krankshaft Documentation

Release 0.3.13

Dan LaMotte

December 18, 2014

Contents

1 Guide 3
1.1 Quickstart 3

2 API 7
2.1 krankshaft.api 7
2.2 krankshaft.auth 7
2.3 krankshaft.authn 7
2.4 krankshaft.authz 7
2.5 krankshaft.models 7
2.6 krankshaft.query 7
2.7 krankshaft.resource 7
2.8 krankshaft.serializer 7
2.9 krankshaft.throttle 7
2.10 krankshaft.util 7
2.11 krankshaft.valid 7

3 Resources 9

A Web API Framework.

Currently only supports Django, but designed to work for other frameworks with some modification. At some point, other framework support will be built in directly.

krankshaft was designed to make the frustrating and unnecessarily complicated parts of Web APIs simple and beautiful by default. It's built in layers that allow the programmer to easily opt-in/out of. From "Expose this model via a web api and handle all the details" to "hands off my API, I'll opt-into the basics as I need them".

krankshaft is meant to be a framework to build Web APIs and grow with your application.

Note: These docs are still very young and need a lot of work. I'll be giving it some much needed attention in the future.

CHAPTER 1

Guide

1.1 Quickstart

krankshaft isn't limited to the given patterns, but it's helpful to see how it was envisioned to be used so you can make some of your own.

1.1.1 Simple View

In app/api.py:

```
from django.conf import settings
from krankshaft import API

api = API('v1', debug=settings.DEBUG)
```

In app/views.py:

```
from app.api import api

@api
def view(request):
    return api.serialize(request, 200, {
        'key': 'value'
    })
```

In app/urls.py:

```
from django.conf.urls import patterns, include, url

urlpatterns = patterns('app.views',
    url('^view/$', 'view'),
)
```

Doesn't seem like we did a whole lot right?

Let's talk to this API:

```
% curl http://localhost:8000/view/
{"key": "value"}
```

How about protecting the API to only authenticated users? Change app/api.py:

```
from django.conf import settings
from krankshaft import API as APIBase, authn, authz
from krankshaft.auth import Auth as AuthBase

class Auth(AuthBase):
    authn = authn.AuthnDjango()
    authz = authz.AuthzDjango(require_authned=True)

class API(APIBase):
    Auth = Auth

api = API('v1', debug=settings.DEBUG)
```

So now we need to authenticate to our api:

```
% curl -u user:password http://localhost:8000/view/
{"key": "value"}
```

1.1.2 Resources

Continuing from our above example, we can hook up a resource (which is simply an class/object versus a simple function):

Append to `app/api.py`:

```
# optional arguments to pass when registering your endpoint
@api(url='^model/(?P<id>\d+)/$')
class ModelResource(object):
    def get(self, request, id):
        ...

    def put(self, request, id):
        ...

    def delete(self, request, id):
        ...
```

Append to `app/urls.py` (since we registered the endpoint using url, the api takes care of pulling in all of those so we can register everything in one go):

```
urlpatterns += patterns('',
    url('^api/', include(api.urls)),
)
```

This enables clients to make GET/PUT/DELETE requests to the endpoint:

```
/api/v1/model/<id>/
```

If a `POST` is made, the client will receive a `405` response with the `Allow` header set to `GET, PUT, DELETE`.

1.1.3 Model Resource

The model resource is simply a built in resource that has special handling for Django models. All that you need to do is subclass the resource, hook up the model and register it:

Append to `app/api.py`:

```
from krankshaft.resource import DjangoModelResource
from app.models import Model

@api
class ModelResource(DjangoModelResource):
    model = Model
```

Again, this is registered in `app/urls.py` automatically because this resource defines a `urls` property (vs using the `url=...` option when decorating).

This resource implementation should be ideal for _most_ situations, but you're free to reimplement parts or all of it. It's meant only as a pattern you can follow and is not required by the framework at all.

CHAPTER 2

API

2.1 krankshaft.api

2.2 krankshaft.auth

2.3 krankshaft.authn

2.4 krankshaft.authz

2.5 krankshaft.models

2.6 krankshaft.query

2.7 krankshaft.resource

2.8 krankshaft.serializer

2.9 krankshaft.throttle

2.10 krankshaft.util

2.11 krankshaft.valid

CHAPTER 3

Resources

- Code
- Docs
- Issues