

## Step 1: Project Setup

- Install Django and DRF (do not forget about venv).
- Create a new Django project (I hope you know the command).
- Create a new Django app (I hope you know the command).

## Step 2: Model Setup

- User model (do not inherit from Django User model !!!):
  - username (CharField): The username of the user.
  - email (EmailField): The email address of the user.
  - password (CharField): The password of the user.
- Product model:
  - name (CharField): The name of the product.
  - description (TextField): A description or details about the product.
  - price (DecimalField): The price of the product.
  - quantity (PositiveIntegerField): The quantity of the product available in stock.
- Order model:
  - user (ForeignKey): A reference to the User model, representing the user who placed the order.
  - products (ManyToManyField): A reference to the Product model, indicating the products included in the order.
  - created\_at (DateTimeField): The date and time when the order was created.
  - status (CharField): The status of the order (e.g., "Pending", "Shipped", "Delivered").
- Category model:
  - name (CharField): The name of the category.
  - description (TextField): A description or details about the category.
- Review model:
  - product (ForeignKey): A reference to the Product model, representing the product being reviewed.
  - user (ForeignKey): A reference to the User model, representing the user who wrote the review.
  - rating (PositiveIntegerField): The rating given by the user for the product (e.g., on a scale of 1 to 5).

- comment (TextField): A comment or review text provided by the user.

### **Step 3: Configure database**

- Set up the database connection in the project's settings file (if you want to use a db engine other than sqlite ).
- Create database migrations (I hope you know the command).
- Apply the migrations (I hope you know the command).

### **Step 4: Create serializers**

- User serializer, including all fields.
- Product serializer: including all fields.
- Order serializer: including all fields. For the 'products' field, use a nested serializer for the Product model.
- Category serializer: including all fields.
- Review serializer: including all fields. For the 'product' field, use a nested serializer for the Product model.

### **Step 5: Views**

- Create views to handle the creation, retrieval, update, and deletion of users for all models: User, Product, Order, Category, Review.
- Use the proper serializers for request/response data serialization and validation.
- Apply relevant permission classes to ensure proper access control.

### **Step 6: URL Configuration**

- Create URL patterns for creation, retrieval, update and deletion for all views. Some might also have a listing url.

### **Step 7: Run and test Application**

- Start the Django development server (I hope you know the command).
- Use tools like Postman or cURL to send HTTP requests to the defined endpoints and verify the responses.
- Test various scenarios like creating users, tasks, categories, and comments, updating and deleting them, and retrieving lists or individual instances.

**Good luck :)**

**Try to find and search on your own as much as you can. Only in critical situations ask me. Try to be independent, because this will result and benefit in the future.**