

Djnago Training - 5. Django REST Framework

Materials

1. [Settings](#)
2. [REST Framework](#) Django REST Framework (or DRF) is your gateway to create a REST API, essentially, we want to create a set of endpoints that do the following:
 - Take the data from the python domain in a `dict` or `django Model` object, transform it into a standard unified format like `JSON`, `XML`, or `YAML` and return it to the user, this is usually a `GET` request. This process is called *serialization*
 - Take the data from the user in `JSON` or `XML` formats, transform it into a pythonic format like a python `dict` datastructure or `django Model` object so that we can do something with it like store it in the database. This process is called *deserialization*
 - Django already comes with support for serialization and deserialization ([see](#)), but DRF comes with support for customization and easy to use `Views` that provide the features above in minimal code
3. [Tutorial](#) from 1.Serilization to 4.Authentication and Permissions
4. [Requests](#) and [Responses](#)
5. [Views](#)
6. [Generic Views](#)

Task

1. Feel free to remove any non-API views that we created from before
2. Create a class-based view at the path `/artists/` that returns a list of artists in JSON format for `GET` requests, the artist data should include the following fields.

```
{
  "id": ...
  "stage_name": ...
  "social_link": ...
}
```

3. The same view above should accept `POST` requests and accept all the fields on the artist model (excluding the id)
 - Include proper validation for each field as listed on the artist model:
 - this field is required
 - this field value already exists (for unique fields)
 - If the request passes the validation process, the given data should be used to create and save an artist instance
4. You may want to use a tool to make the HTTP requests and inspect them, I personally use [Postman](#) but there are other options like [curl](#) and [Insomnia](#)
5. Feel free to write the API views yourself or use DRF's generic views.
6. Feel free to write the serializers yourself or use DRF's `ModelSerializer`

Guidelines

1. Use this gitignore [template](#)
2. List your environment variables (if any) in a `.env.example` file
3. Don't forget to run `manage.py makemigrations` and `python manage.py migrate`
4. You are allowed to refer to any articles, documentation source, questions on StackOverFlow
5. You are not allowed to consult other people through any medium of communication
6. Your grade will be affected if your code isn't clean, you should maintain code cleanliness as much as you can