

# **Title: E-commerce Django REST API Task**

Task Description: You are tasked with developing a Django REST API for an e-commerce website. The API should handle various functionalities related to managing products, categories, sub-categories, and multiple images for each product. Additionally, you need to implement a module for managing customer reviews for products.

**Time Limit: 24 hours**

Requirements:

* Category and Sub-Category Management:
  + Create, Read, Update, and Delete (CRUD) operations for categories and sub-categories.
  + Each category should have fields like name, description, and an optional parent category for hierarchical classification.
  + Sub-categories should be associated with parent categories.
* Product Management:
  + Create, Read, Update, and Delete (CRUD) operations for products.
  + Each product should have fields like name, description, price, stock quantity, category, sub-category, and multiple image upload option.
  + Allow users to upload multiple images for each product.
  + Products should be filterable by category, sub-category, and searchable by name.
* Adding Products under Subcategories:
  + Implement functionality to add products under specific sub-categories.
* Customer Reviews Management:
  + Create, Read, Update, and Delete (CRUD) operations for customer reviews on products.
  + Each review should have fields like a rating (1 to 5), comment, and timestamp of submission.
  + Allow customers to add reviews for products they have purchased.

Submission:

* Provide the complete source code of the Django project.
* Include a brief README file with instructions on how to set up and run the project locally.

Evaluation: Your submission will be evaluated based on the following criteria:

* Completeness and correctness of implemented functionalities.
* Code quality, organization, and readability.
* Proper usage of Django and Django REST framework features.
* Handling multiple image uploads for products.
* Category, sub-category management, and adding products under sub-categories.
* Customer reviews management for products.

# **API Documentation - E-commerce Django REST API**

## **Base URL**

<https://example.com/api/>

## **Endpoints**

### **1. Categories**

#### GET /api/categories/

**Get a list of all categories.**

#### POST /api/categories/

Create a new category.

* Request Body:

{

{  
 "name": "Electronics",  
 "description": "Electronics products"  
}

* Response Body:

{

{  
 "id": 1,  
 "name": "Electronics",  
 "description": "Electronics products"  
}

#### GET /api/categories/{category\_id}/

Get details of a specific category.

#### PUT /api/categories/{category\_id}/

Update an existing category.

* Request Body:

{

{  
 "name": "New Name",  
 "description": "New description"  
}

#### DELETE /api/categories/{category\_id}/

Delete a category.

### **2. Sub-Categories**

#### GET /api/categories/{category\_id}/subcategories/

Get a list of sub-categories under a specific category.

#### POST /api/categories/{category\_id}/subcategories/

Create a new sub-category under a specific category.

* Request Body:

{

{{  
 "name": "Smartphones",  
 "description": "Mobile phones with advanced features"  
}

* Response Body:

{

{  
 "id": 1,  
 "name": "Smartphones",  
 "description": "Mobile phones with advanced features",  
 "category": 1  
}

### **3. Products**

#### GET /api/products/

Get a list of all products.

#### POST /api/products/

Create a new product.

* Request Body:

{{  
 "name": "iPhone 12",  
 "description": "Apple iPhone 12 with A14 Bionic chip",  
 "price": 999,  
 "stock\_quantity": 100,  
 "category": 1,  
 "sub\_category": 1  
}

* Response Body:

{{  
 "id": 1,  
 "name": "iPhone 12",  
 "description": "Apple iPhone 12 with A14 Bionic chip",  
 "price": 999,  
 "stock\_quantity": 100,  
 "category": 1,  
 "sub\_category": 1  
}

#### GET /api/products/{product\_id}/

Get details of a specific product.

#### PUT /api/products/{product\_id}/

Update an existing product.

* Request Body:

{

{  
 "name": "New iPhone",  
 "price": 1099  
}

#### DELETE /api/products/{product\_id}/

Delete a product.

### **4. Customer Reviews**

#### GET /api/products/{product\_id}/reviews/

Get all customer reviews for a specific product.

#### POST /api/products/{product\_id}/reviews/

Add a new customer review for a product.

* Request Body:

{

{  
 "rating": 5,  
 "comment": "Great product!"  
}

* Response Body:

{

{  
 "id": 1,  
 "rating": 5,  
 "comment": "Great product!",  
 "product": 1,  
 "created\_at": "2023-08-07T12:00:00Z"  
}

#### GET /api/products/{product\_id}/reviews/{review\_id}/

Get details of a specific customer review for a product.

#### PUT /api/products/{product\_id}/reviews/{review\_id}/

Update an existing customer review for a product.

* Request Body:

{{

"rating": 4,  
 "comment": "Good product!"  
}

#### DELETE /api/products/{product\_id}/reviews/{review\_id}/

Delete a customer review for a product.

## **Error Handling**

* 400 Bad Request: Invalid request data or missing required fields.
* 404 Not Found: Resource not found.
* 405 Method Not Allowed: Request method not supported for the endpoint.
* 500 Internal Server Error: Unexpected server error.

## **Response Format**

Responses will be in JSON format.

## **Rate Limiting**

There is no rate limiting applied to this API.

## **Pagination**

Collections with multiple items will be paginated with 10 items per page.

Example:

|  |  |  |
| --- | --- | --- |
| **Category** | **Sub-Category** | **Product** |
| Electronics |  |  |
|  | Smartphones |  |
|  |  | iPhone 12 |
|  |  | Samsung Galaxy S21 |
|  | Laptops |  |
|  |  | MacBook Pro |
|  |  | Dell XPS 13 |
| Clothing |  |  |
|  | Men's Clothing |  |
|  |  | T-shirt |
|  |  | Jeans |
|  | Women's Clothing |  |
|  |  | Dress |
|  |  | Skirt |

In this example, we have three main categories: Electronics and Clothing. Under Electronics, we have two subcategories: Smartphones and Laptops. Under Clothing, we have two subcategories: Men's Clothing and Women's Clothing. Under each subcategory, there are specific products listed.