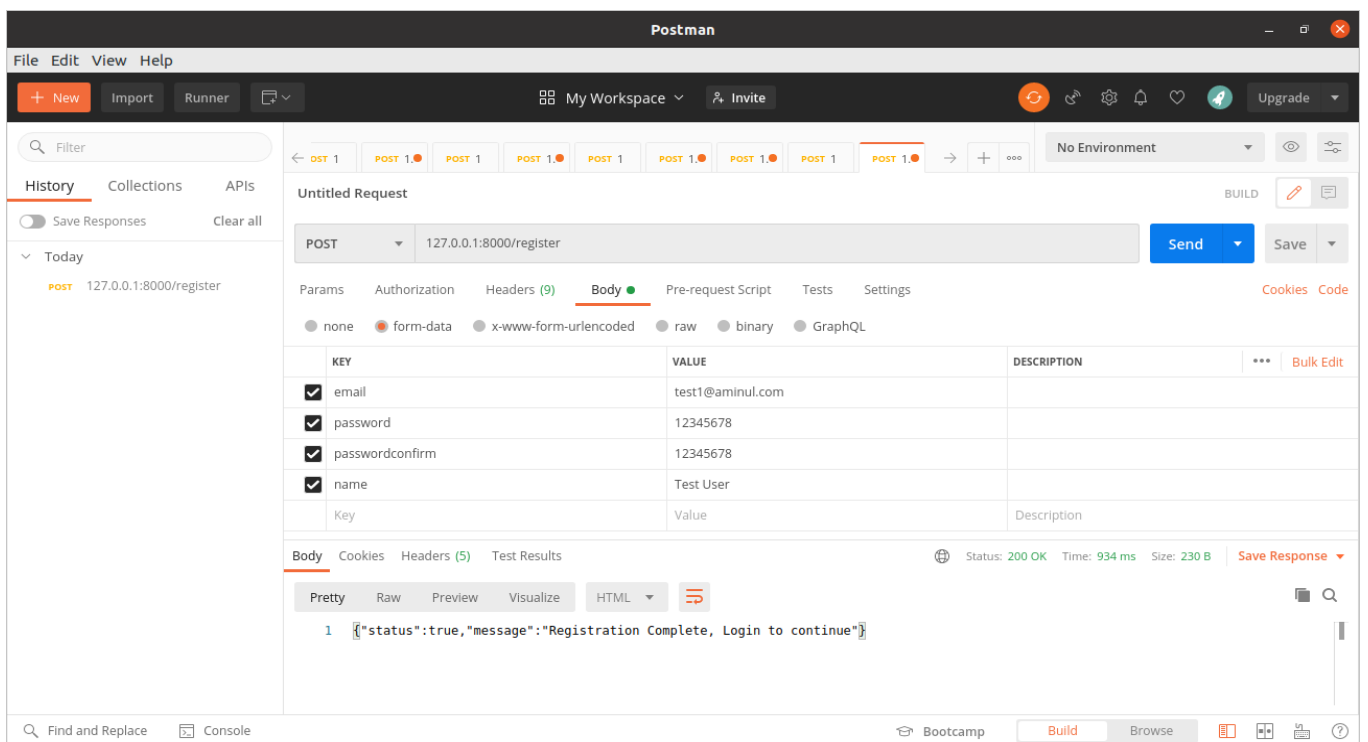
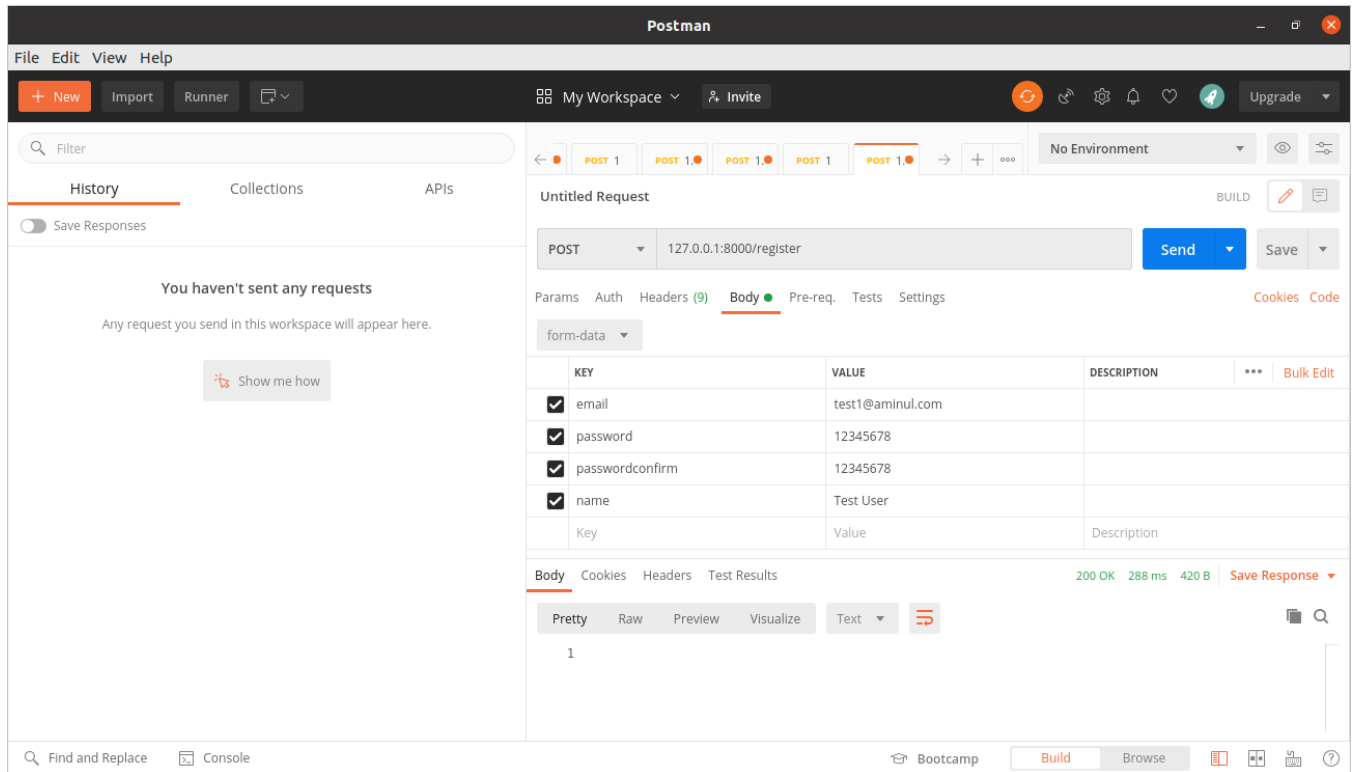


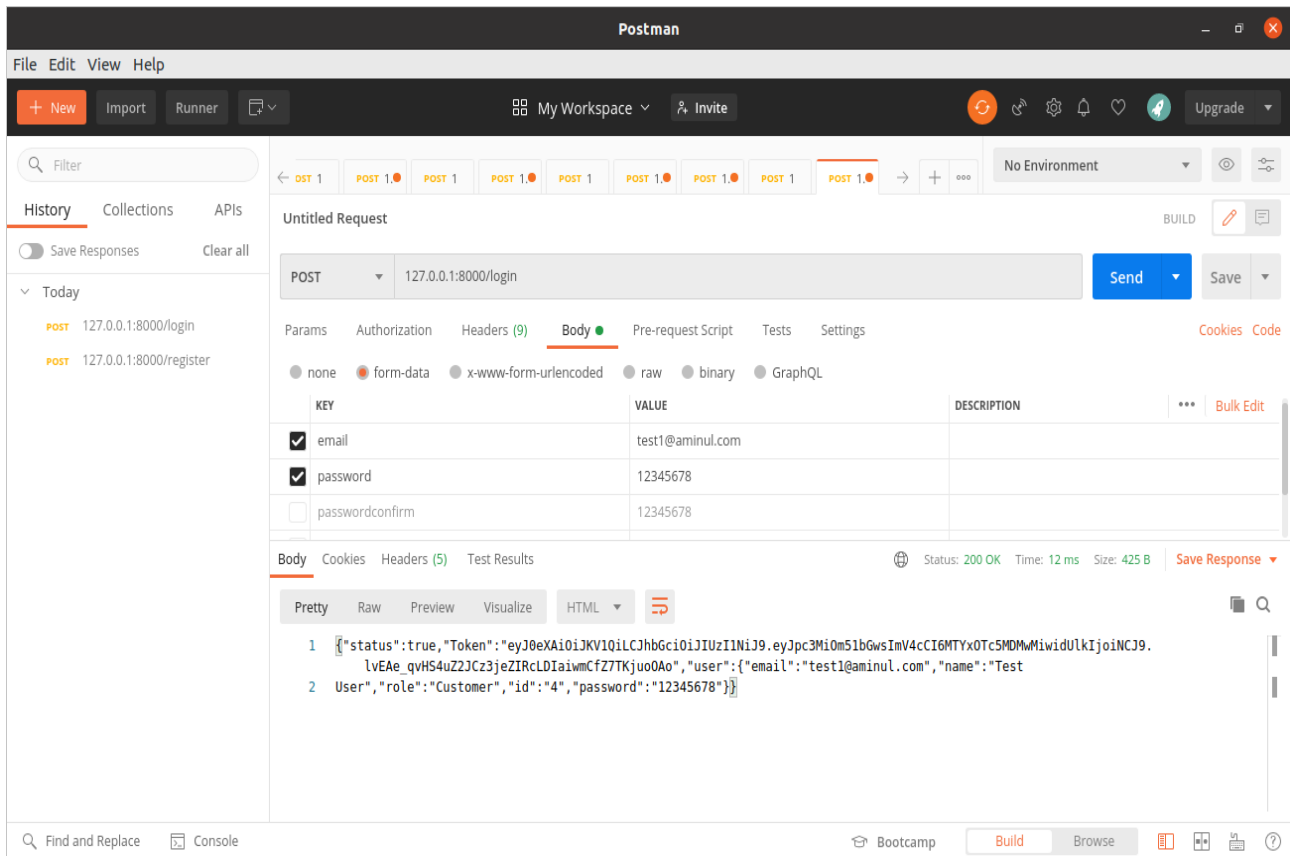
## We Devs Assignment Api Document

Registration Process:

Only Customer type of user can be registered. Admin user can be created directly from database.

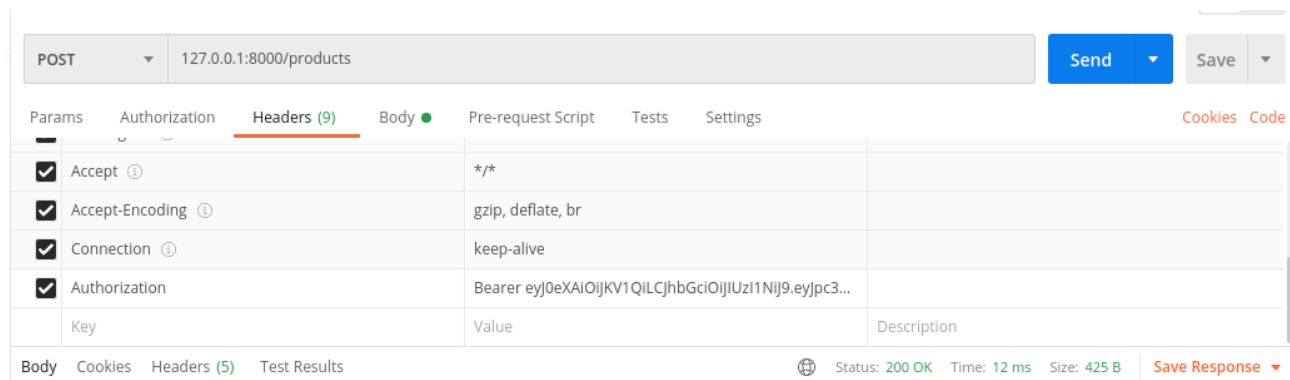


## Login

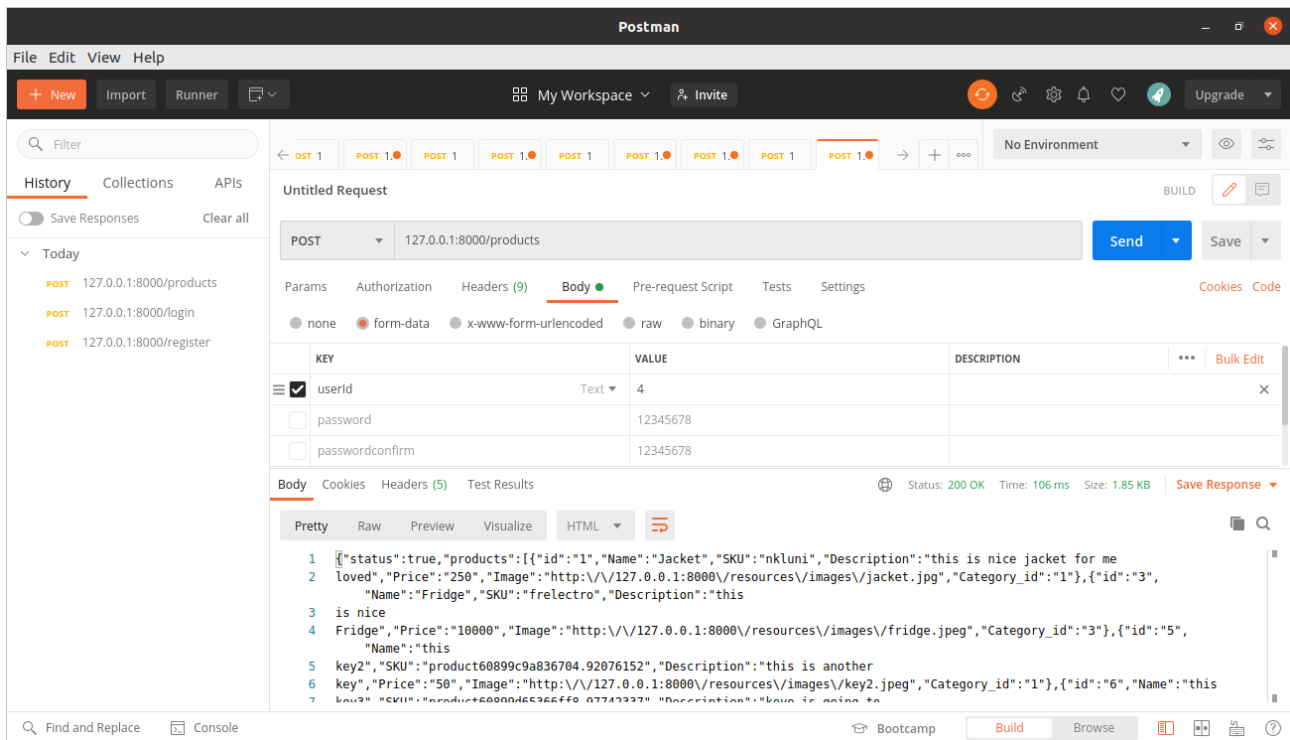


## Setting a Authorization Header:

For sending authorized request Authorization header is necessary.



## Getting Products



Postman interface showing a POST request to 127.0.0.1:8000/products. The request body is a JSON array of product objects. The response is a 200 OK status with a JSON array of product objects.

**Request:**

```
POST 127.0.0.1:8000/products
```

**Body (form-data):**

KEY	VALUE	DESCRIPTION
userid	4	
password	12345678	
passwordconfirm	12345678	

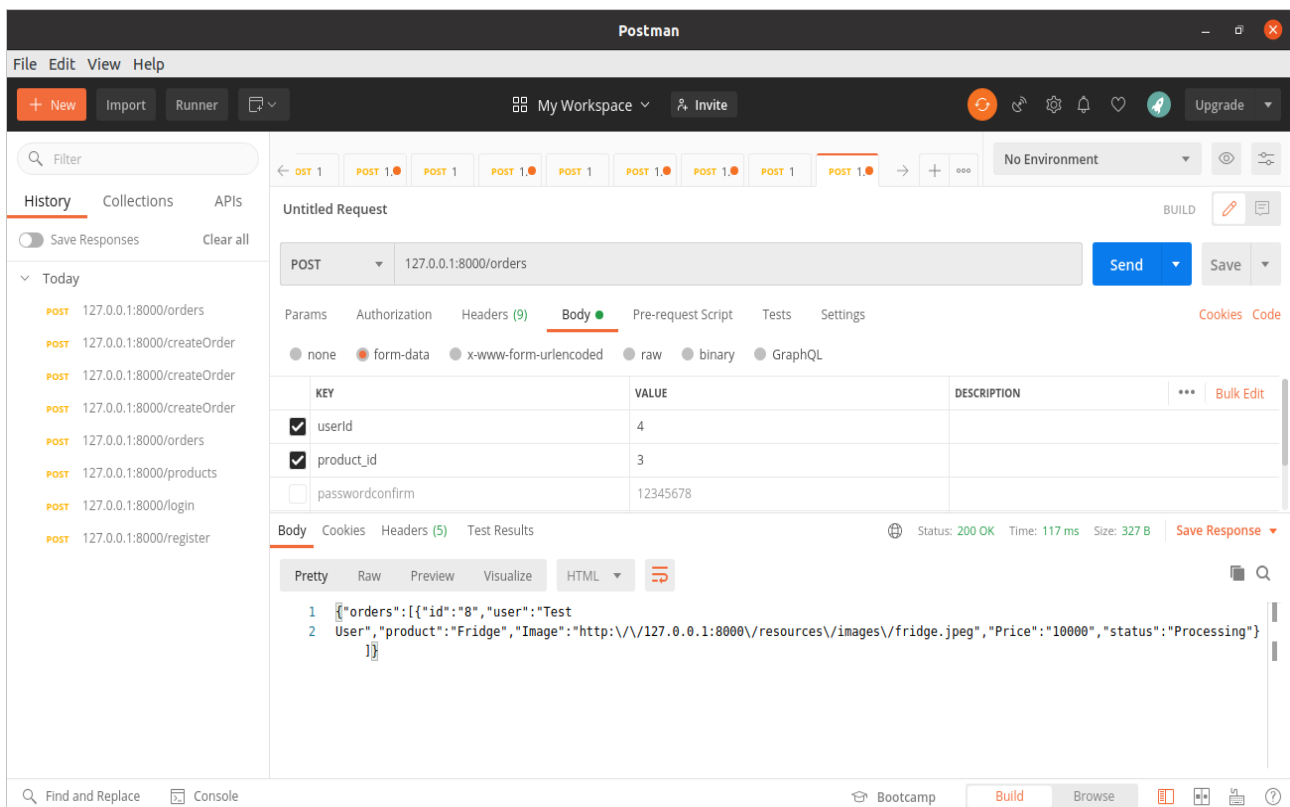
**Body (JSON):**

```
{
  "status": true,
  "products": [
    {
      "id": "1",
      "Name": "Jacket",
      "SKU": "nkluni",
      "Description": "this is nice jacket for me loved",
      "Price": "250",
      "Image": "http://127.0.0.1:8000/resources/images/jacket.jpg",
      "Category_id": "1"
    },
    {
      "id": "3",
      "Name": "Fridge",
      "SKU": "frelectro",
      "Description": "this is nice",
      "Price": "10000",
      "Image": "http://127.0.0.1:8000/resources/images/fridge.jpeg",
      "Category_id": "3"
    },
    {
      "id": "5",
      "Name": "this key2",
      "SKU": "product60899c9a836704.92076152",
      "Description": "this is another key",
      "Price": "50",
      "Image": "http://127.0.0.1:8000/resources/images/key2.jpeg",
      "Category_id": "1"
    },
    {
      "id": "6",
      "Name": "this key3",
      "SKU": "product60899c9a836704.92076152",
      "Description": "this is another key",
      "Price": "50",
      "Image": "http://127.0.0.1:8000/resources/images/key2.jpeg",
      "Category_id": "1"
    }
  ]
}
```

**Response:** Status: 200 OK, Time: 106 ms, Size: 1.85 KB

## Getting Orders

Admin can View all Orders. User can only see their orders.



Postman interface showing a POST request to 127.0.0.1:8000/orders. The request body is a JSON object with order details. The response is a 200 OK status with a JSON object containing order details.

**Request:**

```
POST 127.0.0.1:8000/orders
```

**Body (form-data):**

KEY	VALUE	DESCRIPTION
userid	4	
product_id	3	
passwordconfirm	12345678	

**Body (JSON):**

```
{
  "orders": [
    {
      "id": "8",
      "user": "Test User",
      "product": "Fridge",
      "Image": "http://127.0.0.1:8000/resources/images/fridge.jpeg",
      "Price": "10000",
      "status": "Processing"
    }
  ]
}
```

**Response:** Status: 200 OK, Time: 117 ms, Size: 327 B

## Order Update

Only Admin can update a order, User can't update order

The screenshot shows the Postman application with a POST request to `127.0.0.1:8000/editOrder`. The request body is a JSON object with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> userid	4	
<input checked="" type="checkbox"/> id	3	
<input checked="" type="checkbox"/> status	Shipped	

The response is a 200 OK status with the following JSON body:

```
1 {
2   "error": {
3     "status": 403,
4     "message": "You need to be an admin"
5   }
6 }
```

The screenshot shows the Postman application with a POST request to `127.0.0.1:8000/editOrder`. The request body is a JSON object with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> userid	1	
<input checked="" type="checkbox"/> id	4	
<input checked="" type="checkbox"/> status	Shipped	

The response is a 200 OK status with the following JSON body:

```
1 [{"status":true,"order":{"id":"4","user_id":"3","product_id":"10","status":"Shipped"}}]
```