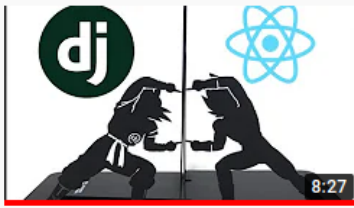


Iván Simarro

Bibliografía Proyecto de Final de Ciclo

DAW

## Videos clave en mi aprendizaje

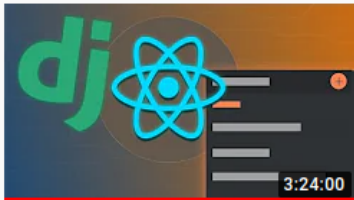


### Cómo implementar Django con React

Julio Mendez • 11.309 visualizaciones

Así como dice el título, hoy haremos la implementación de Django con React. Si te gustó, recuerda dejarme un like o suscribirte :)

▶ Cómo implementar Django con ...



### Django + React Notes App

Dennis Ivy • 69.296 visualizaciones

Continuation of my last stream (React JS Crash Course) but now with a Django backend using the Django REST Framework. Previous...

▶ Django + React Notes App

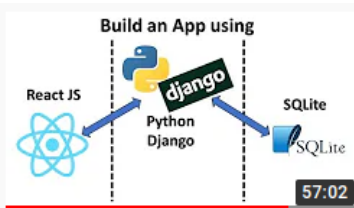


### Full Stack Project with React.js and Django (Task Manager...)

Bek Brace • 31.986 visualizaciones

This is Full-Stack Project with React.js for the frontend UI and Django Framework for the backend. We are using Django for the structure...

▶ Full Stack Project with React.js ...



### React JS + Python Django + SQLite | full-stack app tutorial

Art of Engineer • 12.352 visualizaciones

Code and Github link: <https://art-of-engineer.blogspot.com/2021/07/react-js-python-django-sqlite.html>

▶ React JS + Python Django + SQ...



### Curso Django Rest Framework | Basico y Avanzado (Curso...

Solo Python • 40.927 visualizaciones

Aprende Django Rest Framework y como crear un API desde cero a experto. Tendras las habilidades para recrear el backend de...

▶ Curso Django Rest Framework | ...

# Webs clave en mi aprendizaje

## Requisitos previos

Paso 1: Creación de un entorno virtual de Python e instalación de dependencias

Paso 2: Creación del proyecto Django

Paso 3: creación del frontend de React

Paso 4: creación del modelo de cliente y datos iniciales

Paso 5: creación de la API REST

Incorporación de la clase de serialización

Incorporación de vistas de API

Incorporación de pun-

```
(env) sammy@ubuntu:~$ python manage.py migrate
```

[Copy](#)

Inicie el servidor de desarrollo local:

```
(env) sammy@ubuntu:~$ python manage.py runserver
```

[Copy](#)

El resultado debe ser similar a lo siguiente:

```
Output
Performing system checks...

System check identified no issues (0 silenced).
October 22, 2018 - 15:14:50
Django version 2.1.2, using settings 'djangoreactproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

[Cómo crear una aplicación web moderna para gestionar la información de clientes con Django y React on Ubuntu 18.04 | DigitalOcean](#)

## Academia de programación

Índice del archivo

Python para principiantes

Python para interfaces gráficas

Gestor de clientes con Python

Más tutoriales sobre Python

Módulo analítico NumPy

Módulo gráfico Matplotlib

Django para principiantes

[Server API pelis en Django Rest](#)

Más tutoriales sobre Django

JavaScript para principiantes

Conversor de divisas en JS

Introducción a P5.js

Introducción a Node.js

Chat online con Node.js

Cliente API pelis en Nuxt.js

api/views.py

```
from .models import Pelicula
from .serializers import PeliculaSerializer
from rest_framework import viewsets

class PeliculaViewSet(viewsets.ModelViewSet):
    queryset = Pelicula.objects.all()
    serializer_class = PeliculaSerializer
```

Y ahora añadimos a las urls la ruta de la viewset en la API:

api\_pelis/urls.py

```
from django.contrib import admin
from django.urls import path, include

from api import views
from rest_framework import routers
```

[Proyecto API con Django Rest Framework](#)

```
from django import forms
from .models import AlbumImage

class UploadImageForm(forms.ModelForm):

    class Meta:
        model = AlbumImage
        fields = ['album', 'image']
```

Creamos la función de la vista en **mi\_proyecto/albums/views.py**

```
from django.contrib.auth.decorators import login_required
from django.shortcuts import render_to_response, RequestContext

from .forms import UploadImageForm

@login_required
def upload_image_view(request):
    if request.method == 'POST':
        form = UploadImageForm(request.POST, request.FILES)
        if form.is_valid():
```

[Upload de imágenes con Django, mediante formulario](#)

Python3

```
from rest_framework import serializers
from rest_framework import status

@api_view(['POST'])
def add_items(request):
    item = ItemSerializer(data=request.data)

    # validating for already existing data
    if Item.objects.filter(**request.data).exists():
        raise serializers.ValidationError('This data already exist')

    if item.is_valid():
        item.save()
        return Response(item.data)
    else:
        return Response(status=status.HTTP_404_NOT_FOUND)
```

<https://www.geeksforgeeks.org/django-rest-api-crud-with-drf/>

```
from rest_framework import serializers
from .models import Todo

class TodoSerializer(serializers.ModelSerializer):
    class Meta:
        model = Todo
        fields = ('id', 'title', 'description', 'completed')
```

From the rest\_framework package, we import the serializers. We create a class, TodoSerializer that extends from the ModelSerializer class. We then go ahead and specify the model and fields we want to be returned.

Let's also update the todo/views.py:

```
from django.shortcuts import render
from .serializers import TodoSerializer
from rest_framework import viewsets
from .models import Todo

class TodoView(viewsets.ModelViewSet):
    serializer_class = TodoSerializer
    queryset = Todo.objects.all()
```

<https://www.section.io/engineering-education/react-and-django-rest-framework/>