



Universidad Autónoma de Baja California
Campus Tijuana
Facultad de contaduría y administración



Alumno:
García Bretado David Antonio

Grupo:
372

API Django

Programación en Python

Tijuana B.C. a 9 de noviembre de 2018

API Django

Introducción

Una forma muy común de obtener información de diferentes fuentes es mediante una **API REST**, que proporciona distintos **API endpoints** los cuales son accesibles con tal de conseguir unos datos determinados. **Django REST Framework** es un framework que nos permite el fácil desarrollo de una API REST en **Python**. Actualmente esta tecnología se usa en diversos proyectos, como por ejemplo la API del **Racó** o el proyecto **UOC index**.

Instalación

Una buena práctica es crear un **entorno virtual** con todo lo necesario para poder ejecutar la API REST creada con Django:

?

```
1 # Create the project directory
2 mkdir tutorial
3 cd tutorial
4 # Create a virtualenv to isolate our package dependencies locally
5 virtualenv env
6 source env/bin/activate # On Windows use `env\Scripts\activate`
```

Una vez instalado el entorno, aún nos hace falta instalar las **librerías** necesarias de Python para poder usar Django:

?

```
1 # Install Django and Django REST framework into the virtualenv
2 pip install django
3 pip install djangorestframework
4 # Set up a new project with a single application
5 django-admin.py startproject tutorial . # Note the trailing '.' character
6 cd tutorial
7 django-admin.py startapp quickstart
```

Ahora ya lo tenemos todo a punto para empezar nuestro proyecto con Django.

Estructura del proyecto

Analizando lo que se ha creado dentro de la carpeta tutorial podemos ver como es la estructura de un proyecto con Django:

- env: entorno virtual con las librerías instaladas y los ficheros necesarios para la correcta ejecución del proyecto.
- tutorial: carpeta donde podemos encontrar toda la configuración del proyecto. Destacar settings.py y urls.py, ficheros que respectivamente controlan la configuración del proyecto

(aplicaciones instaladas, módulos que se pueden usar, tipos de bases de datos soportadas...) y las urls disponibles.

- `quickstart`: carpeta donde podemos encontrar la configuración de la aplicación `quickstart` del proyecto.
- `manage.py`: ejecutable en Python que permite la puesta en marcha de la API.

Estructura de la aplicación

Analizando lo que se ha creado dentro de la carpeta `quickstart` podemos ver como es la estructura de una aplicación Django:

- `models.py`: fichero con la información sobre los modelos de datos que usa nuestra aplicación.
- `views.py`: archivo que contiene todas las vistas empleadas por la aplicación.
- `tests.py`: fichero para desarrollar posibles tests para la aplicación.
- `migrations`: directorio encargado de detectar cambios en las bases de datos que usa la aplicación.

Es muy recomendable crear ficheros adicionales, especialmente estos dos:

- `urls.py`: fichero con las posibles urls o endpoints de la API.
- `queries.py`: archivo con las distintas llamadas a las bases de datos.

Fuentes

<https://inlab.fib.upc.edu/es/blog/django-api-rest>