Intro a Python Django y Django REST framework

Meetup Computación y Algoritmos Monterrey

Facultad de Ciencias Físico Matemáticas, UANL, Nuevo León, México

12 de mayo de 2017

Rafael Rodríguez Morales (rafarodrz [@] gmail.com)

django

python dja...
Search term

php laravel
Search term

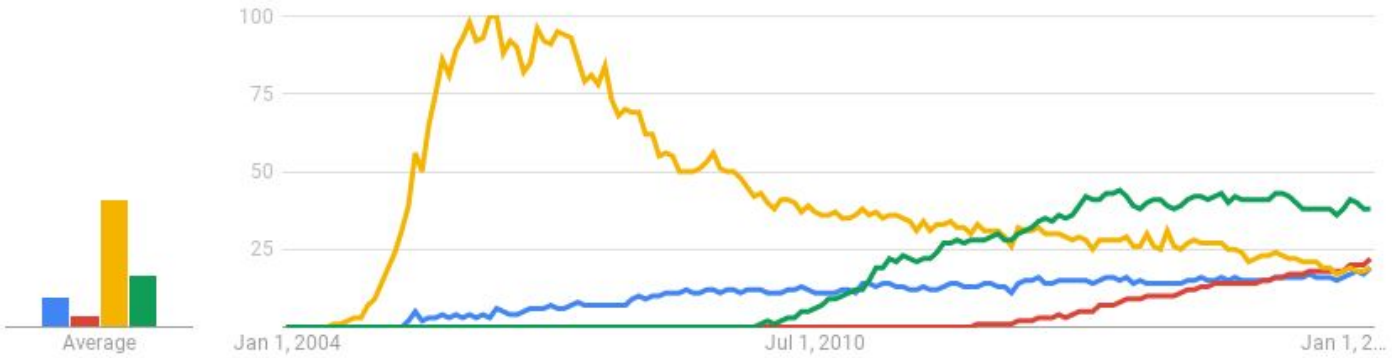
ruby on rails
Search term

node.js
Search term

+

Worldwide | 2004 - present | All categories | Web Search

Interest over time



Información de Google Trends

python dja...
Search term

php laravel
Search term

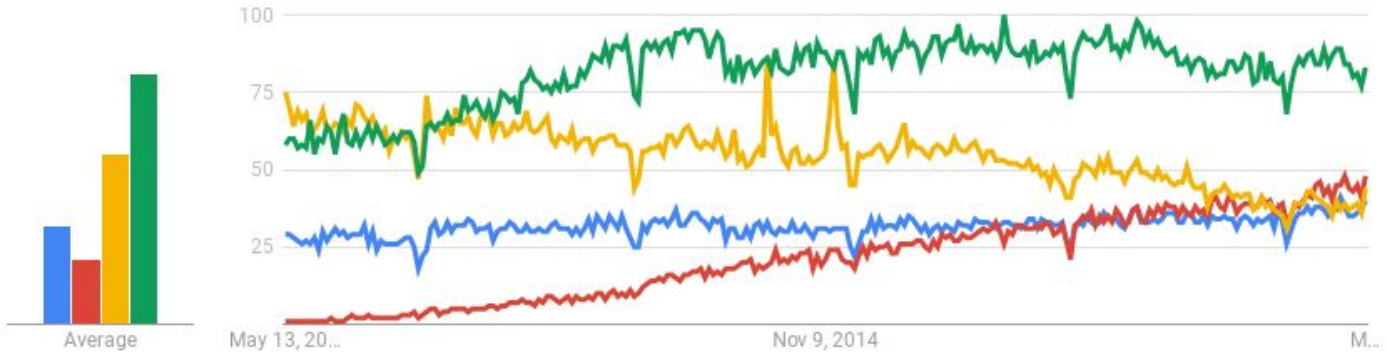
ruby on rails
Search term

node.js
Search term

+

Worldwide | Past 5 years | All categories | Web Search

Interest over time



Información de Google Trends

request

low-level interface

Web Server

WSGI

Web Server

response

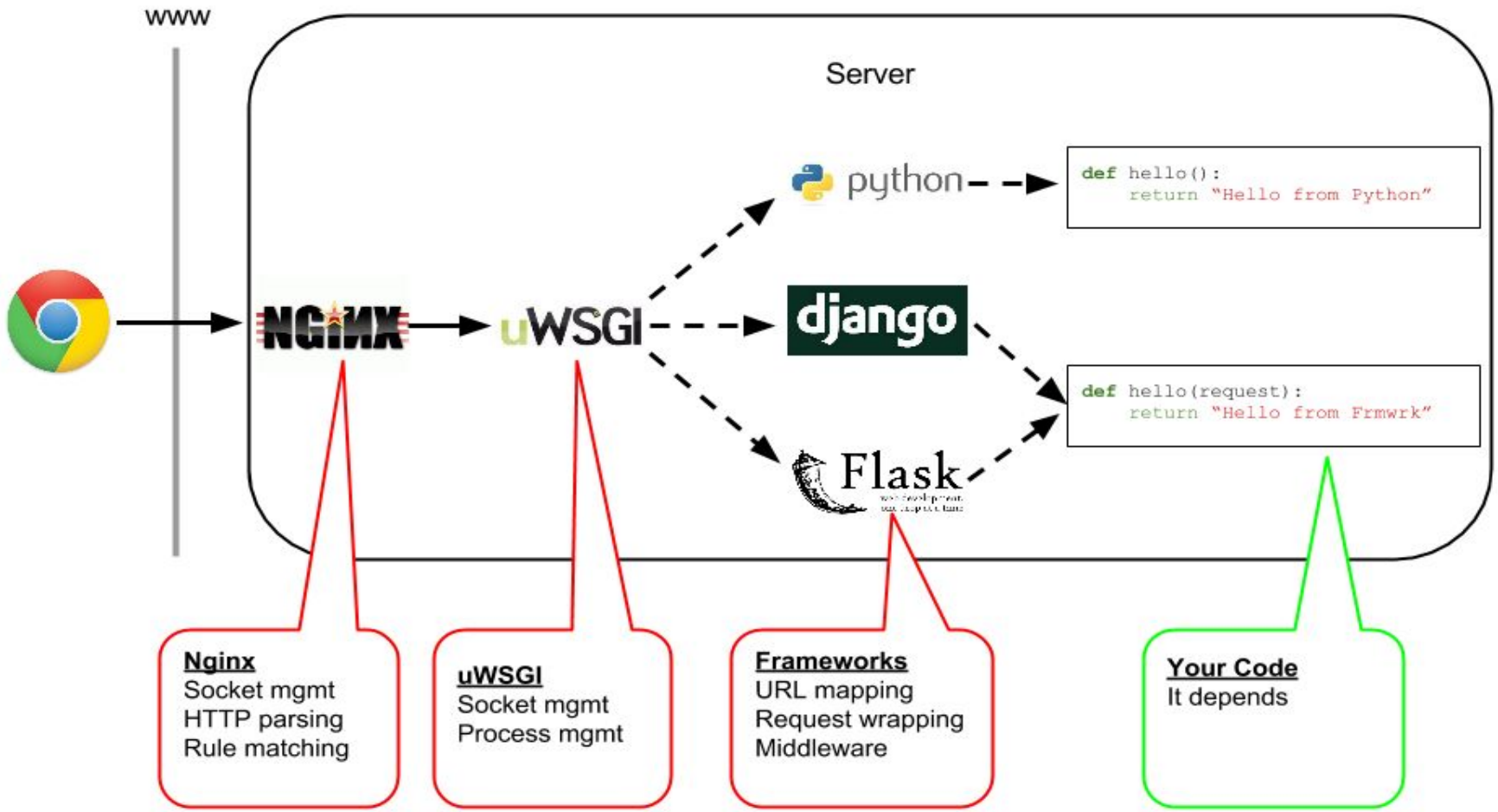
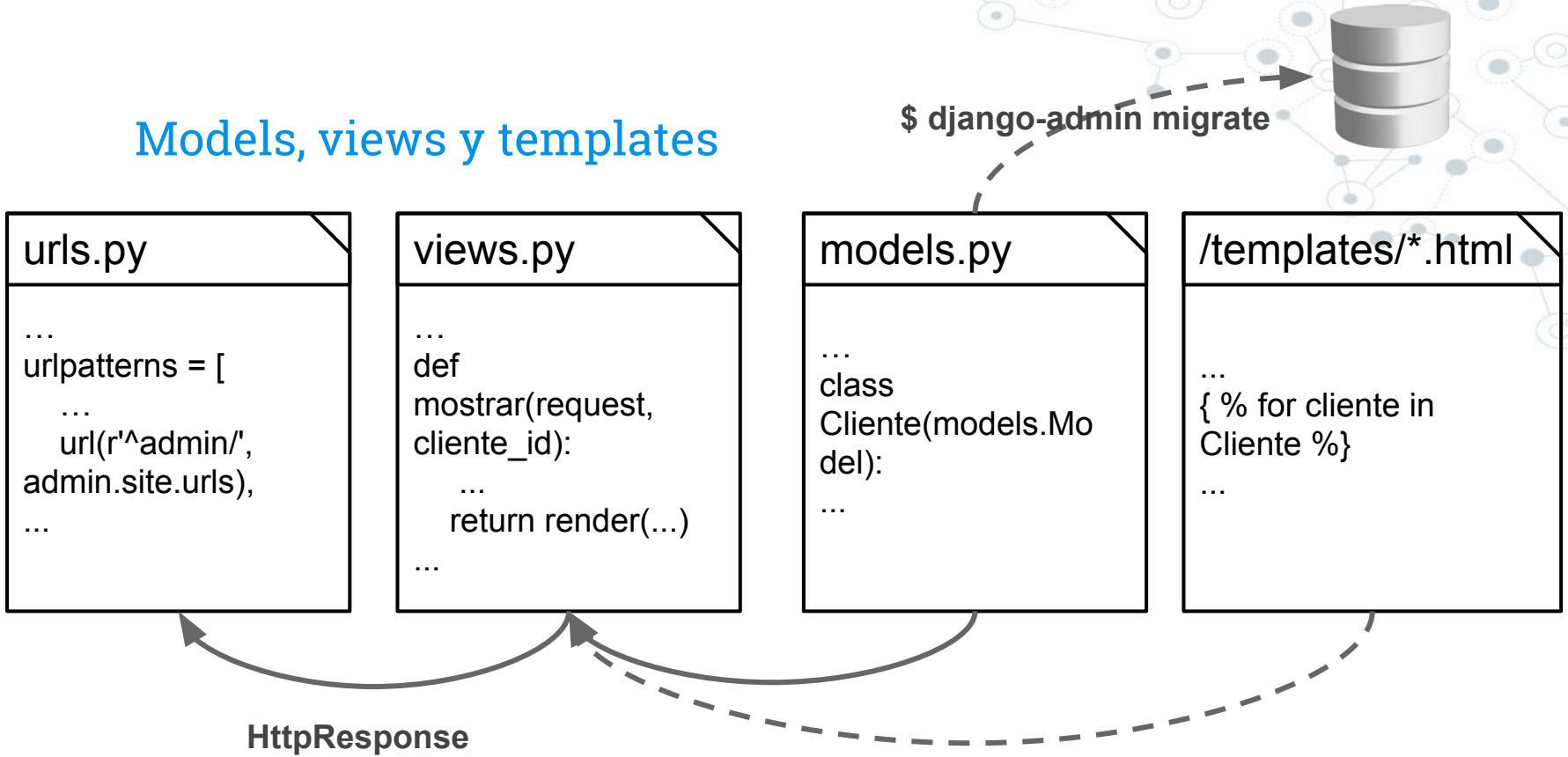


Imagen de http://my-django-python.blogspot.mx/2015/04/host-your-django-website-in-ubuntu.html

Models, views y templates



Crear una API sencilla con Django

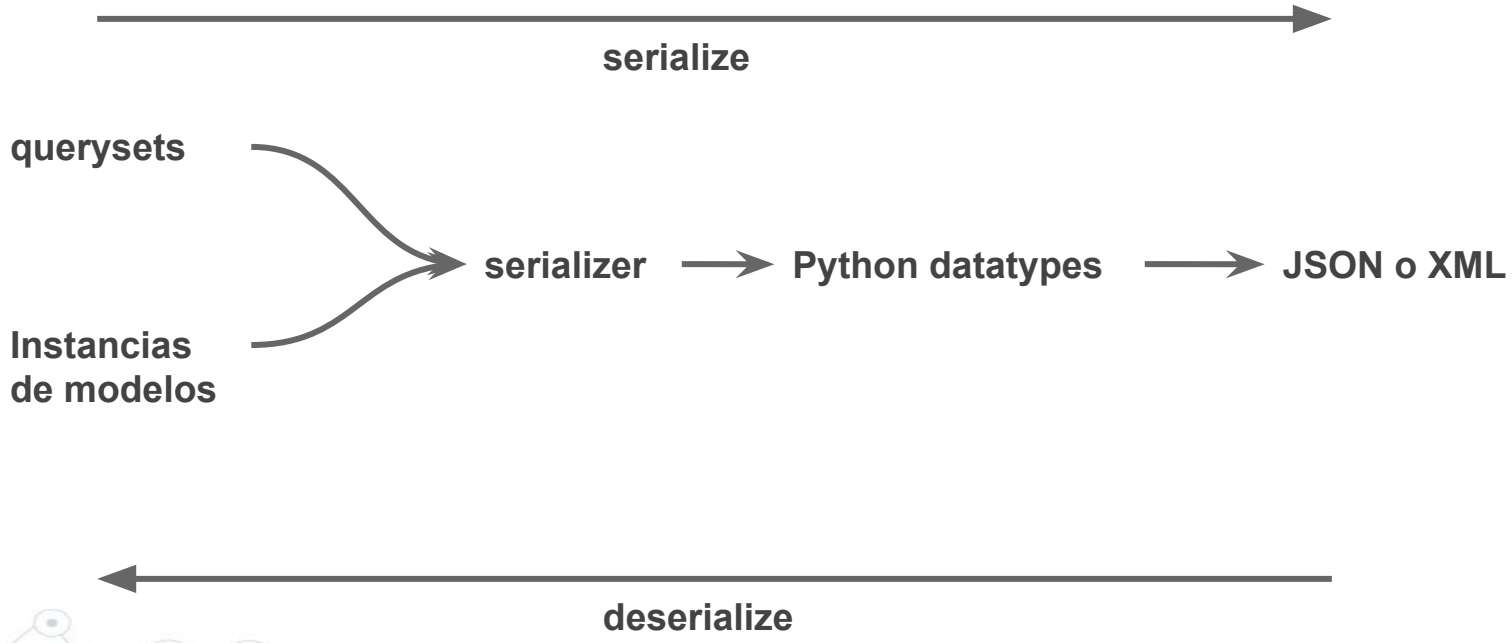
Ejemplo basado en http://www.django-rest-framework.org/tutorial/quickstart/

django REST framework

Frameworks para crear APIs con Django

- Django REST framework
- Tastypie
- Flask-RESTful
- Flask API
- Sandman
- Cornice (para Pyramid)
- Restless (framework agnostic)
- Eve
- ...

Serializers de Django REST framework



Serializing

```
#Definir el model y crear una instancia

class Comment(object):
    def __init__(self, email, content, created=None):
        self.email = email
        self.content = content
        self.created = created or datetime.now()

comment = Comment(email='leila@example.com', content='foo bar')
```

```
from rest_framework import serializers

class CommentSerializer(serializers.Serializer):
    email = serializers.EmailField()
    content = serializers.CharField(max_length=200)
    created = serializers.DateTimeField()


serializer = CommentSerializer(comment)

from rest_framework.renderers import JSONRenderer

json = JSONRenderer().render(serializer.data)
# b'{"email":"leila@example.com","content":"foo
bar","created":"2017-01-27T15:17:10.375877"}'
```

Deserializing

```
from django.utils.six import BytesIO
from rest_framework.parsers import JSONParser

stream = BytesIO(json)
data = JSONParser().parse(stream)

serializer = CommentSerializer(data=data)
serializer.is_valid()
# True
serializer.validated_data
# {'content': 'foo bar', 'email': 'leila@example.com', 'created':
datetime.datetime(2012, 08, 22, 16, 20, 09, 822243)}
```

Cómo funciona Django Rest framework

urls.py

```
…

from rest_framework import
routers

urlpatterns = [
    …
router.register(r'users',views.
UserViewSet)
    ….

...
```

views.py

```
from django.contrib.auth.models
import User, Group

from serializers import
UserSerializer, GroupSerializer

from rest_framework import
viewsets
…
class
UserViewSet(viewsets.ModelView
Set):
        """API Endpoint"""
        queryset = ....
        serializer = UserSerializer
```

serializers.py

```
from
django.contrib.auth.models
import User, Group
from rest_framework import
serializers

class
UserSerializer(serializers.H
yperlinkedModelSerializer):
        class Meta:
            model = User
            fields = ('url',
'username', 'email', 'groups')
```

Levantar el servidor

```
$ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
May 12, 2017 - 17:01:15
Django version 1.11.1, using settings 'apidemo.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Obtener los usuarios desde curl

```
$ curl -i -H "Accept: application/json; indent=4" -u admin:thisisthepassword!
http://127.0.0.1:8000/users/
```

```
HTTP/1.0 200 OK
Date: Fri, 12 May 2017 17:04:58 GMT
Server: WSGIServer/0.2 CPython/3.6.1
Content-Type: application/json
Allow: GET, POST, OPTIONS
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Length: 938

{
    "count": 5,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/users/5/",
            "username": "Brenda",
            "email": "alice@company.com",
            "groups": []
        },
```

Obtener los grupos desde curl

```
$ curl -i -H "Accept: application/json; indent=4" -u admin:thisisthepassword!
http://127.0.0.1:8000/groups/
```

```
HTTP/1.0 200 OK
Date: Fri, 12 May 2017 22:54:42 GMT
Server: WSGIServer/0.2 CPython/3.6.1
Content-Type: application/json
Allow: GET, POST, OPTIONS
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Length: 183

{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/groups/1/",
            "name": "alfa"
        }
    ]
}
```

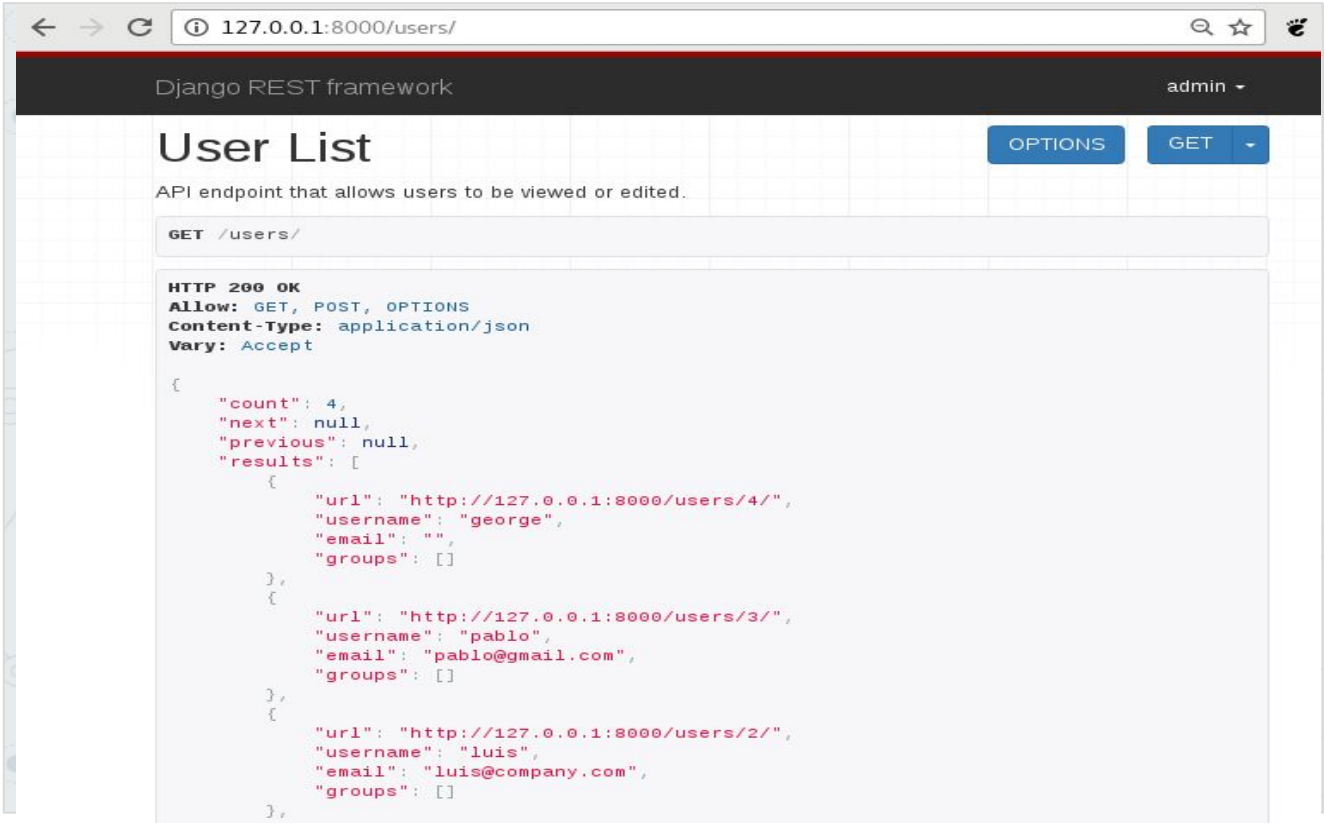
Crear un usuario desde curl

```
$ curl -i -H "Accept: application/json; inent=4" -H "Content-Type:
application/json" -u admin:thisisthepassword! -X POST -d
"{\"username\":\"Joseph\", \"email\":\"lejoseph@company.com\",
\"groups\":[\"http://127.0.0.1:8000/groups/1/\"]}"
http://127.0.0.1:8000/users/
```

```
HTTP/1.0 201 Created
Date: Fri, 12 May 2017 22:52:25 GMT
Server: WSGIServer/0.2 CPython/3.6.1
Content-Type: application/json
Location: http://127.0.0.1:8000/users/6/
Allow: GET, POST, OPTIONS
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Length: 175

{
    "url": "http://127.0.0.1:8000/users/6/",
    "username": "Joseph",
    "email": "lejoseph@company.com",
    "groups": [
        "http://127.0.0.1:8000/groups/1/"
    ]
```

Acceso desde el web browser


127.0.0.1:8000/users/
Django REST framework
admin
User List
OPTIONS
GET
API endpoint that allows users to be viewed or edited.
GET /users/
HTTP 200 OK
Allow: GET, POST, OPTIONS
Content-Type: application/json
Vary: Accept
{
"count": 4,
"next": null,
"previous": null,
"results": [
{
"url": "http://127.0.0.1:8000/users/4/",
"username": "george",
"email": "",
"groups": []
},
{
"url": "http://127.0.0.1:8000/users/3/",
"username": "pablo",
"email": "pablo@gmail.com",
"groups": []
},
{
"url": "http://127.0.0.1:8000/users/2/",
"username": "luis",
"email": "luis@company.com",
"groups": []
},

Obtener los usuarios con request de Python

```python
import requests
from requests.auth import HTTPBasicAuth
import json

url = "http://127.0.0.1:8000/users/"
username = 'admin'
password = 'thisisthepassword!'

response = requests.get(url,auth=HTTPBasicAuth(username, password))
print (response.status_code)

if response.ok:
    print(response.json())
```

Crear un grupo con request de Python

```
import requests
from requests.auth import HTTPBasicAuth

url = "http://127.0.0.1:8000/groups/"
username = 'admin'
password = 'thisisthepassword!'

jsonContent = {"name":"beta"}
response = requests.post(url,auth=HTTPBasicAuth(username, password),
json=jsonContent)

print (response.status_code)

if response.ok:
    print(response.json())
```

```
201
{'url': 'http://127.0.0.1:8000/groups/2/', 'name': 'beta'}
```

Gracias!

Intro a Python Django y Django REST framework

Meetup Computación y Algoritmos Monterrey

Facultad de Ciencias Físico Matemáticas, UANL, Nuevo León, México

12 de mayo de 2017

Rafael Rodríguez Morales (rafarodrz [@] gmail.com)