# Music Library REST API (Backend)

## Introduction

## The Music Library Backend REST API project is the first part of your first full stack application project! Your REST API will serve endpoints accessible through HTTP requests over the internet allowing external applications to interact with your application. You will be connecting your Music Library frontend application to this API upon completion of the project.

## Technologies

Django, Django REST Framework (DRF), Postman

## Learning Objective

## The learning objective of this project is to learn how to build your own REST API using the Django REST Framework. In this project you will learn how to build your own REST API complete with models, serializers, endpoint pathing, class views, and database integration.

💡 It is important to setup the VS Code launch.json file so you can run your API server in debug mode and utilize breakpoints to become familiar with the applications behavior.

## Resources

**PowerPoints**

* Django REST API
* Web Services
* Postman

**Documents**

* User Stories

**Relevant Projects**

Django REST Tutorial Series

Products API

Heroes & Villains

## Tasks

1. **Review all videos and documentation.** Make sure to have a clear direction regarding the end result of the project as well as all supporting documentation that is available to you.
2. **Build and test the API by running the application and testing each endpoint in Postman.** Remember to export the collection once you have tested each endpoint successfully and add the exported collection to your GitHub repository!

## Setup Steps

1. Start by creating a Django REST backend project as you have with previous projects. If needed, refer back to the Django REST API tutorial series to refresh your memory on how to set this up.
2. We will need to install an additional package on our Django back end to ensure our front end’s requests are not blocked by CORS (Cross-Origin Resource Sharing) policy:
   1. Inside your back-end project directory: **pipenv install django-cors-headers**
   2. Inside your back-end project settings.py:
      1. Add **‘corsheaders’** to INSTALLED\_APPS
      2. Add ‘**corsheaders.middleware.CorsMiddleware**’ to MIDDLEWARE
      3. Add new setting anywhere in settings.py file: **CORS\_ORIGIN\_ALLOW\_ALL=True**

## End Result

The final result of the project will be a full CRUD REST API for creating, reading, updating and deleting songs from your MySQL database.