De paseo por Django

Agenda

- Django in a nutshell
- ORM
- Admin
- Django Rest Framework

django

Django

El framework web para perfeccionistas

con fechas limite

¿Quién usa Django?

YouTube

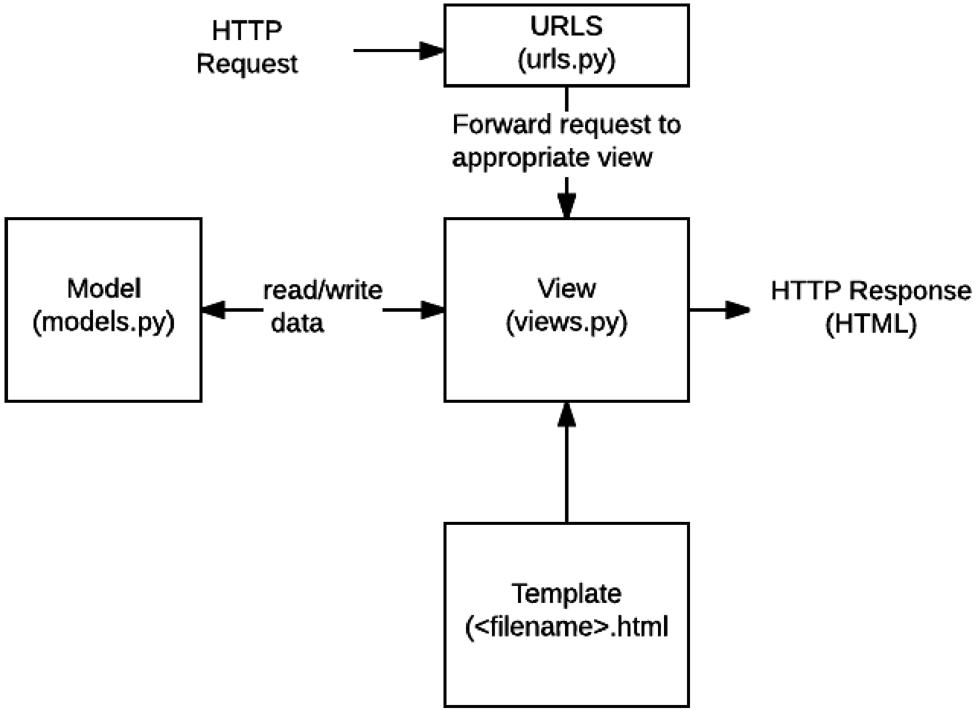
The Washington Post

NATIONAL GEOGRAPHIC

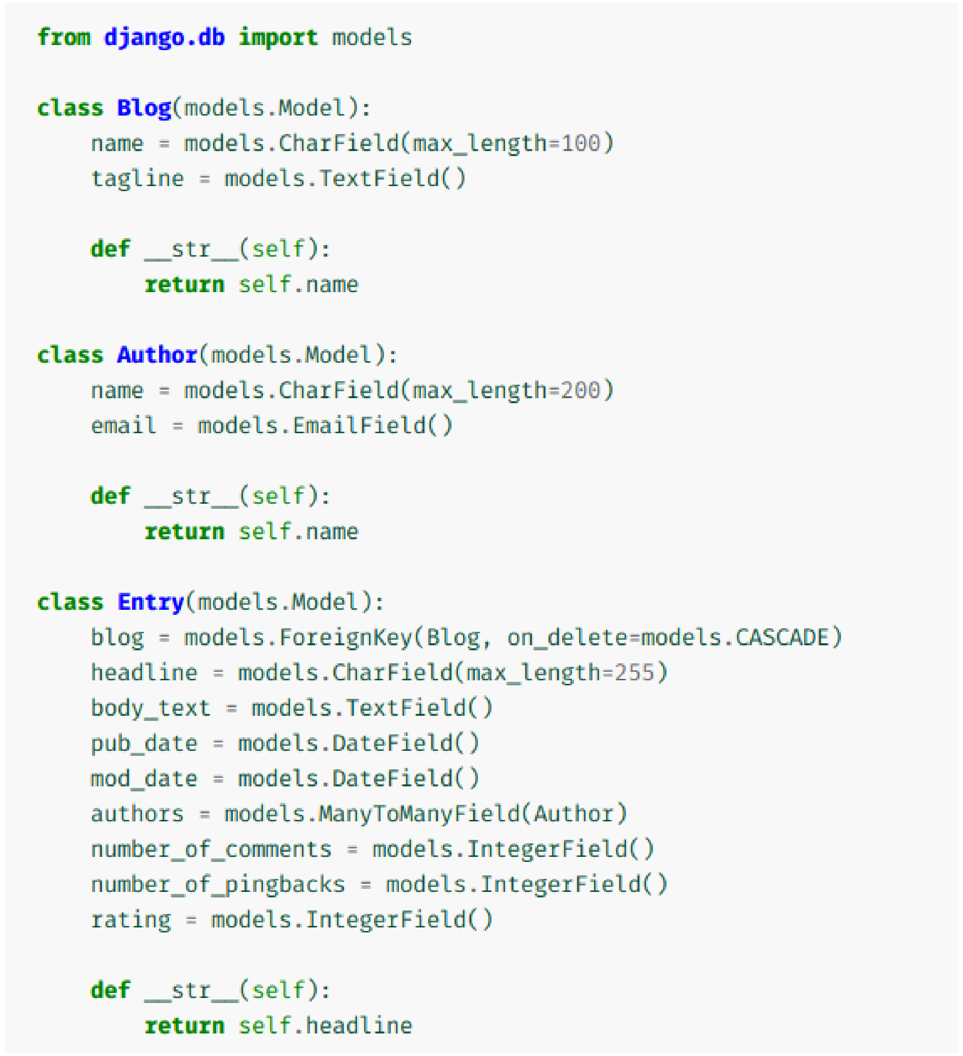
MVC: no - MTV: si

- Model → Django ORM
- Template → Django Template Engine
- View → Clase/Función, Request, Response

El ciclo del request



Modelos



```python
from django.db import models

class Blog(models.Model):
    name = models.CharField(max_length=100)
    tagline = models.TextField()

    def __str__(self):
        return self.name

class Author(models.Model):
    name = models.CharField(max_length=200)
    email = models.EmailField()

    def __str__(self):
        return self.name

class Entry(models.Model):
    blog = models.ForeignKey(Blog, on_delete=models.CASCADE)
    headline = models.CharField(max_length=255)
    body_text = models.TextField()
    pub_date = models.DateField()
    mod_date = models.DateField()
    authors = models.ManyToManyField(Author)
    number_of_comments = models.IntegerField()
    number_of_pingbacks = models.IntegerField()
    rating = models.IntegerField()

    def __str__(self):
        return self.headline
```

Query

```python
from django.db import models

class Blog(models.Model):
    name = models.CharField(max_length=100)
    tagline = models.TextField()

    def __str__(self):
        return self.name

class Author(models.Model):
    name = models.CharField(max_length=200)
    email = models.EmailField()

    def __str__(self):
        return self.name

class Entry(models.Model):
    blog = models.ForeignKey(Blog, on_delete=models.CASCADE)
    headline = models.CharField(max_length=255)
    body_text = models.TextField()
    pub_date = models.DateField()
    mod_date = models.DateField()
    authors = models.ManyToManyField(Author)
    number_of_comments = models.IntegerField()
    number_of_pingbacks = models.IntegerField()
    rating = models.IntegerField()

    def __str__(self):
        return self.headline
```

```python
>>> from blog.models import Blog
>>> b = Blog(name='Beatles Blog', tagline='All the latest Beatles news.')
>>> b.save()
```

```python
>>> john = Author.objects.create(name="John")
>>> paul = Author.objects.create(name="Paul")
>>> george = Author.objects.create(name="George")
>>> ringo = Author.objects.create(name="Ringo")
>>> entry.authors.add(john, paul, george, ringo)
```

```python
>>> from blog.models import Author
>>> joe = Author.objects.create(name="Joe")
>>> entry.authors.add(joe)
```

Query

```
from django.db import models

class Blog(models.Model):
    name = models.CharField(max_length=100)
    tagline = models.TextField()

    def __str__(self):
        return self.name

class Author(models.Model):
    name = models.CharField(max_length=200)
    email = models.EmailField()

    def __str__(self):
        return self.name

class Entry(models.Model):
    blog = models.ForeignKey(Blog, on_delete=models.CASCADE)
    headline = models.CharField(max_length=255)
    body_text = models.TextField()
    pub_date = models.DateField()
    mod_date = models.DateField()
    authors = models.ManyToManyField(Author)
    number_of_comments = models.IntegerField()
    number_of_pingbacks = models.IntegerField()
    rating = models.IntegerField()

    def __str__(self):
        return self.headline
```

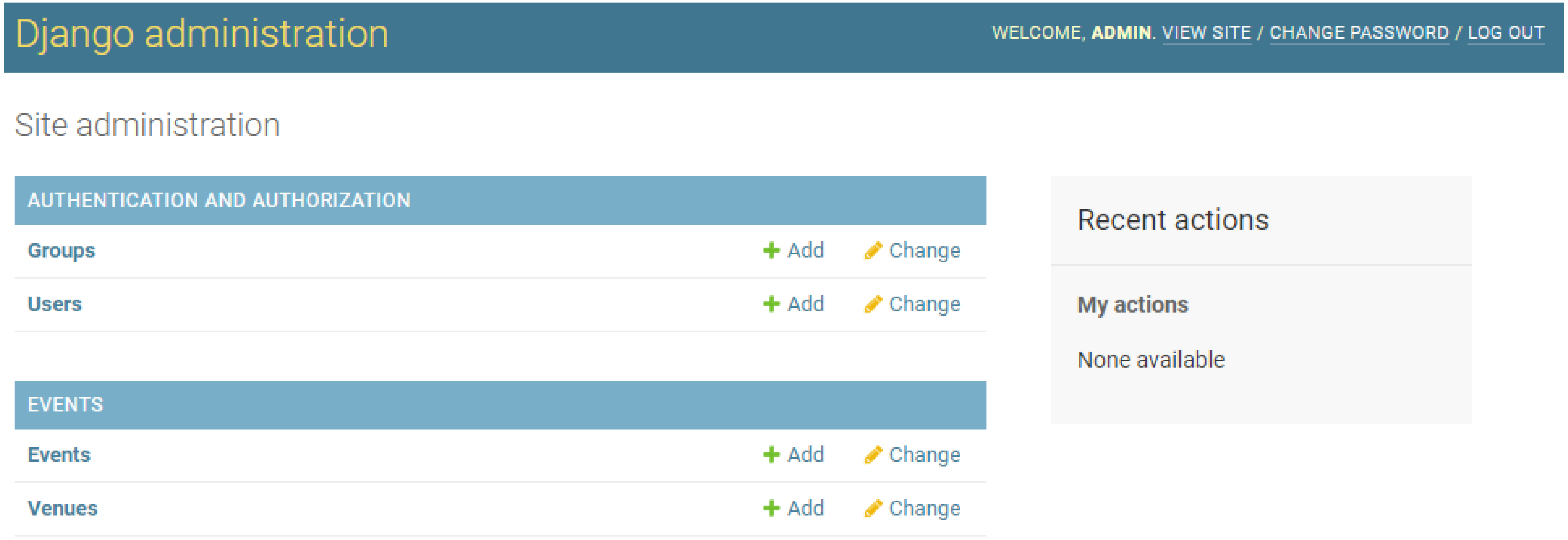
Lookups

```
>>> Entry.objects.filter(
...     headline__startswith='What'
... ).exclude(
...     pub_date__gte=datetime.date.today()
... ).filter(
...     pub_date__gte=datetime.date(2005, 1, 30)
... )
```

Lazy

```
>>> q = Entry.objects.filter(headline__startswith="What")
>>> q = q.filter(pub_date__lte=datetime.date.today())
>>> q = q.exclude(body_text__icontains="food")
>>> print(q)
```

Admin



Django administration

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	✚ Add	✎ Change
Users	✚ Add	✎ Change

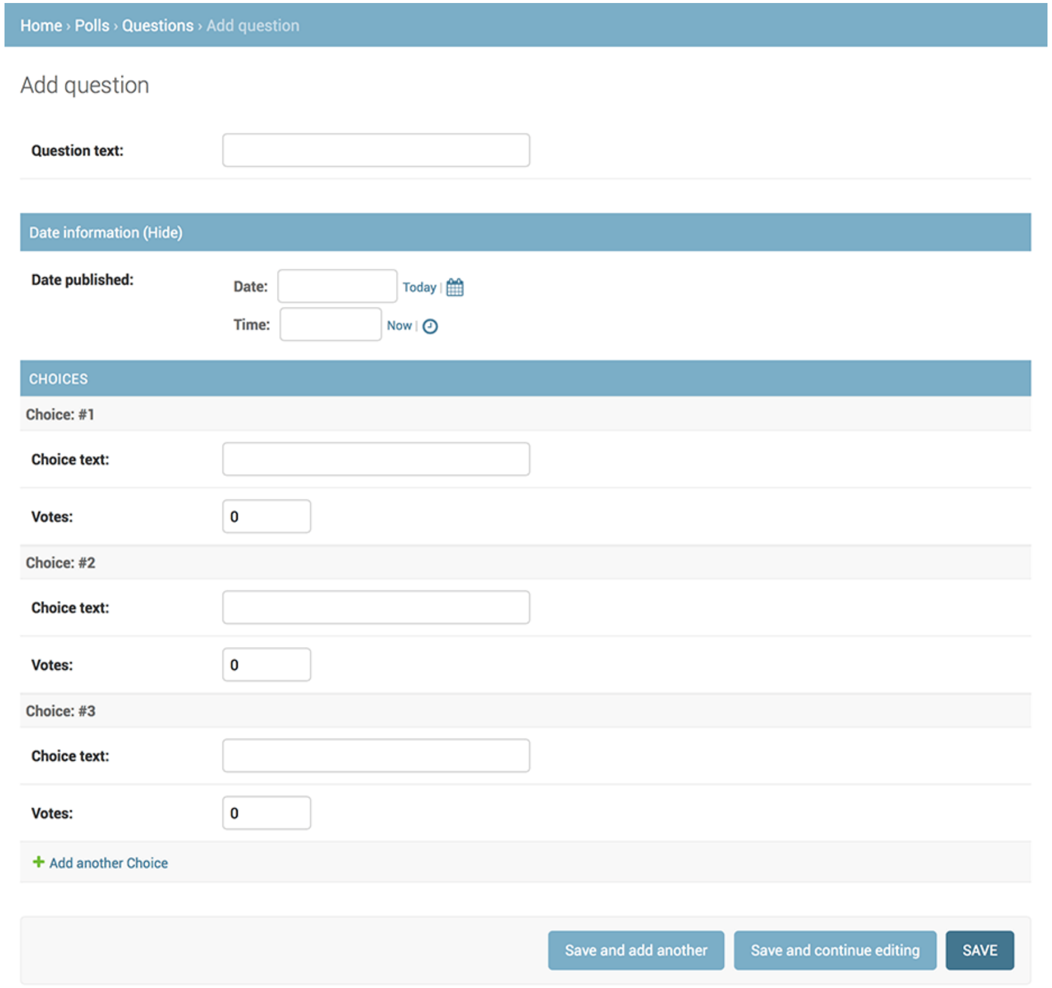
EVENTS		
Events	✚ Add	✎ Change
Venues	✚ Add	✎ Change

Recent actions

My actions

None available

Admin



Home › Polls › Questions › Add question

Add question

Question text:	

Date information (Hide)

Date published: Date: Today | Time: Now |

CHOICES

Choice: #1

Choice text:	
Votes:	0

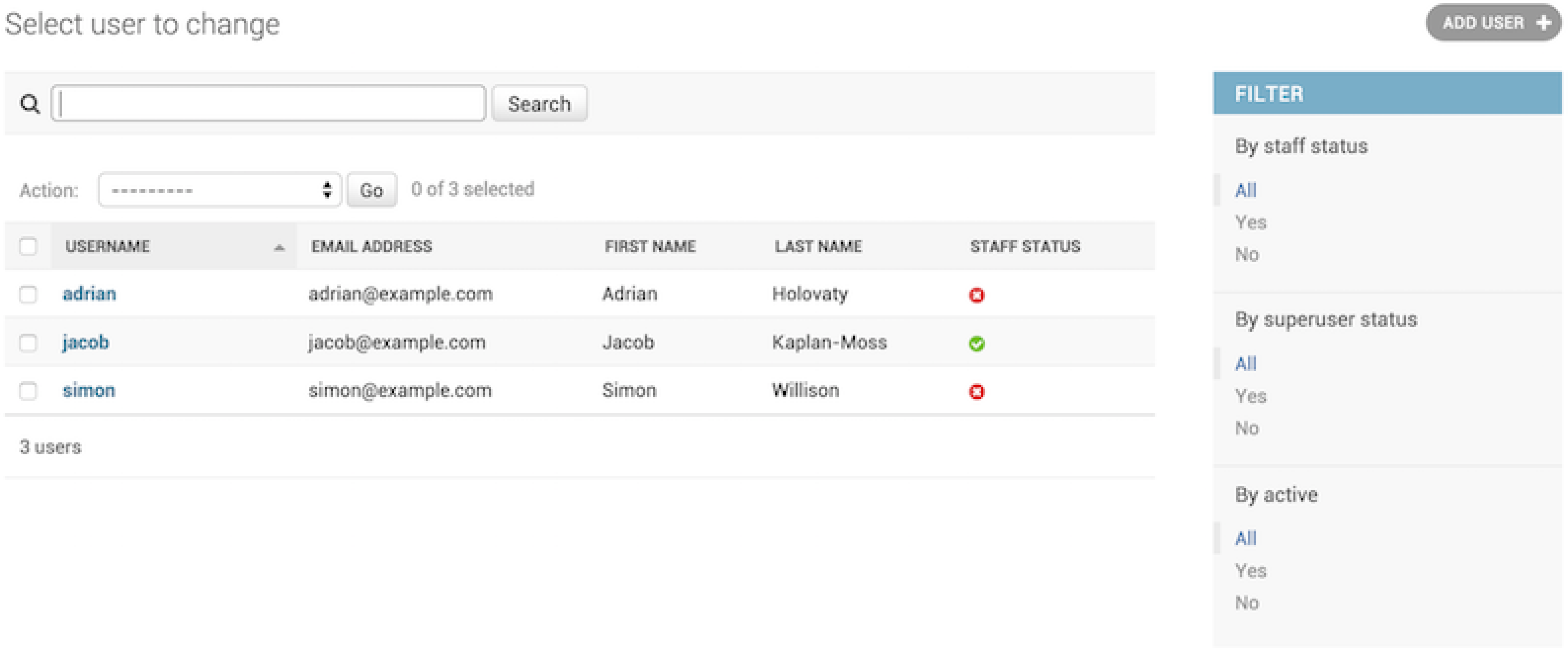
Choice: #2

Choice text:	
Votes:	0

Choice: #3

Choice text:	
Votes:	0

Admin



Select user to change

ADD USER +

Search

Action: --------- Go 0 of 3 selected

	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
☐	adrian	adrian@example.com	Adrian	Holovaty	✗
☐	jacob	jacob@example.com	Jacob	Kaplan-Moss	✓
☐	simon	simon@example.com	Simon	Willison	✗

3 users

FILTER

By staff status

All
Yes
No

By superuser status

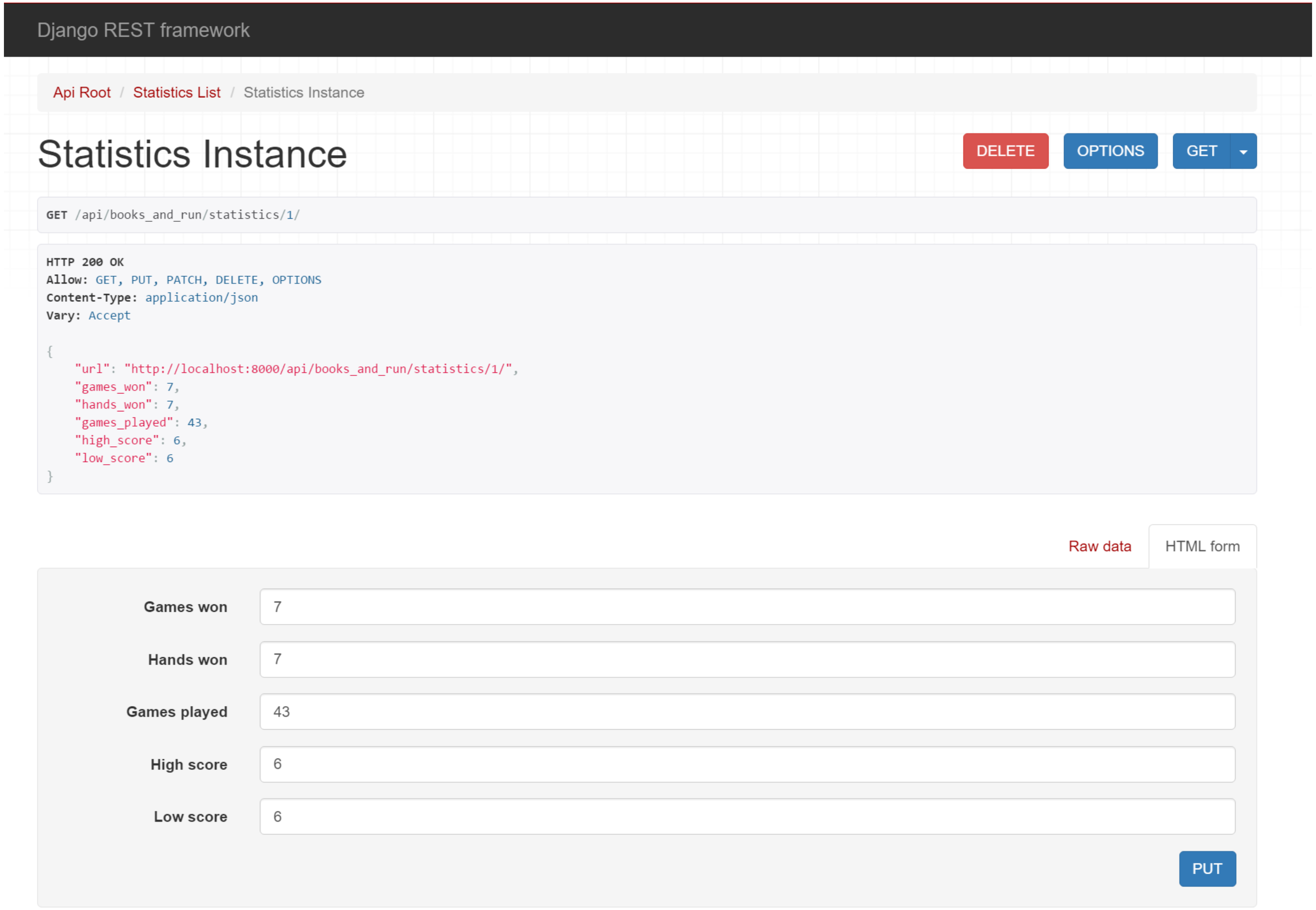
All
Yes
No

By active

All
Yes
No

DRF – Lo mismo pero REST

- Auth
- Serializadores
- REST y JSON
- Navegación API



Django REST framework

Api Root / Statistics List / Statistics Instance

Statistics Instance

DELETE OPTIONS GET

```
GET /api/books_and_run/statistics/1/
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "url": "http://localhost:8000/api/books_and_run/statistics/1/",
    "games_won": 7,
    "hands_won": 7,
    "games_played": 43,
    "high_score": 6,
    "low_score": 6
}
```

Raw data HTML form

Games won	7
Hands won	7
Games played	43
High score	6
Low score	6

PUT

Para seguir viendo

- Django Channels
- Two scoops of Django (Libro)
- Django Girls (Tutorial)
- Documentación oficial
- Toneladas de app “plugueables”
- ….

PREGUNTAS?

RESPUESTAS?

Matias Barriento

- Programador / Payaso / Amante de listar cosas
- Socio de Python Argentina
- Trabajo en Kilimo
- Doy charlas

¡MUCHAS GRACIAS!