Guitar-Store-Backend

Developer Documentation

Overview

This documentation provides an in-depth look at the backend API for an e-commerce platform. It covers the API views, their functionalities, the models used, and various utilities integrated into the system.

Table of Contents

API Views

CountryAPIView

Description

Provides a list of all countries of origin.

Methods

- **GET**: Retrieves all countries of origin from the database.

Example Request

`GET /api/countries/`

Example Response

`[ { "id": 1, "name": "USA" }, { "id": 2, "name": "Germany" } ]`

PayAPIView

Description

Updates the status of an order to "закрыт" (closed).

Methods

- **POST**: Updates the status of the specified order.

Example Request

`POST /api/pay/ Content-Type: application/json { "order_id": 123 }`

Example Response

`{ "message": "Статус заказа успешно обновлен" }`

SessionAPIView

Description

Handles session-related operations, such as retrieving user session information.

Methods

- **GET**: Retrieves session information based on the session ID.

Example Request

`GET /api/session/?session_id=abc123`

Example Response

`{ "user": { "user_id": 1, "firstname": "John", "lastname": "Doe", "email": "john.doe@example.com", "role": "client" }, "status_code": 200 }`

ShoppingCartAPIView

Description

Manages shopping cart operations, including retrieving and modifying cart items.

Methods

- **GET**: Retrieves the shopping cart items for a specified user.
- **POST**: Adds or removes a product from the shopping cart.

Example Request (GET)

`GET /api/shopping_cart/?id=1`

Example Response (GET)

json

Копировать код

`{ "products": [{ "product_id": 1, "product_name": "Laptop", "product_description": "High-end gaming laptop", "price": 1500.00, "in_stock": true, "vendor": "TechVendor", "country_of_origin": "USA", "year_of_production": 2021, "image_url": "http://example.com/image.jpg", "quantity": 2 }], "status_code": 200 }`

Example Request (POST)

http

Копировать код

`POST /api/shopping_cart/ Content-Type: application/json { "user_id": 1, "product_id": 1, "quantity": 1 }`

Example Response (POST)

`{ "message": "Product added to the shopping cart successfully.", "status_code": 201 }`

UserAPIView

Description

Handles user-related operations such as login and registration.

Methods

- **GET**: Authenticates a user based on email and password, and creates a session.
- **POST**: Registers a new user and creates a session.

Example Request (GET)

`GET /api/user/?email=john.doe@example.com&password=123456`

Example Response (GET)

`{ "user": { "user_id": 1, "firstname": "John", "lastname": "Doe", "email": "john.doe@example.com", "role": "client" }, "session_id": "abc123", "status_code": 200 }`

Example Request (POST)

`POST /api/user/ Content-Type: application/json { "firstname": "John", "lastname": "Doe", "email": "john.doe@example.com", "password": "123456" }`

Example Response (POST)

`{ "user": { "id": 1, "firstname": "John", "lastname": "Doe", "email": "john.doe@example.com", "role": "client" }, "session_id": "abc123", "message": "User created successfully", "status_code": 201 }`

UserOrdersView

Description

Manages user orders, including retrieving and creating orders.

Methods

- **GET**: Retrieves orders for a specified user.
- **POST**: Creates a new order for the specified user and products.

Example Request (GET)

http

Копировать код

`GET /api/user_orders/?user_id=1`

Example Response (GET)

`[ { "order_id": 1, "user_id": 1, "order_status": "открыт", "order_date": "2024-05-21", "total_price": 1500.00, "products": [ { "product_id": 1, "product_name": "Laptop", "quantity": 2 } ] } ]`

Example Request (POST)

`POST /api/user_orders/ Content-Type: application/json { "user_id": 1, "products": [ { "product_id": 1, "quantity": 2 } ] }`

Example Response (POST)

`{ "order_id": 1, "user_id": 1, "order_status": "открыт", "order_date": "2024-05-21", "total_price": 3000.00, "products": [ { "product_id": 1, "quantity": 2 } ] }`

VendorAPIView

Description

Provides a list of all vendors.

Methods

- **GET**: Retrieves all vendors from the database.

Example Request

http

Копировать код

`GET /api/vendors/`

Example Response

json

Копировать код

`[ { "id": 1, "name": "TechVendor" }, { "id": 2, "name": "FoodVendor" } ]`

WishlistAPIView

Description

Manages wishlist operations, including retrieving and modifying wishlist items.

Methods

- **GET**: Retrieves the wishlist items for a specified user.
- **POST**: Adds or removes a product from the wishlist.

Example Request (GET)

http

Копировать код

`GET /api/wishlist/?id=1`

Example Response (GET)

json

Копировать код

`{ "products": [ { "product_id": 1, "product_name": "Laptop", "product_description": "High-end gaming laptop", "price": 1500.00, "in_stock": true, "vendor": "TechVendor", "country_of_origin": "USA", "year_of_production": 2021, "image_url": "http://example.com/image.jpg" } ], "status_code": 200 }`

Example Request (POST)

`POST /api/wishlist/ Content-Type: application/json { "user_id": 1, "product_id": 1 }`

Example Response (POST)

`{ "message": "Product added to the wishlist successfully.", "status_code": 201 }`

Models

User

Description

Represents a user in the system.

Fields

- `user_id`: Auto-incrementing primary key.
- `first_name`: First name of the user.
- `last_name`: Last name of the user.

- `email`: Email address of the user (unique).
- `password`: Password of the user.
- `role`: Role of the user (choices: 'admin', 'client').

Methods

- `__str__()`: Returns the full name of the user.

ProductCategory

Description

Represents a category of products.

Fields

- `category_id`: Auto-incrementing primary key.
- `category_name`: Name of the category (unique).

Methods

- `__str__()`: Returns the name of the category.

Vendor

Description

Represents a vendor in the system.

Fields

- `vendor_id`: Auto-incrementing primary key.
- `vendor_name`: Name of the vendor (unique).

Methods

- `__str__()`: Returns the name of the vendor.

CountryOfOrigin

Description

Represents a country of origin for products.

Fields

- `country_id`: Auto-incrementing primary key.
- `country_name`: Name of the country (unique).

Methods

- `__str__()`: Returns the name of the country.

Product

Description

Represents a product in the system.

Fields

- `product_id`: Auto-incrementing primary key.
- `product_name`: Name of the product.
- `product_description`: Description of the product.
- `price`: Price of the product.
- `in_stock`: Boolean indicating if the product is in stock.
- `category`: Foreign key to `ProductCategory`.
- `vendor`: Foreign key to `Vendor`.
- `country_of_origin`: Foreign key to `CountryOfOrigin`.
- `year_of_production`: Year the product was produced.
- `image_url`: URL of the product's image.

Methods

- `__str__()`: Returns the name of the product.

Order

Description

Represents an order placed by a user.

Fields

- `order_id`: Auto-incrementing primary key.
- `user`: Foreign key to `User`.
- `order_status`: Status of the order (choices: 'открыт', 'закрыт').
- `order_date`: Date the order was placed.
- `total_price`: Total price of the order.

Methods

- `__str__()`: Returns the ID and status of the order.

OrderProduct

Description

Represents a product within an order.

Fields

- `order_product_id`: Auto-incrementing primary key.
- `order`: Foreign key to `Order`.
- `product`: Foreign key to `Product`.
- `quantity`: Quantity of the product in the order.

Methods

- `__str__()`: Returns the ID of the order product.

ShoppingCart

Description

Represents a shopping cart for a user.

Fields

- `cart_id`: Auto-incrementing primary key.
- `user`: Foreign key to `User`.
- `product`: Foreign key to `Product`.
- `quantity`: Quantity of the product in the cart.

Methods

- `__str__()`: Returns the ID of the cart.

Wishlist

Description

Represents a wishlist for a user.

Fields

- `wishlist_id`: Auto-incrementing primary key.
- `user`: Foreign key to `User`.
- `product`: Foreign key to `Product`.

Methods

- `__str__()`: Returns the ID of the wishlist.

Session

Description

Represents a user session in the system.

Fields

- `session_id`: Auto-incrementing primary key.
- `session_key`: Unique session key.
- `user`: Foreign key to `User`.

Methods

- `__str__()`: Returns the session key.

Utilities

Generate Unique Session ID

Description

Generates a unique session ID for user sessions.

Usage

python

Копировать код

```
from your_module import generate_unique_session_id session_id =
generate_unique_session_id()
```

Logging

Description

Provides logging functionalities for the application.

Usage

python

Копировать код

```
import logging # Create a logger logger = logging.getLogger(__name__) # Log an error message logger.error("This is an error message")
```

Error Handling

All API views should handle errors gracefully and return appropriate HTTP status codes along with descriptive error messages.

Example Error Response

json

Копировать код

```
{ "error": "Invalid request", "status_code": 400 }
```

Common Error Codes

- `400`: Bad Request
- `401`: Unauthorized
- `403`: Forbidden
- `404`: Not Found
- `500`: Internal Server Error