1 Getting Started
   1.1 Terminology ................................................................. 3

2 Rest API .............................. 5

3 Developer ............................. 7
   3.1 How does it work ....................................................... 7

4 Indices and tables .................. 9
Slack is a key distribution and validation server. It’s main purpose is to verify is there’s a support agreement with a customer for a given server.

Slack does not pretend to be unbreakable and it is not our main objective to be.

Contents:
CHAPTER 1

Getting Started

This section describes all you need to know to start working with Slack. It will describe some of the common flows that we think will occur in production situations as well as some terminology to be able to understand examples and descriptions given throughout this documentation.

1.1 Terminology

Some terms used in this documentation may have different interpretation depending on the context. In the slack documentation, API and source code, we will use the following definitions unless otherwise stated.

User

User is someone who as access to the service offered by slack. It can be a customer or an administrator

Key

The key is the relation between a server under a service agreement and a contract. A configuration with two servers, a high availability setup for example, will have one key for each server.

Server

A server is a single unique computer. The MAC address is used as part of the key generation.

Customer

A customer is a person or an enterprise who has bought one or many contracts.

Contract

The contract is the description of the service that is available to a customer.

Token

A token is a character string that is given to the customer to allow the creation of the key.

Seller

The seller is the person or enterprise that will give the service on the server covered by the service agreement.
Slack is based on a back-end, front-end architecture. This section documents the REST interface of the back-end. All operations possible in Slack are available as HTTP calls. With an integration with information system should not require change to Slack.
This section is meant for developers who need to contribute to Slack. It presents the internal python API that should be used by the REST interface. It also include documentation about Slack’s architecture and coding standards.

3.1 How does it work

3.1.1 Key

A key is generated from the HMAC of the MAC address, expiration date using a sha256. Here is an example for a key expiring on 2016-02-25 and a MAC address 56:84:7a:fe:97:99 and a key ‘test’

```python
'648c0965c5cd17e3f94a61c50d50eeb6d6fcf41b27f6cc02d991b9a8cd41060f'
```

3.1.2 How is a key created

When a customer agrees on a service agreement with the seller, an authorization token is given to the customer. This authorization token is stored in the slack database in order to be checked later.

When the customer is ready to activate his key, an HTTP request to the slack key service will generate a new key and return it to the customer with other information including the expiration date of the key.

Essential parameters to that request are the MAC and the token.

The generated key will be kept by the client and stored in the slack database.

The authorization token should be deleted or marked as used at this time.

The authorization token could be used to carry information since it was generated by a human being or a payment system. Some information I can think of include the client id, contract id, key lifetime, etc.

3.1.3 How is a token created

A token is issued by the seller when an agreement is reached with the customer on the terms and condition of the service. The conditions that must be met before issuing a token are not covered by slack.

Here’s an example of the flow that could occur when selling service to a customer.
1. A is a new customer
2. A calls the seller to get service on his server
3. The seller creates a new customer in slack
4. The seller determines which contract is best suited for the needs of A
5. The seller sends an invoice or receives a payment (this is a business rule, not a part of slack)
6. The seller generates a new token that is emailed, fax or told to A
7. A can activate the service from his server using the HTTP API
Indices and tables

- genindex
- modindex
- search