engine.pattern_recognizer.PatternRecognizer(  
    anchor=None, fuzzy=None, gadget_ids=None, actions=None, pattern=None):  
    Bases: ask_sdk_model.services.game_engine.recognizer.Recognizer
```

This recognizer is true when all of the specified events have occurred in the specified order.

**Parameters**

- `anchor` *(optional) ask_sdk_model.services.game_engine.pattern_recognizer_anchor_type.PatternRecognizerAnchorType* –
- `fuzzy` *(optional) bool* – When true, the recognizer will ignore additional events that occur between the events specified in the pattern.
- `gadget_ids` *(optional) list[str]* – The gadget IDs of the Echo Buttons to consider in this pattern recognizer.
- `actions` *(optional) list[str]* – The actions to consider in this pattern recognizer. All other actions will be ignored.
• **pattern** *(optional)* `list[ask_sdk_model.services.game_engine.pattern.Pattern]` – An object that provides all of the events that need to occur, in a specific order, for this recognizer to be true. Omitting any parameters in this object means "match anything".

```python
to_dict()
Returns the model properties as a dict
```

```python
to_str()
Returns the string representation of the model
```

---

**ask_sdk_model.services.game_engine.pattern_recognizer_anchor_type module**

```python
class ask_sdk_model.services.game_engine.pattern_recognizer_anchor_type.PatternRecognizerAnchorType:
Bases: enum.Enum

Where the pattern must appear in the history of this input handler. * start - (Default) The first event in the pattern must be the first event in the history of raw Echo Button events. * end - The last event in the pattern must be the last event in the history of raw Echo Button events. * anywhere - The pattern may appear anywhere in the history of raw Echo Button events.

Allowed enum values: [start, end, anywhere]
```

```python
to_dict()
Returns the model properties as a dict
```

```python
to_str()
Returns the string representation of the model
```

---

**ask_sdk_model.services.game_engine.progress_recognizer module**

```python
class ask_sdk_model.services.game_engine.progress_recognizer.ProgressRecognizer:
Bases: ask_sdk_model.services.game_engine.recognizer.Recognizer

This recognizer consults another recognizer for the degree of completion, and is true if that degree is above the specified threshold. The completion parameter is specified as a decimal percentage.

Parameters

• **recognizer** *(optional)* `str` – The name of a recognizer for which to track the progress.

• **completion** *(optional)* `float` – The completion threshold, as a decimal percentage, of the specified recognizer before which this recognizer becomes true.

```python
to_dict()
Returns the model properties as a dict
```

```python
to_str()
Returns the string representation of the model
```
ask_sdk_model.services.game_engine.recognizer module

class ask_sdk_model.services.game_engine.recognizer.Recognizer(object_type=None)
    Bases: object

    Recognizers are conditions that, at any moment, are either true or false, based on all the raw button events that
    the Input Handler has received in the time elapsed since the Input Handler session started.

    Parameters object_type ((optional) str) –

    Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that
    sets type variable.

    progress:
    ask_sdk_model.services.game_engine.progress_recognizer.ProgressRecognizer,

    match:
    ask_sdk_model.services.game_engine.pattern_recognizer.PatternRecognizer,

    deviation: ask_sdk_model.services.game_engine.deviation_recognizer.DeviationRecognizer

    @classmethod get_real_child_model(data)
        Returns the real base class specified by the discriminator

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model

ask_sdk_model.services.list_management package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class
directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b
import C.
ask_sdk_model.services.list_management.alexa_list module

class ask_sdk_model.services.list_management.alexa_list.AlexaList(list_id=None, name=None, state=None, version=None, items=None, links=None)

Bases: object

Parameters

• list_id (optional) str-
• name (optional) str-
• state (optional) ask_sdk_model.services.list_management.list_state.ListState-
• version (optional) int-
• items (optional) list[ask_sdk_model.services.list_management.alexa_list_item.AlexaListItem]-
• links (optional) ask_sdk_model.services.list_management.links.Links-

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

ask_sdk_model.services.list_management.alexa_list_item module

class ask_sdk_model.services.list_management.alexa_list_item.AlexaListItem(id=None, version=None, value=None, status=None, created_time=None, updated_time=None, href=None)

Bases: object

Parameters

• id (optional) str-
• version (optional) int-
• value (optional) str-
• status (optional) ask_sdk_model.services.list_management.list_item_state.ListItemState-
• created_time (optional) str-
**updated_time** *(optional) str* –

**href** *(optional) str* – URL to retrieve the item from.

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

**ask_sdk_model.services.list_management.alexa_list_metadata module**

class ask_sdk_model.services.list_management.alexa_list_metadata.AlexaListMetadata(list_id=None, name=None, state=None, version=None, status_map=None):

Bases: object

Parameters

• **list_id** *(optional) str*–

• **name** *(optional) str*–

• **state** *(optional) ask_sdk_model.services.list_management.list_state.ListState*–

• **version** *(optional) int*–

• **status_map** *(optional) list[ask_sdk_model.services.list_management.status.Status]*–

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

**ask_sdk_model.services.list_management.alexa_lists_metadata module**

class ask_sdk_model.services.list_management.alexa_lists_metadata.AlexaListsMetadata(lists=None):

Bases: object

Parameters **lists** *(optional) list[ask_sdk_model.services.list_management.alexa_list_metadata.AlexaListMetadata]*–

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

2.14. Hosting Skills as Webservice
ask_sdk_model.services.list_management.create_list_item_request module

class ask_sdk_model.services.list_management.create_list_item_request.CreateListItemRequest

    Bases: object

    Parameters
    • value ((optional) str)-
    • status ((optional) ask_sdk_model.services.list_management.
      list_item_state.ListItemState)-

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.services.list_management.create_list_request module

class ask_sdk_model.services.list_management.create_list_request.CreateListRequest

    Bases: object

    Parameters
    • name ((optional) str)-
    • state ((optional) ask_sdk_model.services.list_management.
      list_state.ListState)-

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.services.list_management.error module

class ask_sdk_model.services.list_management.error.Error

    Bases: object

    Parameters
    • object_type ((optional) str)-
    • message ((optional) str)-

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model
ask_sdk_model.services.list_management.forbidden_error module

class ask_sdk_model.services.list_management.forbidden_error.ForbiddenError (message=None)
    Bases: object
    Parameters message (optional) str -
    to_dict ()
        Returns the model properties as a dict
    to_str ()
        Returns the string representation of the model

ask_sdk_model.services.list_management.links module

class ask_sdk_model.services.list_management.links.Links (next=None)
    Bases: object
    Parameters next (optional) str -
    to_dict ()
        Returns the model properties as a dict
    to_str ()
        Returns the string representation of the model

ask_sdk_model.services.list_management.list_body module

class ask_sdk_model.services.list_management.list_body.ListBody (list_id=None)
    Bases: object
    Parameters list_id (optional) str -
    to_dict ()
        Returns the model properties as a dict
    to_str ()
        Returns the string representation of the model

ask_sdk_model.services.list_management.list_created_event_request module

class ask_sdk_model.services.list_management.list_created_event_request.ListCreatedEventRequest (request_id=None,
                                                                                              times-tamp=None,
                                                                                              lo-cale=None,
                                                                                              body=None,
                                                                                              event_creation_time=None,
                                                                                              event_publishing_time=None)
    Bases: ask_sdk_model.request.Request
    Parameters
        • request_id (optional) str - Represents the unique identifier for the specific request.
**timestamp** *(optional) datetime* – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

**locale** *(optional) str* – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

**body** *(optional) ask_sdk_model.services.list_management.list_body.ListBody* –

**event_creation_time** *(optional) datetime* –

**event_publishing_time** *(optional) datetime* –

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model

```python
ask_sdk_model.services.list_management.list_deleted_event_request module

class ask_sdk_model.services.list_management.list_deleted_event_request.ListDeletedEventRequest

Bases: ask_sdk_model.request.Request

Parameters

**request_id** *(optional) str* – Represents the unique identifier for the specific request.

**timestamp** *(optional) datetime* – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

**locale** *(optional) str* – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

**body** *(optional) ask_sdk_model.services.list_management.list_body.ListBody* –

**event_creation_time** *(optional) datetime* –

**event_publishing_time** *(optional) datetime* –

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model
```
ask_sdk_model.services.list_management.list_item_body module

class ask_sdk_model.services.list_management.list_item_body.ListItemBody(list_id=None, list_item_ids=None)

Bases: object

Parameters

- **list_id** (optional `str`) -
- **list_item_ids** (optional `list[str]`) -

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model

ask_sdk_model.services.list_management.list_item_state module

class ask_sdk_model.services.list_management.list_item_state.ListItemState

Bases: enum.Enum

Allowed enum values: [active, completed]

**to_dict()**
Returns the model properties as a dict

**to_str()**
Returns the string representation of the model

ask_sdk_model.services.list_management.list_items_created_event_request module

class ask_sdk_model.services.list_management.list_items_created_event_request.ListItemsCreatedEventRequest(request_id=None, timestamp=None, locale=None, body=None, event_creation_time=None, event_publishing_time=None)

Bases: ask_sdk_model.request.Request

Parameters

- **request_id** (optional `str`) – Represents the unique identifier for the specific request.
- **timestamp** (optional `datetime`) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (optional `str`) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
- **body** (optional `ask_sdk_model.services.list_management.list_item_body.ListItemBody`) –
- **event_creation_time** (optional `datetime`) –
• `event_publishing_time`((optional) `datetime`)–

`to_dict()`
Returns the model properties as a dict

`to_str()`
Returns the string representation of the model

`ask_sdk_model.services.list_management.list_items_deleted_event_request` module

class `ask_sdk_model.services.list_management.list_items_deleted_event_request.ListItemsDeletedEventRequest`

Bases: `ask_sdk_model.request.Request`

Parameters

• `request_id`((optional) `str`) – Represents the unique identifier for the specific request.

• `timestamp`((optional) `datetime`) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• `locale`((optional) `str`) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• `body`((optional) `ask_sdk_model.services.list_management.list_item_body.ListItemBody`)–

• `event_creation_time`((optional) `datetime`)–

• `event_publishing_time`((optional) `datetime`)–

`to_dict()`
Returns the model properties as a dict

`to_str()`
Returns the string representation of the model

`ask_sdk_model.services.list_management.list_items_updated_event_request` module

class `ask_sdk_model.services.list_management.list_items_updated_event_request.ListItemsUpdatedEventRequest`

Bases: `ask_sdk_model.request.Request`

Parameters
• request_id ((optional) str) – Represents the unique identifier for the specific request.

• timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• body ((optional) ask_sdk_model.services.list_management.list_item_body.ListItemBody) –

• event_creation_time ((optional) datetime) –

• event_publishing_time ((optional) datetime) –

to_dict()  
Returns the model properties as a dict
to_str()  
Returns the string representation of the model

ask_sdk_model.services.list_management.list_management_service_client module

class ask_sdk_model.services.list_management.list_management_service_client.ListManagementServiceClient  
Bases: ask_sdk_model.services.base_service_client.BaseServiceClient

ServiceClient for calling the ListManagementService APIs.

Parameters  
api_configuration (ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of ask_sdk_model.services.api_configuration.ApiConfiguration

create_list (create_list_request, **kwargs)  
This API creates a custom list. The new list name must be different than any existing list name.

Parameters  
create_list_request (ask_sdk_model.services.list_management.create_list_request.CreateListRequest) – (required)

Return type Union[Error, AlexaListMetadata]

create_list_item (list_id, create_list_item_request, **kwargs)  
This API creates an item in an active list or in a default list.

Parameters

• list_id (str) – (required) The customer’s listId retrieved from a getListsMetadata call.

• create_list_item_request (ask_sdk_model.services.list_management.create_list_item_request.CreateListItemRequest) – (required)

Return type Union[AlexaListItem, Error]

delete_list (list_id, **kwargs)  
This API deletes a customer custom list.

Parameters  
list_id (str) – (required) Value of the customer’s listId retrieved from a getListsMetadata call

Return type None
delete_list_item \((list_id, item_id, **kwargs)\)

This API deletes an item in the specified list.

**Parameters**

- \(list_id\) \((str)\) – (required) The customer’s listId is retrieved from a getListsMetadata call.
- \(item_id\) \((str)\) – (required) The customer’s itemId is retrieved from a GetList call.

**Return type** None

get_list \((list_id, status, **kwargs)\)

Retrieves the list metadata including the items in the list with requested status.

**Parameters**

- \(list_id\) \((str)\) – (required) Retrieved from a call to GetListsMetadata to specify the listId in the request path.
- \(status\) \((str)\) – (required) Specify the status of the list.

**Return type** Union[AlexaList, Error]

get_list_item \((list_id, item_id, **kwargs)\)

This API can be used to retrieve single item with in any list by listId and itemId. This API can read list items from an archived list. Attempting to read list items from a deleted list return an ObjectNotFound 404 error.

**Parameters**

- \(list_id\) \((str)\) – (required) Retrieved from a call to getListsMetadata
- \(item_id\) \((str)\) – (required) itemId within a list is retrieved from a getList call

**Return type** Union[AlexaListItem, Error]

get_lists_metadata \((**kwargs)\)

Retrieves the metadata for all customer lists, including the customer’s default lists.

**Return type** Union[ForbiddenError, Error, AlexaListsMetadata]

update_list \((list_id, update_list_request, **kwargs)\)

This API updates a custom list. Only the list name or state can be updated. An Alexa customer can turn an archived list into an active one.

**Parameters**

- \(list_id\) \((str)\) – (required) Value of the customer’s listId retrieved from a getListsMetadata call.
- \(update_list_request\) \((ask_sdk_model.services.list_management.UpdateListRequest)\) – (required)

**Return type** Union[Error, AlexaListMetadata]

update_list_item \((list_id, item_id, update_list_item_request, **kwargs)\)

API used to update an item value or item status.

**Parameters**

- \(list_id\) \((str)\) – (required) Customer’s listId
- \(item_id\) \((str)\) – (required) itemId to be updated in the list
update_list_item_request (ask_sdk_model.services.
list_management.update_list_item_request.
UpdateListItemRequest) – (required)

Return type Union[AlexaListItem, Error]

ask_sdk_model.services.list_management.list_state module

class ask_sdk_model.services.list_management.list_state.ListState
    Bases: enum.Enum
    Allowed enum values: [active, archived]
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model

ask_sdk_model.services.list_management.list_updated_event_request module

class ask_sdk_model.services.list_management.list_updated_event_request.ListUpdatedEventRequest
    Bases: ask_sdk_model.request.Request

Parameters
    • request_id ((optional) str) – Represents the unique identifier for the specific request.
    • timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
    • locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
    • body ((optional) ask_sdk_model.services.list_management.list_body.ListBody) –
    • event_creation_time ((optional) datetime) –
    • event_publishing_time ((optional) datetime) –

    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model
ask_sdk_model.services.list_management.status module

class ask_sdk_model.services.list_management.status.Status(url=None, status=None):
    Bases: object

    Parameters
    • url ((optional) str) –
    • status ((optional) ask_sdk_model.services.list_management.list_item_state.ListItemState) –

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.services.list_management.update_list_item_request module

class ask_sdk_model.services.list_management.update_list_item_request.UpdateListItemRequest(value=None, status=None, version=None):
    Bases: object

    Parameters
    • value ((optional) str) – New item value
    • status ((optional) ask_sdk_model.services.list_management.list_item_state.ListItemState) – Item Status
    • version ((optional) int) – Item version when it was read.

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.services.list_management.update_list_request module

class ask_sdk_model.services.list_management.update_list_request.UpdateListRequest(name=None, state=None, version=None):
    Bases: object

    Parameters
    • name ((optional) str) –
    • state ((optional) ask_sdk_model.services.list_management.list_state.ListState) –
    • version ((optional) int) –
to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

ask_sdk_model.services.monetization package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.
For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.services.monetization.entitled_state module

class ask_sdk_model.services.monetization.entitled_state.EntitledState  
Bases: enum.Enum
  
State determining if the user is entitled to the product. Note - Any new values introduced later should be treated as 'NOT_ENTITLED'. * 'ENTITLED' - The user is entitled to the product. * 'NOT_ENTITLED' - The user is not entitled to the product.
  
Allowed enum values: [ENTITLED, NOT_ENTITLED]

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

ask_sdk_model.services.monetization.error module

class ask_sdk_model.services.monetization.error.Error(message=None)  
Bases: object
  
Describes error detail

Parameters message (optional) str – Readable description of error

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model
class ask_sdk_model.services.monetization.in_skill_product.InSkillProduct:

Bases: object

Parameters

- **product_id** *(optional) str* – Product Id
- **reference_name** *(optional) str* – Developer selected in-skill product name. This is for developer reference only.
- **name** *(optional) str* – Name of the product in the language from the `Accept-Language` header
- **object_type** *(optional) ask_sdk_model.services.monetization.product_type.ProductType* –
- **summary** *(optional) str* – Product summary in the language from the `Accept-Language` header
- **purchasable** *(optional) ask_sdk_model.services.monetization.purchasable_state.PurchasableState* –
- **entitled** *(optional) ask_sdk_model.services.monetization.entitled_state.EntitledState* –
- **entitlement_reason** *(optional) ask_sdk_model.services.monetization.entitlement_reason.EntitlementReason* –
- **active_entitlement_count** *(optional) int* – Total active purchases of the product made by the user. Note - For ENTITLEMENT and SUBSCRIPTION product types, the value is either zero(NOT_ENTITLED) or one(ENTITLED). For CONSUMABLE product type the value is zero or more, as CONSUMABLE can be re-purchased.
- **purchase_mode** *(optional) ask_sdk_model.services.monetization.purchase_mode.PurchaseMode* –
**ask_sdk_model.services.monetization.in_skill_products_response module**

```python
class ask_sdk_model.services.monetization.in_skill_products_response.InSkillProductsResponse:
```

Bases: `object`

Parameters

- `in_skill_products`  
  (optional) `list[ask_sdk_model.services.monetization.in_skill_product.InSkillProduct]`) – List of In-Skill Products
- `is_truncated`  
  (optional) `bool`
- `next_token`  
  (optional) `str`

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

**ask_sdk_model.services.monetization.monetization_service_client module**

```python
class ask_sdk_model.services.monetization.monetization_service_client.MonetizationServiceClient:
```

Bases: `ask_sdk_model.services.base_service_client.BaseServiceClient`

ServiceClient for calling the MonetizationService APIs.

Parameters `api_configuration`  
(ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of `ask_sdk_model.services.api_configuration.ApiConfiguration`

**get_in_skill_product**(accept_language, product_id, **kwargs)

Get In-Skill Product information based on user context for the Skill.

Parameters

- `accept_language`  
  (str) – (required) User’s locale/language in context
- `product_id`  
  (str) – (required) Product Id.

Return type  
`Union[Error, InSkillProduct]`

**get_in_skill_products**(accept_language, **kwargs)

Gets In-Skill Products based on user’s context for the Skill.

Parameters

- `accept_language`  
  (str) – (required) User’s locale/language in context
- `purchasable`  
  (str) – Filter products based on whether they are purchasable by
  the user or not.  
  * ‘PURCHASABLE’ - Products that are purchasable by the user.  
  * ‘NOT_PURCHASABLE’ - Products that are not purchasable by the user.
• **entitled** *(str)* – Filter products based on whether they are entitled to the user or not.
  * ‘ENTITLED’ - Products that the user is entitled to. * ‘NOT_ENTITLED’ - Products that the user is not entitled to.

• **product_type** *(str)* – Product type. * ‘SUBSCRIPTION’ - Once purchased, customers will own the content for the subscription period. * ‘ENTITLEMENT’ - Once purchased, customers will own the content forever. * ‘CONSUMABLE’ - Once purchased, customers will be entitled to the content until it is consumed. It can also be re-purchased.

• **next_token** *(str)* – When response to this API call is truncated (that is, isTruncated response element value is true), the response also includes the nextToken element, the value of which can be used in the next request as the continuation-token to list the next set of objects. The continuation token is an opaque value that In-Skill Products API understands. Token has expiry of 24 hours.

• **max_results** *(float)* – sets the maximum number of results returned in the response body. If you want to retrieve fewer than upper limit of 100 results, you can add this parameter to your request. maxResults should not exceed the upper limit. The response might contain fewer results than maxResults, but it will never contain more. If there are additional results that satisfy the search criteria, but these results were not returned because maxResults was exceeded, the response contains isTruncated = true.

**Return type**  Union*[Error, InSkillProductsResponse]*

**ask_sdk_model.services.monetization.product_type module**

```python
class ask_sdk_model.services.monetization.product_type.ProductType
    Bases: enum.Enum

    Product type. * 'SUBSCRIPTION' - Once purchased, customers will own the content for the subscription period. * 'ENTITLEMENT' - Once purchased, customers will own the content forever. * 'CONSUMABLE' - Once purchased, customers will be entitled to the content until it is consumed. It can also be re-purchased.

    Allowed enum values: [SUBSCRIPTION, ENTITLEMENT, CONSUMABLE]

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.services.monetization.purchasable_state module**

```python
class ask_sdk_model.services.monetization.purchasable_state.PurchasableState
    Bases: enum.Enum

    State determining if the product is purchasable by the user. Note - Any new values introduced later should be treated as 'NOT_PURCHASABLE'. * 'PURCHASABLE' - The product is purchasable by the user. * 'NOT_PURCHASABLE' - The product is not purchasable by the user.

    Allowed enum values: [PURCHASABLE, NOT_PURCHASABLE]

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```
ask_sdk_model.services.monetization.purchase_mode module

```python
class ask_sdk_model.services.monetization.purchase_mode.PurchaseMode
    Bases: enum.Enum
    Indicates if the entitlements are for TEST or LIVE purchases. * 'TEST' - test purchases made by developers or
    beta testers. Purchase not sent to payment processing. * 'LIVE' - purchases made by live customers. Purchase
    sent to payment processing.
    Allowed enum values: [TEST, LIVE]
```

```python
to_dict()
    Returns the model properties as a dict
```

```python
to_str()
    Returns the string representation of the model
```

ask_sdk_model.servicesreminder_management package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class
directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b
import C.

ask_sdk_model.services.reminder_management.alert_info module

```python
class ask_sdk_model.services.reminder_management.alert_info.AlertInfo(spoken_info=None)
    Bases: object
    Alert info for VUI / GUI
    Parameters spoken_info ((optional) ask_sdk_model.services.
        reminder_management.alert_info_spoken_info.SpokenInfo)
```

```python
to_dict()
    Returns the model properties as a dict
```

```python
to_str()
    Returns the string representation of the model
```

ask_sdk_model.services.reminder_management.alert_info_spoken_info module

```python
class ask_sdk_model.services.reminder_management.alert_info_spoken_info.SpokenInfo(content=None)
    Bases: object
    Parameters for VUI presentation of the reminder
    Parameters content ((optional) list[ask_sdk_model.services.
        reminder_management.spoken_text.SpokenText])
```

2.14. Hosting Skills as Webservice
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.services.reminder_management.error module

class ask_sdk_model.services.reminder_management.error.Error(code=None, message=None)
    Bases: object
    Parameters
    • code ((optional) str) – Domain specific error code
    • message ((optional) str) – Detailed error message

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.services.reminder_management.event module

class ask_sdk_model.services.reminder_management.event.Event(status=None, alert_token=None)
    Bases: object
    Parameters
    • status ((optional) ask_sdk_model.services.reminder_management.status.Status)
    • alert_token ((optional) str)

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
module ask_sdk_model.services.reminder_management.get_reminder_response:

class GetReminderResponse:
    Bases: ask_sdk_model.services.reminder_management.reminder.Reminder
    Response object for get reminder request
    Parameters
    • alert_token (optional str) – Unique id of this reminder alert
    • created_time (optional datetime) – Valid ISO 8601 format - Creation time of this reminder alert
    • updated_time (optional datetime) – Valid ISO 8601 format - Last updated time of this reminder alert
    • status (optional ask_sdk_model.services.reminder_management.status.Status)
    • trigger (optional ask_sdk_model.services.reminder_management.trigger.Trigger)
    • alert_info (optional ask_sdk_model.services.reminder_management.alert_info.AlertInfo)
    • push_notification (optional ask_sdk_model.services.reminder_management.push_notification.PushNotification)
    • version (optional str) – Version of reminder alert

to_dict()  
Returns the model properties as a dict

to_str()   
Returns the string representation of the model

module ask_sdk_model.services.reminder_management.get_reminders_response:

class GetRemindersResponse:
    Bases: object
    Response object for get reminders request
    Parameters
    • total_count (optional str) – Total count of reminders returned
• **alerts**  
  
  ```python
  (optional) list[ask_sdk_model.services.
  reminder_management.reminder.Reminder] – List of reminders
  ```

• **links**  
  
  ```python
  (optional) str – Link to retrieve next set of alerts if total count is greater
  than max results
  ```

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.services.reminder_management.push_notification module**

class ask_sdk_model.services.reminder_management.push_notification.PushNotification(status=None)

Bases: object

Enable / disable reminders push notifications to Alexa mobile apps

```python
Parameters

status
  (optional) ask_sdk_model.services.
  reminder_management.push_notification_status.
  PushNotificationStatus
```

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.services.reminder_management.push_notification_status module**

class ask_sdk_model.services.reminder_management.push_notification_status.PushNotificationStatus

Bases: enum.Enum

Push notification status - Enabled/Disabled

Allowed enum values: [ENABLED, DISABLED]

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

**ask_sdk_model.services.reminder_management.recurrence module**

class ask_sdk_model.services.reminder_management.recurrence.Recurrence(freq=None, 
by_day=None, 
in-
ter-
val=None)

Bases: object

Recurring date/time using the RFC 5545 standard in JSON object form

```python
Parameters
```
• **freq**  
  *(optional)*  
  `ask_sdk_model.services.reminder_management.recurrence_freq.RecurrenceFreq`–

• **by_day**  
  *(optional)*  
  `list[ask_sdk_model.services.reminder_management.recurrence_day.RecurrenceDay]`–

• **interval** *(optional)*  
  `int`– contains a positive integer representing at which intervals the recurrence rule repeats

  ```python
to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model
```

### ask_sdk_model.services.reminder_management.recurrence_day module

class `ask_sdk_model.services.reminder_management.recurrence_day.RecurrenceDay`

Bases: `enum.Enum`

Day of recurrence

Allowed enum values: [SU, MO, TU, WE, TH, FR, SA]

  ```python
to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model
```

### ask_sdk_model.services.reminder_management.recurrence_freq module

class `ask_sdk_model.services.reminder_management.recurrence_freq.RecurrenceFreq`

Bases: `enum.Enum`

Frequency of recurrence

Allowed enum values: [WEEKLY, DAILY]

  ```python
to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model`
class ask_sdk_model.services.reminder_management.reminder.Reminder(alert_token=None, created_time=None, updated_time=None, status=None, trigger=None, alert_info=None, push_notification=None, version=None)

Bases: object

Reminder object

Parameters

- alert_token ((optional) str) – Unique id of this reminder alert
- created_time ((optional) datetime) – Valid ISO 8601 format - Creation time of this reminder alert
- updated_time ((optional) datetime) – Valid ISO 8601 format - Last updated time of this reminder alert
- status ((optional) ask_sdk_model.services.reminder_management.status.Status)
- trigger ((optional) ask_sdk_model.services.reminder_management.trigger.Trigger)
- alert_info ((optional) ask_sdk_model.services.reminder_management.alert_info.AlertInfo)
- push_notification ((optional) ask_sdk_model.services.reminder_management.push_notification.PushNotification)
- version ((optional) str) – Version of reminder alert

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

class ask_sdk_model.services.reminder_management.reminder_created_event_request.ReminderCreatedEventRequest(request_id=None, timesamp=None, local=None, body=None)

Bases: ask_sdk_model.request.Request

Parameters
• request_id (optional str) – Represents the unique identifier for the specific request.

• timestamp (optional datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• locale (optional str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• body (optional) ask_sdk_model.services.reminder_management.event.Event

  to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_deleted_event module

class ask_sdk_model.services.reminder_management.reminder_deleted_event.ReminderDeletedEvent
  Bases: object

  Parameters

  alert_tokens (optional list[str])

  to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_deleted_event_request module

class ask_sdk_model.services.reminder_management.reminder_deleted_event_request.ReminderDeletedEventRequest
  Bases: ask_sdk_model.request.Request

  Parameters

  • request_id (optional str) – Represents the unique identifier for the specific request.

  • timestamp (optional datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

  • locale (optional str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

  • body (optional) ask_sdk_model.services.reminder_management.reminder_deleted_event.ReminderDeletedEvent

  to_dict()
  Returns the model properties as a dict
to_str()

Returns the string representation of the model

```
ask_sdk_model.services.reminder_management.reminder_management_service_client module

class ask_sdk_model.services.reminder_management.reminder_management_service_client.ReminderManagementServiceClient
    Bases: ask_sdk_model.services.base_service_client.BaseServiceClient

ServiceClient for calling the ReminderManagementService APIs.

    Parameters
        api_configuration
            (ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of ask_sdk_model.

    create_reminder (reminder_request, **kwargs)

This API is invoked by the skill to create a new reminder.

    Parameters
        reminder_request
            (ask_sdk_model.services.reminder_management.reminder_request.ReminderRequest) – (required)

    Return type
        Union[ReminderResponse, Error]

delete_reminder (alert_token, **kwargs)

This API is invoked by the skill to delete a single reminder.

    Parameters
        alert_token
            (str) – (required)

    Return type
        None

get_reminder (alert_token, **kwargs)

This API is invoked by the skill to get a single reminder.

    Parameters
        alert_token
            (str) – (required)

    Return type
        Union[GetReminderResponse, Error]

get_reminders (**kwargs)

This API is invoked by the skill to get all reminders created by the caller.

    Return type
        Union[GetRemindersResponse, Error]

update_reminder (alert_token, reminder_request, **kwargs)

This API is invoked by the skill to update a reminder.

    Parameters
        • alert_token
            (str) – (required)
        • reminder_request
            (ask_sdk_model.services.reminder_management.reminder_request.ReminderRequest) – (re-

    Return type
        Union[ReminderResponse, Error]
ask_sdk_model.services.reminder_management.reminder_request module

class ask_sdk_model.services.reminder_management.reminder_request.ReminderRequest(request_time=None, trigger=None, alert_info=None, push_notification=None):

    Bases: object

    Input request for creating a reminder

    Parameters

    • request_time ((optional) datetime) – Valid ISO 8601 format - Creation time of this reminder alert

    • trigger ((optional) ask_sdk_model.services.reminder_management.trigger.Trigger) –

    • alert_info ((optional) ask_sdk_model.services.reminder_management.alert_info.AlertInfo) –

    • push_notification ((optional) ask_sdk_model.services.reminder_management.push_notification.PushNotification) –

    to_dict()

        Returns the model properties as a dict

    to_str()

        Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_response module

class ask_sdk_model.services.reminder_management.reminder_response.ReminderResponse(alert_token=None, created_time=None, updated_time=None, status=None, version=None, href=None):

    Bases: object

    Response object for post/put/delete reminder request

    Parameters

    • alert_token ((optional) str) – Unique id of this reminder alert

    • created_time ((optional) str) – Valid ISO 8601 format - Creation time of this reminder alert

    • updated_time ((optional) str) – Valid ISO 8601 format - Last updated time of this reminder alert

    • status ((optional) ask_sdk_model.services.reminder_management.status.Status) –
• version((optional) str) – Version of reminder alert
• href((optional) str) – URI to retrieve the created alert

to_dict()
  Returns the model properties as a dict
to_str()
  Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_started_event_request module

class ask_sdk_model.services.reminder_management.reminder_started_event_request.ReminderStartedEventRequest(request_id=None,
  timestamp=None,
  locale=None,
  body=None)

Bases: ask_sdk_model.request.Request

Parameters

  • request_id((optional) str) – Represents the unique identifier for the specific request.
  • timestamp((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
  • locale((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
  • body((optional) ask_sdk_model.services.reminder_management.event.Event)

  to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_status_changed_event_request module

class ask_sdk_model.services.reminder_management.reminder_status_changed_event_request.ReminderStatusChangedEventRequest(request_id=None,
  timestamp=None,
  locale=None,
  body=None)

Bases: ask_sdk_model.request.Request

Parameters

  • request_id((optional) str) – Represents the unique identifier for the specific request.
  • timestamp((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
• **locale** *(optional) str* — A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• **body** *(optional) ask_sdk_model.services.reminder_management.event.Event* —

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

### ask_sdk_model.services.reminder_management.reminder_updated_event_request module

#### class ask_sdk_model.services.reminder_management.reminder_updated_event_request.ReminderUpdatedEventRequest

Bases: `ask_sdk_model.request.Request`

Parameters

• **request_id** *(optional) str* — Represents the unique identifier for the specific request.

• **timestamp** *(optional) datetime* — Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

• **locale** *(optional) str* — A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

• **body** *(optional) ask_sdk_model.services.reminder_management.event.Event* —

```python
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

### ask_sdk_model.services.reminder_management.spoken_text module

#### class ask_sdk_model.services.reminder_management.spoken_text.SpokenText

Bases: `object`

Parameters

• **locale** *(optional) str* — The locale in which the spoken text is rendered. e.g. en-US

• **ssml** *(optional) str* — Spoken text in SSML format.

• **text** *(optional) str* — Spoken text in plain text format.
Alexa SKills Kit SDK for Python Documentation, Release 1.10.2

```python
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.services.reminder_management.status module

class ask_sdk_model.services.reminder_management.status.Status
    Bases: enum.Enum
    Status of reminder
    Allowed enum values: [ON, COMPLETED]
to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model

ask_sdk_model.services.reminder_management.trigger module

class ask_sdk_model.services.reminder_management.trigger.Trigger
    Bases: object
    Trigger information for Reminder
    Parameters
        • object_type ((optional) ask_sdk_model.services.
            reminder_management.trigger_type.TriggerType) –
        • scheduled_time ((optional) datetime) – Valid ISO 8601 format - Intended
            trigger time
        • offset_in_seconds ((optional) int) – If reminder is set using relative time, use
            this field to specify the time after which reminder will ring (in seconds)
        • time_zone_id ((optional) str) – Intended reminder’s timezone
        • recurrence ((optional) ask_sdk_model.services.
            reminder_management.recurrence.Recurrence) –

to_dict()
    Returns the model properties as a dict
to_str()
    Returns the string representation of the model
```
ask_sdk_model.services.reminder_management.trigger_type module

```python
class ask_sdk_model.services.reminder_management.trigger_type.TriggerType
    Bases: enum.Enum
    Type of reminder - Absolute / Relative
    Allowed enum values: [SCHEDULED_ABSOLUTE, SCHEDULED_RELATIVE]
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model
```

ask_sdk_model.services.ups package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.services.ups.distance_units module

```python
class ask_sdk_model.services.ups.distance_units.DistanceUnits
    Bases: enum.Enum
    Allowed enum values: [METRIC, IMPERIAL]
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model
```

ask_sdk_model.services.ups.error module

```python
class ask_sdk_model.services.ups.error.Error(code=None, message=None)
    Bases: object
    Parameters
        • code (optional) ask_sdk_model.services.ups.error_code.ErrorCode
        • message (optional) str - A human readable description of error.
    to_dict()
        Returns the model properties as a dict
    to_str()
        Returns the string representation of the model
```

2.14. Hosting Skills as Webservice
ask_sdk_model.services.ups.error_code module

class ask_sdk_model.services.ups.error_code.ErrorCode
    Bases: enum.Enum

    A more precise error code. Some of these codes may not apply to some APIs.
    - INVALID_KEY: the setting key is not supported
    - INVALID_VALUE: the setting value is not valid
    - INVALID_TOKEN: the token is invalid
    - INVALID_URI: the uri is invalid
    - DEVICE_UNREACHABLE: the device is offline
    - UNKNOWN_ERROR: internal service error

    Allowed enum values: [INVALID_KEY, INVALID_VALUE, INVALID_TOKEN, INVALID_URI, DEVICE_UNREACHABLE, UNKNOWN_ERROR]

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model

ask_sdk_model.services.ups.phone_number module

class ask_sdk_model.services.ups.phone_number.PhoneNumber
    country_code=None,
    phone_number=None)
    Bases: object

    Parameters
    • country_code ((optional) str)
    • phone_number ((optional) str)

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model

ask_sdk_model.services.ups.ups_service_client module

class ask_sdk_model.services.ups.ups_service_client.UpsServiceClient
    api_configuration)
    Bases: ask_sdk_model.services.base_service_client.BaseServiceClient

    ServiceClient for calling the UpsService APIs.

    Parameters
    • api_configuration (ask_sdk_model.services.api_configuration.ApiConfiguration)
        - Instance of ask_sdk_model.services.api_configuration.ApiConfiguration

    get_profile_email(**kwargs)
        Gets the email address of the customer associated with the current enablement. Requires customer consent for scopes: [alexa::profile:email:read]

        Return type Union[str, Error]

    get_profile_given_name(**kwargs)
        Gets the given name (first name) of the customer associated with the current enablement. Requires customer consent for scopes: [alexa::profile:given_name:read]

        Return type Union[str, Error]
**get_profile_mobile_number**(**kwargs)**

Gets the mobile phone number of the customer associated with the current enablement. Requires customer consent for scopes: [alexa::profile:mobile_number:read]

**Return type** Union[PhoneNumber, Error]

**get_profile_name**(**kwargs)**

Gets the full name of the customer associated with the current enablement. Requires customer consent for scopes: [alexa::profile:name:read]

**Return type** Union[str, Error]

**get_system_distance_units**(device_id, **kwargs)**

Gets the distance measurement unit of the device. Does not require explicit customer consent.

**Parameters**

 device_id (str) – (required) The device Id

**Return type** Union[Error, DistanceUnits]

**get_system_temperature_unit**(device_id, **kwargs)**

Gets the temperature measurement units of the device. Does not require explicit customer consent.

**Parameters**

 device_id (str) – (required) The device Id

**Return type** Union[TemperatureUnit, Error]

**get_system_time_zone**(device_id, **kwargs)**

Gets the time zone of the device. Does not require explicit customer consent.

**Parameters**

 device_id (str) – (required) The device Id

**Return type** Union[str, Error]

**ask_sdk_model.services.ups.temperature_unit module**

**class** ask_sdk_model.services.ups.temperature_unit.TemperatureUnit

**Bases:** enum.Enum

Allowed enum values: [CELSIUS, FAHRENHEIT]

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

**ask_sdk_model.services.proactive_events package**

**Submodules**

**Note:** Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.
class ask_sdk_model.services.proactive_events.create_proactive_event_request.CreateProactiveEventRequest:

Bases: object

Parameters

• **timestamp** *(optional) datetime*—The date and time of the event associated with this request, in ISO 8601 format.

• **reference_id** *(optional) str*—Client-supplied ID for correlating the event with external entities. The allowed characters for the referenceId field are alphanumeric and ~, and the length of the referenceId field must be 1-100 characters.

• **expiry_time** *(optional) datetime*—The date and time, in ISO 8601 format, when the service will automatically delete the notification if it is still in the pending state.

• **event** *(optional) ask_sdk_model.services.proactive_events.event.Event*—

• **localized_attributes** *(optional) list[object]*—A list of items, each of which contains the set of event attributes that requires localization support.

• **relevant_audience** *(optional) ask_sdk_model.services.proactive_events.relevant_audience.RelevantAudience*—

  to_dict()
  Returns the model properties as a dict

  to_str()
  Returns the string representation of the model

class ask_sdk_model.services.proactive_events.error.Error:

Bases: object

Parameters

• **code** *(optional) int*—

• **message** *(optional) str*—

  to_dict()
  Returns the model properties as a dict
to_str()

Returns the string representation of the model

ask_sdk_model.services.proactive_events.event module

class ask_sdk_model.services.proactive_events.event.Event(name=None, payload=None)

Bases: object

The event data to be sent to customers, conforming to the schema associated with this event.

Parameters
- name ((optional) str)-
- payload ((optional) object)-


to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.proactive_events.proactive_events_service_client module

class ask_sdk_model.services.proactive_events.proactive_events_service_client.ProactiveEventsServiceClient(api_configuration, authentication_configuration)

Bases: ask_sdk_model.services.base_service_client.BaseServiceClient

ServiceClient for calling the ProactiveEventsService APIs.

Parameters
- api_configuration (ask_sdk_model.services.api_configuration.ApiConfiguration) - Instance of ask_sdk_model.services.api_configuration.ApiConfiguration

create_proactive_event (create_proactive_event_request, stage, **kwargs)

Create a new proactive event in live stage.

Parameters
- create_proactive_event_request (ask_sdk_model.services.proactive_events.create_proactive_event_request.CreateProactiveEventRequest) - (required) Request to create a new proactive event.

Return type None

ask_sdk_model.services.proactive_events.relevant_audience module

class ask_sdk_model.services.proactive_events.relevant_audience.RelevantAudience(object_type=None, payload=None)

Bases: object

The audience for this event.

2.14. Hosting Skills as Webservice
Parameters

- `object_type` *(optional)* `ask_sdk_model.services.proactive_events.relevant_audience_type.RelevantAudienceType` -

- `payload` *(optional)* `object` – If `relevantAudience.type` is set to Multicast, then the payload object is empty. Otherwise, the userId value for which the event is targeted is required.

`to_dict()`

Returns the model properties as a dict

`to_str()`

Returns the string representation of the model

`ask_sdk_model.services.proactive_events.relevant_audience_type` module

class `ask_sdk_model.services.proactive_events.relevant_audience_type.RelevantAudienceType`

Bases: `enum.Enum`

The audience for this event. Use Multicast to target information to all customers subscribed to that event, or use Unicast to target information containing the actual userId for individual events.

Allowed enum values: [Unicast, Multicast]

`to_dict()`

Returns the model properties as a dict

`to_str()`

Returns the string representation of the model

`ask_sdk_model.services.proactive_events.skill_stage` module

class `ask_sdk_model.services.proactive_events.skill_stage.SkillStage`

Bases: `enum.Enum`

Stage for creating Proactive events.

Since proactive events can be created on the DEVELOPMENT and LIVE stages of the skill, this enum provides the stage values that can be used to pass to the service call.

Allowed enum values: [DEVELOPMENT, LIVE]

`to_dict()`

Returns the model properties as a dict

`to_str()`

Returns the string representation of the model

`ask_sdk_model.services.skill_messaging` package

Submodules

---

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.
For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

**ask_sdk_model.services.skill_messaging.error module**

class ask_sdk_model.services.skill_messaging.error.Error(code=None, message=None)

Bases: object

Parameters

- code ((optional) int)
- message ((optional) str)

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model

**ask_sdk_model.services.skill_messaging.send_skill_messaging_request module**

class ask_sdk_model.services.skill_messaging.send_skill_messaging_request.SendSkillMessagingRequest(data=None, expires_after_seconds=None)

Bases: object

The message that needs to be sent to the skill

Parameters

- data ((optional) object) – The payload data to send with the message. The data must be in the form of JSON-formatted key-value pairs. Both keys and values must be of type String. The total size of the data cannot be greater than 6KB. For calculation purposes, this includes keys and values, the quotes that surround them, the "\" character that separates them, the commas that separate the pairs, and the opening and closing braces around the field. However, any whitespace between key/value pairs is not included in the calculation of the payload size. If the message does not include payload data, as in the case of a sync message, you can pass in an empty JSON object "{}".

- expires_after_seconds ((optional) int) – The number of seconds that the message will be retained to retry if message delivery is not successful. Allowed values are from 60 (1 minute) to 86400 (1 day), inclusive. The default is 3600 (1 hour). Multiple retries may occur during this interval. The retry logic is exponential. The first retry executes after 30 seconds, and this time period doubles on every retry. The retries will end when the total time elapsed since the message was first sent has exceeded the value you provided for expiresAfterSeconds. Message expiry is rarely a problem if the message handler has been set up correctly. With a correct setup, you will receive the message once promptly. This mechanism for retries is provided as a safeguard in case your skill goes down during a message delivery.

**to_dict()**

Returns the model properties as a dict

**to_str()**

Returns the string representation of the model
class ask_sdk_model.services.skill_messaging.skill_messaging_service_client.SkillMessagingServiceClient(
    api_configuration, authentication_configuration)

     Bases: ask_sdk_model.services.base_service_client.BaseServiceClient

ServiceClient for calling the SkillMessagingService APIs.

Parameters
api_configuration (ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of ask_sdk_model.services.api_configuration.ApiConfiguration

send_skill_message (user_id, send_skill_messaging_request, **kwargs)

Send a message request to a skill for a specified user.

Parameters
• user_id (str) – (required) The user Id for the specific user to send the message
• send_skill_messaging_request (ask_sdk_model.services.skill_messaging.send_skill_messaging_request.SendSkillMessagingRequest) – (required) Message Request to be sent to the skill.

Return type None

ask_sdk_model.services.lwa package

Submodules

ask_sdk_model.services.lwa.access_token module

class ask_sdk_model.services.lwa.access_token.AccessToken (token=None, expiry=None)

     Bases: object

Represents the access token provided by LWA (Login With Amazon).

This is a wrapper class over ask_sdk_model.services.lwa.access_token_response.AccessTokenResponse that retrieves and stores the access token, the expiry time from LWA response.

Parameters
• token (str) – access token from LWA
• expiry (datetime) – exact timestamp in UTC datetime, which is the expiry time for this access token. This is set as the combined datetime of current system time when the LWA response is received and the expiry time in seconds, provided in the LWA response.
ask_sdk_model.services.lwa.access_token_request module

class ask_sdk_model.services.lwa.access_token_request.AccessTokenRequest(client_id=None, client_secret=None, scope=None)

Bases: object
Request for retrieving an access token from LWA.

Parameters

• client_id (str) – The ClientId value from developer console
• client_secret (str) – The ClientSecret value from developer console
• scope (str) – The required scope for which the access token is requested for

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.lwa.access_token_response module

class ask_sdk_model.services.lwa.access_token_response.AccessTokenResponse(access_token=None, expires_in=None, scope=None, token_type=None)

Bases: object
LWA response for retrieving an access token.

Parameters

• access_token (str) – The access token from LWA
• expires_in (int) – The duration in seconds of the access token lifetime
• scope (str) – The scope specified in the access token request
• token_type (str) – The type of token issued

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.lwa.error module

class ask_sdk_model.services.lwa.error.Error(error_description=None, error_type=None)

Bases: object
Error from LWA Client request.

Parameters
• **error_description** *(optional) str* – Description of the error
• **error_type** *(optional) str* – Type of error

```python
to_dict()
   Returns the model properties as a dict
to_str()
   Returns the string representation of the model
```

**ask_sdk_model.services.lwa.lwa_client module**

class ask_sdk_model.services.lwa.lwa_client.LwaClient(*api_configuration, authentication_configuration)*

_bases: ask_sdk_model.services.base_service_client.BaseServiceClient*

Client to call Login with Amazon (LWA) to retrieve access tokens.

Parameters

• **api_configuration** *(ask_sdk_model.services.api_configuration.ApiConfiguration)* – ApiConfiguration instance with valid Serializer and ApiClient. The authorization value and api endpoint is not used by the LWA Client.
• **authentication_configuration** *(ask_sdk_model.services.authentication_configuration.AuthenticationConfiguration)* – AuthenticationConfiguration instance with valid client id and client secret, for making LWA calls.

Raises **ValueError** if authentication configuration is not provided.

```python
get_access_token_for_scope(scope)*
   Retrieve access token for given scope.

   Return the scoped access token from the scoped_token_cache if the token is unexpired. If it is expired or is not present, then retrieve a new access token for the given scope, using the client id and client secret in the input ask_sdk_model.services.authentication_configuration.AuthenticationConfiguration instance.

   Parameters **scope** *(str)* – Target scope for the access token

   Raises **ValueError** if no scope is passed and **ValueError** if LWA AccessTokenResponse is None.
```

**Submodules**

**Note:** Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.
ask_sdk_model.services.api_client module

class ask_sdk_model.services.api_client.ApiClient
   Bases: object

   Represents a basic contract for API request invocation.

   invoke(request)
      Dispatches a request to an API endpoint described in the request.

      The ApiClient is expected to resolve in the case an API returns a non-200 HTTP status code. The responsibility of translating a particular response code to an error lies with the caller.

      Parameters
      request (ApiClientRequest) – Request to dispatch to the ApiClient

      Returns
      Response from the client call

      Return type
      ApiClientResponse

ask_sdk_model.services.api_client_message module

class ask_sdk_model.services.api_client_message.ApiClientMessage(headers=None, body=None)
   Bases: object

   Represents the interface between ask_sdk_model.services.api_client.ApiClient implementation and a Service Client.

   Parameters

   • headers (list[tuple[str, str]]) – List of header tuples

   • body (str) – Body of the message

ask_sdk_model.services.api_client_request module

class ask_sdk_model.services.api_client_request.ApiClientRequest(headers=None, body=None, url=None, method=None)
   Bases: ask_sdk_model.services.api_client_message.ApiClientMessage

   Represents a request sent from Service Clients to an ask_sdk_model.services.api_client.ApiClient implementation.

   Parameters

   • headers (list[tuple[str, str]]) – List of header tuples

   • body (str) – Body of the message

   • url (str) – Url of the request

   • method (str) – Method called with the request
ask_sdk_model.services.api_client_response module

class ask_sdk_model.services.api_client_response.ApiClientResponse(headers=None, body=None, status_code=None)

Bases: ask_sdk_model.services.api_client_message.ApiClientMessage

Represent a request sent from Service Clients to an ask_sdk_model.services.api_client. ApiClient class implementation.

Parameters

- headers (list[tuple[str, str]]) – List of header tuples
- body (str) – Body of the message
- status_code (int) – Status code of the response

ask_sdk_model.services.api_configuration module

class ask_sdk_model.services.api_configuration.ApiConfiguration(serializer=None, api_client=None, authorization_value=None, api_endpoint=None)

Bases: object

Represents a class that provides API configuration options needed by service clients.

Parameters

- serializer ((optional) ask_sdk_model.services.serializer.Serializer) – serializer implementation for encoding/decoding JSON from/to Object models.
- api_client ((optional) ask_sdk_model.services.api_client.ApiClient) – API Client implementation
- authorization_value ((optional) str) – Authorization value to be used on any calls of the service client instance
- api_endpoint ((optional) str) – Endpoint to hit by the service client instance

ask_sdk_model.services.authentication_configuration module

class ask_sdk_model.services.authentication_configuration.AuthenticationConfiguration(client_id=None, client_secret=None)

Bases: object

Represents a class that provides authentication configuration.

Parameters

- client_id (str) – Client ID required for authentication.
- client_secret (str) – Client Secret required for authentication.
ask_sdk_model.services.base_service_client module

class ask_sdk_model.services.base_service_client.BaseServiceClient(api_configuration)
   Bases: object
   
   Class to be used as the base class for the generated service clients.
   
   The class has to be implemented by the service clients and this class instantiation is not supported

   Parameters
   api_configuration (ask_sdk_model.services.api_configuration.ApiConfiguration) – ApiConfiguration implementation

   invoke (method, endpoint, path, query_params, header_params, path_params, response_definitions, body, response_type)
   
   Calls the ApiClient based on the ServiceClient specific data provided as well as handles the well-known responses from the Api.

   Parameters
   • method (str) – Http method
   • endpoint – Base endpoint to make the request to
   • path (str) – Specific path to hit. It might contain variables to be interpolated with path_params
   • query_params (list(tuple(str, str))) – Parameter values to be sent as part of query string
   • header_params (list(tuple(str, str))) – Parameter values to be sent as headers
   • path_params (dict(str, str)) – Parameter values to be interpolated in the path
   • response_definitions (list(ask_sdk_model.services.service_client_response.ServiceClientResponse)) – Well-known expected responses by the ServiceClient
   • body (object) – Request body
   • response_type (class) – Type of the expected response if applicable

   Returns
   Response object instance of the response_type provided

   Return type
   object

   Raises
   ask_sdk_model.services.service_exception.ServiceException if service fails and ValueError if serializer or API Client is not configured in api_configuration # noqa: E501

ask_sdk_model.services.serializer module

class ask_sdk_model.services.serializer.Serializer
   Bases: object

   Represents an abstract object used for Serialization tasks

   deserialize (payload, obj_type)
   
   Deserializes the payload to object of provided obj_type.

   Parameters
   • payload (str) – String to deserialize
• **obj** *(type)* – Target type of deserialization

  Returns Deserialized object

  Return type object

  `serialize(obj)` Serializes an object into a string.

  Parameters `obj` – object to serialize

  Returns serialized object in string format

  Return type str

ask_sdk_model.services.service_client_factory module

class ask_sdk_model.services.service_client_factory.ServiceClientFactory(api_configuration)

  Bases: object

  ServiceClientFactory class to help build service clients.

  Parameters `api_configuration` *(ApiConfiguration)* – API Configuration for calling services

  `get_device_address_service()` Get DeviceAddressServiceClient for device_address_service.

  Returns Client for calling the service

  Return type DeviceAddressServiceClient

  Raises ValueError

  `get_directive_service()` Get DirectiveServiceClient for directive_service.

  Returns Client for calling the service

  Return type DirectiveServiceClient

  Raises ValueError

  `get_list_management_service()` Get ListManagementServiceClient for list_management_service.

  Returns Client for calling the service

  Return type ListManagementServiceClient

  Raises ValueError

  `get_monetization_service()` Get MonetizationServiceClient for monetization_service.

  Returns Client for calling the service

  Return type MonetizationServiceClient

  Raises ValueError

  `get_reminder_management_service()` Get ReminderManagementServiceClient for reminder_management_service.

  Returns Client for calling the service

  Chapter 2. SDK Features
Return type  ReminderManagementServiceClient
Raises  ValueError

get_ups_service()
Get UpsServiceClient for ups_service.

Returns  Client for calling the service

Return type  UpsServiceClient
Raises  ValueError

ask_sdk_model.services.service_client_response module

class  ask_sdk_model.services.service_client_response.ServiceClientResponse (response_type, status_code, message)

Bases: object
Represents a well-known response object by Service Client.

Parameters

- response_type  (Response class) – Well-known representation of the response
- status_code  (int) – Status code to be attached to the response
- message  (str) – Message to be attached to the response

ask_sdk_model.services.service_exception module

exception  ask_sdk_model.services.service_exception.ServiceException (message, status_code, headers, body)

Bases: Exception
Exception thrown by a Service client when an error response was received or some operation failed.

Parameters

- message  (str) – Description of the error
- status_code  (int) – Status code of the HTTP Response
- headers  (list(tuple(str, str))) – Headers of the HTTP response that return the failure
- body  (object) – Body of the HTTP Response

ask_sdk_model.slu package

Subpackages

2.14. Hosting Skills as Webservice
ask_sdk_model.slu.entityresolution package

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

ask_sdk_model.slu.entityresolution.resolution module

class ask_sdk_model.slu.entityresolution.resolution.Resolution

    Bases: object

    Represents a possible authority for entity resolution

    Parameters

        • authority ((optional) str)-
        • status ((optional) ask_sdk_model.slu.entityresolution.status.Status)-
        • values ((optional) list[ask_sdk_model.slu.entityresolution.value_wrapper.ValueWrapper])- 

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model

ask_sdk_model.slu.entityresolution.resolutions module

class ask_sdk_model.slu.entityresolution.resolutions.Resolutions

    Bases: object

    Represents the results of resolving the words captured from the user’s utterance. This is included for slots that use a custom slot type or a built-in slot type that you have extended with your own values. Note that resolutions is not included for built-in slot types that you have not extended.

    Parameters resolutions_per_authority ((optional) list[ask_sdk_model.slu.entityresolution.resolution.Resolution])- 

    to_dict()
    Returns the model properties as a dict

    to_str()
    Returns the string representation of the model
ask_sdk_model.slu.entityresolution.status module

class ask_sdk_model.slu.entityresolution.status.Status (code=None)
    Bases: object
    Parameters code : (optional) ask_sdk_model.slu.entityresolution.status_code.StatusCode – Indication of the results of attempting to resolve the user utterance against the defined slot types.

    to_dict ()
    Returns the model properties as a dict

    to_str ()
    Returns the string representation of the model

ask_sdk_model.slu.entityresolution.status_code module

class ask_sdk_model.slu.entityresolution.status_code.StatusCode
    Bases: enum.Enum
    Indication of the results of attempting to resolve the user utterance against the defined slot types.
    Allowed enum values: [ER_SUCCESS_MATCH, ER_SUCCESS_NO_MATCH, ER_ERROR_TIMEOUT, ER_ERROR_EXCEPTION]

    to_dict ()
    Returns the model properties as a dict

    to_str ()
    Returns the string representation of the model

ask_sdk_model.slu.entityresolution.value module

class ask_sdk_model.slu.entityresolution.value.Value (name=None, id=None)
    Bases: object
    Represents the resolved value for the slot, based on the user’s utterance and slot type definition.
    Parameters
    • name : (optional) str – The name for the resolution value.
    • id : (optional) str – The id for the resolution value.

    to_dict ()
    Returns the model properties as a dict

    to_str ()
    Returns the string representation of the model

ask_sdk_model.slu.entityresolution.value_wrapper module

class ask_sdk_model.slu.entityresolution.value_wrapper.ValueWrapper (value=None)
    Bases: object
    A wrapper class for an entity resolution value used for JSON serialization.
Parameters `value`  

- *(optional)* `ask_sdk_model.slu.entityresolution.value.Value` -

`to_dict()`  

- Returns the model properties as a dict

`to_str()`  

- Returns the string representation of the model

**ask_sdk_model.ui package**

**Submodules**

**Note:** Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with class `C`, you can do `from a import C` instead of `from a.b import C`.

**ask_sdk_model.ui.ask_for_permissions_consent_card module**

```python
class ask_sdk_model.ui.ask_for_permissions_consent_card.AskForPermissionsConsentCard(permissions=None)
    bases: ask_sdk_model.ui.card.Card

Parameters `permissions`  

- *(optional)* `list[str]` -

`to_dict()`  

- Returns the model properties as a dict

`to_str()`  

- Returns the string representation of the model

**ask_sdk_model.ui.card module**

```python
class ask_sdk_model.ui.card.Card(object_type=None)
    bases: object

Parameters `object_type`  

- *(optional)* `str` -

**Note:** This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

- `LinkAccount`: `ask_sdk_model.ui.link_account_card.LinkAccountCard`
- `Standard`: `ask_sdk_model.ui.standard_card.StandardCard`
- `AskForPermissionsConsent`: `ask_sdk_model.ui.ask_for_permissions_consent_card.AskForPermissionsConsentCard`
- `Simple`: `ask_sdk_model.ui.simple_card.SimpleCard`
classmethod get_real_child_model(data)
    Returns the real base class specified by the discriminator

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.ui.image module

class ask_sdk_model.ui.image.Image(small_image_url=None, large_image_url=None)
    Bases: object

    Parameters
    • small_image_url (optional) str–
    • large_image_url (optional) str–

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model

ask_sdk_model.ui.link_account_card module

class ask_sdk_model.ui.link_account_card.LinkAccountCard
    Bases: ask_sdk_model.ui.card.Card

    to_dict()
        Returns the model properties as a dict

    to_str()
        Returns the string representation of the model

ask_sdk_model.ui.output_speech module

class ask_sdk_model.ui.output_speech.OutputSpeech(object_type=None, play_behavior=None)
    Bases: object

    Parameters
    • object_type (optional) str–

    • play_behavior ((optional) ask_sdk_model.ui.play_behavior.PlayBehavior)–

    Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

    SSML: ask_sdk_model.ui.ssml_output_speech.SsmlOutputSpeech,
PlainText: `ask_sdk_model.ui.plain_text_output_speech.PlainTextOutputSpeech`

classmethod `get_real_child_model(data)`
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.plain_text_output_speech module

class `ask_sdk_model.ui.plain_text_output_speech.PlainTextOutputSpeech(play_behavior=None, text=None)`
Bases: `ask_sdk_model.ui.output_speech.OutputSpeech`

Parameters

• `play_behavior` ((optional) `ask_sdk_model.ui.play_behavior.PlayBehavior`) -

• `text` (optional) `str` -

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.reprompt module

class `ask_sdk_model.ui.reprompt.Reprompt(output_speech=None)`
Bases: `object`

Parameters `output_speech` (optional) `ask_sdk_model.ui.output_speech.OutputSpeech` -

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.simple_card module

class `ask_sdk_model.ui.simple_card.SimpleCard(title=None, content=None)`
Bases: `ask_sdk_model.ui.card.Card`

Parameters

• `title` (optional) `str` -

• `content` (optional) `str` -
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.ui.ssml_output_speech module

class ask_sdk_model.ui.ssml_output_speech.SsmlOutputSpeech(play_behavior=None, ssml=None)
    Bases: ask_sdk_model.ui.output_speech.OutputSpeech
    Parameters
    • play_behavior ((optional) ask_sdk_model.ui.play_behavior.PlayBehavior) -
    • ssml (optional) str -
    to_dict()
        Returns the model properties as a dict

to_str()
        Returns the string representation of the model

ask_sdk_model.ui.standard_card module

class ask_sdk_model.ui.standard_card.StandardCard(title=None, text=None, image=None)
    Bases: ask_sdk_model.ui.card.Card
    Parameters
    • title (optional) str -
    • text (optional) str -
    • image (optional) ask_sdk_model.ui.image.Image -
    to_dict()
        Returns the model properties as a dict

to_str()
        Returns the string representation of the model

Submodules

Note: Canonical imports have been added in the __init__.py of the package. This helps in importing the class directly from the package, than through the module.

For eg: if package a has module b with class C, you can do from a import C instead of from a.b import C.

2.14. Hosting Skills as Webservice 253
ask_sdk_model.application module

```python
class ask_sdk_model.application.Application(application_id=None)
    Bases: object
    An object containing an application ID. This is used to verify that the request was intended for your service.

    Parameters
    application_id ((optional) str) – A string representing the application identifier for your skill.

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

ask_sdk_model.context module

```python
class ask_sdk_model.context.Context(system=None, audio_player=None, automotive=None, display=None, geolocation=None, viewport=None)
    Bases: object
    Parameters

    • system ((optional) ask_sdk_model.interfaces.system.system_state.SystemState) – Provides information about the current state of the Alexa service and the device interacting with your skill.

    • audio_player ((optional) ask_sdk_model.interfaces.audioplayer.audio_player_state.AudioPlayerState) – Provides the current state for the AudioPlayer interface.

    • automotive ((optional) ask_sdk_model.interfaces.automotive.automotive_state.AutomotiveState) – Provides the automotive specific information of the device.

    • display ((optional) ask_sdk_model.interfaces.display.display_state.DisplayState) – Provides the current state for the Display interface.

    • geolocation ((optional) ask_sdk_model.interfaces.geolocation.geolocation_state.GeolocationState) – Provides the last gathered geolocation information of the device.

    • viewport ((optional) ask_sdk_model.interfaces.viewport.viewport_state.ViewportState) – Provides the characteristics of a device’s viewport.

to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model
```

ask_sdk_model.device module

```python
class ask_sdk_model.device.Device(device_id=None, supported_interfaces=None)
    Bases: object
```
An object providing information about the device used to send the request. The device object contains both deviceId and supportedInterfaces properties. The deviceId property uniquely identifies the device. The supportedInterfaces property lists each interface that the device supports. For example, if supportedInterfaces includes AudioPlayer {}, then you know that the device supports streaming audio using the AudioPlayer interface.

Parameters

- **device_id** *(optional)* `str` – The deviceId property uniquely identifies the device.
- **supported_interfaces** *(optional)* `ask_sdk_model.supported_interfaces.SupportedInterfaces` – Lists each interface that the device supports. For example, if supportedInterfaces includes AudioPlayer {}, then you know that the device supports streaming audio using the AudioPlayer interface.

`to_dict()`

Returns the model properties as a dict

`to_str()`

Returns the string representation of the model

`ask_sdk_model.dialog_state` module

class `ask_sdk_model.dialog_state.DialogState`

Bases: `enum.Enum`

Enumeration indicating the status of the multi-turn dialog. This property is included if the skill meets the requirements to use the Dialog directives. Note that COMPLETED is only possible when you use the Dialog.Delegate directive. If you use intent confirmation, dialogState is considered COMPLETED if the user denies the entire intent (for instance, by answering “no” when asked the confirmation prompt). Be sure to also check the confirmationStatus property on the Intent object before fulfilling the user’s request.

Allowed enum values: [STARTED, IN_PROGRESS, COMPLETED]

`to_dict()`

Returns the model properties as a dict

`to_str()`

Returns the string representation of the model

`ask_sdk_model.directive` module

class `ask_sdk_model.directive.Directive` *(object_type=None)*

Bases: `object`

Parameters `object_type` *(optional)* `str` –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

AudioPlayer.Stop:

`ask_sdk_model.interfaces.audioplayer.stop_directive.StopDirective`

Dialog.ConfirmSlot:

`ask_sdk_model.dialog.confirm_slot_directive.ConfirmSlotDirective`
AudioPlayer.Play:
`ask_sdk_model.interfaces.audioplayer.play_directive.PlayDirective`,


Connections.SendRequest: `ask_sdk_model.interfaces.connections.send_request_directive.SendRequestDirective`,

Dialog.UpdateDynamicEntities:
`ask_sdk_model.dialog.dynamic_entities_directive.DynamicEntitiesDirective`,

Display.RenderTemplate: `ask_sdk_model.interfaces.display.render_template_directive.RenderTemplateDirective`,

GadgetController.SetLight: `ask_sdk_model.interfaces.gadget_controller.set_light_directive.SetLightDirective`,

Dialog.Delegate: `ask_sdk_model.dialog.delegate_directive.DelegateDirective`,

Hint: `ask_sdk_model.interfaces.display.hint_directive.HintDirective`,

Dialog.ConfirmIntent:
`ask_sdk_model.dialog.confirm_intent_directive.ConfirmIntentDirective`,


GameEngine.StartInputHandler: `ask_sdk_model.interfaces.game_engine.start_input_handler_directive.StartInputHandlerDirective`,

VideoApp.Launch: `ask_sdk_model.interfaces.videoapp.launch_directive.LaunchDirective`,

GameEngine.StopInputHandler: `ask_sdk_model.interfaces.game_engine.stop_input_handler_directive.StopInputHandlerDirective`,

Tasks.CompleteTask: `ask_sdk_model.interfaces.tasks.complete_task_directive.CompleteTaskDirective`,


Dialog.ElicitSlot:
`ask_sdk_model.dialog.elicit_slot_directive.ElicitSlotDirective`,

Chapter 2. SDK Features

`classmethod get_real_child_model(data)`
Returns the real base class specified by the discriminator

`to_dict()`
Returns the model properties as a dict

`to_str()`
Returns the string representation of the model

`ask_sdk_model.intent module`

`class ask_sdk_model.intent.Intent (name=None, slots=None, confirmation_status=None)`
Bases: object
An object that represents what the user wants.

Parameters
- `name ((optional) str)` – A string representing the name of the intent.
- `slots ((optional) dict(str, ask_sdk_model.slot.Slot))` – A map of key-value pairs that further describes what the user meant based on a predefined intent schema. The map can be empty.
- `confirmation_status ((optional) ask_sdk_model.intent_confirmation_status.IntentConfirmationStatus)`

`to_dict()`
Returns the model properties as a dict

`to_str()`
Returns the string representation of the model

`ask_sdk_model.intent_confirmation_status module`

`class ask_sdk_model.intent_confirmation_status.IntentConfirmationStatus`  
Bases: enum.Enum
Indication of whether an intent or slot has been explicitly confirmed or denied by the user, or neither.

Allowed enum values: [NONE, DENIED, CONFIRMED]

`to_dict()`
Returns the model properties as a dict

`to_str()`
Returns the string representation of the model
ask_sdk_model.intent_request module

class ask_sdk_model.intent_request.IntentRequest(request_id=None, timestamp=None, locale=None, dialog_state=None, intent=None):
    Bases: ask_sdk_model.request.Request

    An IntentRequest is an object that represents a request made to a skill based on what the user wants to do.

    Parameters
    • request_id ((optional) str) – Represents the unique identifier for the specific request.
    • timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
    • locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
    • dialog_state ((optional) ask_sdk_model.dialog_state.DialogState) – Enumeration indicating the status of the multi-turn dialog. This property is included if the skill meets the requirements to use the Dialog directives. Note that COMPLETED is only possible when you use the Dialog.Delegate directive. If you use intent confirmation, dialogState is considered COMPLETED if the user denies the entire intent (for instance, by answering “no” when asked the confirmation prompt). Be sure to also check the confirmationStatus property on the Intent object before fulfilling the user’s request.
    • intent ((optional) ask_sdk_model.intent.Intent) – An object that represents what the user wants.

    to_dict()  
    Returns the model properties as a dict

    to_str()  
    Returns the string representation of the model

ask_sdk_model.launch_request module

class ask_sdk_model.launch_request.LaunchRequest(request_id=None, timestamp=None, locale=None, task=None):
    Bases: ask_sdk_model.request.Request

    Represents that a user made a request to an Alexa skill, but did not provide a specific intent.

    Parameters
    • request_id ((optional) str) – Represents the unique identifier for the specific request.
    • timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
    • locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
    • task ((optional) ask_sdk_model.task.Task) –
to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.permissions module

class ask_sdk_model.permissions.Permissions(consent_token=None, scopes=None)
    Bases: object
    
    Contains a consentToken allowing the skill access to information that the customer has consented to provide, such as address information. Note that the consentToken is deprecated. Use the apiAccessToken available in the context object to determine the user’s permissions.

    Parameters

    • consent_token ((optional) str) – A token listing all the permissions granted for this user.
    • scopes ((optional) dict(str, ask_sdk_model.scope.Scope)) – A map where the key is a LoginWithAmazon(LWA) scope and value is a list of key:value pairs which describe the state of user actions on the LWA scope. For e.g. &quot;scopes&quot; :{
      &quot;alexa::devices:all:geolocation:read&quot;: {
        &quot;status&quot;: &quot;GRANTED&quot;
      }
    }

    This value of &quot;alexa::devices:all:geolocation:read&quot; will determine if the Geolocation data access is granted by the user, or else it will show a card of type AskForPermissionsConsent to the user to get this permission.

    to_dict()
    Returns the model properties as a dict

to_str()
    Returns the string representation of the model

ask_sdk_model.request module

class ask_sdk_model.request.Request(object_type=None, request_id=None, timestamp=None, locale=None)
    Bases: object
    
    A request object that provides the details of the user’s request. The request body contains the parameters necessary for the service to perform its logic and generate a response.

    Parameters

    • object_type ((optional) str) – Describes the type of the request.
    • request_id ((optional) str) – Represents the unique identifier for the specific request.
    • timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
    • locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

playback_finished_request.PlaybackFinishedRequest,

AlexaSkillEvent.SkillEnabled: ask_sdk_model.events.skillevents.
skill_enabled_request.SkillEnabledRequest,

AlexaHouseholdListEvent.ListUpdated: ask_sdk_model.services.list_management.
list_updated_event_request.ListUpdatedEventRequest,

AlexaSkillEvent.ProactiveSubscriptionChanged:
ask_sdk_model.events.skillevents.proactive_subscription_changed_request.
ProactiveSubscriptionChangedRequest,

Alexa.Presentation.APL.UserEvent:
ask_sdk_model.interfaces.alexa.presentation.apl.user_event.UserEvent,

skill_disabled_request.SkillDisabledRequest,

Display.ElementSelected: ask_sdk_model.interfaces.display.
element_selected_request.ElementSelectedRequest,

AlexaSkillEvent.SkillPermissionChanged: ask_sdk_model.events.skillevents.
permission_changed_request.PermissionChangedRequest,

AlexaHouseholdListEvent.ItemsCreated: ask_sdk_model.services.list_management.
list_items_created_event_request.ListItemsCreatedEventRequest,

Reminders.ReminderUpdated: ask_sdk_model.servicesreminder_management.
reminder_updated_event_request.ReminderUpdatedEventRequest,

SessionResumedRequest:
ask_sdk_model.session_resumed_request.SessionResumedRequest,

SessionEndedRequest: ask_sdk_model.session_ended_request.SessionEndedRequest,

IntentRequest: ask_sdk_model.intent_request.IntentRequest,

playback_failed_request.PlaybackFailedRequest,

CanFulfillIntentRequest: ask_sdk_model.canfulfill.can_fulfill_intent_request.
CanFulfillIntentRequest,
Reminders.ReminderStarted: ask_sdk_model.services.reminder_management.
reminder_started_event_request.ReminderStartedEventRequest,

LaunchRequest: ask_sdk_model.launch_request.LaunchRequest,

Reminders.ReminderCreated: ask_sdk_model.services.reminder_management.
reminder_created_event_request.ReminderCreatedEventRequest,

playback_stopped_request.PlaybackStoppedRequest,

PlaybackController.PreviousCommandIssued:
ask_sdk_model.interfaces.playbackcontroller.
previous_command_issued_request.PreviousCommandIssuedRequest,

AlexaHouseholdListEvent.ItemsUpdated: ask_sdk_model.services.list_management.
list_items_updated_event_request.ListItemsUpdatedEventRequest,

account_linked_request.AccountLinkedRequest,

AlexaHouseholdListEvent.ListCreated: ask_sdk_model.services.list_management.
list_created_event_request.ListCreatedEventRequest,

playback_started_request.PlaybackStartedRequest,

playback_nearly_finished_request.PlaybackNearlyFinishedRequest,

Reminders.ReminderStatusChanged: ask_sdk_model.services.reminder_management.
reminder_status_changed_event_request.ReminderStatusChangedEventRequest,

AlexaHouseholdListEvent.ItemsDeleted: ask_sdk_model.services.list_management.
list_items_deleted_event_request.ListItemsDeletedEventRequest,

Reminders.ReminderDeleted: ask_sdk_model.services.reminder_management.
reminder_deleted_event_request.ReminderDeletedEventRequest,

ConnectionsResponse,

Messaging.MessageReceived: ask_sdk_model.interfaces.messaging.
message_received_request.MessageReceivedRequest,

Connections.Request: ask_sdk_model.interfaces.connections.connections_request.
ConnectionsRequest,

System.ExceptionEncountered: ask_sdk_model.interfaces.system.
exception_encountered_request.ExceptionEncounteredRequest,

AlexaSkillEvent.SkillPermissionAccepted: ask_sdk_model.events.skillevents.
permission_accepted_request.PermissionAcceptedRequest,

AlexaHouseholdListEvent.ListDeleted: ask_sdk_model.services.list_management.
list_deleted_event_request.ListDeletedEventRequest,

GameEngine.InputHandlerEvent: ask_sdk_model.interfaces.game_engine.
input_handler_event_request.InputHandlerEventRequest,

PlaybackController.NextCommandIssued: ask_sdk_model.interfaces.playbackcontroller.
next_command_issued_request.NextCommandIssuedRequest,

PlaybackController.PauseCommandIssued: ask_sdk_model.interfaces.playbackcontroller.
pause_command_issued_request.PauseCommandIssuedRequest,

PlaybackController.PlayCommandIssued: ask_sdk_model.interfaces.playbackcontroller.
play_command_issued_request.PlayCommandIssuedRequest

classmethod get_real_child_model(data)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.request_envelope module

class ask_sdk_model.request_envelope.RequestEnvelope (version=None, session=None,
context=None, request=None)

Bases: object

Request wrapper for all requests sent to your Skill.

Parameters

- version (optional) str – The version specifier for the request.
- session (optional) ask_sdk_model.session.Session – The session object
  provides additional context associated with the request.
- context (optional) ask_sdk_model.context.Context – The context object
  provides your skill with information about the current state of the Alexa
  service and device at the time the request is sent to your service. This is included
  on all requests. For requests sent in the context of a session (LaunchRequest and
  IntentRequest), the context object duplicates the user and application information
  that is also available in the session.
- request (optional) ask_sdk_model.request.Request – A request object
  that provides the details of the user’s request.
to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.response module

class ask_sdk_model.response.Response (output_speech=None, card=None, reprompt=None, directives=None, should_end_session=None, can_fulfill_intent=None)

Bases: object

Parameters

• output_speech ((optional) ask_sdk_model.ui.output_speech.OutputSpeech)-
• card ((optional) ask_sdk_model.ui.card.Card)-
• reprompt ((optional) ask_sdk_model.ui.reprompt.Reprompt)-
• directives ((optional) list(ask_sdk_model.directive.Directional)-
• should_end_session ((optional) bool)-
• can_fulfill_intent ((optional) ask_sdk_model.canfulfill.can_fulfill_intent.CanFulfillIntent)-

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.response_envelope module

class ask_sdk_model.response_envelope.ResponseEnvelope (version=None, session_attributes=None, user_agent=None, response=None)

Bases: object

Parameters

• version ((optional) str)-
• session_attributes ((optional) dict(str, object))-  
• user_agent ((optional) str)-
• response ((optional) ask_sdk_model.response.Response)-

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

2.14. Hosting Skills as Webservice 263
ask_sdk_model.session module

class ask_sdk_model.session.Session(new=None, session_id=None, user=None, attributes=None, application=None)

Bases: object

Represents a single execution of the alexa service

Parameters

- **new** *(optional) bool* – A boolean value indicating whether this is a new session. Returns true for a new session or false for an existing session.

- **session_id** *(optional) str* – A string that represents a unique identifier per a user’s active session.

- **user** *(optional) ask_sdk_model.user.User* – An object that describes the user making the request.

- **attributes** *(optional) dict(str, object)* – A map of key-value pairs. The attributes map is empty for requests where a new session has started with the property new set to true. When returning your response, you can include data you need to persist during the session in the sessionAttributes property. The attributes you provide are then passed back to your skill on the next request.

- **application** *(optional) ask_sdk_model.application.Application* –

  to_dict()

  Returns the model properties as a dict

  to_str()

  Returns the string representation of the model

ask_sdk_model.session_ended_error module

class ask_sdk_model.session_ended_error.SessionEndedError(object_type=None, message=None)

Bases: object

An error object providing more information about the error that occurred.

Parameters

- **object_type** *(optional) ask_sdk_model.session_ended_error_type.SessionEndedErrorType* – A string indicating the type of error that occurred.

- **message** *(optional) str* – A string providing more information about the error.

  to_dict()

  Returns the model properties as a dict

  to_str()

  Returns the string representation of the model

ask_sdk_model.session_ended_error_type module

class ask_sdk_model.session_ended_error_type.SessionEndedErrorType

Bases: enum.Enum
A string indicating the type of error that occurred.

Allowed enum values: [INVALID_RESPONSE, DEVICE_COMMUNICATION_ERROR, INTERNAL_SERVICE_ERROR, ENDPOINT_TIMEOUT]

to_dict()
   Returns the model properties as a dict
to_str()
   Returns the string representation of the model

ask_sdk_model.session_ended_reason module

class ask_sdk_model.session_ended_reason.SessionEndedReason
   Bases: enum.Enum
   The reason why session ended when not initiated from the Skill itself.
   Allowed enum values: [USER_INITIATED, ERROR, EXCEEDED_MAX_REPROMPTS]
   to_dict()
      Returns the model properties as a dict
to_str()
      Returns the string representation of the model

ask_sdk_model.session_ended_request module

class ask_sdk_model.session_ended_request.SessionEndedRequest
   Bases: ask_sdk_model.request.Request
   A SessionEndedRequest is an object that represents a request made to an Alexa skill to notify that a session was ended. Your service receives a SessionEndedRequest when a currently open session is closed for one of the following reasons: <ol><li>The user says “exit”</li><li>The user does not respond or says something that does not match an intent defined in your voice interface while the device is listening for the user’s response</li><li>an error occurs</li></ol>

Parameters
   • request_id ((optional) str) – Represents the unique identifier for the specific request.
   • timestamp ((optional) datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
   • locale ((optional) str) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.
   • reason ((optional) ask_sdk_model.session_ended_reason.SessionEndedReason) – Describes why the session ended.
   • error ((optional) ask_sdk_model.session_ended_error.SessionEndedError) – An error object providing more information about the error that occurred.
to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

ask_sdk_model.slot module

class ask_sdk_model.slot.Slot(name=None, value=None, confirmation_status=None, resolutions=None)
Bases: object

Parameters

- **name** *(optional) str* – A string that represents the name of the slot.
- **value** *(optional) str* – A string that represents the value the user spoke for the slot. This is the actual value the user spoke, not necessarily the canonical value or one of the synonyms defined for the entity. Note that AMAZON.LITERAL slot values sent to your service are always in all lower case.
- **confirmation_status** *(optional) ask_sdk_model.slot_confirmation_status.SlotConfirmationStatus* – Indication of whether an intent or slot has been explicitly confirmed or denied by the user, or neither.
- **resolutions** *(optional) ask_sdk_model.slu.entityresolution.resolutions.Resolutions* – Contains the results of entity resolution. These are organized by authority. An authority represents the source for the data provided for the slot. For a custom slot type, the authority is the slot type you defined.

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

ask_sdk_model.slot_confirmation_status module

class ask_sdk_model.slot_confirmation_status.SlotConfirmationStatus
Bases: enum.Enum

An enumeration indicating whether the user has explicitly confirmed or denied the value of this slot.

Allowed enum values: [NONE, DENIED, CONFIRMED]

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model
ask_sdk_model.supported_interfaces module

class ask_sdk_model.supported_interfaces.SupportedInterfaces (alexa_presentation_apl=None, audio_player=None, display=None, video_app=None, geolocation=None)

Bases: object

An object listing each interface that the device supports. For example, if supportedInterfaces includes AudioPlayer {}, then you know that the device supports streaming audio using the AudioPlayer interface.

Parameters

• alexa_presentation_apl ((optional) ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface.AlexaPresentationAplInterface)
• audio_player ((optional) ask_sdk_model.interfaces.audioplayer.audio_player_interface.AudioPlayerInterface)
• display ((optional) ask_sdk_model.interfaces.display_interface.DisplayInterface)
• video_app ((optional) ask_sdk_model.interfaces.videoapp.video_app_interface.VideoAppInterface)
• geolocation ((optional) ask_sdk_model.interfaces.geolocation_interface.GeolocationInterface)

to_dict ()
    Returns the model properties as a dict
to_str ()
    Returns the string representation of the model

ask_sdk_model.user module

class ask_sdk_model.user.User (user_id=None, access_token=None, permissions=None)

Bases: object

Represents the user registered to the device initiating the request.

Parameters

• user_id (optional str) – A string that represents a unique identifier for the user who made the request. The length of this identifier can vary, but is never more than 255 characters. The userId is automatically generated when a user enables the skill in the Alexa app. Note: Disabling and re-enabling a skill generates a new identifier.

• access_token (optional str) – A token identifying the user in another system. This is only provided if the user has successfully linked their account. See Linking an Alexa User with a User in Your System for more details.

• permissions (optional ask_sdk_model.permissions.Permissions) –
to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

**ask_sdk_model.permission_status module**

class ask_sdk_model.permission_status.PermissionStatus  
Bases: enum.Enum

This denotes the status of the permission scope.
Allowed enum values: [GRANTED, DENIED]

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model

**ask_sdk_model.scope module**

class ask_sdk_model.scope.Scope(status=None)  
Bases: object

This is the value of LoginWithAmazon(LWA) consent scope. This object is used as in the key-value pairs that
are provided in user.permissions.scopes object

Parameters status  
((optional) ask_sdk_model.permission_status.PermissionStatus) –

to_dict()  
Returns the model properties as a dict

to_str()  
Returns the string representation of the model
Got Feedback?

- We would like to hear about your bugs, feature requests, questions or quick feedback. Please search for existing issues before opening a new one. It would also be helpful if you follow the templates for issue and pull request creation. Please follow the contributing guidelines for pull requests!!
- Request and vote for Alexa features!
4.1 Community

- Amazon Developer Forums: Join the conversation!
- Hackster.io - See what others are building with Alexa.

4.2 Tutorials & Guides

- Voice Design Guide - A great resource for learning conversational and voice user interface design.
CHAPTER 5

Indices and tables

• genindex
• modindex
Python Module Index

ask_sdk.standard, 89
ask_sdk_core.api_client, 80
ask_sdk_core.dispatch_components.exception_components, 71
ask_sdk_core.dispatch_components.request_components, 70
ask_sdk_core.exceptions, 81
ask_sdk_core.serialize, 81
ask_sdk_core.skill, 74
ask_sdk_core.skill_builder, 75
ask_sdk_core.utils.predicate, 82
ask_sdk_core.utils.request_util, 84
ask_sdk_core.utils.viewport, 83
ask_sdk_dynamodb.adapter, 88
ask_sdk_dynamodb.partition_keygen, 89
ask_sdk_model.application, 254
ask_sdk_model.canfulfill.can_fulfill_intent, 102
ask_sdk_model.canfulfill.can_fulfill_intent_request, 99
ask_sdk_model.canfulfill.can_fulfill_intent_values, 100
ask_sdk_model.canfulfill.can_fulfill_slot, 101
ask_sdk_model.canfulfill.can_fulfill_slot_values, 101
ask_sdk_model.canfulfill.can_understand_slot_values, 107
ask_sdk_model.context, 254
ask_sdk_model.device, 254
ask_sdk_model.dialog.confirm_intent_directive, 102
ask_sdk_model.dialog.confirm_intent_directive, 102
ask_sdk_model.dialog.confirm_slot_directive, 102
ask_sdk_model.dialog.delegateDirective, 102
ask_sdk_model.dialog.elicit_slot_directive, 102
ask_sdk_model.dialog.state, 255
ask_sdk_model.directive, 255
ask_sdk_model.events.skillevents.account_linked_body, 103
ask_sdk_model.events.skillevents.account_linked_request, 103
ask_sdk_model.events.skillevents.permission, 104
ask_sdk_model.events.skillevents.permission_accepted_request, 104
ask_sdk_model.events.skillevents.permission_body, 105
ask_sdk_model.events.skillevents.permission_changed_request, 105
ask_sdk_model.events.skillevents.skill_disabled_request, 106
ask_sdk_model.events.skillevents.skill_enabled_request, 106
ask_sdk_model.intent, 257
ask_sdk_model.intent_confirmation_status, 257
ask_sdk_model.intent_request, 258
ask_sdk_model.interfaces.alexa, 124
ask_sdk_model.interfaces.alexa.presentation, 124
ask_sdk_model.interfaces.alexa.presentation.apl, 107
ask_sdk_model.interfaces.alexa.presentation.apl.audio_track, 114
ask_sdk_model.interfaces.alexa.presentation.apl.auto_page_command, 108
ask_sdk_model.interfaces.alexa.presentation.apl.command, 109
ask_sdk_model.interfaces.alexa.presentation.apl.component_state, 115
ask_sdk_model.interfaces.alexa.presentation.apl.control_media_command, 115
ask_sdk_model.interfaces.alexa.presentation.apl.align, 107
ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface, 107
ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface, 107
ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface, 107
ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface, 107
ask_sdk_model.interfaces.alexa.presentation.apl.execute_commands_directive

ask_sdk_model.interfaces.alexa.presentation.apl.highlight_mode

ask_sdk_model.interfaces.alexa.presentation.apl.idle_command

ask_sdk_model.interfaces.alexa.presentation.apl.media_command_type

ask_sdk_model.interfaces.alexa.presentation.apl.parallel_command

ask_sdk_model.interfaces.alexa.presentation.apl.play_media_command

ask_sdk_model.interfaces.alexa.presentation.apl.position

ask_sdk_model.interfaces.alexa.presentation.apl.render_document_directive

ask_sdk_model.interfaces.alexa.presentation.apl.runtime

ask_sdk_model.interfaces.alexa.presentation.apl.scroll_command

ask_sdk_model.interfaces.alexa.presentation.apl.scroll_to_index_command

ask_sdk_model.interfaces.alexa.presentation.apl.send_event_command

ask_sdk_model.interfaces.alexa.presentation.apl.sequential_command

ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command

ask_sdk_model.interfaces.alexa.presentation.apl.set_state_command

ask_sdk_model.interfaces.alexa.presentation.apl.set_value_command

ask_sdk_model.interfaces.alexa.presentation.apl.speak_item_command

ask_sdk_model.interfaces.alexa.presentation.apl.speak_list_command

ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes

ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity

ask_sdk_model.interfaces.amazonpay.model.request.billing_agreement_attributes

ask_sdk_model.interfaces.amazonpay.model.request.payment_action

ask_sdk_model.interfaces.amazonpay.model.request.price

ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes

ask_sdk_model.interfaces.amazonpay.model.request.provider_credit

ask_sdk_model.interfaces.amazonpay.model.request.seller_billing_agreement_attributes

ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes

ask_sdk_model.interfaces.amazonpay.model.response.authorization_details

ask_sdk_model.interfaces.amazonpay.model.response.authorization_status

ask_sdk_model.interfaces.amazonpay.model.response.billing_agreement_details

ask_sdk_model.interfaces.amazonpay.model.response.destination

ask_sdk_model.interfaces.amazonpay.model.response.price

ask_sdk_model.interfaces.amazonpay.model.response.release_environment

ask_sdk_model.interfaces.amazonpay.model.response.state

ask_sdk_model.interfaces.amazonpay.model.v1.authorization_details

ask_sdk_model.interfaces.amazonpay.model.v1.authorization_status

ask_sdk_model.interfaces.amazonpay.model.v1.authorize_attributes

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_attributes

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status

ask_sdk_model.interfaces.amazonpay.model.v1.destination

ask_sdk_model.interfaces.amazonpay.model.v1.payment_action

ask_sdk_model.interfaces.amazonpay.model.v1.price

ask_sdk_model.interfaces.amazonpay.model.v1.provider_attributes

ask_sdk_model.interfaces.amazonpay.model.v1.provider_credit

ask_sdk_model.interfaces.amazonpay.model.v1.release_environment

ask_sdk_model.interfaces.amazonpay.model.v1.seller_billing_agreement_attributes

ask_sdk_model.interfaces.amazonpay.model.v1.seller_order_attributes

ask_sdk_model.interfaces.amazonpay.model.v1.state
Python Module Index
AbstractExceptionHandler (class in ask_sdk_core.dispatch_components.exception_components), 71
AbstractExceptionHandler (class in ask_sdk_runtime.dispatch_components.exception_components), 62
AbstractExceptionHandler (class in ask_sdk_runtime.dispatch_components.exception_components), 62
AbstractExceptionHandler (class in ask_sdk_runtime.dispatch_components.exception_components), 62
AbstractExceptionMapper (class in ask_sdk_runtime.dispatch_components.exception_components), 62
AbstractHandlerAdapter (class in ask_sdk_runtime.dispatch_components.request_components), 61
AbstractHandlerAdapter (class in ask_sdk_runtime.dispatch_components.request_components), 61
AbstractPersistenceAdapter (class in ask_sdk_core.attributes_manager), 80
AbstractRequestDispatcher (class in ask_sdk_runtime.dispatch), 59
AbstractRequestDispatcher (class in ask_sdk_runtime.dispatch_components.request_components), 70
AbstractRequestHandler (class in ask_sdk_runtime.dispatch_components.request_components), 70
AbstractRequestHandlingChain (class in ask_sdk_runtime.dispatch_components.request_components), 61
AbstractRequestInterceptor (class in ask_sdk_core.dispatch_components.request_components), 70
AbstractRequestInterceptor (class in ask_sdk_core.dispatch_components.request_components), 70
AbstractRequestInterceptor (class in ask_sdk_core.dispatch_components.request_components), 60
AbstractRequestInterceptor (class in ask_sdk_core.dispatch_components.request_components), 60
AbstractRequestInterceptor (class in ask_sdk_core.dispatch_components.request_components), 60
AbstractResponseInterceptor (class in ask_sdk_core.dispatch_components.request_components), 71
AbstractResponseInterceptor (class in ask_sdk_core.dispatch_components.request_components), 71
AbstractResponseInterceptor (class in ask_sdk_runtime.dispatch_components.exception_components.GenericExceptionMapper method), 65
AbstractResponseInterceptor (class in ask_sdk_runtime.dispatch_components.request_components), 71
AbstractResponseInterceptor (class in ask_sdk_runtime.dispatch_components.request_components), 71
AbstractResponseInterceptor (class in ask_sdk_runtime.dispatch_components.request_components), 71
add_exception_handler ()
(ask_sdk_runtime.skill_builder.AbstractSkillBuilder
method), 66

add_request_interceptor ()
(ask_sdk_runtime.dispatch_components.request_components.Gener
method), 66

add_global_request_interceptor ()
(ask_sdk.standard.StandardSkillBuilder
method), 68

add_global_response_interceptor ()
(ask_sdk.standard.StandardSkillBuilder
method), 90

add_global_request_interceptor ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 77

add_global_response_interceptor ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 77

add_global_request_interceptor ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_global_response_interceptor ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_handler ()
(ask_sdk.standard.StandardSkillBuilder
method), 90

add_request_handler ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 78

add_request_handler ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_handler ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_handler ()
(ask_sdk_runtime.skill_runtime.dispatch_components.request_components.Gener
method), 64

add_request_handler ()
(ask_sdk_runtime.dispatch_components.request_components.Gener
method), 64

add_request_interceptor ()
(ask_sdk_runtime.dispatch_components.request_components.Gener
method), 64

add_response_interceptor ()
(ask_sdk_runtime.dispatch_components.request_components.Gener
method), 64

add_global_request_interceptor ()
(ask_sdk.standard.StandardSkillBuilder
method), 90

add_global_response_interceptor ()
(ask_sdk.standard.StandardSkillBuilder
method), 90

add_global_request_interceptor ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 77

add_global_response_interceptor ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 77

add_global_request_interceptor ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_global_response_interceptor ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_handler ()
(ask_sdk.standard.StandardSkillBuilder
method), 90

add_request_handler ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 78

add_request_handler ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_handler ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_interceptor ()
(ask_sdk_runtime.dispatch_components.request_components.Gener
method), 64

add_response_interceptor ()
(ask_sdk_runtime.dispatch_components.request_components.Gener
method), 64

add_global_request_interceptor ()
(ask_sdk.standard.StandardSkillBuilder
method), 90

add_global_response_interceptor ()
(ask_sdk.standard.StandardSkillBuilder
method), 90

add_global_request_interceptor ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 77

add_global_response_interceptor ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 77

add_global_request_interceptor ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_global_response_interceptor ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_handler ()
(ask_sdk.standard.StandardSkillBuilder
method), 90

add_request_handler ()
(ask_sdk_core.skill_builder.CustomSkillBuilder
method), 78

add_request_handler ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_handler ()
(ask_sdk_runtime.skill.RuntimeConfigurationBuilder
method), 66

add_request_interceptor ()
(ask_sdk_runtime.dispatch_components.request_components.Gener
method), 64

add_response_interceptor ()
(ask_sdk_runtime.dispatch_components.request_components.Gener
method), 64

Address (class in ask_sdk_model.services.device_address.address), 194

AlertInfo (class in ask_sdk_model.services.reminder_management.alert_info), 221

AlexaList (class in ask_sdk_model.services.list_management.alexa_list), 206

AlexaListItem (class in ask_sdk_model.services.list_management.alexa_list_item), 206

AlexaListMetadata (class in ask_sdk_model.services.list_management.alexa_list_metadata), 207

AlexaListsMetadata (class in ask_sdk_model.services.list_management.alexa_lists_metadata), 207

AlexaPresentationAplInterface (class in ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl), 107

Align (class in ask_sdk_model.interfaces.alexa.presentation.apl.align), 107

AnimationStep (class in ask_sdk_model.interfaces.gadget_controller.animation_step), 178

AmazonPayErrorResponse (class in ask_sdk_model.interfaces.amazonpay.response.amazon_pay_error_response), 149

AmazonPayErrorResponse (class in ask_sdk_model.interfaces.amazonpay.v1.amazon_pay_error_response), 144

ApiClient (class in ask_sdk_model.services.api_client), 243

ApiClientException, 81

ApiClientMessage (class in ask_sdk_model.services.api_client_message), 243

ApiClientRequest (class in ask_sdk_model.services.api_client_request), 243

ApiClientResponse (class in ask_sdk_model.services.api_client_response), 244

ApiClient (class in ask_sdk_model.services.api_client), 243

ApiClientException, 81

ApiClientMessage (class in ask_sdk_model.services.api_client_message), 243

ApiClientRequest (class in ask_sdk_model.services.api_client_request), 243

ApiClientResponse (class in ask_sdk_model.services.api_client_response), 244

ApiClient (class in ask_sdk_model.services.api_client), 243

ApiClientException, 81

ApiClientMessage (class in ask_sdk_model.services.api_client_message), 243

ApiClientRequest (class in ask_sdk_model.services.api_client_request), 243

ApiClientResponse (class in ask_sdk_model.services.api_client_response), 244

ApiClient (class in ask_sdk_model.services.api_client), 243

ApiClientException, 81

ApiClientMessage (class in ask_sdk_model.services.api_client_message), 243

ApiClientRequest (class in ask_sdk_model.services.api_client_request), 243

ApiClientResponse (class in ask_sdk_model.services.api_client_response), 244

ApiClient (class in ask_sdk_model.services.api_client), 243

ApiClientException, 81

ApiClientMessage (class in ask_sdk_model.services.api_client_message), 243

ApiClientRequest (class in ask_sdk_model.services.api_client_request), 243

ApiClientResponse (class in ask_sdk_model.services.api_client_response), 244
Application (class in ask_sdk_model.application), 254
apply (class method), 103
ask (class method), 72
ask_sdk.core.response_helper.ResponseFactory (module), 106
ask_sdk_model.events.skillevents.account_linked_request (module), 103
ask_sdk_model.events.skillevents.permission (module), 104
ask_sdk_model.events.skillevents.permission_accepted_request (module), 104
ask_sdk_model.events.skillevents.permission_body (module), 105
ask_sdk_model.events.skillevents.permission_change_request (module), 105
ask_sdk_model.events.skillevents.skill_enabled_request (module), 106
ask_sdk_model.events.skillevents.skill_disabled_request (module), 106
ask_sdk_model.intent (module), 257
ask_sdk_model.intent_confirmation_status (module), 257
ask_sdk_model.intent_request (module), 258
ask_sdk_model.interfaces.alexa (module), 124
ask_sdk_model.interfaces.alexa.presentation (module), 124
ask_sdk_model.interfaces.alexa.presentation.apl.alexapresentationutility (module), 107
ask_sdk_model.interfaces.alexa.presentation.apl.apl (module), 107
ask_sdk_model.interfaces.alexa.presentation.apl.angular (module), 114
ask_sdk_model.interfaces.alexa.presentation.apl.audio (module), 108
ask_sdk_model.interfaces.alexa.presentation.apl.button (module), 109
ask_sdk_model.interfaces.alexa.presentation.apl.canfulfill_can_fulfill_intent (module), 100
ask_sdk_model.interfaces.alexa.presentation.apl.canfulfill_can_fulfill_intent_slotslotvalue (module), 100
ask_sdk_model.interfaces.alexa.presentation.apl.canfulfill_can_fulfill_intent_slotvalue (module), 101
ask_sdk_model.interfaces.alexa.presentation.apl.canfulfill_can_understand_slotvalue (module), 101
ask_sdk_model.interfaces.alexa.presentation.apl.component (module), 254
ask_sdk_model.interfaces.alexa.presentation.apl.device (module), 254
ask_sdk_model.interfaces.alexa.presentation.apl.dialog.confirm_intent_directive (module), 116
ask_sdk_model.interfaces.alexa.presentation.apl.dialog.confirm_slot_directive (module), 116
ask_sdk_model.interfaces.alexa.presentation.apl.delegateDirective (module), 116
ask_sdk_model.interfaces.alexa.presentation.apl.elicitation_slotdialog (module), 117
ask_sdk_model.interfaces.alexa.presentation.apl.elicitation_slotdialog_slot (module), 117
ask_sdk_model.interfaces.alexa.presentation.apl.session (module), 111
ask_sdk_model.interfaces.alexa.presentation.apl.state (module), 111
ask_sdk_model.events.skillevents.account_linked_request (module), 103
ask_sdk_model.events.skillevents.account_linked_request ATTRIBUTE (module), 255
ask_sdk_model.events.skillevents.account_linked_request METHOD (module), 255
ask_sdk_model.events.skillevents.account_linked_request CLASS METHOD (module), 255
ask_sdk_model.events.skillevents.account_linked_request MODULE (module), 255
ask_sdk_model.events.skillevents.permission (module), 103
ask_sdk_model.events.skillevents.permission_accepted_request (module), 104
ask_sdk_model.events.skillevents.permission_body (module), 105
ask_sdk_model.events.skillevents.permission_change_request (module), 105
ask_sdk_model.events.skillevents.skill_enabled_request (module), 106
ask_sdk_model.events.skillevents.skill_disabled_request (module), 106
ask_sdk_model.intent (module), 257
ask_sdk_model.intent_confirmation_status (module), 257
ask_sdk_model.intent_request (module), 258
ask_sdk_model.interfaces.alexa (module), 124
ask_sdk_model.interfaces.alexa.presentation (module), 124
ask_sdk_model.interfaces.alexa.presentation.apl.alexapresentationutility (module), 107
ask_sdk_model.interfaces.alexa.presentation.apl.apl (module), 107
ask_sdk_model.interfaces.alexa.presentation.apl.angular (module), 114
ask_sdk_model.interfaces.alexa.presentation.apl.audio (module), 108
ask_sdk_model.interfaces.alexa.presentation.apl.button (module), 109
ask_sdk_model.interfaces.alexa.presentation.apl.canfulfill_can_fulfill_intent (module), 100
ask_sdk_model.interfaces.alexa.presentation.apl.canfulfill_can_fulfill_intent_slotslotvalue (module), 100
ask_sdk_model.interfaces.alexa.presentation.apl.canfulfill_can_fulfill_intent_slotvalue (module), 101
ask_sdk_model.interfaces.alexa.presentation.apl.canfulfill_can_understand_slotvalue (module), 101
ask_sdk_model.interfaces.alexa.presentation.apl.component (module), 254
ask_sdk_model.interfaces.alexa.presentation.apl.device (module), 254
ask_sdk_model.interfaces.alexa.presentation.apl.dialog.confirm_intent_directive (module), 116
ask_sdk_model.interfaces.alexa.presentation.apl.dialog.confirm_slot_directive (module), 116
ask_sdk_model.interfaces.alexa.presentation.apl.delegateDirective (module), 116
ask_sdk_model.interfaces.alexa.presentation.apl.elicitation_slotdialog (module), 117
ask_sdk_model.interfaces.alexa.presentation.apl.elicitation_slotdialog_slot (module), 117
ask_sdk_model.interfaces.alexa.presentation.apl.session (module), 111
ask_sdk_model.interfaces.alexa.presentation.apl.state (module), 111
ask_sdk_model.events.skillevents.account_linked_request (module), 103
ask_sdk_model.events.skillevents.account_linked_request ATTRIBUTE (module), 255
ask_sdk_model.events.skillevents.account_linked_request METHOD (module), 255
ask_sdk_model.events.skillevents.account_linked_request CLASS METHOD (module), 255
ask_sdk_model.events.skillevents.account_linked_request MODULE (module), 255
ask_sdk_model.interfaces.alexa.presentation.apl.audio_item
(module), 118
ask_sdk_model.interfaces.alexa.presentation.apl.audio_item_metadata
(module), 118
ask_sdk_model.interfaces.alexa.presentation.apl.command
(module), 119
ask_sdk_model.interfaces.amazonpay.v1.billing_agreement_status
(module), 120
ask_sdk_model.interfaces.amazonpay.v1.billing_agreement_details
(module), 120
ask_sdk_model.interfaces.amazonpay.v1.billing_agreement_attributes
(module), 121
ask_sdk_model.interfaces.amazonpay.v1.authorization_details
(module), 121
ask_sdk_model.interfaces.amazonpay.v1.authorization_status
(module), 123
ask_sdk_model.interfaces.amazonpay.v1.authorize_attributes
(module), 123
ask_sdk_model.interfaces.amazonpay.response.amazon_pay_error_response
(module), 123
ask_sdk_model.interfaces.amazonpay.response.charge_amazon_pay_result
(module), 123
ask_sdk_model.interfaces.amazonpay.response.charge_amazon_pay_request
(module), 123
ask_sdk_model.interfaces.amazonpay.response.release_environment
(module), 123
ask_sdk_model.interfaces.amazonpay.response.request.seller_order_attributes
(module), 123
ask_sdk_model.interfaces.amazonpay.response.request.seller_billing_agreement_attributes
(module), 123
ask_sdk_model.interfaces.amazonpay.response.request.provider_credit
(module), 123
ask_sdk_model.interfaces.amazonpay.response.request.provider_attributes
(module), 123
ask_sdk_model.interfaces.amazonpay.response.request.price
(module), 123
ask_sdk_model.interfaces.amazonpay.response.request.payment_action
(module), 123
ask_sdk_model.interfaces.amazonpay.response.request.billing_agreement_attributes
(module), 123
ask_sdk_model.interfaces.amazonpay.v1.command
(module), 124
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 124
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 124
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 124
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 125
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 125
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 125
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 126
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 126
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 126
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 127
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 127
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 127
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 128
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 128
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 128
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 129
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 129
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 129
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 130
ask_sdk_modelinterfaces.amazonpay.v1.authorization_response
(module), 130
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 130
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 131
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 131
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 131
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 132
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 132
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 132
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 133
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 133
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 133
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 134
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 134
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 134
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 135
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 135
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 135
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 136
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 136
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 136
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 137
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 137
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 137
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 138
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 138
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 138
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 139
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 139
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 139
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 140
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 140
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 140
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 141
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 141
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 141
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 142
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 142
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 142
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 143
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 143
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 143
ask_sdk_model.interfaces.amazonpay.v1.authorization_request
(module), 144
ask_sdk_model.interfaces.amazonpay.v1.authorization_response
(module), 144
ask_sdk_model.interfaces.amazonpay.v1.authorization
(module), 144
ask_sdk_model.interfaces.audioplayer.audio_player_interface (module), 152
ask_sdk_model.interfaces.audioplayer.audio_player_state (module), 152
ask_sdk_model.interfaces.audioplayer.clear_behavior (module), 152
ask_sdk_model.interfaces.audioplayer.clear_queue_directive (module), 152
ask_sdk_model.interfaces.audioplayer.current_playback_state (module), 153
ask_sdk_model.interfaces.audioplayer.error (module), 153
ask_sdk_model.interfaces.audioplayer.error_type (module), 153
ask_sdk_model.interfaces.audioplayer.play_behavior (module), 154
ask_sdk_model.interfaces.audioplayer.play_directive (module), 154
ask_sdk_model.interfaces.audioplayer.playback_failed_request (module), 154
ask_sdk_model.interfaces.audioplayer.playback_finished_request (module), 155
ask_sdk_model.interfaces.audioplayer.playback_nearly_finished_request (module), 156
ask_sdk_model.interfaces.audioplayer.playback_started_request (module), 156
ask_sdk_model.interfaces.audioplayer.playback_stopped_request (module), 157
ask_sdk_model.interfaces.audioplayer.player_activity (module), 157
ask_sdk_model.interfaces.audioplayer.stop_directive (module), 158
ask_sdk_model.interfaces.audioplayer.stream (module), 158
ask_sdk_model.interfaces.connections.connections_request (module), 164
ask_sdk_model.interfaces.connections.connections_response (module), 165
ask_sdk_model.interfaces.connections.connections_status (module), 166
ask_sdk_model.interfaces.connections.entities.base_entity (module), 159
ask_sdk_model.interfaces.connections.entities.postal_address (module), 159
ask_sdk_model.interfaces.connections.entities.restaurant (module), 160
ask_sdk_model.interfaces.connections.requests.base_request (module), 160
ask_sdk_model.interfaces.connections.requests.print_image_request (module), 161
ask_sdk_model.interfaces.connections.requests.print_pdf_request (module), 162
ask_sdk_model.interfaces.connections.requests.print_web_page_request (module), 162
ask_sdk_model.interfaces.connections.requests.schedule_food_establishment_reservation_request (module), 163
ask_sdk_model.interfaces.connections.requests.schedule_taxi_reservation_request (module), 163
ask_sdk_model.interfaces.connections.send_request_directive (module), 166
ask_sdk_model.interfaces.connections.send_response_directive (module), 167
ask_sdk_model.interfaces.display.back_button_behavior (module), 167
ask_sdk_model.interfaces.display.body_template1 (module), 168
ask_sdk_model.interfaces.display.body_template2 (module), 168
ask_sdk_model.interfaces.display.body_template3 (module), 169
ask_sdk_model.interfaces.display.body_template6 (module), 169
ask_sdk_model.interfaces.display.body_template7 (module), 170
ask_sdk_model.interfaces.display.display_interface (module), 170
ask_sdk_model.interfaces.display.display_state (module), 171
ask_sdk_model.interfaces.display.display_element_selected_request (module), 171
ask_sdk_model.interfaces.display.display_hint (module), 171
ask_sdk_model.interfaces.display.display_hint_directive (module), 172
ask_sdk_model.interfaces.display.display_image (module), 172
ask_sdk_model.interfaces.display.display_image_instance (module), 173
ask_sdk_model.interfaces.display.display_image_size (module), 173
ask_sdk_model.interfaces.display.display_list_item (module), 173
ask_sdk_model.interfaces.display.display_list_template1 (module), 174
ask_sdk_model.interfaces.display.display_list_template2 (module), 174
ask_sdk_model.interfaces.display.display_plain_text (module), 175
ask_sdk_model.interfaces.display.display_plain_text_hint (module), 175
ask_sdk_model.interfaces.display.display_render_template_directive (module), 175
ask_sdk_model.interfaces.display.display_rich_text (module), 175
ask_sdk_model.interfaces.display.display_request (module), 176
ask_sdk_model.interfaces.display.display_text_content (module), 176
ask_sdk_model.interfaces.display.text_field (module), 177
ask_sdk_model.interfaces.gadget_controller (module), 181
ask_sdk_model.interfaces.game_engine.input_handler_event_request (module), 182
ask_sdk_model.interfaces.game_engine.stop_input_handler_directive (module), 183
ask_sdk_model.interfaces.game_engine.start_input_handler_directive (module), 183
ask_sdk_model.interfaces.geolocation.access (module), 178
ask_sdk_model.interfaces.geolocation.altitude (module), 178
ask_sdk_model.interfaces.geolocation.coordinate (module), 178
ask_sdk_model.interfaces.geolocation.geolocation_interface (module), 179
ask_sdk_model.interfaces.geolocation.geolocation_state (module), 179
ask_sdk_model.interfaces.geolocation.heading (module), 180
ask_sdk_model.interfaces.geolocation.location_services (module), 180
ask_sdk_model.interfaces.geolocation.speed (module), 181
ask_sdk_model.interfaces.geolocation.status (module), 181
ask_sdk_model.interfaces.messaging.message_received_request (module), 184
ask_sdk_model.interfaces.monetization.v1.in_skill_product (module), 185
ask_sdk_model.interfaces.monetization.v1.purchase_result (module), 185
ask_sdk_model.interfaces.playbackcontroller.next_command_issued_request (module), 186
ask_sdk_model.interfaces.playbackcontroller.pause_command_issued_request (module), 186
ask_sdk_model.interfaces.playbackcontroller.play_command_issued_request (module), 187
ask_sdk_model.interfaces.playbackcontroller.previous_command_issued_request (module), 187
ask_sdk_model.interfaces.system.error (module), 188
ask_sdk_model.interfaces.system.error_cause (module), 188
ask_sdk_model.interfaces.system.error_type (module), 188
ask_sdk_model.interfaces.system.exception_encountered_request (module), 189
ask_sdk_model.interfaces.system.system_state (module), 189
ask_sdk_model.interfaces.videoapp.launch_directive (module), 190
ask_sdk_model.interfaces.videoapp.metadata (module), 190
ask_sdk_model.interfaces.videoapp.video_app_interface (module), 190
ask_sdk_model.interfaces.videoapp.video_item (module), 191
ask_sdk_model.interfaces.viewport (module), 194
ask_sdk_model.interfaces.viewport.experience (module), 191
ask_sdk_model.interfaces.viewport.keyboard (module), 192
ask_sdk_model.interfaces.viewport.shape (module), 192
ask_sdk_model.interfaces.viewport.touch (module), 192
ask_sdk_model.interfaces.viewport.viewport_state (module), 193
Index
ask_sdk_model.services.monetization.purchase_mode (module), 220
ask_sdk_model.services.proactive_events.create_proactive_event_request (module), 236
ask_sdk_model.services.proactive_events.error (module), 236
ask_sdk_model.services.proactive_events.event (module), 237
ask_sdk_model.services.proactive_events.proactive_events_service_client (module), 237
ask_sdk_model.services.proactive_events.relevant_audience (module), 237
ask_sdk_model.services.proactive_events.relevant_audience_type (module), 238
ask_sdk_model.services.proactive_events.skill_stage (module), 238
ask_sdk_model.services.reminder_management.alert_info (module), 221
ask_sdk_model.services.reminder_management.alert_info_spoken_info (module), 221
ask_sdk_model.services.reminder_management.error (module), 222
ask_sdk_model.services.reminder_management.event (module), 222
ask_sdk_model.services.reminder_management.get_reminder_response (module), 223
ask_sdk_model.services.reminder_management.get_reminders_response (module), 223
ask_sdk_model.services.reminder_management.push_notification (module), 224
ask_sdk_model.services.reminder_management.push_notification_status (module), 224
ask_sdk_model.services.reminder_management.recurrence (module), 224
ask_sdk_model.services.reminder_management.recurrence_day (module), 225
ask_sdk_model.services.reminder_management.recurrence_freq (module), 225
ask_sdk_model.services.reminder_management.reminder (module), 226
ask_sdk_model.services.reminder_management.reminder_created_event_request (module), 226
ask_sdk_model.services.reminder_management.reminder_deleted_event (module), 227
ask_sdk_model.services.reminder_management.reminder_deleted_event_request (module), 227
ask_sdk_model.services.reminder_management.reminder_management_service_client (module), 228
ask_sdk_model.services.reminder_management.reminder_request (module), 229
ask_sdk_model.services.reminder_management.reminder_response (module), 229
ask_sdk_model.services.reminder_management.reminder_started_event_request (module), 230
ask_sdk_model.services.reminder_management.reminder_status_changed_event_request (module), 230
ask_sdk_model.services.reminder_management.reminder_updated_event_request (module), 231
ask_sdk_model.services.reminder_management.spoken_text (module), 231
ask_sdk_model.services.reminder_management.status (module), 232
ask_sdk_model.services.reminder_management.trigger (module), 232
ask_sdk_model.services.reminder_management.trigger_type (module), 233
ask_sdk_model.services.serializer (module), 245
ask_sdk_model.services.service_client_factory (module), 246
ask_sdk_model.services.service_client_response (module), 247
ask_sdk_model.services.service_exception (module), 247
ask_sdk_model.services.skill_messaging.error (module), 239
ask_sdk_model.services.skill_messaging.send_skill_messaging_request (module), 239
ask_sdk_model.services.skill_messaging.skill_messaging_service_client (module), 240
ask_sdk_model.services.ups.distance_units (module), 233
ask_sdk_model.services.ups.error (module), 233
ask_sdk_model.services.ups.error_code (module), 234
ask_sdk_model.services.ups.phone_number (module), 234
ask_sdk_model.services.ups.temperature_unit (module), 235
ask_sdk_model.services.ups.ups_service_client (module), 234
ask_sdk_model.session (module), 264
ask_sdk_model.session_ended_error (module), 264
ask_sdk_model.session_ended_error_type (module), 264
ask_sdk_model.session_ended_reason (module), 264
ask_sdk_model.session_ended_request (module), 264
ask_sdk_model.session_service_client (module), 266
ask_sdk_model.slot (module), 266
ask_sdk_model.slot_confirmation_status (module), 266
ask_sdk_model.slu.entityresolution.resolution (module), 248
ask_sdk_model.slu.entityresolution.resolutions (module), 248
ask_sdk_model.slu.entityresolution.species (module), 248
ask_sdk_model.slu.entityresolution.species_entityresolution.resolution (module), 248
ask_sdk_model.slu.entityresolution.species_entityresolution.resolutions (module), 248
BillingAgreementAttributes (class in ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_attributes), 127
BillingAgreementDetails (class in ask_sdk_model.interfaces.amazonpay.model.response.billing_agreement_details), 141
BillingAgreementDetails (class in ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details), 128
BillingAgreementStatus (class in ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status), 129
BodyTemplate1 (class in ask_sdk_model.interfaces.display.body_template1), 168
BodyTemplate2 (class in ask_sdk_model.interfaces.display.body_template2), 168
BodyTemplate3 (class in ask_sdk_model.interfaces.display.body_template3), 169
BodyTemplate6 (class in ask_sdk_model.interfaces.display.body_template6), 169
BodyTemplate7 (class in ask_sdk_model.interfaces.display.body_template7), 170
CanFulfillIntent (class in ask_sdk_model.canfulfill.can_fulfill_intent), 99
CanFulfillIntentRequest (class in ask_sdk_model.canfulfill.can_fulfill_intent_request), 99
CanFulfillIntentValues (class in ask_sdk_model.canfulfill.can_fulfill_intent_values), 100
CanFulfillSlot (class in ask_sdk_model.canfulfill.can_fulfill_slot), 100
CanFulfillSlotValues (class in ask_sdk_model.canfulfill.can_fulfill_slot_values), 101
CanUnderstandSlotValues (class in ask_sdk_model.canfulfill.can_understand_slot_values), 101
ChargeAmazonPayResult (class in ask_sdk_model.interfaces.amazonpay.response.charge_amazon_pay_result), 150
ChargeAmazonPay (class in ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay), 145
ChargeAmazonPayRequest (class in ask_sdk_model.interfaces.amazonpay.request.charge_amazon_pay_request), 147
ClearBehavior (class in ask_sdk_model.interfaces.audioplayer.clear_behavior), 152
ClearQueueDirective (class in ask_sdk_model.interfaces.audioplayer.clear_queue_directive), 152
CanFulfillIntent (class in ask_sdk_model.interfaces.alexa.presentation.apl.command), 109
CanFulfillIntentRequest (class in ask_sdk_model.interfaces.alexa.presentation.apl.component_state), 115
CanFulfillIntentValues (class in ask_sdk_model.interfaces.alexa.presentation.apl.component_state), 115
CanUnderstandSlotValues (class in ask_sdk_model.interfaces.alexa.presentation.apl.component_state), 115
ChargeAmazonPayResult (class in ask_sdk_model.interfaces.alexa.presentation.apl.component_state), 150
ConnectionsStatus (class in ask_sdk_model.interfaces.connections.connections_status), 102

Context (class in ask_sdk_model.context), 254

ControlMediaCommand (class in ask_sdk_model.interfaces.alexa.presentation.apl.control_media_command), 115

Coordinate (class in ask_sdk_model.interfaces.geolocation.coordinate), 178

create() (ask_sdk.standard.StandardSkillBuilder method), 90

create() (ask_sdk_core.skill_builder.CustomSkillBuilder method), 78

create() (ask_sdk_core.skill_builder.SkillBuilder method), 75

create() (ask_sdk_runtime.skill_builder.AbstractSkillBuilder method), 69

create_list() (ask_sdk_model.services.list_management.list_management_service_client.ListManagementServiceClient method), 213

create_list_item() (ask_sdk_model.services.list_management.list_management_service_client.ListManagementServiceClient method), 213

create_proactive_event() (ask_sdk_model.services.proactive_events.proactive_events_service_client.ProactiveEventsServiceClient in ask_sdk_model.services.proactive_events.proactive_events_service_client), 237

create_reminder() (ask_sdk_model.services.reminder_management.reminder_management_service_client.ReminderManagementServiceClient in ask_sdk_model.services.amazonpay.model.response.destination), 142

CreateListItemRequest (class in ask_sdk_model.services.list_management.create_list_item_request), 208

CreateListRequest (class in ask_sdk_model.services.list_management.create_list_request), 208

CreateProactiveEventRequest (class in ask_sdk_model.services.proactive_events.create_proactive_event_request), 236

CurrentPlaybackState (class in ask_sdk_model.interfaces.audioplayer.current_playback_state), 153

CustomSkill (class in ask_sdk_core.skill), 75

CustomSkillBuilder (class in ask_sdk_core.skill_builder), 77

Density (class in ask_sdk_core.utils.viewport), 83

deserialize() (ask_sdk_core.serialize.DefaultSerializer method), 82

deserialize() (ask_sdk_model.services.serializer.Serializer method), 245

DeviationRecognizer (class in ask_sdk_model.services.game_engine.deviation_recognizer), 200

Device (class in ask_sdk_model.device), 254

device_id_partition_keygen() (in module ask_sdk_dynamodb.partition_keygen), 89

CreateDirectiveRequest (class in ask_sdk_model.services.directive.directive), 196

Directive (class in ask_sdk_model.directive), 255

DirectiveServiceClient (class in ask_sdk_model.services.directive.directive_service_client), 196

dispatch() (ask_sdk_runtime.dispatch.AbstractRequestDispatcher method), 59

dispatch() (ask_sdk_runtime.dispatch.GenericRequestDispatcher method), 63

dispatch() (django_ask_sdk.skill_adapter.SkillAdapter method), 97
dispatch_request() (flask_ask_sdk.skill_adapter.SkillAdapter method), 96
DispatchException, 69
DisplayInterface (class in ask_sdk_model.interfaces.display.display_interface), 170
DisplayState (class in ask_sdk_model.interfaces.display.display_state), 171
DistanceUnits (class in ask_sdk_model.services.ups.distance_units), 233
DynamoDbAdapter (class in ask_sdk_dynamodb.adapter), 88
ElementSelectedRequest (class in ask_sdk_model.interfaces.display.element_selected_request), 171
ElicitSlotDirective (class in ask_sdk_model.dialog.elicit_slot_directive), 102
enqueue() (ask_sdk_model.services.directive.directive_service_client.DirectiveServiceClient method), 196
EntitledState (class in ask_sdk_model.services.monetization.entitled_state), 217
EQUAL (ask_sdk_core.utils.viewport.Orientation attribute), 83
Error (class in ask_sdk_model.interfaces.audioplayer.error), 153
Error (class in ask_sdk_model.interfaces.system.error), 188
Error (class in ask_sdk_model.services.device_address.error), 195
Error (class in ask_sdk_model.services.directive.error), 197
Error (class in ask_sdk_model.services.list_management.error), 208
Error (class in ask_sdk_model.services.lwa.error), 241
Error (class in ask_sdk_model.services.monetization.error), 217
Error (class in ask_sdk_model.services.proactive_events.error), 236
Error (class in ask_sdk_model.services.reminder_management.error), 222
Error (class in ask_sdk_model.services.skill_messaging.error), 239
Error (class in ask_sdk_model.services.ups.error), 233
ErrorCause (class in ask_sdk_model.interfaces.system.error_cause), 188
ErrorCode (class in ask_sdk_model.services.ups.error_code), 234
ExceptionEncounteredRequest (class in ask_sdk_model.interfaces.system.exception_encountered_request), 189
execute() (ask_sdk_runtime.dispatch_components.request_components.AbstractHandlerAdapter method), 61
execute() (ask_sdk_runtime.dispatch_components.request_components.GenericHandlerAdapter method), 65
ExecuteCommandsDirective (class in ask_sdk_model.interfaces.alexa.presentation.apl.execute_commands_directive), 110
Experience (class in ask_sdk_model.interfaces.viewport.experience), 191
EXTENSION_NAME (in module flask_ask_sdk.skill_adapter), 96
ForbiddenError (class in ask_sdk_model.services.list_management.forbidden_error), 209

294
get_attributes()
get_api_access_token()
get_account_linking_access_token()
get_access_token_for_scope()
get_access_token_for_scope()
get_account_linking_access_token()
get_api_access_token()
get_attributes()
get_address()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_attributes()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
get_address()
Index
handle() (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler.method), 60

HandlerInput (class in ask_sdk_core.handler_input), 69

Header (class in ask_sdk_model.services.directive.header), 197

Heading (class in ask_sdk_model.interfaces.geolocation.heading), 197

HIGH (ask_sdk_core.utils.viewport.Density attribute), 83

HighlightMode (class in ask_sdk_model.interfaces.alexa.presentation.apl.highlight_mode), 111

Hint (class in ask_sdk_model.interfaces.display.hint), 171

HintDirective (class in ask_sdk_model.interfaces.display.hint_directive), 172

http_method_names (django_ask_sdk.skill_adapter.SkillAdapter.attribute), 98

http_method_not_allowed() (django_ask_sdk.skill_adapter.SkillAdapter.method), 98

HUB_LANDSCAPE_LARGE (ask_sdk_core.utils.viewport.ViewportProfile.attribute), 83

HUB_LANDSCAPE_MEDIUM (ask_sdk_core.utils.viewport.ViewportProfile.attribute), 83

HUB_LANDSCAPE_SMALL (ask_sdk_core.utils.viewport.ViewportProfile.attribute), 83

HUB_ROUND_SMALL (ask_sdk_core.utils.viewport.ViewportProfile.attribute), 83

IdleCommand (class in ask_sdk_model.interfaces.alexa.presentation.apl.idle_command), 116

Image (class in ask_sdk_model.interfaces.display.image), 172

Image (class in ask_sdk_model.ui.image), 251

ImageInstance (class in ask_sdk_model.interfaces.display.image_instance), 173

ImageSize (class in ask_sdk_model.interfaces.display.image_size), 173

init_app() (flask_ask_sdk.skill_adapter.SkillAdapter method), 96

InputEvent (class in ask_sdk_model.services.game_engine.input_event), 201

InputEventActionType (class in ask_sdk_model.services.game_engine.input_event_action_type), 202

InputEventHandler (class in ask_sdk_model.services.game_engine.input_handler_event), 202

InputEventHandlerRequest (class in ask_sdk_model.interfaces.game_engine.input_handler_event_request), 182

InSkillProduct (class in ask_sdk_model.interfaces.monetization.v1.in_skill_product), 185

InSkillProductsResponse (class in ask_sdk_model.services.monetization.in_skill_products_response), 219

Intent (class in ask_sdk_model.intent), 257

IntentConfirmationStatus (class in ask_sdk_model.intent_confirmation_status), 257

IntentRequest (class in ask_sdk_model.intent_request), 258

invoke() (ask_sdk_core.api_client.DefaultApiClient.method), 80

invoke() (ask_sdk_core.skill.CustomSkill.method), 75

invoke() (ask_sdk_model.services.api_client.ApiClient.method), 243

invoke() (ask_sdk_model.services.base_service_client.BaseServiceClient.method), 245

invoke() (ask_sdk_runtime.skill.AbstractSkill.method), 67

is_canfulfill_intent_name() (in module ask_sdk_core.utils.predicate), 82

is_intent_name() (in module ask_sdk_core.utils.predicate), 82

is_new_session() (in module ask_sdk_core.utils.request_util), 87

is_request_type() (in module ask_sdk_core.utils.predicate), 82

Keyboard (class in ask_sdk_model.interfaces.viewport.keyboard), 192

Image (class in ask_sdk_model.ui.image), 251

Image (class in ask_sdk_model.interfaces.display.image), 172

Image (class in ask_sdk_model.interfaces.display.image_instance), 173

ImageSize (class in ask_sdk_model.interfaces.display.image_size), 173

init_app() (flask_ask_sdk.skill_adapter.SkillAdapter method), 96

InputEvent (class in ask_sdk_model.services.game_engine.input_event), 201

InputEventActionType (class in ask_sdk_model.services.game_engine.input_event_action_type), 202
Alexa SKills Kit SDK for Python Documentation, Release 1.10.2

LARGE (ask_sdk_core.utils.viewport.Size attribute), 83
LaunchDirective (class in ask_sdk_model.interfaces.geolocation.location_services), 180
LaunchRequest (class in ask_sdk_model.launch_request), 258
LightAnimation (class in ask_sdk_model.services.gadget_controller.light_animation), 199
LinkAccountCard (class in ask_sdk_model.ui.link_account_card), 251
Links (class in ask_sdk_model.services.list_management.links), 209
ListBody (class in ask_sdk_model.services.list_management.list_body), 209
ListCreatedEventRequest (class in ask_sdk_model.services.list_management.list_created_event_request), 209
ListDeletedEventRequest (class in ask_sdk_model.services.list_management.list_deleted_event_request), 210
ListItem (class in ask_sdk_model.interfaces.display.list_item), 173
ListItemBody (class in ask_sdk_model.services.list_management.list_item_body), 211
ListItemsCreatedEventRequest (class in ask_sdk_model.services.list_management.list_items_created_event_request), 211
ListItemsDeletedEventRequest (class in ask_sdk_model.services.list_management.list_items_deleted_event_request), 212
ListItemsUpdatedEventRequest (class in ask_sdk_model.services.list_management.list_items_updated_event_request), 212
ListManagementServiceClient (class in ask_sdk_model.services.list_management.list_management_service_client), 213
ListState (class in ask_sdk_model.services.list_management.list_state), 215
ListTemplate1 (class in ask_sdk_model.interfaces.display.list_template1), 174
ListTemplate2 (class in ask_sdk_model.interfaces.display.list_template2), 174
ListUpdatedEventRequest (class in ask_sdk_model.services.list_management.list_updated_event_request), 215
LocationServices (class in ask_sdk_model.interfaces.geolocation.location_services), 215

MONETIZATION_SERVICE_CLIENT

OrderedEnum (class in django_ask_sdk.skill_adapter.SkillAdapter.method), 98
OutputSpeech (class in ask_sdk_model.ui.output_speech), 251

Index
ParallelCommand (class in ask_sdk_model.interfaces.alexa.presentation.apl.parallel_command), 116
Pattern (class in ask_sdk_model.services.game_engine.pattern), 203
PatternRecognizer (class in ask_sdk_model.services.game_engine.pattern_recognizer), 203
PatternRecognizerAnchorType (class in ask_sdk_model.services.game_engine.pattern_recognizer_anchor_type), 204
PauseCommandIssuedRequest (class in ask_sdk_model.interfaces.playbackcontroller.pause_command_issued_request), 186
PaymentAction (class in ask_sdk_model.interfaces.amazonpay.model.request.payment_action), 136
PaymentAction (class in ask_sdk_model.interfaces.amazonpay.model.v1.payment_action), 130
Permission (class in ask_sdk_model.events.skillevents.permission), 104
PermissionAcceptedRequest (class in ask_sdk_model.events.skillevents.permission_accepted_request), 104
PermissionBody (class in ask_sdk_model.events.skillevents.permission_body), 105
PermissionChangedRequest (class in ask_sdk_model.events.skillevents.permission_changed_request), 105
Permissions (class in ask_sdk_model.permissions), 259
PermissionStatus (class in ask_sdk_model.permission_status), 268
PersistenceException, 81
persistent_attributes (ask_sdk_core.attributes_manager.AttributesManager attribute), 79
PhoneNumber (class in ask_sdk_model.services.ups.phone_number), 234
PLAIN_TEXT_TYPE (in module ask_sdk_core.response_helper), 72
PlainText (class in ask_sdk_model.interfaces.display.plain_text), 175
PlainTextHint (class in ask_sdk_model.interfaces.display.plain_text_hint), 175
PlainTextOutputSpeech (class in ask_sdk_model.ui.plain_text_output_speech), 252
PlaybackFailedRequest (class in ask_sdk_model.interfaces.audioplayer.playback_failed_request), 154
PlaybackFinishedRequest (class in ask_sdk_model.interfaces.audioplayer.playback_finished_request), 155
PlaybackNearlyFinishedRequest (class in ask_sdk_model.interfaces.audioplayer.playback_nearly_finished_request), 156
PlaybackStartedRequest (class in ask_sdk_model.interfaces.audioplayer.playback_started_request), 156
PlaybackStoppedRequest (class in ask_sdk_model.interfaces.audioplayer.playback_stopped_request), 157
PlayDirective (class in ask_sdk_model.interfaces.audioplayer.play_behavior), 154
PlayerActivity (class in ask_sdk_model.interfaces.audioplayer.player_activity), 157
PlayMediaCommand (class in ask_sdk_model.interfaces.alexa.presentation.apl.play_media_command), 117
PORTRAIT (ask_sdk_core.utils.viewport.Orientation attribute), 83
PostImageRequest (class in ask_sdk_model.interfaces.connections.requests.print_image_request), 161
PrintImageRequest (class in ask_sdk_model.interfaces.connections.requests.print_pdf_request), 161
PrintPDFRequest
PrintWebPageRequest (class in ask_sdk_model.interfaces.connections.requests.print_web_page_request),
ProactiveEventsServiceClient (class in ask_sdk_model.services.proactive_events.proactive_events_service_client),
process() (ask_sdk_core.dispatch_components.request_components.AbstractRequestInterceptor method),
process() (ask_sdk_core.dispatch_components.request_components.AbstractResponseInterceptor method),
process() (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor method),
process() (ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor method),
ProviderAttributes (class in ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes),
ProviderAttributes (class in ask_sdk_model.interfaces.amazonpay.model.v1.provider_attributes),
ProviderCredit (class in ask_sdk_model.interfaces.amazonpay.model.request.provider_credit),
ProviderCredit (class in ask_sdk_model.interfaces.amazonpay.model.v1.provider_credit),
PurchasableState (class in ask_sdk_model.services.monetization.purchasable_state),
PurchaseMode (class in ask_sdk_model.services.monetization.purchase_mode),
PurchaseResult (class in ask_sdk_model.interfaces.amazonpay.model.v1.purchase_result),
PushNotification (class in ask_sdk_model.services.reminder_management.push_notification),
PushNotificationStatus (class in ask_sdk_model.services.reminder_management.push_notification_status),
Reminder (class in ask_sdk_model.services.reminder_management.reminder),
ReminderCreatedEventRequest (class in ask_sdk_model.services.reminder_management.reminder_created_event_request),
ReminderDeletedEvent (class in ask_sdk_model.services.reminder_management.reminder_deleted_event),
ReminderDeletedEventRequest (class in ask_sdk_model.services.reminder_management.reminder_deleted_event_request),
ReminderManagementServiceClient (class in ask_sdk_model.services.reminder_management.reminder_management_service_client),
ReminderRequest (class in ask_sdk_model.services.reminder_management.reminder_request),
ReminderResponse (class in ask_sdk_model.services.reminder_management.reminder_response),
ReminderStartedEventRequest (class in ask_sdk_model.services.reminder_management.reminder_started_event_request),
ReminderStatusChangedEventRequest (class in ask_sdk_model.services.reminder_management.reminder_status_changed_event_request),
ReminderUpdatedEventRequest (class in ask_sdk_model.services.reminder_management.reminder_updated_event_request),
Recognizer (class in ask_sdk_model.services.game_engine.recognizer),
RenderDocumentDirective (class in ask_sdk_model.interfaces.alexa.presentation.apl.render_document_directive),
ReleaseEnvironment (class in ask_sdk_model.interfaces.amazonpay.model.response.release_environment),
ReleaseEnvironment (class in ask_sdk_model.interfaces.amazonpay.model.v1.release_environment),
RelevantAudience (class in ask_sdk_model.services.proactive_events.relevant_audience),
RelevantAudienceType (class in ask_sdk_model.services.proactive_events.relevant_audience_type),
ProductType (class in ask_sdk_model.services.monetization.product_type),
ProviderAttribute (class in ask_sdk_model.interfaces.amazonpay.model.response.provider_attribute),
RenderTemplateDirective  

`ask_sdk_model.interfaces.display.render_template_directive`  

175

Reprompt  

`ask_sdk_model.ui.reprompt`  

252

Request  

`ask_sdk_model.request`  

259

request_attributes  

`ask_sdk_core.attributes_manager.AttributesManager.attribute`  

79

request_handler  

`ask_sdk_runtime.dispatch_components.request_components.GenericRequestHandlerChain.attribute`  

63

request_handler()  

`ask_sdk.standard.StandardSkillBuilder`  

91

request_handler()  

`ask_sdk_core.skill_builder.CustomSkillBuilder`  

79

request_handler()  

`ask_sdk_core.skill_builder.SkillBuilder`  

77

request_handler()  

`ask_sdk_runtime.dispatch_components.request_components.GenericRequestHandlerChain.method`  

61

request_handler()  

`ask_sdk_runtime.skill_builder.AbstractSkillBuilder.method`  

68

request_handler_chains  

`ask_sdk_runtime.dispatch_components.request_components.GenericRequestMapper.attribute`  

64

request_interceptors  

`ask_sdk_runtime.dispatch_components.request_components.GenericRequestHandlerChain.attribute`  

64

request_interceptors()  

`ask_sdk_runtime.dispatch_components.request_components.GenericRequestHandlerChain.method`  

68

RequestEnvelope  

`ask_sdk_model.interfaces.connections.entities.restaurant`  

160

RICH_TEXT_TYPE  

`ask_sdk_core.response_helper`  

72

RichText  

`ask_sdk_model.interfaces.display.rich_text`  

175

Runtime  

`ask_sdk_model.interfaces.alexa.presentation.apl.runtime`  

112

RuntimeConfiguration  

`ask_sdk_runtime.skill`  

66

RuntimeConfigurationBuilder  

`ask_sdk_runtime.skill`  

66

S

save_attributes()  

`ask_sdk_core.attributes_manager.AbstractPersistenceAdapter.method`  

80

save_persistent_attributes()  

`ask_sdk_core.attributes_manager.AttributesManager.method`  

79

ScheduleFoodEstablishmentReservationRequest  

`ask_sdk_model.interfaces.connections.requests.schedule_food_establishment_reservation_request`  

163

ScheduleTaxiReservationRequest  

`ask_sdk_model.interfaces.connections.requests.schedule_taxi_reservation_request`  

163

Scope  

`ask_sdk_model.scope`  

268

ScrollCommand  

`ask_sdk_model.interfaces.alexa.presentation.apl.scroll_command`  

118

ScrollToIndexCommand  

`ask_sdk_model.interfaces.alexa.presentation.apl.scroll_to_index_command`  

118

SellerBillingAgreementAttributes  

`ask_sdk_model.interfaces.amazonpay.model.request.seller_billing_agreement_attributes`  

137

SellerBillingAgreementAttributes  

`ask_sdk_model.interfaces.amazonpay.model.v1.seller_billing_agreement_attributes`  

132

SellerOrderAttributes  

`ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes`  

138

SellerOrderAttributes  

`ask_sdk_model.interfaces.amazonpay.model.v1.seller_order_attributes`  

132

send_skill_message()  

`ask_sdk_model.services.skill_messaging.skill_messaging_service_client.SkillMessagingServiceClient.method`  

240

Index
SendDirectiveRequest  (class in set_should_end_session()
    ask_sdk_model.services.directive.send_directive_request), (ask_sdk_core.response_helper.ResponseFactory
    method), 73

SendEventCommand  (class in SetLightDirective (class in
    ask_sdk_model.interfaces.alexa.presentation.apl.send_event_command)
    ask_sdk_model.interfaces.gadget_controller.set_light_directive), 119

SendRequestDirective  (class in SetLightParameters (class in
    ask_sdk_model.interfaces.connections.send_request_directive)
    ask_sdk_model.services.gadget_controller.set_light_parameters), 166

SendResponseDirective  (class in SetPageCommand (class in
    ask_sdk_model.interfaces.connections.send_response_directive)
    ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command), 167

SendSkillMessagingRequest  (class in SetStateCommand (class in
    ask_sdk_model.services.skill_messaging.send_skill_messaging_request)
    ask_sdk_model.interfaces.alexa.presentation.apl.set_state_command), 239

SequentialCommand  (class in setup() (django_ask_sdk.skill_adapter.SkillAdapter
    method), 120)

SerializationException, 69, 81
serialize() (ask_sdk_core.serialize.DefaultSerializer
    method), 81
serialize() (ask_sdk_model.services.serializer.Serializer
    method), 246

Serializer  (class in SetupAmazonPayResult (class in
    ask_sdk_model.interfaces.amazonpay.response.setup_amazon_pay), 148)

service_client_factory (ask_sdk_model.services.serializer), 245
setup() (django_ask_sdk.skill_adapter.SkillAdapter
    method), 98

SetupAmazonPay  (class in ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay), 146

SetupAmazonPayRequest (class in ask_sdk_model.interfaces.amazonpay.request.setup_amazon_pay), 148

ServiceClientFactory (class in ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command), 112

SetValueCommand  (class in ask_sdk_model.interfaces.alexa.presentation.apl.set_value_command), 123

Shape  (class in ask_sdk_model.interfaces.viewport.shape), 192

ShortAddress  (class in ask_sdk_model.services.device_address.short_address), 195

SIGNATURE_CERT_CHAIN_URL_HEADER (in module ask_sdk_webservice_support.verifier_constants), 92

SIGNATURE_CERT_CHAIN_URL_KEY (in module django_ask_sdk.skill_adapter), 98

SIGNATURE_HEADER (in module ask_sdk_webservice_support.verifier_constants), 92

SIGNATURE_KEY (in module django_ask_sdk.skill_adapter), 98

SimpleCard  (class in ask_sdk_model.ui.simple_card), 252

set_can_fulfill_intent() (ask_sdk_core.response_helper.ResponseFactory
    method), 73

set_card() (ask_sdk_core.response_helper.ResponseFactory
    method), 72
Alexa SKills Kit SDK for Python Documentation, Release 1.10.2

TimestampVerifier (class in ask_sdk_webservice_support.verifier), 93

to_dict() (ask_sdk_model.application.Application method), 254
to_dict() (ask_sdk_model.canfulfill.can_fulfill_intent.CanFulfillIntent method), 99
to_dict() (ask_sdk_model.canfulfill.can_fulfill_intent_request.CanFulfillIntentRequest method), 100
to_dict() (ask_sdk_model.canfulfill.can_fulfill_intent_values.CanFulfillIntentValues method), 100
to_dict() (ask_sdk_model.canfulfill.can_fulfill_slot.CanFulfillSlot method), 115
to_dict() (ask_sdk_model.canfulfill.can_fulfill_slot_values.CanFulfillSlotValues method), 101
to_dict() (ask_sdk_model.canfulfill.can_understand_slot_values.CanUnderstandSlotValues method), 101
to_dict() (ask_sdk_model.context.Context method), 254
to_dict() (ask_sdk_model.device.Device method), 255
to_dict() (ask_sdk_model.dialog.confirm_intent_directive.ConfirmIntentDirective method), 102
to_dict() (ask_sdk_model.dialog.confirm_slot_directive.ConfirmSlotDirective method), 102
to_dict() (ask_sdk_model.dialog.delegate_directive.DelegateDirective method), 117
to_dict() (ask_sdk_model.dialog.elicit_slot_directive.ElicitSlotDirective method), 111
to_dict() (ask_sdk_model.dialog.state.DialogState method), 255
to_dict() (ask_sdk_model.directive.Directive method), 257
to_dict() (ask_sdk_model.events.skillevents.account_linked_body.AccountLinkedBody method), 103
to_dict() (ask_sdk_model.events.skillevents.account_linked_request.AccountLinkedRequest method), 104
to_dict() (ask_sdk_model.events.skillevents.permission.Permission method), 104
to_dict() (ask_sdk_model.events.skillevents.permission_accepted_request.PermissionAcceptedRequest method), 105
to_dict() (ask_sdk_model.events.skillevents.permission_body.PermissionBody method), 105
to_dict() (ask_sdk_model.events.skillevents.permission_changed_request.PermissionChangedRequest method), 105
to_dict() (ask_sdk_model.events.skillevents.permission_disabled_request.PermissionDisabledRequest method), 106
to_dict() (ask_sdk_model.events.skillevents.permission_enabled_request.PermissionEnabledRequest method), 107
to_dict() (ask_sdk_model.intent.Intent method), 257
to_dict() (ask_sdk_model.intent_confirmation_status.IntentConfirmationStatus method), 257
to_dict() (ask_sdk_model.intent_request.IntentRequest to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface.AlexaPresentationAplInterface method), 93
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.audio_track.AudioTrack method), 108
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.command.Command method), 116
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.command_type.CommandType method), 117
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.condition.Condition method), 118
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.data_adapter.DataAdapter method), 119
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.debug.Debug method), 119
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.display.Display method), 120
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.display_action_artifact.DisplayActionArtifact method), 121
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.display_focus_point(DISPLAY_FOCUS_POINT method), 122
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.display_text_field.DisplayTextField method), 122
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.execute_commands_directive.ExecuteCommandsDirective method), 111
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.runtime.Runtime method), 112
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.scroll_to_index_command.ScrollToIndexCommand method), 113
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.scroll_command.ScrollCommand method), 113
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.send_event_command.SendEventCommand method), 119
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.set_state_command.SetStateCommand method), 120
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.set_value_command.SetValueCommand method), 120
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.speak_list_command.SpeakListCommand method), 121
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.user_event.UserEvent method), 122
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.video_track.VideoTrack method), 122
to_dict() (ask_sdk_model.interfaces.alexa.presentation.apl.video_source.VideoSource method), 122
Alexa Skills Kit SDK for Python Documentation, Release 1.10.2
to_dict() (ask_sdk_model.services.reminder_management.reminder_request.ReminderRequest method), 229

to_dict() (ask_sdk_model.slu.entityresolution.value_wrapper.ValueWrapper method), 230

to_dict() (ask_sdk_model.slu.entityresolution.value.Value method), 231

to_dict() (ask_sdk_model.slu.entityresolution.resolution.Resolution method), 232

to_dict() (ask_sdk_model.slu.entityresolution.resolutions.Resolutions method), 233

to_dict() (ask_sdk_model.device.Device method), 234

to_dict() (ask_sdk_model.slu.entityresolution.status.Status method), 235

to_dict() (ask_sdk_model.device.Device method), 236

to_dict() (ask_sdk_model.device.Device method), 237

to_dict() (ask_sdk_model.device.Device method), 238

to_dict() (ask_sdk_model.device.Device method), 239

to_dict() (ask_sdk_model.device.Device method), 240

to_dict() (ask_sdk_model.device.Device method), 241

to_dict() (ask_sdk_model.device.Device method), 242

to_dict() (ask_sdk_model.device.Device method), 243

to_dict() (ask_sdk_model.device.Device method), 244

to_dict() (ask_sdk_model.device.Device method), 245

to_dict() (ask_sdk_model.device.Device method), 246

to_dict() (ask_sdk_model.device.Device method), 247

to_dict() (ask_sdk_model.device.Device method), 248

to_dict() (ask_sdk_model.device.Device method), 249

to_dict() (ask_sdk_model.device.Device method), 250

to_dict() (ask_sdk_model.device.Device method), 251

to_dict() (ask_sdk_model.device.Device method), 252

to_dict() (ask_sdk_model.device.Device method), 253

to_dict() (ask_sdk_model.device.Device method), 254

to_dict() (ask_sdk_model.device.Device method), 255

to_dict() (ask_sdk_model.device.Device method), 256

to_dict() (ask_sdk_model.device.Device method), 257

to_dict() (ask_sdk_model.device.Device method), 258

to_dict() (ask_sdk_model.device.Device method), 259

to_dict() (ask_sdk_model.device.Device method), 260

to_dict() (ask_sdk_model.device.Device method), 261

to_dict() (ask_sdk_model.device.Device method), 262

to_dict() (ask_sdk_model.device.Device method), 263

to_dict() (ask_sdk_model.device.Device method), 264

to_dict() (ask_sdk_model.device.Device method), 265

to_dict() (ask_sdk_model.device.Device method), 266

to_dict() (ask_sdk_model.device.Device method), 267

to_dict() (ask_sdk_model.device.Device method), 268

to_dict() (ask_sdk_model.device.Device method), 269

to_dict() (ask_sdk_model.device.Device method), 270

to_dict() (ask_sdk_model.device.Device method), 271

to_dict() (ask_sdk_model.device.Device method), 272

to_dict() (ask_sdk_model.device.Device method), 273

to_dict() (ask_sdk_model.device.Device method), 274

to_dict() (ask_sdk_model.device.Device method), 275

to_dict() (ask_sdk_model.device.Device method), 276

to_dict() (ask_sdk_model.device.Device method), 277

to_dict() (ask_sdk_model.device.Device method), 278

to_dict() (ask_sdk_model.device.Device method), 279

to_dict() (ask_sdk_model.device.Device method), 280

to_dict() (ask_sdk_model.device.Device method), 281

to_dict() (ask_sdk_model.device.Device method), 282

to_dict() (ask_sdk_model.device.Device method), 283

to_dict() (ask_sdk_model.device.Device method), 284

to_dict() (ask_sdk_model.device.Device method), 285

to_dict() (ask_sdk_model.device.Device method), 286

to_dict() (ask_sdk_model.device.Device method), 287

to_dict() (ask_sdk_model.device.Device method), 288

to_dict() (ask_sdk_model.device.Device method), 289

to_dict() (ask_sdk_model.device.Device method), 290

to_dict() (ask_sdk_model.device.Device method), 291

to_dict() (ask_sdk_model.device.Device method), 292

to_dict() (ask_sdk_model.device.Device method), 293

to_dict() (ask_sdk_model.device.Device method), 294

to_dict() (ask_sdk_model.device.Device method), 295

to_dict() (ask_sdk_model.device.Device method), 296

to_dict() (ask_sdk_model.device.Device method), 297

to_dict() (ask_sdk_model.device.Device method), 298

to_dict() (ask_sdk_model.device.Device method), 299

to_dict() (ask_sdk_model.device.Device method), 300

to_dict() (ask_sdk_model.device.Device method), 301

to_dict() (ask_sdk_model.device.Device method), 302

to_dict() (ask_sdk_model.device.Device method), 303

to_dict() (ask_sdk_model.device.Device method), 304

to_dict() (ask_sdk_model.device.Device method), 305

to_dict() (ask_sdk_model.device.Device method), 306

to_dict() (ask_sdk_model.device.Device method), 307

to_dict() (ask_sdk_model.device.Device method), 308

to_dict() (ask_sdk_model.device.Device method), 309
to_str() (ask_sdk_model.services.list_management.list_items_versioned.ListItemVersionedRequest.get_reminders method), 213

2o_str() (ask_sdk_model.services.list_management.list_items_versioned.ListItemVersionedRequest.get_reminders method), 213

to_str() (ask_sdk_model.services.list_management.list_status.ListStatusRequest.remember_management.push_notification method), 215

to_str() (ask_sdk_model.services.list_management.list_status.ListStatusRequest.remember_management.push_notification method), 215

to_str() (ask_sdk_model.services.list_management.status.StatusRequest.remember_management.recurrence_request method), 216

to_str() (ask_sdk_model.services.list_management.update_list_request.UpdateListRequest.remember_management.recurrence_data method), 216

to_str() (ask_sdk_model.services.list_management.update_list_request.UpdateListRequest.remember_management.recurrence_data method), 217

to_str() (ask_sdk_model.services.list_management.update_list_request.UpdateListRequest.remember_management.reminder_request method), 225

to_str() (ask_sdk_model.services.lwa.access_token_request.AccessTokenRequest.remember_management.reminder_request method), 226

to_str() (ask_sdk_model.services.lwa.access_token_request.AccessTokenRequest.remember_management.reminder_request method), 226

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227

to_str() (ask_sdk_model.services.lwa.authorization_error.AuthorizationError.remember_management.reminder_request method), 227
verify_request_and_dispatch()
(ask_sdk_webservice_support.webservice_handler.WebserviceSkillHandler
   method), 94
verify_signature(django_ask_sdk.skill_adapter.SkillAdapter
   attribute), 97
verify_timestamp(django_ask_sdk.skill_adapter.SkillAdapter
   attribute), 97

VideoAppInterface (class in
   ask_sdk_model.interfaces.videoapp.video_app_interface), 190
VideoItem (class in
   ask_sdk_model.interfaces.videoapp.video_item), 191
VideoSource (class in
   ask_sdk_model.interfaces.alexa.presentation.apl.video_source), 123
ViewportProfile (class in
   ask_sdk_core.utils.viewport), 83
ViewportState (class in
   ask_sdk_model.interfaces.viewport.viewport_state), 193

WebserviceSkillHandler (class in
   ask_sdk_webservice_support.webservice_handler), 94

with_traceback() (ask_sdk_webservice_support.verifier.VerificationException
   method), 92

X

XHIGH (ask_sdk_core.utils.viewport.Density attribute), 83
XLARGE (ask_sdk_core.utils.viewport.Size attribute), 83
XLOW (ask_sdk_core.utils.viewport.Density attribute), 83
XSMSALL (ask_sdk_core.utils.viewport.Size attribute), 83
XXHIGH (ask_sdk_core.utils.viewport.Density attribute), 83