

Construindo APIs com Django-Tastypie

Roberto Fagá Jr - rfaga@usp.br

Sumário

- API RESTFul
- Tastypie
- Integrando a um projeto Django
- Autenticação
- Testando a API
- Comunicação JSON
- Integração Backbone.js

API RESTFul

- O que é RESTFul?
 - Transferência de Estado Representativo
 - Princípios REST
 - Objetivo de escalabilidade
 - Generalizar interfaces
 - Oferecimento de serviços independentes
 - Protocolo HTTP
 - GET, POST, PUT, PATCH, DELETE

API RESTFul

- Conceitos
 - Cliente-Servidor
 - Stateless
 - Cacheable
 - Camada no sistema
 - Interface uniforme

VIDA DE PROGRAMADOR

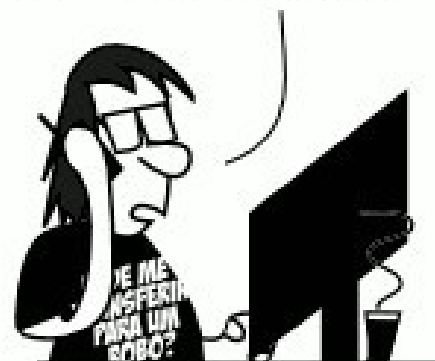
.COM.BR

```
real historia;  
string sender;  
sender = "Herbet Mota";
```



#908

AQUI É O PROGRAMADOR,
ESTOU COM UM PROBLEMA AO
ACESSAR UM DOS WEB SERVICES DE
VOCÊS. O RETORNO ESTÁ VINDO
APENAS COM OS DECIMAIS DO
VALOR, SEM A PARTE INTEIRA.



ENTENDO, MAS QUAL FOI A ABA
DO WEB SERVICE QUE O SENHOR
ESTÁ ACESSANDO?

NÃO, VEJA BEM... WEB
SERVICE NÃO TEM ABA.
É UM SISTEMA MELI QUE
ACESSA O SEU SERVIÇO...



HMMM... ENTENDO... MAS QUAL
A PÁGINA DO WEB SERVICE QUE
O SENHOR ESTÁ NAVEGANDO
E ENCONTRANDO O
ERRO???

CRASH!



Tastypie

- Framework API pra Django
- Auxilia na criação de interface REST

<https://github.com/toastdriven/django-tastypie>

- Outras opções p/ API Django
 - Django REST framework
 - Django-piston



Tastypie

- api.py
- ModelResource / Resource
- Vamos ao código!

<https://github.com/rfaga/tastypie-backbone-example>

- Testando:
- <http://localhost:8000/api/operation/?format=json>

Integrando projeto Django

- Criar arquivo api.py
 - Mapear modelos em Resources
 - Modificar os resources como desejado
- Colocar os Resources no urls.py
- Feito!

Integrando projeto Django

- Código! (step1)

```
### balance/models.py  
# coding: UTF-8
```

```
from django.db import models
```

```
# Create your models here.
```

```
class Operation(models.Model):
```

```
    description = models.TextField(verbose_name="Descrição da operação", null=True)
```

```
    value = models.DecimalField(u'Valor', max_digits=19, decimal_places=2, null=True)
```

```
    date = models.DateField(verbose_name="Data", null=True)
```

Autenticação

- Basic / Web based
- Session based (?)
- OAuth2
- Autenticação / Autorização

Autenticação

- Authentication base class

```
class Authentication(object):  
  
    def is_authenticated(self, request, **kwargs):  
        return True  
  
    def get_identifier(self, request):  
        return identification
```

Pacotes tastypie.authentication.*

Autenticação

- Tarefa: Como autenticar somente usuários?

Autorização

- Código (step2)

Como autenticar para usuários com login na sessão atual?

- Dica: ative o admin pra auxiliar nos testes
- senha padrão do DB: teste / teste

Autorização

- Mesma coisa?

`Tastypie.authorization.Authorization`

Testando a API

- Navegador
 - Chrome e Firefox com ótimos consoles
 - Plugins
- Curl
- Testes:
 - Get detail
 - Post → edit

Testando a API

Advanced Rest Client

Curso Tastybone

PUT operation example

Save

Open

Request

Socket

Projects

Curso Tastybone

Saved

History

Settings

About

Rate this application ♥

Donate

▶ http://localhost:8000/api/operation/2/

☐ GET ☐ POST ☒ PUT ☐ PATCH ☐ DELETE ☐ HEAD ☐ OPTIONS

☐ Other

Raw

Form

Headers

Raw

Form

Files (0)

Payload

Encode payload Decode payload

{"description": "Incoming2"}

application/json

Set "Content-Type" header to overwrite this value.

Clear

Send

Comunicação JSON

- Objetos JSON
- JQuery
 - `$.ajax{url, data, successFunction}`
- Python – simplejson , json

Comunicação JSON

- Tarefa: Via JQuery, pegue os registros e imprima na tela!

```
<html>
  <head>
    <title>JQuery test</title>
    <script src="/static/assets/jquery/jquery.min.js"></script>
  </head>
  <body>
    <script>$.ajax ... </script>
  </body>
</html>
```

Testando a API

- Tastypie-client
- Forma de usar shell Python
- Integração com outros programas Python
- <https://github.com/rfaga/tastypie-client>

Testando a API

- Tastypie-client
- Forma de usar shell Python
- Integração com outros programas Python
- <https://github.com/rfaga/tastypie-client>

Testando a API

- Tastypie-client
- Tarefa mamão-com-açúcar:
 - Pegue as operações
 - Some os valores
 - Imprima o saldo!

Integração Backbone.js

- Backbone.js – Modelar aplicação em JS!
- API para lógica e restrições
- Views/Model/Routers em single app
- Exemplo CRUD – código!



BACKBONE.JS

Integração Backbone.js

- Cuidado de Segurança
- O que é back end e o que é front end?
- Processamento no cliente
- Backbone e JQuery

Deu tempo? Tarefa extra

- Pegue agora a base Backbone e faça seleção de operações por data, em um combo box (select do html)

Links de referência

http://en.wikipedia.org/wiki/Representational_state_transfer
<http://pt.wikipedia.org/wiki/REST>

<https://docs.djangoproject.com/en/1.5/>

<http://django-tastypie.readthedocs.org/en/latest/>
<http://getbootstrap.com/>

<http://backbonejs.org/>
<https://github.com/PaulUithol/backbone-tastypie>
<http://backgridjs.com/>

aa