e

REST and Django Framework

Quick overview

Assumptions

- You know what is HTTP
- You know something about Python (or any HLL)
- You know a little bit about DB

Overview

➔ **What is REST?**

Enjoy and have a REST

➔ **REST API**

Why it is used and how to build one

➔ **Django**

… (REST) Framework

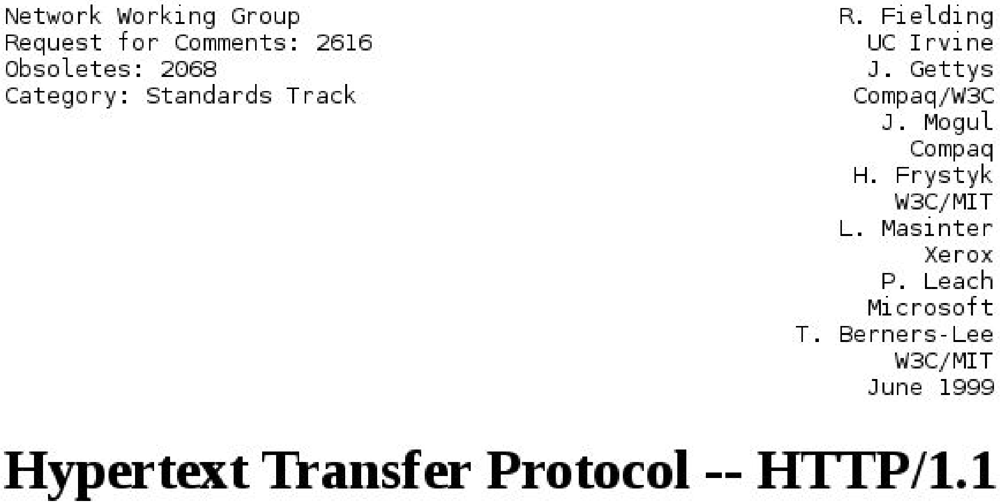
What is REST?

What is REST?

- It’s not a framework
- It’s not a standard
- It’s not a protocol
- It’s not a specification
- It’s about architecture of a web service

Described by Roy T. Fielding

Roy T. Who?



```
Network Working Group                                        R. Fielding
Request for Comments: 2616                                     UC Irvine
Obsoletes: 2068                                                J. Gettys
Category: Standards Track                                     Compaq/W3C
                                                                J. Mogul
                                                                  Compaq
                                                              H. Frystyk
                                                                 W3C/MIT
                                                             L. Masinter
                                                                   Xerox
                                                                P. Leach
                                                               Microsoft
                                                         T. Berners-Lee
                                                                 W3C/MIT
                                                               June 1999
```

Hypertext Transfer Protocol -- HTTP/1.1

RFC 2616

Why should I care?

- Applications are getting bigger
- There are more functions available
- More users
- Higher standards of availability

What is the biggest application in the Internet?

What kind of architecture?

Constraints

Client-Server

Stateless

Cacheable

Uniform Interface

Independent

Security

Resources and URI

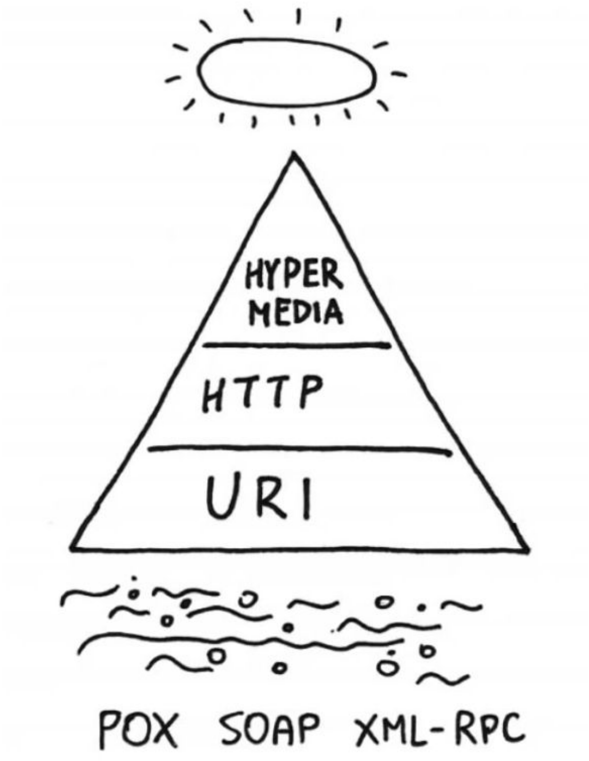
Two types of links

http://www.alledrogo.pl/user/john/auctions - collection address

http://www.alledrogo.pl/user/john/auctions/id/5 - resource address

http://www.alledrogo.pl/user/john/auctions?lowest=5&max=15

RESTFull, or how much REST in REST



Give me right Method

GET

POST

PUT

DELETE

HEAD

OPTIONS

PATCH

Give me right Method

GET

POST

PUT

DELETE

HEAD

OPTIONS

PATCH

Almost like CRUD but not the same

Idempotent

Insanity: doing the same thing over and over again and expecting different results.

Albert Einstein

What they have in common?

Status and Code

USE IT

Status and Code

200 - OK

201 - CREATED

202 - ACCEPTED

204 - NO CONTENT

Status and Code

301 - MOVED PERMANENTLY

303 - SEE OTHER

304 - NOT MODIFIED

307 - TEMPORARY REDIRECT

Status and Code

400 - BAD REQUEST

401 - UNAUTHORIZED

403 - FORBIDDEN

405 - METHOD NOT ALLOWED

409 - CONFLICT

410 - GONE

Status and Code

500 - INTERNAL SERVER ERROR

503 - SERVICE UNAVAILABLE

Do not ignore HTTP Status

- You can also add something extra
    - {“msg”: “Item already exist in the database”, }
- Also for developer
    - {“dev”: “Conflicting item with id=13 exist in the database”}

HATEOAS

Hypermedia as the engine of application state

- Client state between requests is preserved using resources

API Key

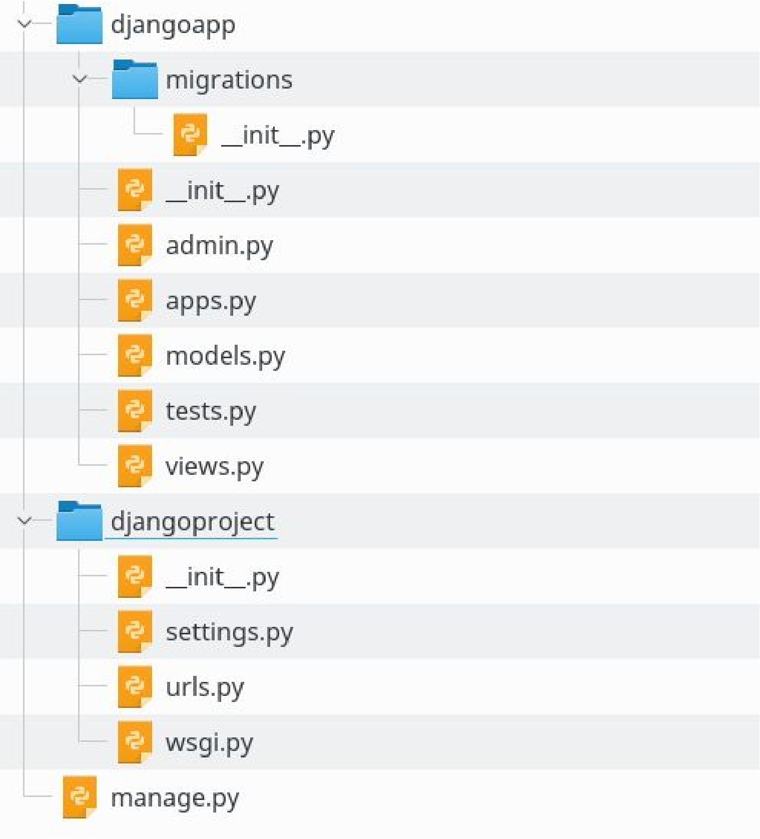
- Safer
- More convenient

About framework

Django Features

- ORM
- Forms
- Templates
- Views
- Admin Portal

Directory structure



```
djangoapp
    migrations
        __init__.py
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
djangoproject
    __init__.py
    settings.py
    urls.py
    wsgi.py
manage.py
```

Why fragmenting?

Because of Models

- Recipes
- Ingredients
- Meals
- Exercises
- Workout plans
- Diets

What is model?

- Typically related to database tables

Django will automatically take care of

- PK
- FK
- uniqueness
- constraints

Queries

Creating objects:

```
Users.object.create(name=’John’)
```

Filter or select relations:

```
User.objects.filter(groups__group__name=’admin’, age__lte=25)
```

DB Inheritance

- Multi-table - do not recomended (create base table, additional queries)

- Abstract - base table does not exist in the database

Foreign Key

- Many to one
- One to many
- One to one (direct access) User.Profile.Sentence
- Many to many mapped through another table (Many to one + one to many) or automatically using through table **BONUS QUESTION, when to use custom through table**

Django debug

django-debug-toolbar

- SQL
- Templates
- Views
- Signals

Forms

- Standard forms - are not related to any model (contact, login, registration)
  - isValid(), custom validators
- Model forms - created automatically based on model class

- Validation - standard fields have built in validators
  - Custom validators (min, max, regex, custom), works both on forms and models
  - clean_{fieldname} method
  - Form.clean()
  - Model.clean() - run only from model form clean method

Formsets

A way to have multiple instances of the same for in the same time

- Registration of many people to an event

Templates

- Context (dictionary)
- Not necessary html (pdf, json)
- {% Functions %} and {{ Values | Filters }}
    - Custom filters
- Tags
    - Simple tags
    - Inclusion tags

Views

Road to view and back:

- Request
- WSGI
- Middleware
- URL
  - RegEx
  - Arguments to kwargs
- View (forms, models, validation) return Response
- Middleware (exceptions)
- WSGI

Views

Views are just functions:

```
@login_required

def get_user(request):

...

    return render(request, “user_form.html”, {“form”: form})


url(r”^user/get/([0-9]+), views.get_user)
```

Class based views

- Hide a lot of things
- Saves a lot of time
- … and space

Admin

- Automatic admin interface
- Filter
- Search
- Customize by enums
- Decide what to display and how to display it
- Custom actions
- Templates (because client does not need to know it is Django)

User handling

- Registration
- Authentication
    - Passwords PBKDF2
    - XSS, CSRF, Clickjacking
- Permissions

All of that is highly customisable

Summary

- High community
- Lots of packages
- REST Framework
- Highly customizable
- Great by default
- Debug Toolbar