Authentication API Documentation

Overview

Documentation for authentication endpoints

Version information

Version : 1.0.0

Tags

- Auth : Endpoints related to user authentication
- Brand : Endpoints related to brand management
- Category : Endpoints related to category management
- Coupon : Endpoints related to coupon management
- Product : Endpoints related to product management

Paths

Create brand

```
POST /api/brands
```

Description

Creates a new brand

Parameters

Type	Name	Description	Schema
Body	**body** *required*	Brand name and description	body

body

Name	Schema
description *optional*	string

Name	Schema
name *optional*	string

Responses

HTTP Code	Description	Schema
201	Created	No Content
422	Unprocessable Entity	Response 422
500	Internal Server Error	Response 500

Response 422

Name	Schema
error *optional*	string

Response 500

Name	Schema
error *optional*	string

Tags

- Brand

Get brands

```
GET /api/brands
```

Description

Retrieves a list of brands

Parameters

Type	Name	Description	Schema
Query	**max_items** *optional*	Maximum items per page	integer

Type	Name	Description	Schema
Query	**page** *optional*	Page number	integer

Responses

HTTP Code	Description	Schema
200	OK	Response 200
500	Internal Server Error	Response 500

Response 200

Name	Schema
data *optional*	< data > array

data

Name	Schema
description *optional*	string
id *optional*	integer
name *optional*	string

Response 500

Name	Schema
error *optional*	string

Tags

- Brand

Get brand by ID

```
GET /api/brands/{id}
```

Description

Retrieves a brand by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Brand ID	integer

Responses

HTTP Code	Description	Schema
200	OK	Response 200
500	Internal Server Error	Response 500

Response 200

Name	Schema
description *optional*	string
id *optional*	integer
name *optional*	string

Response 500

Name	Schema
error *optional*	string

Tags

- Brand

Update brand

```
PUT /api/brands/{id}
```

Description

Updates a brand by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Brand ID	integer
Body	**body** *required*	New brand name and description	body

body

Name	Schema
description *optional*	string
name *optional*	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Brand

Delete brand

```
DELETE /api/brands/{id}
```

Description

Deletes a brand by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Brand ID	integer

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Brand

Create category

```
POST /api/categories
```

Description

Creates a new category

Parameters

Type	Name	Description	Schema
FormData	**description** *required*	Category description	string
FormData	**image** *required*	Category image file	file

Type	Name	Description	Schema
FormData	name *required*	Category name	string

Responses

HTTP Code	Description	Schema
201	Created	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Consumes

- multipart/form-data

Tags

- Category

Get categories

```
GET /api/categories
```

Description

Retrieves a list of categories

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Category

Get category by ID

```
GET /api/categories/{id}
```

Description

Retrieves a category by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Category ID	integer

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Category

Update category

```
PUT /api/categories/{id}
```

Description

Updates a category by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Category ID	integer
FormData	**description** *required*	New category description	string
FormData	**image** *required*	New category image file	file
FormData	**name** *required*	New category name	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Consumes

- `multipart/form-data`

Tags

- Category

Delete category

```
DELETE /api/categories/{id}
```

Description

Deletes a category by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Category ID	integer

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Category

Create coupon

```
POST /api/coupons
```

Description

Creates a new coupon

Parameters

Type	Name	Description	Schema
Body	**body** *required*	Coupon details	body

body

Name	Schema
couponCode *optional*	string
date_end *optional*	string (date)
date_start *optional*	string (date)
id_product *optional*	integer
percentage *optional*	number
type *optional*	enum (product, order)

Responses

HTTP Code	Description	Schema
201	Created	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Coupon

Get coupon by ID

```
GET /api/coupons/{id}
```

Description

Retrieves a coupon by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Coupon ID	integer

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Coupon

Update coupon

```
PUT /api/coupons/{id}
```

Description

Updates a coupon by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Coupon ID	integer
Body	**body** *required*	Updated coupon details	body

body

Name	Schema
couponCode *optional*	string

Name	Schema
date_end *optional*	string (date)
date_start *optional*	string (date)
id_product *optional*	integer
percentage *optional*	number

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Coupon

Delete coupon

```
DELETE /api/coupons/{id}
```

Description

Deletes a coupon by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Coupon ID	integer

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Coupon

Create product

```
POST /api/products
```

Description

Creates a new product

Parameters

Type	Name	Description	Schema
FormData	**brand_id** *required*	Brand ID of the product	integer
FormData	**category_id** *required*	Category ID of the product	integer
FormData	**description** *required*	Product description	string
FormData	**image** *optional*	Product image	file
FormData	**name** *required*	Product name	string
FormData	**price** *required*	Product price	number
FormData	**priceBarre** *required*	Product barre price	number

Type	Name	Description	Schema
FormData	**product_type_id** *required*	Product type ID	integer

Responses

HTTP Code	Description	Schema
201	Created	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Consumes

- multipart/form-data

Tags

- Product

Get products

```
GET /api/products
```

Description

Retrieves products based on filtering criteria

Parameters

Type	Name	Description	Schema
Query	**brandIds** *optional*	Brand IDs	< integer > array
Query	**categoryIds** *optional*	Category IDs	< integer > array
Query	**max_items** *optional*	Maximum items per page	integer

Type	Name	Description	Schema
Query	**page** *optional*	Page number	integer
Query	**priceMax** *optional*	Maximum price	number
Query	**priceMin** *optional*	Minimum price	number

Responses

HTTP Code	Description	Schema
200	OK	Response 200
500	Internal Server Error	Response 500

Response 200

Name	Schema
data *optional*	< data > array

data

Name	Schema
brand_id *optional*	integer
category_id *optional*	integer
description *optional*	string
id *optional*	integer
image_path *optional*	string
image_url *optional*	string
name *optional*	string
price *optional*	number

Name	Schema
priceBarre *optional*	number
product_type_id *optional*	integer

Response 500

Name	Schema
error *optional*	string

Tags

- Product

Get product by ID

```
GET /api/products/{id}
```

Description

Retrieves product details by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Product ID	integer

Responses

HTTP Code	Description	Schema
200	OK	Response 200
500	Internal Server Error	Response 500

Response 200

Name	Schema
brand_id *optional*	integer

Name	Schema
category_id *optional*	integer
description *optional*	string
id *optional*	integer
image_path *optional*	string
image_url *optional*	string
name *optional*	string
price *optional*	number
priceBarre *optional*	number
product_type_id *optional*	integer

Response 500

Name	Schema
error *optional*	string

Tags

- Product

Update product

```
PUT /api/products/{id}
```

Description

Updates an existing product by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Product ID	integer
FormData	**brand_id** *required*	Brand ID of the product	integer
FormData	**category_id** *required*	Category ID of the product	integer
FormData	**description** *required*	Product description	string
FormData	**image** *optional*	Product image	file
FormData	**name** *required*	Product name	string
FormData	**price** *required*	Product price	number
FormData	**priceBarre** *required*	Product barre price	number
FormData	**product_type_id** *required*	Product type ID	integer

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Consumes

- `multipart/form-data`

Tags

- Product

Delete product

```
DELETE /api/products/{id}
```

Description

Deletes a product by its ID

Parameters

Type	Name	Description	Schema
Path	**id** *required*	Product ID	integer

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Product

Initiate Google OAuth 2.0 authentication

```
GET /auth/google
```

Description

Initiates Google OAuth 2.0 authentication flow

Responses

HTTP Code	Description	Schema
200	OK	No Content

Tags

- Auth

Check if user is logged in

```
GET /auth/isloggedin
```

Description

Checks if user is logged in with a session cookie

Responses

HTTP Code	Description	Schema
200	OK	No Content
403	Forbidden	Response 403
500	Internal Server Error	Response 500

Response 403

Name	Schema
error *optional*	string

Response 500

Name	Schema
error *optional*	string

Tags

- Auth

User login

```
POST /auth/login
```

Description

Logs in a user with email and password

Parameters

Type	Name	Description	Schema
Body	**body** *required*	Email and password of the user	body

body

Name	Schema
email *optional*	string
password *optional*	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
403	Forbidden	Response 403
422	Unprocessable Entity	Response 422
500	Internal Server Error	Response 500

Response 403

Name	Schema
error *optional*	string

Response 422

Name	Schema
error *optional*	string

Response 500

Name	Schema
error *optional*	string

Tags

- Auth

User logout

```
GET /auth/logout
```

Description

Logs out the user and clears the session cookie

Responses

HTTP Code	Description	Schema
200	OK	No Content
500	Internal Server Error	Response 500

Response 500

Name	Schema
error *optional*	string

Tags

- Auth

User sign up

```
POST /auth/sign-up
```

Description

Registers a new user with full name, phone number, email, and password

Parameters

Type	Name	Description	Schema
Body	**body** *required*	Full name, phone number, email, and password of the user	body

body

Name	Schema
email *optional*	string
fullName *optional*	string
numberPhone *optional*	string
password *optional*	string

Responses

HTTP Code	Description	Schema
200	OK	Response 200
422	Unprocessable Entity	Response 422
500	Internal Server Error	Response 500

Response 200

Name	Schema
message *optional*	string

Response 422

Name	Schema
error *optional*	string

Response 500

Name	Schema
error *optional*	string

- Auth

Verify user sign-up

```
GET /auth/sign-up/verify
```

Description

Verifies user sign-up with a verification token

Parameters

Type	Name	Description	Schema
Query	t *required*	Verification token	string

Responses

HTTP Code	Description	Schema
201	Created	Response 201
422	Unprocessable Entity	Response 422
500	Internal Server Error	Response 500

Response 201

Name	Schema
message *optional*	string

Response 422

Name	Schema
error *optional*	string

Response 500

Name	Schema
error *optional*	string

Tags

- Auth