

ECOMMERCE API

ALBY REJI

BE TRaine

INTRODUCTION

This e-commerce project presents a dynamic platform designed for administrators, registered users, and non-registered users. Utilizing Django's framework, the project delivers features including user registration, product management, order processing, and communication.

Admins are equipped with tools to manage product categories and products, oversee user accounts, process orders, and promote products via email campaigns. Authentication and permissions ensure secure access to admin-exclusive features.

Registered users can register, reset their passwords, create, and manage their addresses and profiles, view categories and products, add items to their shopping carts, place orders, and access their order history. The shopping experience is enhanced with filtering, pagination, and the ability to receive email confirmations for order placements.

Non-registered users can browse product categories, view product listings, and access detailed product information. This allows users to explore the platform without the need for registration.

The system implements RESTful APIs secured with JWT-based authentication, facilitating seamless communication between the frontend and backend. Additionally, email notifications are sent to users upon registration, password reset, and order placement, enhancing user engagement and interaction.

ADMIN MODULE

Admin Login

[Quiz App](#) / <http://localhost:8000/login/>

Save

Send

POSThttp://localhost:8000/login/

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	email	admin@gmail.com			
<input checked="" type="checkbox"/>	password	admin			
	Key	Value	Description		

BodyCookiesHeaders (10)Test Results

Status: 200 OKTime: 495 msSize: 794 BSave as Example

PrettyRawPreviewVisualizeJSON

```
1{"refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.\n2  eyJ0b2t1b190eXB1IjoicmVmcnVzaCI6ImV4cCI6MTY5MjUwMjA5MSwiaWF0IjoxNjkyNDE1NjksLCJqdGkiOiJjZjEzNDI3OGY0MGE0OTYyOTUwYzYk40GM2ZTNkZGRhNSIsI\n3  nVzZXJfaWQ10jF9.chIu9Sg0sL0qWY9Clwo14yBC00g-bCEdH5H6cxnEh_Y",\n4  "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.\n  eyJ0b2t1b190eXB1IjoicmVmcnVzaCI6ImV4cCI6MTY5MjUwMjA5MSwiaWF0IjoxNjkyNDE1NjksLCJqdGkiOiJjZjEzNDI3OGY0MGE0OTYyOTUwYzYk40GM2ZTNkZGRhNSIsI\n  XN1cl9pZCI6MX0.gq8M3VulJHx16T7_4KQYq3sk3P9SjZnpqI-eFU2qX6w"}
```

Category Management

- Category Create View

This API allows an admin user to create a new category for products. Authentication is required using JWT, and admin user permissions are necessary.

[Quiz App](#) / <http://localhost:8000/admin/category/create/>

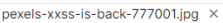
Save

Send

POSThttp://localhost:8000/admin/category/create/

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	category_name	Laptops			
<input checked="" type="checkbox"/>	image				
	Key	Value	Description		

BodyCookiesHeaders (10)Test Results

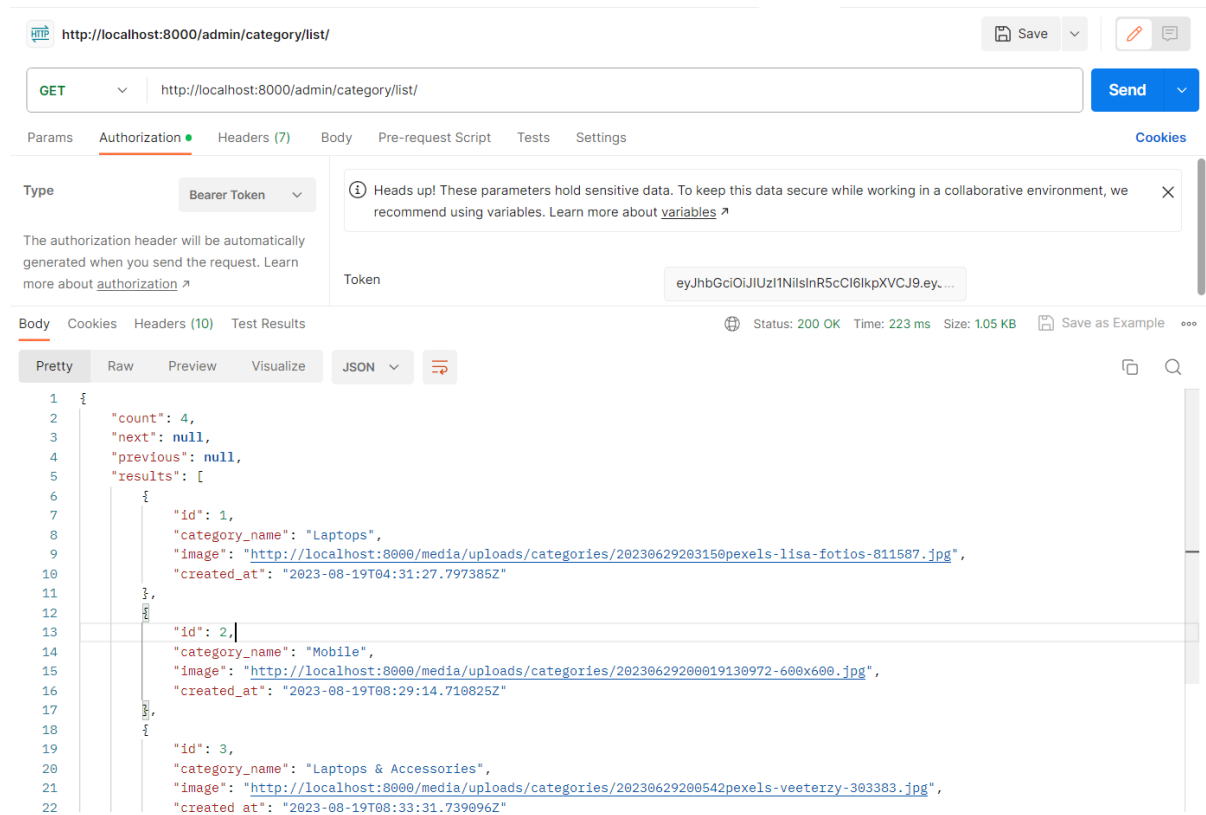
Status: 200 OKTime: 1203 msSize: 341 BSave as Example

PrettyRawPreviewVisualizeJSON

```
1{"Category Added Successfully"}\n2\n3
```

- Category List View

This API lists all existing categories. It requires authentication and admin user permissions. Pagination is implemented to manage a potentially extensive list of categories.



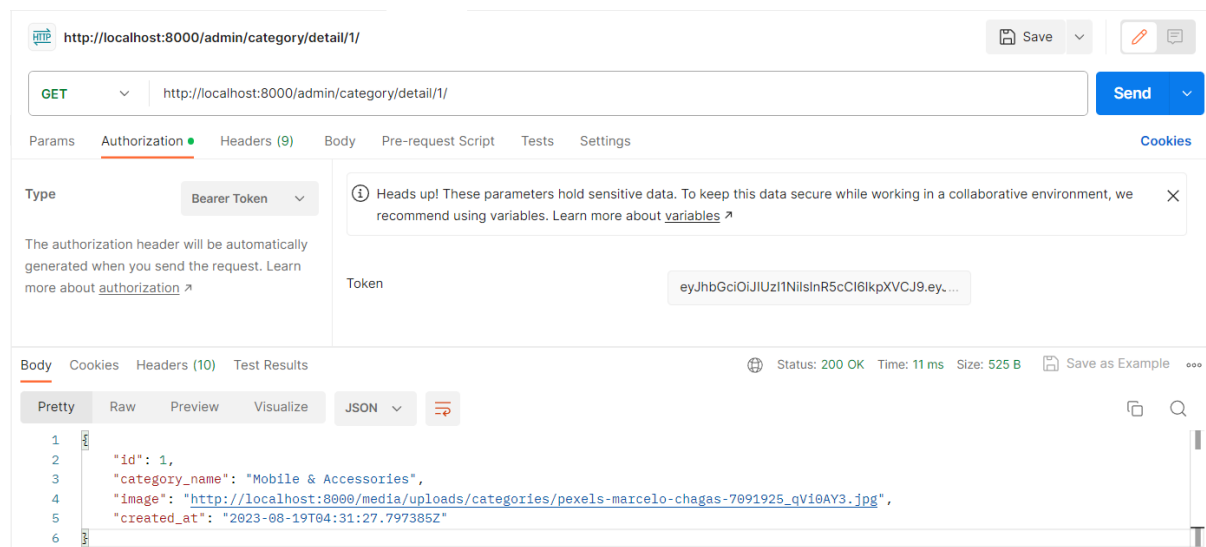
The screenshot shows a REST client interface with the URL `http://localhost:8000/admin/category/list/` and the method `GET`. The response status is `200 OK` with a time of `223 ms` and a size of `1.05 KB`. The response body is displayed in JSON format, showing a list of categories with pagination information.

```
{
  "count": 4,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "category_name": "Laptops",
      "image": "http://localhost:8000/media/uploads/categories/20230629203150pexels-lisa-fotios-811587.jpg",
      "created_at": "2023-08-19T04:31:27.797385Z"
    },
    {
      "id": 2,
      "category_name": "Mobile",
      "image": "http://localhost:8000/media/uploads/categories/20230629200019130972-600x600.jpg",
      "created_at": "2023-08-19T08:29:14.710825Z"
    },
    {
      "id": 3,
      "category_name": "Laptops & Accessories",
      "image": "http://localhost:8000/media/uploads/categories/202306292000542pexels-veeterzy-303383.jpg",
      "created_at": "2023-08-19T08:33:31.739096Z"
    }
  ]
}
```

- Category Detail View

This API allows viewing, updating, and deleting a specific category. It necessitates authentication and admin user permissions.

- Category View



The screenshot shows a REST client interface with the URL `http://localhost:8000/admin/category/detail/1/` and the method `GET`. The response status is `200 OK` with a time of `11 ms` and a size of `525 B`. The response body is displayed in JSON format, showing the details of a single category.

```
{
  "id": 1,
  "category_name": "Mobile & Accessories",
  "image": "http://localhost:8000/media/uploads/categories/pexels-marcelo-chagas-7091925_qVi0AY3.jpg",
  "created_at": "2023-08-19T04:31:27.797385Z"
}
```

Category Update

http://localhost:8000/admin/category/detail/1/

Save

PUT

http://localhost:8000/admin/category/detail/1/

Send

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> category_name	Mobile & Accessories			
<input checked="" type="checkbox"/> image	pexels-marcelo-chagas-7091925_qVi0AY3.jpg			
Key	Value	Description		

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 269 ms

Size: 525 B

Save as Example

...

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": 1,
3   "category_name": "Mobile & Accessories",
4   "image": "http://localhost:8000/media/uploads/categories/pexels-marcelo-chagas-7091925_qVi0AY3.jpg",
5   "created_at": "2023-08-19T04:31:27.797385Z"
6 }
```

Category delete

http://localhost:8000/admin/category/detail/5/

Save

DELETE

http://localhost:8000/admin/category/detail/5/

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 102 ms

Size: 379 B

Save as Example

...

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "message": "Category deleted successfully."
3 }
```

Product Management

- Product Create View

This API allows an admin user to create a new product. Authentication is required using JWT, and admin user permissions are necessary.

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/admin/product/create/`. The request body is a form with the following data:

Key	Value	Description
<input checked="" type="checkbox"/> product_name	Dell Inspiron 15 Laptop	
<input checked="" type="checkbox"/> description	15.6" Full HD Display, Intel Core i5 Processor, 8GB RAM...	
<input checked="" type="checkbox"/> price	50000	
<input checked="" type="checkbox"/> quantity	10	
<input checked="" type="checkbox"/> image	20230629200542pexels-veeterzy-303383.jpg	
<input checked="" type="checkbox"/> categories	1	

The response is a JSON object: `{"Product Added Successfully"}`. The status is 200 OK, time is 215 ms, and size is 340 B.

- Product List View

This API lists all existing categories. It requires authentication and admin user permissions. Pagination is implemented to manage a potentially extensive list of categories.

- Product list

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/admin/product/list/`. The request is authenticated with a Bearer Token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1Ijoi...`. The response is a JSON object with pagination information and a list of products:

```
1 {
2   "count": 4,
3   "next": null,
4   "previous": null,
5   "results": [
6     {
7       "id": 1,
8       "product_name": "Dell Inspiron 15 Laptop",
9       "description": "15.6\" Full HD Display, Intel Core i5 Processor, 8GB RAM, 256GB SSD, Windows 10",
10      "price": "50000.00",
11      "quantity": 10,
12      "image": "http://localhost:8000/media/uploads/products/20230629200542pexels-veeterzy-303383.jpg",
13      "categories": [
14        1
15      ],
16      "created_at": "2023-08-19T08:58:39.468978Z"
17    }
18  ]
19 }
```

The status is 200 OK, time is 16 ms, and size is 1.63 KB.

- Product Detail View

This API allows viewing, updating, and deleting a specific product. It requires authentication and admin user permissions.

- Product

view

http://localhost:8000/admin/product/detail/3/

Save

GET

http://localhost:8000/admin/product/detail/3/

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> price	50000		
Key	Value	Description	

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 12 ms

Size: 671 B

Save as Example

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "id": 3,
3    "product_name": "Logitech Wireless Keyboard and Mouse",
4    "description": "Reliable 2.4GHz Connection, Quiet Typing, Long Battery Life",
5    "price": "2000.00",
6    "quantity": 10,
7    "image": "http://localhost:8000/media/uploads/products/COSMIC-BYTE-CB-GK-17-GALACTIC-BLACK_wq5AfDj.jpg",
8    "categories": [
9      1,
10     3
11   ],
12   "created_at": "2023-08-19T09:01:58.052379Z"
13 }
```

- Product Update

http://localhost:8000/admin/product/detail/1/

Save

PATCH

http://localhost:8000/admin/product/detail/1/

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> price	55000		
Key	Value	Description	

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 103 ms

Size: 670 B

Save as Example

Pretty

Raw


Preview

Visualize



JSON

```
1  {
2    "id": 1,
3    "product_name": "Dell Inspiron 15 Laptop",
4    "description": "15.6\" Full HD Display, Intel Core i5 Processor, 8GB RAM, 256GB SSD, Windows 10",
5    "price": "55000.00",
6    "quantity": 10,
7    "image": "http://localhost:8000/media/uploads/products/20230629200542pexels-veeterzy-303383.jpg",
8    "categories": [
9      1
10     ],
11   "created_at": "2023-08-19T08:58:39.468978Z"
12 }
```

○ Product delete

 http://localhost:8000/admin/product/detail/5/

Save



DELETE

http://localhost:8000/admin/product/detail/5/

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Type

Bearer Token

ⓘ

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

×

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 142 ms

Size: 378 B

Save as Example

Pretty

Raw

Preview

Visualize

JSON

1

2

3

⌵

"message": "Product deleted successfully."

⌵

User Management

- User List View

This API lists all non-admin users. It requires authentication and admin user permissions. It includes pagination.

The screenshot shows a REST client interface for a GET request to `http://localhost:8000/admin/users/list/`. The request is authenticated with a Bearer Token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`. The response is a JSON object with the following structure:

```
1 {
2   "count": 3,
3   "next": null,
4   "previous": null,
5   "results": [
6     {
7       "id": 2,
8       "username": "Amal",
9       "email": "amal@gmail.com",
10      "first_name": "Amal",
11      "last_name": "R"
12    },
13    {
14      "id": 3,
15      "username": "Anoop",
16      "email": "anoop@gmail.com",
17      "first_name": "Anoop",
18      "last_name": "M"
19    }
20  ]
21 }
```

- User Delete View

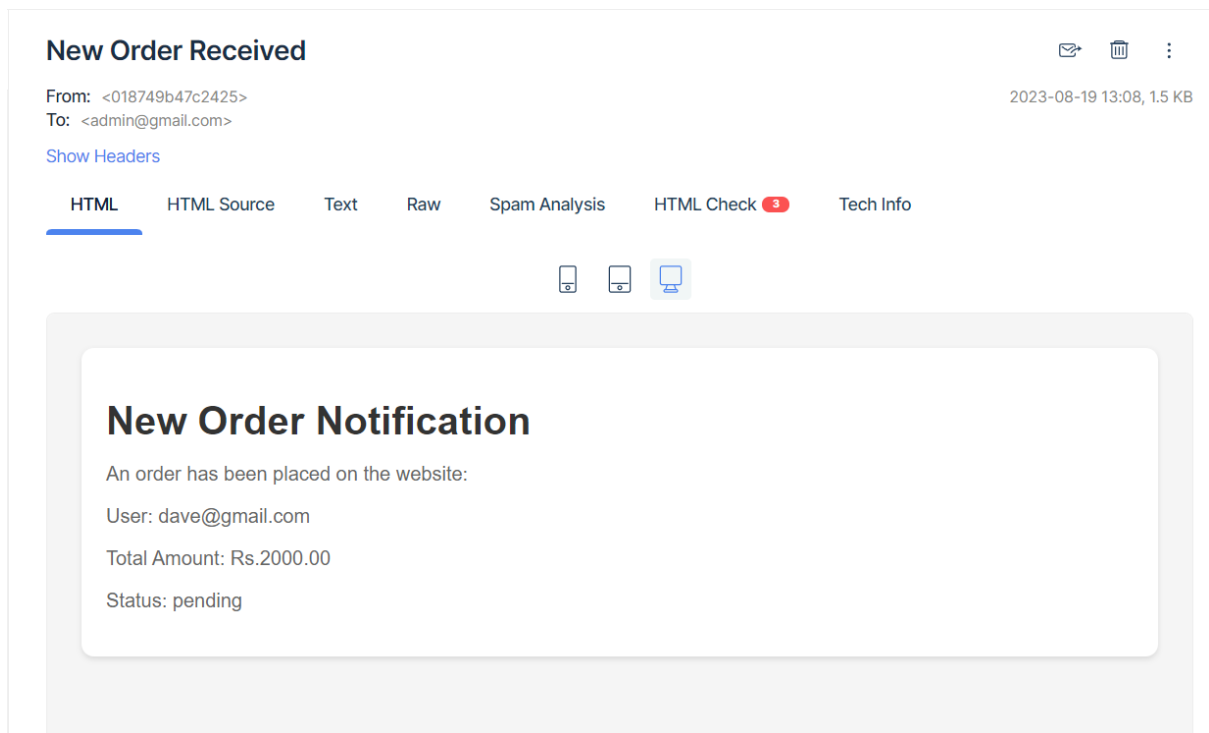
This API enables an admin user to delete a specific user. Authentication and admin user permissions are required.

The screenshot shows a REST client interface for a DELETE request to `http://localhost:8000/admin/users/delete/5/`. The request is authenticated with the same Bearer Token as the previous screenshot. The response is a JSON object with the following structure:

```
1 {
2   "message": "User deleted successfully."
3 }
```

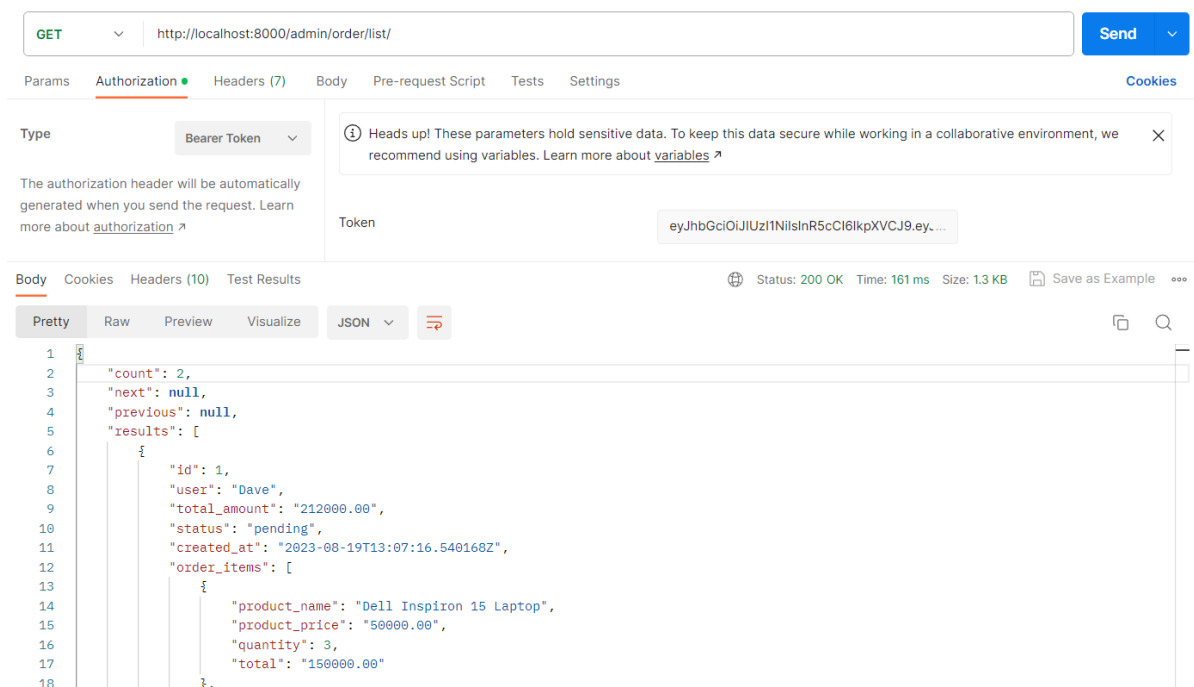
Order Management

- When a user places an order, the admin receives an email



- Order List View

This API displays a list of all orders. It demands authentication and admin user permissions. Pagination is implemented to manage a potentially extensive list of orders.



- Order Detail View

This API allows the viewing of a specific order along with its associated order items. It necessitates authentication and admin user permissions.

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/admin/order/list/2/`. The request is authenticated with a Bearer Token. The response is a JSON object with the following structure:

```
1 {
2   "id": 2,
3   "user": "Dave",
4   "total_amount": "2000.00",
5   "status": "pending",
6   "created_at": "2023-08-19T13:08:21.403047Z",
7   "order_items": [
8     {
9       "product": "Logitech Wireless Keyboard and Mouse",
10      "quantity": 1,
11      "total": 2000.0
12    }
13  ],
14  "shipping_address": {
15    "id": 1,
16    "user": "Dave",
17    "house_name": "daves House",
18    "city": "Kottayam",
19    "state": "Kerala",
20    "postal_code": "12345",
21    "country": "India"
22  }
23 }
```

- Order Update View

This API allows the update of an order's status and sends an email confirmation to the user. Authentication and admin user permissions are required.

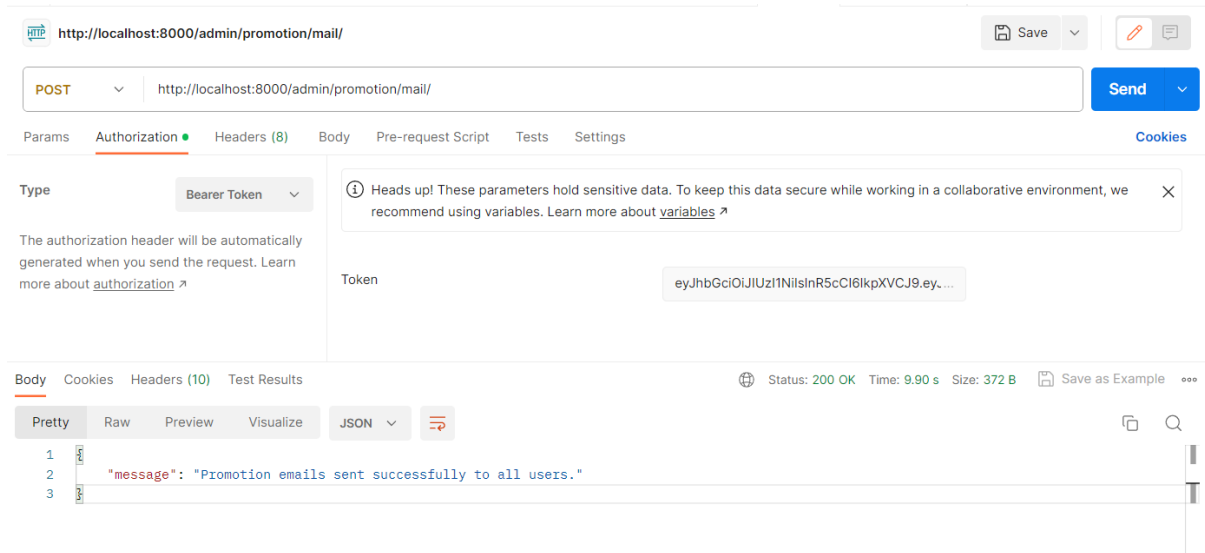
The screenshot shows a REST client interface with a PUT request to `http://localhost:8000/admin/order/update/2/`. The request body is `{ "status": "shipped" }`. The response is a JSON object with a message:

```
1 {
2   "message": "Order status is updated"
3 }
```

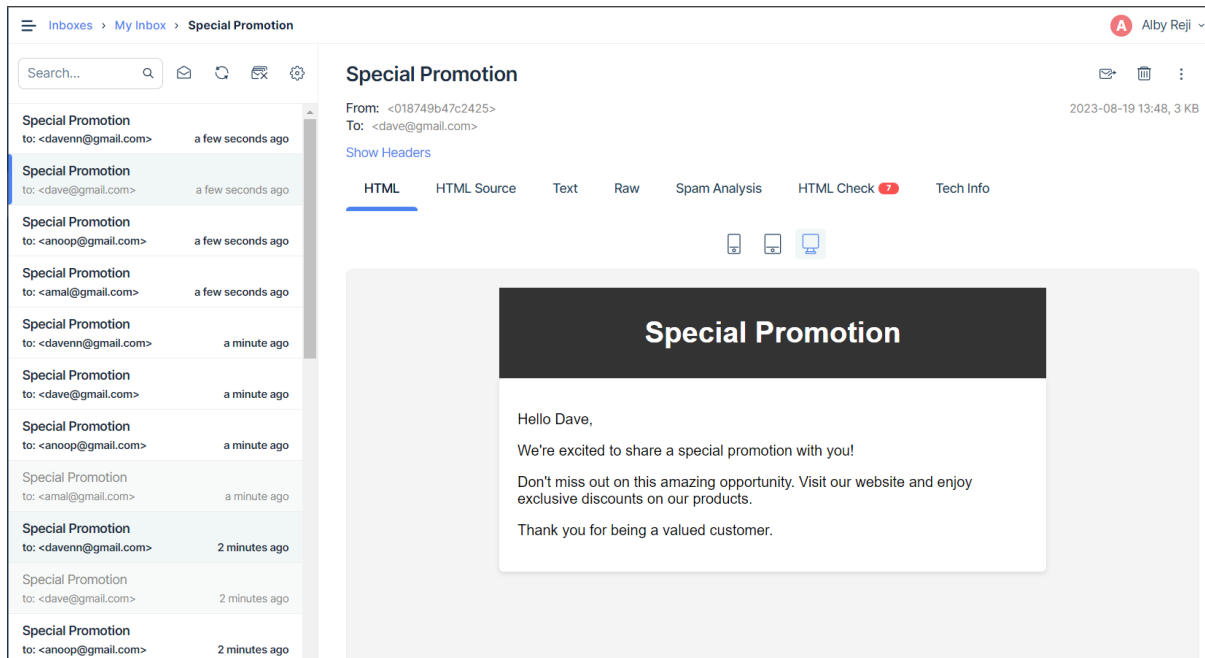
Promotion Email

- Send Promotion Email View

This API permits admin to send promotion emails to all registered users in this system. Authentication and admin user permissions are needed. Django's email functionality is utilized to send HTML emails.



- Promotion Mail with html



USER MODULE

Registration and Authentication View

- Registration

This API allows users to register by sending a registration confirmation email after a successful registration.

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8000/register/`
- Method:** `POST`
- Body (JSON):**

```
{  "username": "Amal",  "email": "amal@gmail.com",  "first_name": "Amal",  "last_name": "R",  "password": "amal123",  "password2": "amal123"}
```
- Status:** `200 OK`, Time: `3.73 s`, Size: `353 B`
- Response Body (JSON):**

```
{  "response": "message:User created"}
```

- Registration Mail

The screenshot shows an email interface with the following details:

- Sender:** Alby Reji
- Subject:** User Registration
- From:** `<018749b47c2425>`
- To:** `<dave@gmail.com>`
- Date:** 2023-08-19 03:23, 1.6 KB
- Content Type:** HTML
- Body:**

Registration Successful

Congratulations **Dave**.... Your registration on our website was successful. We are thrilled to welcome you to our community.

- Login

Quiz App / http://localhost:8000/login/

Save

POST

http://localhost:8000/login/

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	email	dave@gmail.com			
<input checked="" type="checkbox"/>	password	dave123			
	Key	Value	Description		

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 632 ms

Size: 794 B

Save as Example

Pretty

Raw

Preview

Visualize

JSON

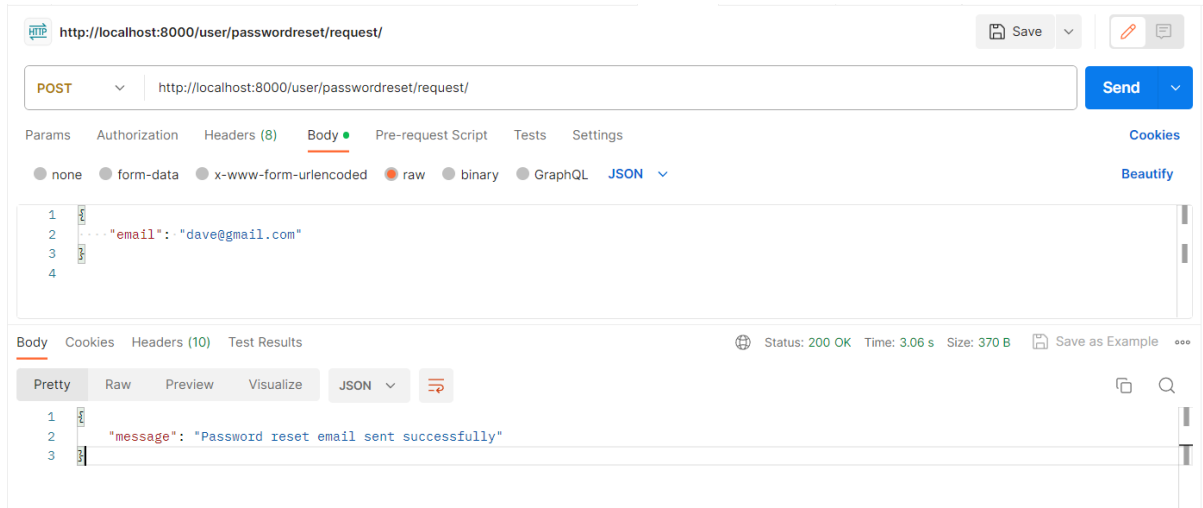
```

1  [ref]
2  "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b190eXB1IjoicmVmcVzaCIsImV4cCI6MTY5MjUwMTg2OjI1IiwiaWF0IjoxNjkyNDE1NDY4LjQqdGkiOiIiZmVhY2I3MThlYzE0ZjE4OWVlYWNhNmIyZyZg5MWJjMyIsInVzZXJfawQ1OjR9.lwn195fDrGKW2nAjumMMgtD3hh_U50CHOTfYQugdk2k",
3  "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b190eXB1IjoicmVmcVzaCIsImV4cCI6MTY5MjUwMTg2OjI1IiwiaWF0IjoxNjkyNDE1NDY4LjQqdGkiOiIiZmVhY2I3MThlYzE0ZjE4OWVlYWNhNmIyZyZg5MWJjMyIsInVzZXJfawQ1OjR9.lwn195fDrGKW2nAjumMMgtD3hh_U50CHOTfYQugdk2k"
4  [ref]
```

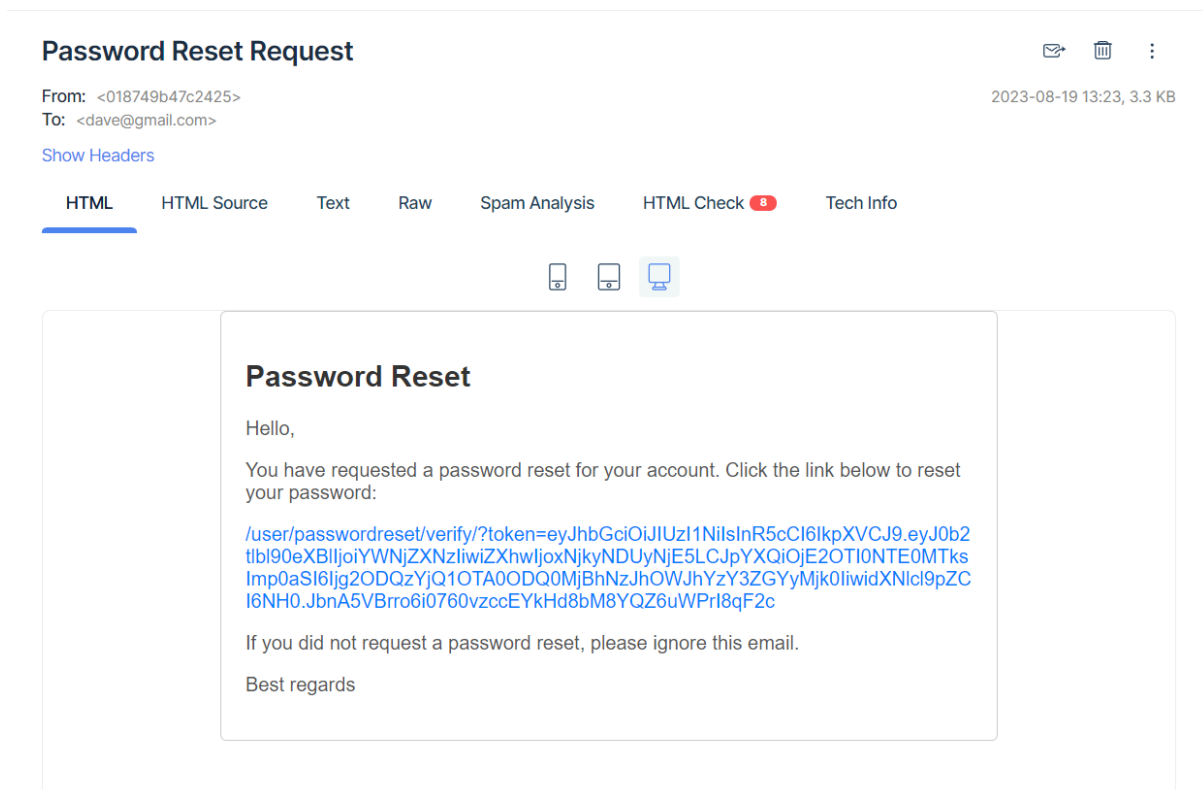
Password Reset

- Password Reset Request View

This API handles requests for resetting a user's password and sends a reset email containing a token generated using the user's access token.



- Password reset Mail



- Password Reset Verify View

This API verifies whether the provided password reset token is valid.

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/user/passwordreset/verify/?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b90eXBlljoiYWVjZWNjZXRnZliwiZXhwIj...`. The request is sent, and the response is displayed in the Body tab as a JSON object: `{ "message": "Token is valid" }`. The status is 200 OK, and the response size is 351 B.

Key	Value	Description
token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b90eXBlljoiYWVjZWNjZXRnZliwiZXhwIj...	

```
1 {
2   "message": "Token is valid"
3 }
```

- Password Reset View

This API enables users to reset their passwords using a valid reset token.

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/user/passwordreset/reset/?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b90eXBlljoiYWVjZWNjZXRnZliwiZXhwIj...`. The request body is `{ "password": "dave456", "confirm_password": "dave456" }`. The request is sent, and the response is displayed in the Body tab as a JSON object: `{ "message": "Password reset successfully" }`. The status is 200 OK, and the response size is 359 B.

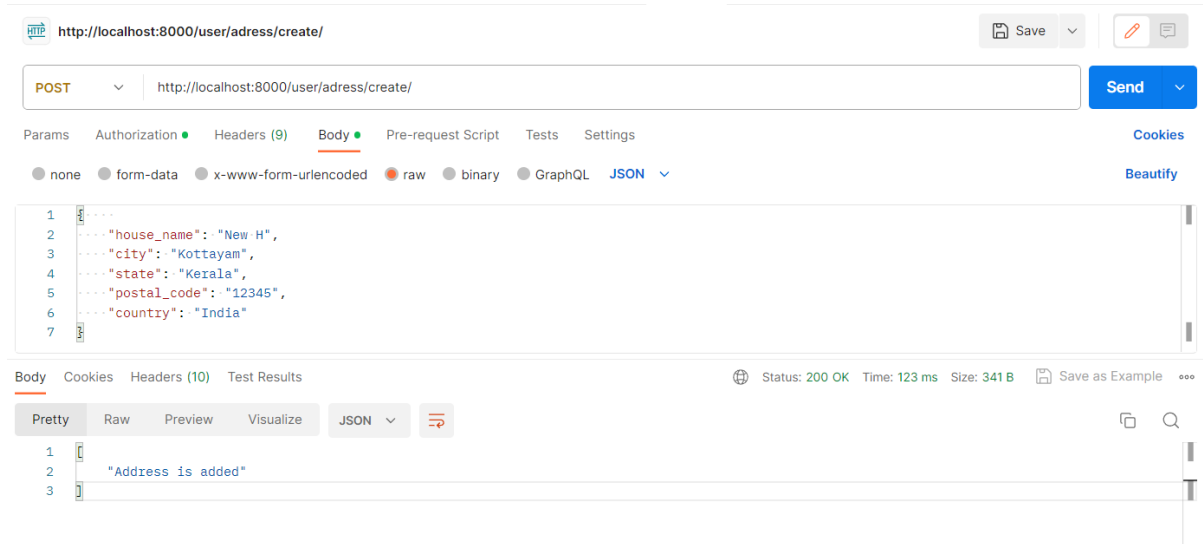
```
1 {
2   "password": "dave456",
3   "confirm_password": "dave456"
4 }
5
```

```
1 {
2   "message": "Password reset successfully"
3 }
```


Address and Profile Management

- Address Create view

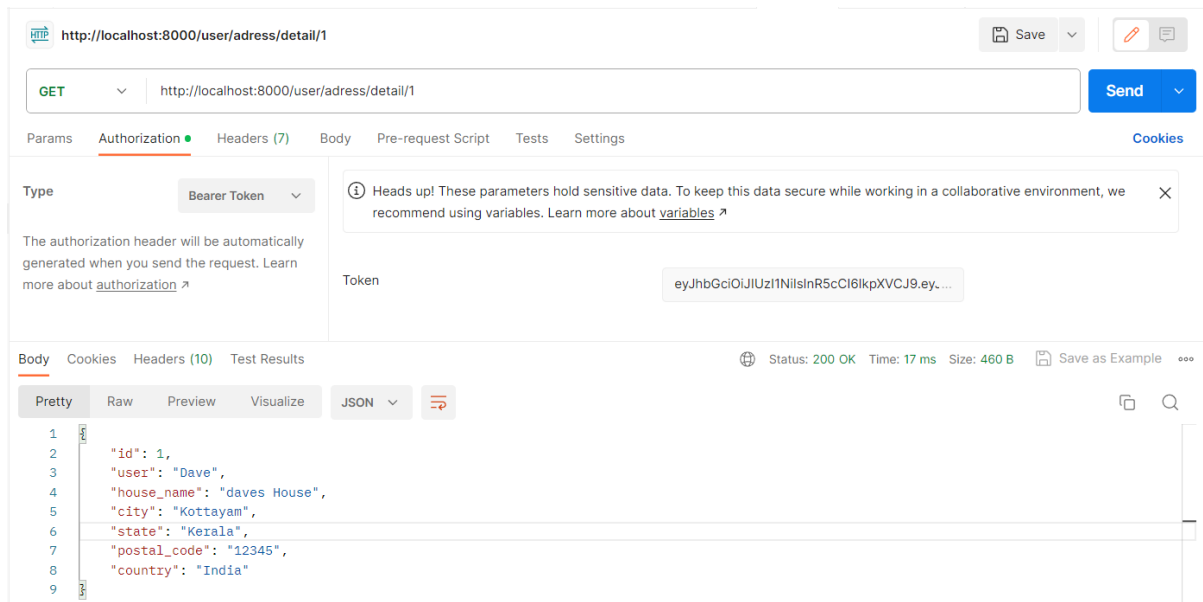
This API allows authenticated users to create and list their addresses.



- Address Detail View

This API allows authenticated users to retrieve, update, and delete their addresses.

- Retrieve Addresses



○ Update Adress

The screenshot shows a REST client interface with a PUT request to `http://localhost:8000/user/adress/detail/1/`. The request body is a JSON object with the following fields: `house_name`, `city`, `state`, `postal_code`, and `country`. The response status is 200 OK, and the response body is a JSON object containing the updated user profile information.

```
PUT http://localhost:8000/user/adress/detail/1/

{
  "house_name": "daves House ",
  "city": "Kottayam",
  "state": "Kerala",
  "postal_code": "12345",
  "country": "India"
}
```

Status: 200 OK Time: 150 ms Size: 460 B

```
{
  "id": 1,
  "user": "Dave",
  "house_name": "daves House",
  "city": "Kottayam",
  "state": "Kerala",
  "postal_code": "12345",
  "country": "India"
}
```

● Profile Create view

This API allows authenticated users to create and list their user profiles.

○ Profile Create

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/user/profile/create/`. The request body is a form with two fields: `profile_picture` and `mobile_number`. The response status is 200 OK, and the response body is a JSON object with the message `"Profile is created"`.

```
POST http://localhost:8000/user/profile/create/

{
  "profile_picture": "team-1_ABFyCk4.jpg",
  "mobile_number": "9876543212"
}
```

Status: 200 OK Time: 852 ms Size: 343 B

```
{
  "Profile is created"
}
```

○ Profile View

HTTP

http://localhost:8000/user/profile/create/

Save

⋮

Ⓞ

Ⓞ

GET

http://localhost:8000/user/profile/create/

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#).

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#).

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 16 ms

Size: 597 B

Save as Example

⋮

Pretty

Raw

Preview

Visualize

JSON

⋮

⋮

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

[

{

"id": 1,

"user": {

"id": 4,

"username": "Dave",

"email": "dave@gmail.com",

"first_name": "Dave",

"last_name": "M"

},

"profile_picture": "http://localhost:8000/media/uploads/profile_images/team-1_ABfYck4_8pCSZ4g.jpg",

"mobile_number": "9876543212",

"created_at": "2023-08-19T12:23:21.049554Z"

}

]

User Category View

- Category list View

This API lists categories for authenticated users with pagination to manage a potentially extensive list of categories.

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8000/user/category/list/`
- Method:** `GET`
- Authorization:** Bearer Token. The token value is `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`
- Status:** 200 OK, Time: 11 ms, Size: 1.05 KB
- Response Body (JSON):**

```
{
  "count": 4,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "category_name": "Laptops",
      "image": "http://localhost:8000/media/uploads/categories/20230629203150pexels-lisa-fotios-811587.jpg",
      "created_at": "2023-08-19T04:31:27.797385Z"
    },
    {
      "id": 2,
      "category_name": "Mobile",
      "image": "http://localhost:8000/media/uploads/categories/20230629200019130972-600x600.jpg",
      "created_at": "2023-08-19T08:29:14.710825Z"
    },
    {
      "id": 3,
      "category_name": "Laptops & Accessories",
      "image": "http://localhost:8000/media/uploads/categories/20230629200542pexels-veeterzy-303383.jpg",
      "created_at": "2023-08-19T08:33:31.739096Z"
    }
  ]
}
```

User Product View

- Product List View

This API lists products for authenticated users with filtering options and pagination for effective navigation. Additionally, filtering options are available to refine the displayed categories based on specific criteria.

- Product list

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/user/product/list`. The request is authenticated with a Bearer Token. The response is a JSON object with the following structure:

```
1 {
2   "count": 9,
3   "next": "http://localhost:8000/user/product/list?Pg=2",
4   "previous": null,
5   "results": [
6     {
7       "id": 1,
8       "product_name": "Dell Inspiron 15 Laptop",
9       "description": "15.6\" Full HD Display, Intel Core i5 Processor, 8GB RAM, 256GB SSD, Windows 10",
10      "price": "50000.00",
11      "quantity": 10,
12      "image": "http://localhost:8000/media/uploads/products/20230629200542pexels-veeterzy-303383.jpg",
13      "categories": [
14        1
15      ],
16      "created_at": "2023-08-19T08:58:39.468978Z"
17    }
18  ]
19 }
```

- Filtering Based on Category

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/user/product/list?name=Laptops`. The request is authenticated with a Bearer Token. The response is a JSON object with the following structure:

```
1 {
2   "count": 9,
3   "next": "http://localhost:8000/user/product/list?Pg=2&name=Laptops",
4   "previous": null,
5   "results": [
6     {
7       "id": 1,
8       "product_name": "Dell Inspiron 15 Laptop",
9       "description": "15.6\" Full HD Display, Intel Core i5 Processor, 8GB RAM, 256GB SSD, Windows 10",
10      "price": "50000.00",
11      "quantity": 10,
12      "image": "http://localhost:8000/media/uploads/products/20230629200542pexels-veeterzy-303383.jpg",
13      "categories": [
14        1
15      ],
16      "created_at": "2023-08-19T08:58:39.468978Z"
17    }
18  ]
19 }
```

○ Filtering Based on Price

HTTP <http://localhost:8000/user/product/list/?price=50000> Save

GET <http://localhost:8000/user/product/list/?price=50000> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Type Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body Cookies Headers (10) Test Results Status: 200 OK Time: 54 ms Size: 702 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "count": 1,
3   "next": null,
4   "previous": null,
5   "results": [
6     {
7       "id": 1,
8       "product_name": "Dell Inspiron 15 Laptop",
9       "description": "15.6\" Full HD Display, Intel Core i5 Processor, 8GB RAM, 256GB SSD, Windows 10",
10      "price": "50000.00",
11      "quantity": 10,
12      "image": "http://localhost:8000/media/uploads/products/20230629200542pexels-veeterzy-303383.jpg",
13      "categories": [
14        1
15      ],
16      "created_at": "2023-08-19T08:58:39.468978Z"
17    }
18  ]
19 }
```

● User Product View

This API retrieves details of a specific product for authenticated users.

HTTP <http://localhost:8000/user/product/retrieve/4/> Save

GET <http://localhost:8000/user/product/retrieve/4/> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Type Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body Cookies Headers (10) Test Results Status: 200 OK Time: 15 ms Size: 619 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 4,
3   "product_name": "Seagate 1TB External Hard Drive",
4   "description": "USB 3.0, Plug-and-Play, Back Up Your Data",
5   "price": "4000.00",
6   "quantity": 10,
7   "image": "http://localhost:8000/media/uploads/products/pexels-polina-tankilevitch-4523013.jpg",
8   "categories": [
9     1,
10    3
11  ],
12   "created_at": "2023-08-19T09:04:58.944180Z"
13 }
```

Cart Management

- Add To Cart View

This API allows authenticated users to add products to their cart and calculates the cart total.

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8000/user/addto/cart/`
- Method:** `POST`
- Authorization:** Bearer Token `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`
- Status:** 200 OK, Time: 202 ms, Size: 455 B
- Response Body (JSON):**

```
1 {
2   "cart_items": {
3     "id": 4,
4     "user": "Dave",
5     "product_name": "Logitech Wireless Keyboard and Mouse",
6     "quantity": 1,
7     "total": "2000.00"
8   },
9   "cart_total": 212000.0
10 }
```

- Cart List View

This API lists cart items for authenticated users with pagination to manage a potentially extensive list of cart items.

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8000/user/view/cart/`
- Method:** `GET`
- Authorization:** Bearer Token `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`
- Status:** 200 OK, Time: 53 ms, Size: 661 B
- Response Body (JSON):**

```
1 {
2   "count": 3,
3   "next": null,
4   "previous": null,
5   "results": [
6     {
7       "id": 1,
8       "user": "Dave",
9       "product_name": "Dell Inspiron 15 Laptop",
10      "quantity": 1,
11      "total": "50000.00"
12    },
13    {
14      "id": 2,
15      "user": "Dave",
16      "product_name": "Apple MacBook Air",
17      "quantity": 1,
18      "total": "60000.00"
19    }
20  ]
21 }
```

- Cart Update View

This API updates cart item quantities for authenticated users.

HTTP http://localhost:8000/user/cart/update/1/ Save

PUT http://localhost:8000/user/cart/update/1/ Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "quantity": 3
3 }
4
```

Body Cookies Headers (10) Test Results Status: 200 OK Time: 140 ms Size: 412 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "user": "Dave",
4   "product_name": "Dell Inspiron 15 Laptop",
5   "quantity": 3,
6   "total": "150000.00"
7 }
```

- Cart Delete View

This API Deletes cart items for authenticated users.

HTTP http://localhost:8000/user/cart/delete/3/ Save

DELETE http://localhost:8000/user/cart/delete/3/ Send

Params **Authorization** Headers (7) Body Pre-request Script Tests Settings

Type Bearer Token Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body Cookies Headers (10) Test Results Status: 200 OK Time: 116 ms Size: 357 B Save as Example

Pretty Raw Preview Visualize JSON

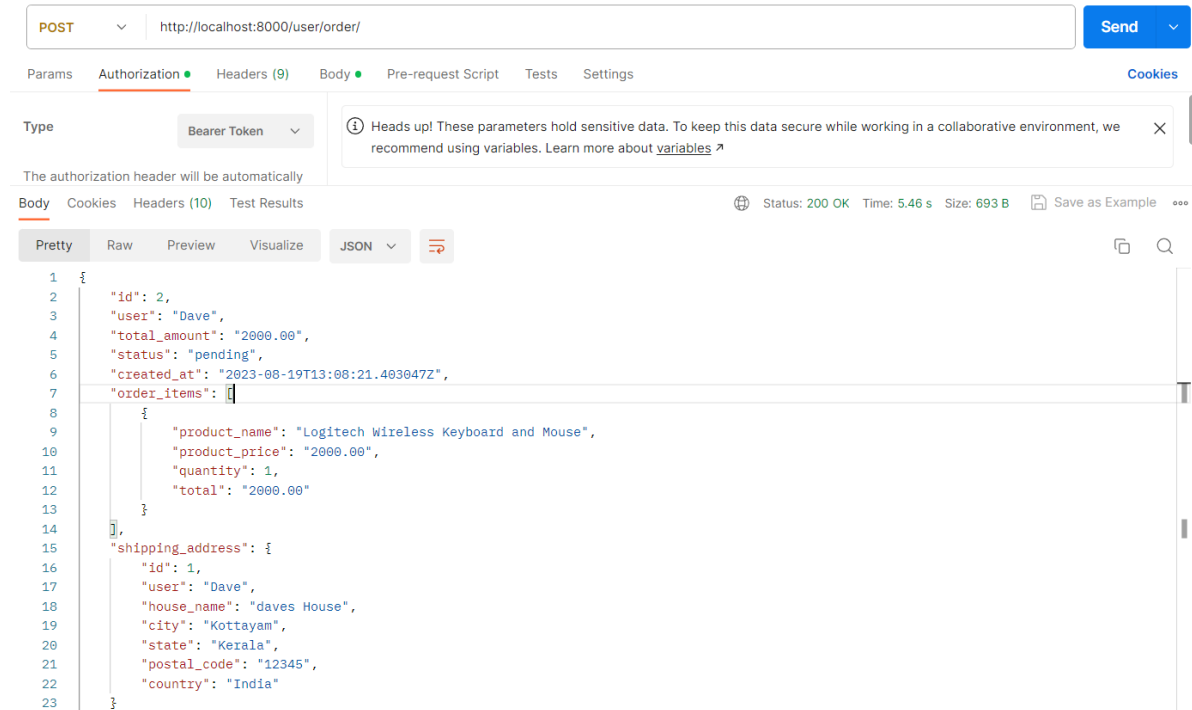
```
1 {
2   "message": "Cart item removed successfully."
3 }
```


Order Management

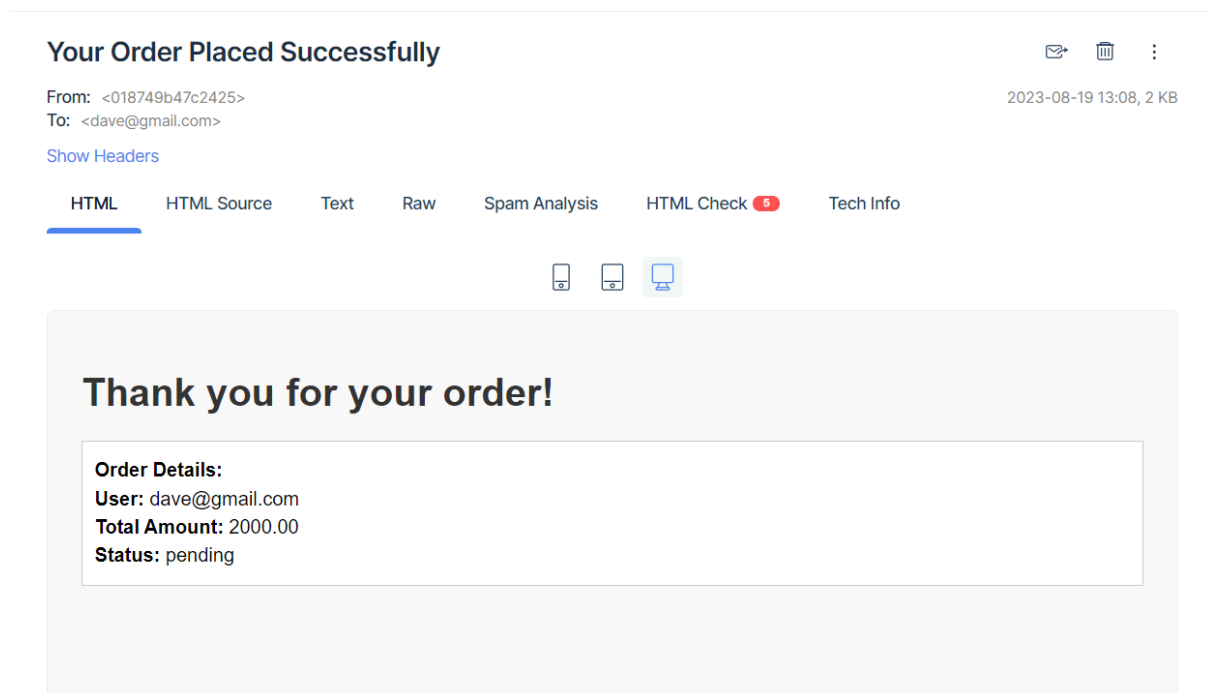
- Order View

Allows authenticated users to place orders, subtracts ordered quantities from product stock, and sends order confirmation emails to both users and admins.

- Making order



- Users receive confirmation mail



- Order List View

This API allows authenticated users to view a list of their orders. Pagination is implemented to manage the display of potentially numerous orders.

The screenshot shows a REST client interface. The method is GET and the URL is `http://localhost:8000/user/order/list/`. The Authorization header is set to Bearer Token with the value `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`. The response status is 200 OK, and the body is a JSON object:

```
1 {
2   "count": 2,
3   "next": null,
4   "previous": null,
5   "results": [
6     {
7       "id": 1,
8       "user": "Dave",
9       "total_amount": "212000.00",
10      "status": "pending",
11      "created_at": "2023-08-19T13:07:16.540168Z",
12      "order_items": [
13        {
14          "product_name": "Dell Inspiron 15 Laptop",
15          "product_price": "50000.00",
16          "quantity": 3,
17          "total": "150000.00"
18        }
19      ]
20    }
21  ]
22 }
```

- The user will receive an email notification when the admin updates the order status.

Order Confirmation

From: <admin@gmail.com>
To: <dave@gmail.com>

2023-08-19 13:39, 3.1 KB

[Show Headers](#)

HTML

[HTML Source](#)

[Text](#)

[Raw](#)

[Spam Analysis](#)

[HTML Check](#) 7

[Tech Info](#)



Order Confirmation

Hello ,

Your order is on its way.....

Order ID: 2

Order Status: shipped

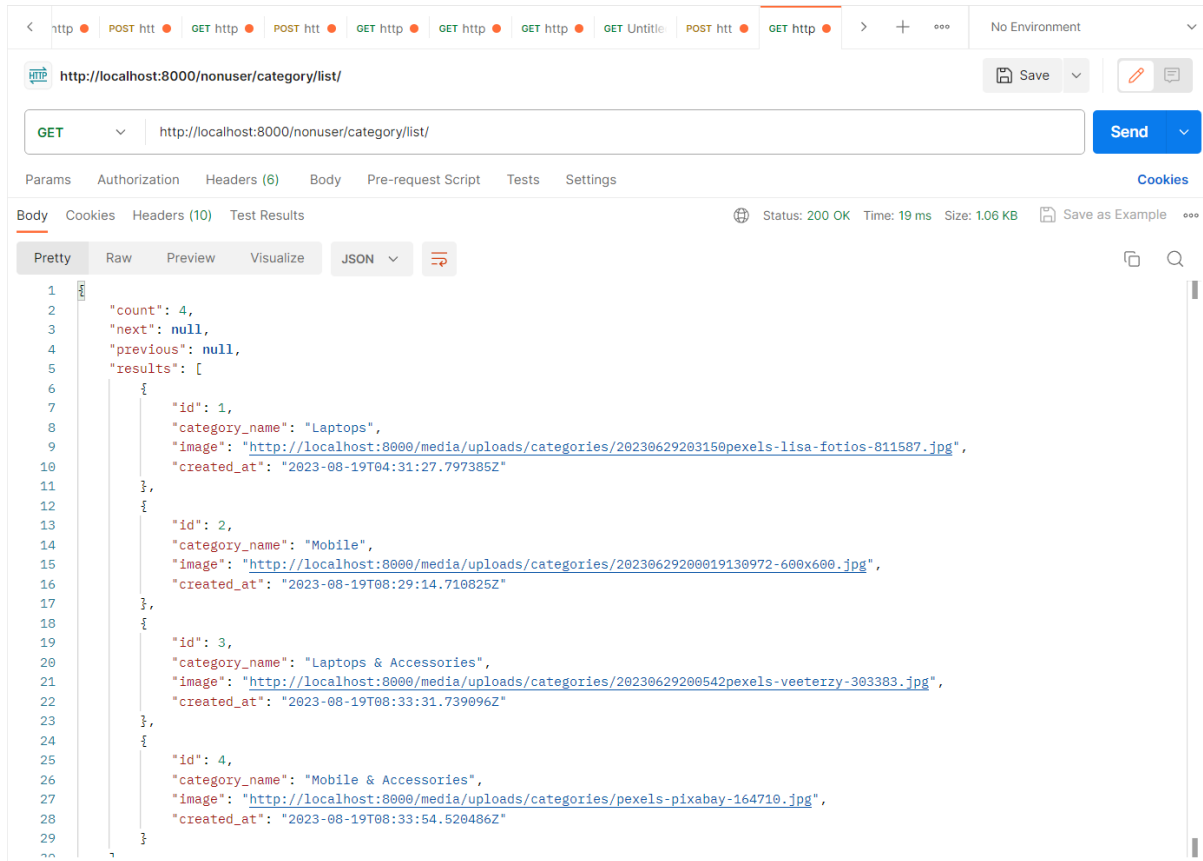
Total Amount: 2000.00

We appreciate your business and look forward to serving you again.

NON-REGISTERED USERS' MODULE

- Non-User Category list View

Lists categories for non-registered users.



- Non-User Product list view

Lists products for non-registered users with filtering options.

- Product list view

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/nonuser/product/list/`. The response is a JSON object with the following structure:

```
1 {
2   "count": 9,
3   "next": "http://localhost:8000/nonuser/product/list/?Pg=2",
4   "previous": null,
5   "results": [
6     {
7       "id": 1,
8       "product_name": "Dell Inspiron 15 Laptop",
9       "description": "15.6\" Full HD Display, Intel Core i5 Processor, 8GB RAM, 256GB SSD, Windows 10",
10      "price": "50000.00",
11      "quantity": 7,
12      "image": "http://localhost:8000/media/uploads/products/20230629200542pexels-veeterzy-303383.jpg",
13      "categories": [
14        1
15      ],
16      "created_at": "2023-08-19T08:58:39.468978Z"
17    },
18    {
19      "id": 2,
20      "product_name": "Apple MacBook Air",
21      "description": "13.3\" Retina Display, Apple M1 Chip, 8GB RAM, 256GB SSD, macOS",
22      "price": "60000.00",
23      "quantity": 9,
24      "image": "http://localhost:8000/media/uploads/products/20230629200542pexels-veeterzy-303383_gUhX1US.jpg",
25      "categories": [
26        1
27      ],
28      "created_at": "2023-08-19T09:00:24.225277Z"
29    }
30  ]
31 }
```

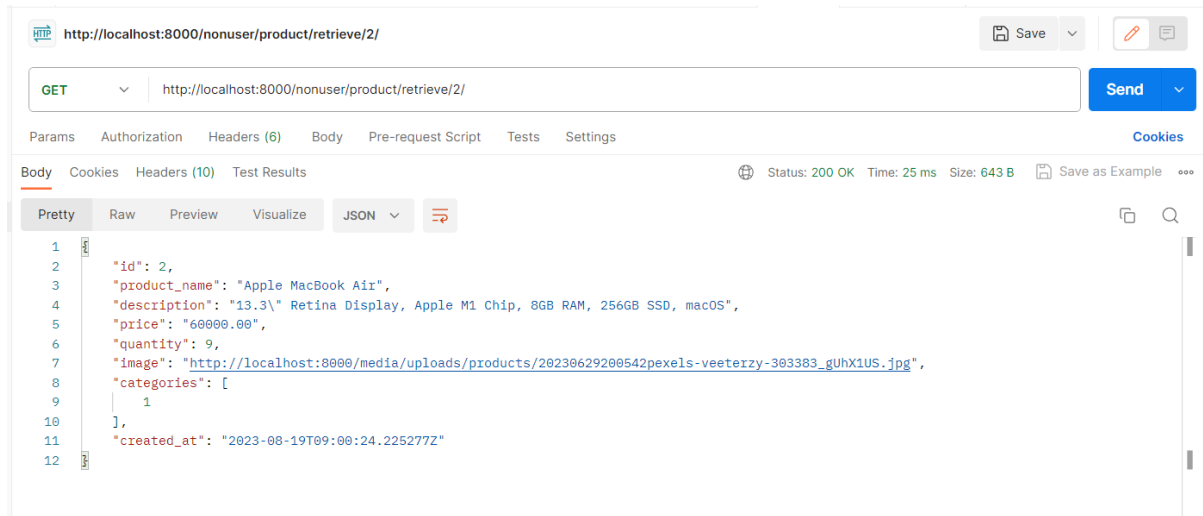
- Product list Filtering

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/nonuser/product/list/?price=50000`. The response is a JSON object with the following structure:

```
1 {
2   "count": 1,
3   "next": null,
4   "previous": null,
5   "results": [
6     {
7       "id": 1,
8       "product_name": "Dell Inspiron 15 Laptop",
9       "description": "15.6\" Full HD Display, Intel Core i5 Processor, 8GB RAM, 256GB SSD, Windows 10",
10      "price": "50000.00",
11      "quantity": 7,
12      "image": "http://localhost:8000/media/uploads/products/20230629200542pexels-veeterzy-303383.jpg",
13      "categories": [
14        1
15      ],
16      "created_at": "2023-08-19T08:58:39.468978Z"
17    }
18  ]
19 }
```

- Non-User Product View

Retrieves details of a specific product for non-registered users.



Links

Project: <https://albyecom.pythonanywhere.com/>

GitHub: <https://github.com/AlbyReji/Ecommerce-API>