API DOCUMENTATION

ItemList API Endpoint Documentation:

Overview
The ItemList endpoint provides a list of items and allows for filtering based on various attributes such as SKU, name, category, and tags.

Base URL
http://127.0.0.1:8000/items/

HTTP Method
GET

Query Parameters
- sku (string, optional): The stock keeping unit identifier for the item.
- name (string, optional): The name of the item, which supports partial matches.
- category_id (integer, optional): The identifier of the category to which the item belongs.
- tag_ids (list of integers, optional): A list of tag identifiers associated with the item.

Request Examples

Filter by SKU
sql

GET /items/?sku=NEW-SKU
Filter by Name
sql

GET /items/?name=New%20Item
Filter by Category ID
bash

GET /items/?category_id=1
Filter by Multiple Tag IDs
bash

GET /items/?tag_ids=1&tag_ids=2

Response Format

Upon a successful request, the API returns a list of items, each with the following structure:
json

```
[
  {
    "sku": "string",
```

```
        "name": "string",
        "category": {
            "id": "integer",
            "name": "string"
        },
        "tags": [
            {
                "id": "integer",
                "name": "string"
            },
            {
                "id": "integer",
                "name": "string"
            }
        ],
        "in_stock": "decimal",
        "available_stock": "decimal"
    }
    // ... additional items
]
```


Successful Response Example

A successful response for a request to GET /items/?name=New%20Item might look like:
json


```
HTTP/1.1 200 OK
Content-Type: application/json

[
    {
        "sku": "NEW-SKU",
        "name": "New Item Name",
        "category": {
            "id": 1,
            "name": "New Category"
        },
        "tags": [
            {
                "id": 1,
                "name": "New"
            },
            {
                "id": 2,
                "name": "Sample"
            }
        ],
```

```
        "in_stock": "100.000",
        "available_stock": "80.000"
    }
]
```


Error Response Example

If there are any errors (e.g., invalid query parameters), the API might return a response like:
json


```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "Invalid query parameters."
}
```


Attached are the screenshots of the execution:

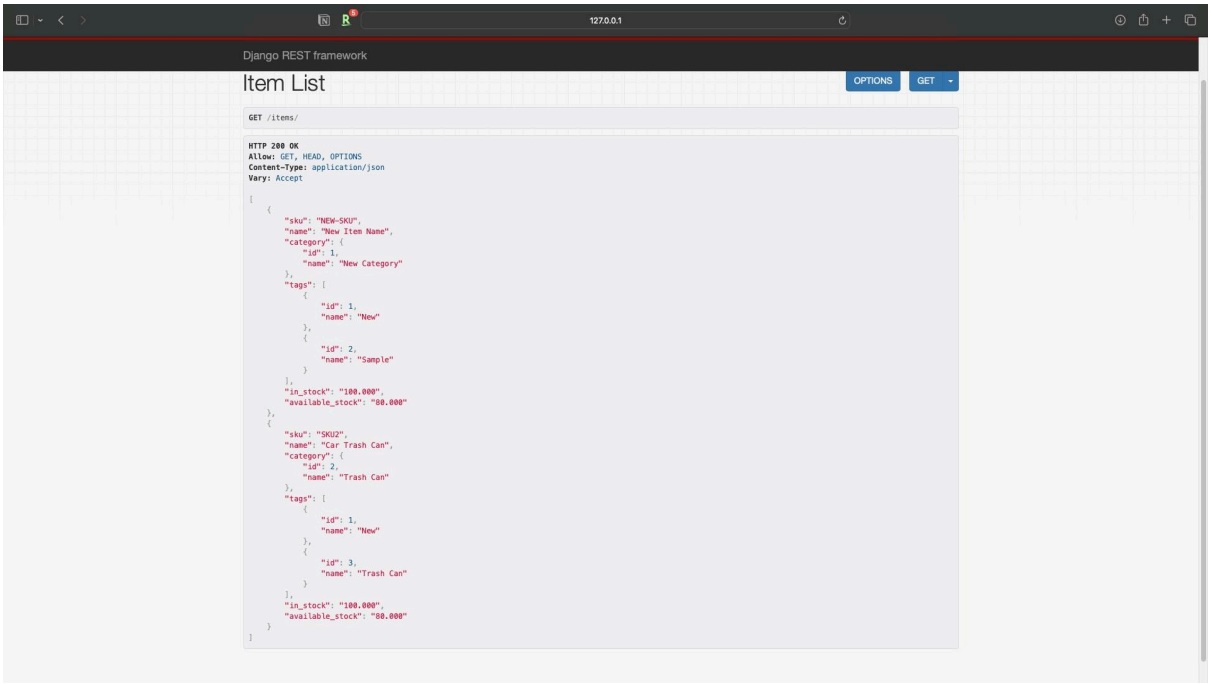

Fig 1: Result of Successful execution of the endpoint

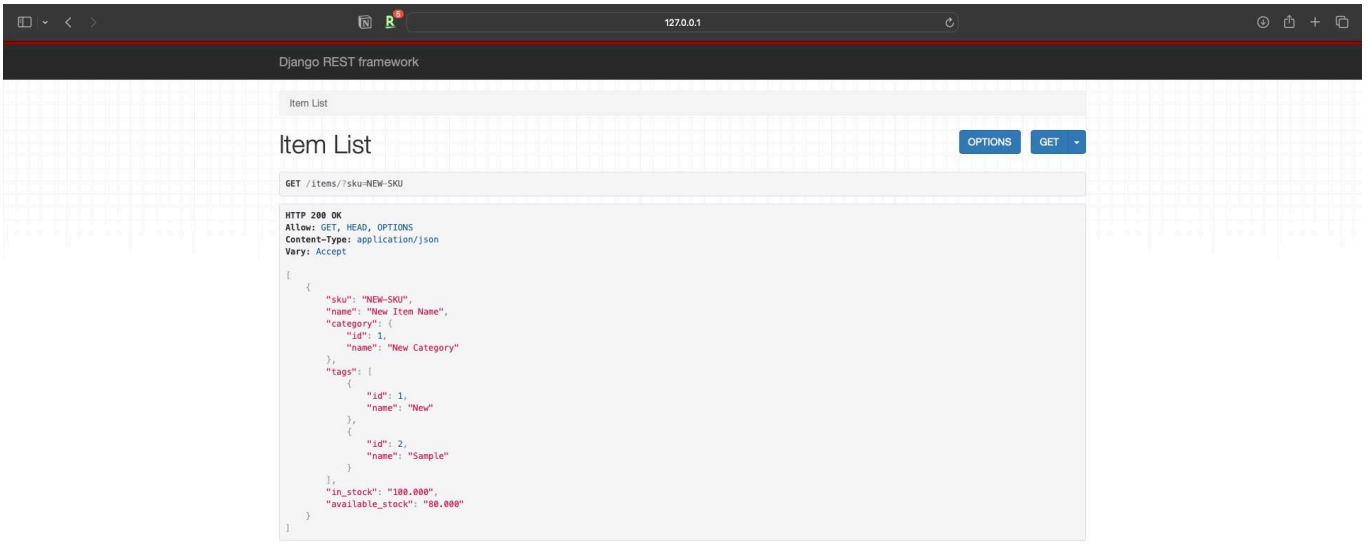

Fig 2: Result of Successful execution of the endpoint with Filter

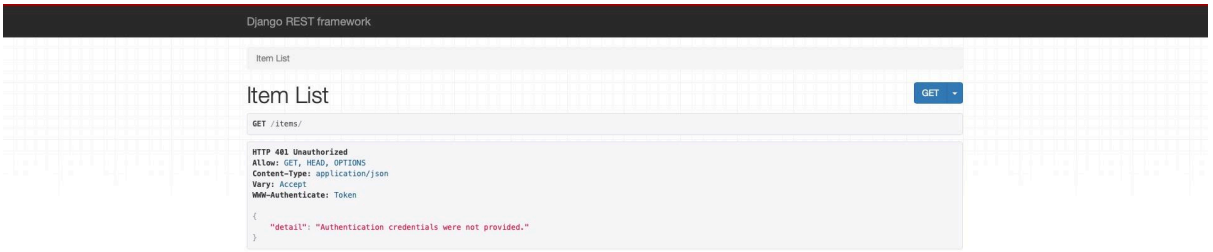

Fig 3: Result of Successful execution of the endpoint with user Authentication