# YouTube Clone Backend Framework

## Introduction

Backend development is typically the first step after the planning phase of the creation of a new application. The frontend application is dependent on the backend server’s data and functionality. This is an exercise of building a backend Web API prior to the start of the frontend of the application. We will be testing and interacting with this backend Web API using Postman to ensure everything is working as expected. This will also help us visualize what the structure of the data will be that the frontend will be consuming.

## Learning Objectives

Understand how to build a Web API for a full-stack application with database integration.

## Task

Create a Django REST API connected to a MySQL database allowing a frontend application to POST comments to the MySQL database relative to the video the user is currently watching.

## Resources

**Lectures**

* All previous Django lectures
* Django & JWT

**Documents**

* User Stories document – contains information and links relating to making requests to the YouTube Data API
* [React Django Starter Code Setup Guide](https://devcodecamp-my.sharepoint.com/:w:/p/michael/EdEG2flLvNxAmeQB-MeXL-oBuri3zyhSqXoF0Hw8nAWbFw?e=6gXxGF)

**Relevant Projects**

* Music Library (both front and back end)

**Other Resources**

* **Video – Working with User Data as a Foreign Key in Django REST Framework**

## Setup Steps

1. Follow the setup steps in the React Django Starter Code Setup Guide to get your full project up and running. [LINK TO DOCUMENT](https://devcodecamp-my.sharepoint.com/:w:/p/michael/EdEG2flLvNxAmeQB-MeXL-oBuri3zyhSqXoF0Hw8nAWbFw?e=6gXxGF)
2. Watch the videos on your portal that walk through the backend portions of the starter code.
3. Read over the user stories for YouTube Clone Flex Backend closely to understand the features required.
4. Create a new app for comments using the django-admin startapp command
   1. Be sure to register the new app in your settings.py file
   2. Be sure to add the new app to your admin site
5. Write a Model class called Comment
6. Write a Serializer for the Comment Model
7. Write two view functions to satisfy the requirements of the user stories.
   1. Be sure they are tied to URL configurations in a urls.py file
   2. Make sure the URLs for this app are included in the base urls.py file for your project
8. Test endpoints in Postman!

## End Result

The result of your YouTube Clone API backend will be the execution of requests made in Postman.

You must test your YouTube Clone API by executing each request you create in Postman.