

# E-Commerce API Documentation

## 💡 Overview

This is a production-ready RESTful API for an E-Commerce platform. It features secure authentication, role-based access control, product management, reviews, and image uploads.

**Base URL:** `http://localhost:5000/api`

## 🔒 Authentication

### 1. Register User

Create a new user account.

- **Endpoint:** POST `/auth/register`
- **Body:**

```
{  
  "name": "John Doe",  
  "email": "john@example.com",  
  "password": "securepassword123",  
  "role": "customer" // Optional, default: customer. Use 'admin' for admin privileges.  
}
```

- **Response (201 Created):**

```
{  
  "success": true,  
  "token": "eyJhbGciOiJIUzI1NiIs... ",  
  "data": { "user": { ... } }  
}
```

### 2. Login User

Authenticate existing user and receive access tokens.

- **Endpoint:** POST `/auth/login`
- **Body:**

```
{  
  "email": "john@example.com",  
  "password": "securepassword123"  
}
```

- **Response (200 OK):**

- Returns token in Body.
- Sets jwt HttpOnly Cookie.
- Sets Authorization Header.

### 3. Forgot Password

Request a password reset link (Simulated email).

- **Endpoint:** POST `/auth/forgotPassword`
- **Body:** `{"email": "john@example.com"}`
- **Response:** Check console logs or email inbox (if SMTP configured).

### 4. Reset Password

Reset password using the token received in email.

- **Endpoint:** PATCH `/auth/resetPassword/:token`
- **Body:** `{"password": "newpassword123"}`

## 📦 Products

### 1. Get All Products

Retrieve a list of products with advanced filtering.

- **Endpoint:** GET `/products`
- **Query Parameters:**

- `page`: Page number (default: 1)
- `limit`: Items per page (default: 10)
- `sort`: Sort field (e.g., `price`, `-price` for desc)
- `search`: Search by name or description
- `price[gte]=100`: Filter by price range

- **Example:** `/products?page=1&limit=5&sort=-price&price[gte]=500`

### 2. Get Single Product

- **Endpoint:** GET /products/:id

### 3. Create Product (Admin Only)

Upload images and create a product.

- **Endpoint:** POST /products
- **Headers:**
  - Authorization: Bearer <token>
  - Content-Type: multipart/form-data
- **Body (FormData):**
  - name: "Gaming Laptop"
  - price: 1500
  - category: "Electronics"
  - description: "High-end gaming laptop"
  - stock: 10
  - images: (Select files)

### 4. Add Review

Add a rating and comment to a product.

- **Endpoint:** POST /products/:id/reviews
- **Headers:** Authorization: Bearer <token>
- **Body:**

```
{  
  "rating": 5,  
  "comment": "Amazing product!"  
}
```

## 🔒 Security Features

- **Rate Limiting:** Limits requests per IP to prevent abuse.
- **HPP:** Protects against HTTP Parameter Pollution.
- **MongoSanitize:** Prevents NoSQL Injection attacks.
- **XSS Protection:** Helmet headers set automatically.
- **Validation:** Strict input validation using Zod schemas.

## ⚠ Error Codes

- **400:** Bad Request (Invalid Input, Duplicate Data).
- **401:** Unauthorized (Invalid/Expired Token).
- **403:** Forbidden (Admin access required).
- **404:** Not Found (Resource does not exist).
- **500:** Internal Server Error.