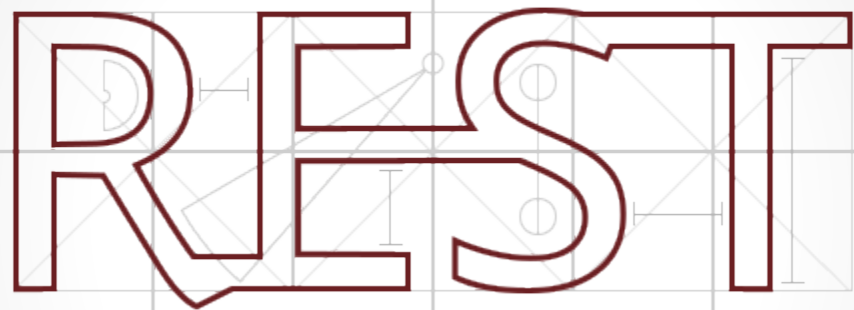


django

The word "REST" is rendered in a large, dark red, outlined serif font. The letters are superimposed on a light gray technical drawing grid. This grid includes a primary square frame, a smaller inner square, and a series of construction lines such as arcs and perpendiculars, which are typical of engineering or architectural drafting. The overall effect is that of a precise, technical illustration of the word.

framework

# who am I?

- developer@keycrunch & founder@offerqa
- built REST api in Django since 2013

follow me on twitter @linuxcity

# What is REST?

[http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer)

Web API that uses GET, POST, PUT, PATCH, DELETE operations over HTTP.

# What do you need from a framework

- serialization and deserialization
- pagination
- data validation
- authentication
- custom queryset
- proper HTTP response handling (errors, format and such)
- caching
- throttling
- permissions

# Good to have

- Really good test coverage of their code
- Decent performance
- Documentation
- An active community to advance and support the framework

And Django REST Framework has them all!!!

# Why Django REST framework

- Class-based views
- it's very django-ish

*...As it's based off Django 1.3 style Class Based Views (CBVs), it has a very familiar pattern. Actually, because of the quality of the documentation, I really prefer using django-rest-framework CBVs more than using Django's actual CBVs...*

*-from <http://pydanny.com/choosing-an-api-framework-for-django.html>*

- neat design

...

**And Most of All**

I enjoyed working with it



# Requirement

Django >1.4.11

so if you are still on Django 1.3, it's time to update. And you should have done that three years ago!

# a minimal REST API

## models.py

```
class Post(models.Model):  
    post = models.TextField()
```

```
class PostSerializer(serializers.HyperlinkedModelSerializer):  
    class Meta:  
        model = Post
```

# a minimal REST API

## views.py

```
class PostViewSet(viewsets.ModelViewSet):  
    queryset = Post.objects.all()  
    serializer_class = PostSerializer
```

# a minimal REST API

## urls.py

```
router = routers.SimpleRouter()
router.register(r'posts', views.PostViewSet)
```

```
urlpatterns = patterns("",
    url(r'^$', include(router.urls)),
    ....
```

# a minimal REST API

This project can be found at:

<https://github.com/zheli/django-rest-kog>

tag: v1

# Key elements

serializer

views

# serializer

serializing:

querysets, model instances->python native datatypes

deserializing:

data string->model instances

# **serializer**

automatic serializers from model:

ModelSerializer, HyperlinkedModelSerializer



# views

automatic views from model:

ModelViewSet: CRUD + list

ReadOnlyModelViewSet: retrieve + list

# test

`APIRequestFactory()` from `RequestFactory`

`APIClient()` from `Client`

Test POST request:

```
class PostTests(APITestCase): ←- self.client == APIClient()
    def test_create_post(self):
        response = self.client.post('/posts/', {'post': 'this is a test post'}, format='json')
        self.assertEqual(response.status_code, status.HTTP_201_CREATED)
        assert response.data is not None
```

# documentation

Django REST Swagger



## /countries

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

GET	/api/countries/	Gets a list of countries
-----	-----------------	--------------------------

POST	/api/countries/	Gets a list of countries
------	-----------------	--------------------------

GET	/api/countries/{pk}/	Detailed view of the country
-----	----------------------	------------------------------

PUT	/api/countries/{pk}/	Detailed view of the country
-----	----------------------	------------------------------

DELETE	/api/countries/{pk}/	Detailed view of the country
--------	----------------------	------------------------------

PATCH	/api/countries/{pk}/	Detailed view of the country
-------	----------------------	------------------------------

## /manufacturers

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

## /cigars

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

## /custom

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

# documentation

REST Framework Docs

Django Rest Framework L x

localhost:8000

jQuery plugin: T...openeGauge3812 Ce...TDD Kata 1 - Str...egaug94.e.bn...AS XML to Python ...Deploying Djan...Index of /xmlGood eGauge R...FacebookValidation with ...

Django Rest Framework Docs

/api/cigars

List Of Cigars

Allowed Methods: [ GET, POST, OPTIONS ]

Model: Cigar

Description:  
Lists and creates cigars from the database.

Fields:

Name	Type	Read-only	Default	Min	Max
manufacturer	Related Field	x			
manufacturer_id	Writable Field				
url	U R L Field	x			200
id	Field				
name	Char Field				25
colour	Char Field				30
gauge	Integer Field				
length	Integer Field				
price	Model Field				
notes	Char Field				

/api/cigars/{pk\_id}

Cigar Details

/api/countries

List Of Countries

/api/countries/{pk\_id}

Countries Details

/api/custom

A Custom View

# Questions & Feedbacks

Interesting?

What's next?