📦 Laravel 12 Inventory API

A RESTful Laravel 12 backend for managing products with CRUD operations and user authentication using \*\*Laravel Sanctum\*\*. Supports basic role-based access control (admin vs staff).

## 🚀 Features

- Laravel 12 + Sanctum authentication

- Register/login endpoints

- Product CRUD API

- Role-based access (`admin` can create/delete, `staff` can only view)

- SKU generation

- Validation and structured API responses

## 🛠️ Requirements

- PHP >= 8.2

- Composer

- MySQL or any supported DB

- Laravel CLI

Steps to integrate the project

1. Unzip the file

2. Create a DB called Crud on your local host

3. Migrate the db (php artisan migrate)

4. Insert dummy data into the db (php artisan migrate:fresh --seed)

5. Login details for the following users

admin => email: admin@example.com, password: password

staff => email: staff@example.com, password: password

Endpoint using Postman

BASE\_URL = http://127.0.0.1:8000

📌 Register

POST BASE\_URL/api/register

Body (JSON):

{

    "name": "Daniel",

    "email": "daniechinedu766@gmail.com",

    "password": "Password@",

    "role": "staff"

}

{

"name": "Jane Doe",

"email": "jane@example.com",

"password": "password",

"role": "admin"

}

Json(Response)

{

    "message": "User registered successfully"

}

📌 Login

POST BASE\_URL/api/login

Body (JSON):

{

    "email": "admin@example.com",

    "password": "password"

}

{

"email": "jane@example.com",

"password": "password"

}

Json

{

    "token": "1|UGwjhmBo8u6w6pMGpkaOfzf9xFeziB8JytOnF7sI7a93a5df"

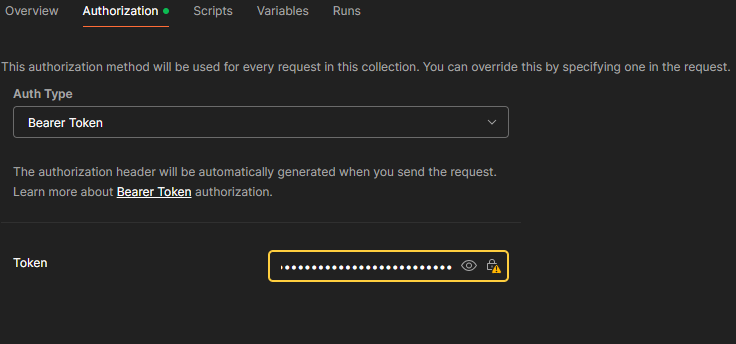
}

Response :{

"token": "eyJ0eXAiOiJKV1QiLCJhb..."

}

Authentication

The token gotten from the login will be used as a bearer token

Paste the token in the space provided.

📦 Product Management (Requires Auth token)

✅ List all Products

GET BASE\_URL/api/v1/products

json response

{

    "current\_page": 1,

    "data": [

        {

            "id": 1,

            "name": "aspernatur",

            "sku": "SKU042",

            "quantity": "195",

            "status": "A",

            "cost\_price": 58138,

            "selling\_price": 122827,

            "created\_by": 1,

            "updated\_by": null,

            "created\_at": "2025-06-24T07:46:12.000000Z",

            "updated\_at": "2025-06-24T07:46:12.000000Z"

        },

        }

    ],

    "next\_page\_url": "http://127.0.0.1:8000/api/v1/products?page=2",

    "path": "http://127.0.0.1:8000/api/v1/products",

    "per\_page": 10,

    "prev\_page\_url": null,

    "to": 10,

    "total": 20

}

✅ List Products by id

GET BASE\_URL/api/products/view/{product\_id}

json

{

    "id": 1,

    "name": "Test Product 1",

    "sku": "SKU042",

    "quantity": "14",

    "status": "U",

    "cost\_price": 90,

    "selling\_price": 100,

    "created\_by": 1,

    "updated\_by": 1,

    "created\_at": "2025-06-24T07:46:12.000000Z",

    "updated\_at": "2025-06-24T11:42:09.000000Z"

}

➕ Create Product

POST BASE\_URL/api/products/create

Body (JSON):

{

    "name": "Test Product 3",

    "quantity": 32,

    "status": "A",

    "cost\_price": 89.77,

    "selling\_price": 99.900

}

json

{

    "name": "Test Product 3",

    "sku": "TEST-PRODUCT-3-9ICXP",

    "quantity": "32",

    "status": "A",

    "cost\_price": "89.77",

    "selling\_price": "99.9",

    "created\_by": 1,

    "updated\_at": "2025-06-24T09:27:40.000000Z",

    "created\_at": "2025-06-24T09:27:40.000000Z",

    "id": 21

}

➕ Update Product

POST BASE\_URL/api/v1/products/update/{product\_id}

{

    "name": "Test Product 1",

    "quantity": 14,

    "status": "A",

    "cost\_price": 89.77,

    "selling\_price": 99.900

}

Json

{

    "id": 1,

    "name": "Test Product 1",

    "sku": "SKU042",

    "quantity": "14",

    "status": "A",

    "cost\_price": "89.77",

    "selling\_price": "99.9",

    "created\_by": 1,

    "updated\_by": 1,

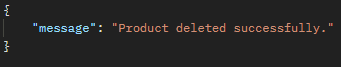
    "created\_at": "2025-06-24T07:46:12.000000Z",

    "updated\_at": "2025-06-24T11:25:38.000000Z"

}

➕ Delete Product

GET BASE\_URL/api/v1/products/delete/{product\_id}



json

{

    "message": "Product deleted successfully."

}

➕ Set Product Status to Available

GET BASE\_URL/api/v1/products/avail/{product\_id}

{

    "message": "Product set to available successfully."

}

➕ Set Product Status to Unavailable

GET BASE\_URL/api/v1/products/unavail/{product\_id}

{

    "message": "Product set to unavailable successfully."

}

➕ Get User Details

GET BASE\_URL/api/get\_user

json

{

    "id": 1,

    "name": "Admin User",

    "email": "admin@example.com",

    "role": "1",

    "email\_verified\_at": null,

    "created\_at": "2025-06-24T07:46:12.000000Z",

    "updated\_at": "2025-06-24T07:46:12.000000Z"

}

➕ Low Stock

GET BASE\_URL/api/v1/products/lowstock

[

    {

        "id": 1,

        "name": "Test Product 1",

        "sku": "SKU042",

        "quantity": 14,

        "status": "U",

        "cost\_price": 90,

        "selling\_price": 100,

        "created\_by": 1,

        "updated\_by": 1,

        "created\_at": "2025-06-24T07:46:12.000000Z",

        "updated\_at": "2025-06-24T11:42:09.000000Z"

    },

]

POST BASE\_URL/api/v1/ai\_query

Body

{

    "question": "Whats is total product"

}

Response

{

    "status": "success",

    "answer": "The total product value is 21",

    "breakdown": {

        "total\_Product": 21

    }

}

➕ Export xlsx file (Excel file)

GET BASE\_URL/api/v1/products/export

In the feature when report start getting huge a job will be created to handle the file to prevent crashing or hanging of ones pc or laptop

➕ Logout

GET BASE\_URL/api/logout