djangorestframework-queryfields Documentation

Release latest

Feb 19, 2018

Contents

Allows clients to control which fields will be sent in the API response. The idea is based on a recipe described in the official documentation, Dynamically modifying fields, adapted to allow API users to specify fields via GET request query parameters.

Doing the filtering server-side instead of client-side can have advantages:

- Fewer bytes down the wire = snappier ajax for your webapps
- Less backend server load if expensive-to-serialize fields go unneeded
- For thin clients; a better choice if client-side is a dumb/low-power device

CHAPTER 1

Contents

1.1 Installation

```
pip install djangorestframework-queryfields
```

1.2 Quickstart

Specify your base model serializer like this:

```
from rest_framework.serializers import ModelSerializer
from drf_queryfields import QueryFieldsMixin

class MyModelSerializer(QueryFieldsMixin, ModelSerializer):
    pass
```

Yeah, that's pretty much it.

1.3 Usage

```
GET http://127.0.0.1:8000/things/

HTTP/1.1 200 OK
...
[
  {
    "id": 1,
    "key1": "val1",
    "key2": "val2",
    "key3": "val3",
```

```
  },
  {
    "id": 2,
    "key1": "valA",
    "key2": "valB",
    "key3": "valC",
  }
]


GET http://127.0.0.1:8000/things/?fields=id,key2

HTTP/1.1 200 OK
...
[
  {
    "id": 1,
    "key2": "val2",
  },
  {
    "id": 2,
    "key2": "valB",
  }
]


GET http://127.0.0.1:8000/things/?fields!=key2

HTTP/1.1 200 OK
...
[
  {
    "id": 1,
    "key1": "val1",
    "key3": "val3",
  },
  {
    "id": 2,
    "key1": "valA",
    "key3": "valC",
  }
]
```

1.4 FAQ

Q: Can I use this with vanilla serializers as well as `ModelSerializer`?

A: Sure. You'll need include the request in the context, to provide access on the querystring:

```
MySerializer(obj, context={'request': request})
```

Q: The name `fields` conflicts with some other functionality in my API (e.g. django-filter). Can I change it to something else?

A: Yep. Override a couple of attributes on the class, and then Python's MRO will take care of the rest. For example:

```
class MyModelSerializer(QueryFieldsMixin, ModelSerializer):

    include_arg_name = 'include'
    exclude_arg_name = 'exclude'
    delimiter = '|'
```

Now request like `GET /things/?exclude=key2|key3` instead of the default `GET /things/?fields!=key2,key3`.

Q: This thing broke, you suck... / Hey, wouldn't it be cool if...

A: Well, that's not really a question, pal. For feature requests or bug reports, please create an issue here.