**Inventory Management API Documentation**

**Table of Contents**

1. [Setup](#setup)
2. [Endpoints](#endpoints)
   * [Inventory Items](#inventory-items)
   * [Suppliers](#suppliers)
3. [Usage Examples](#usage-examples)

**Setup**

**Prerequisites**

* Python 3.12.1
* Django 5.0.6
* Django REST Framework

#### Installation

1. **Clone the repository**

git clone https://github.com/tmabs-94/Inventory-management-API.git

cd inventory\_project

1. **Create and activate a virtual environment**

python -m venv venv

source venv/bin/activate # On Windows use `venv\Scripts\activate`

1. **Install dependencies**

pip install -r requirements.txt

1. **Apply migrations**

python manage.py migrate

1. **Run the development server**

python manage.py runserver

### Endpoints

**Inventory Items**

* **List all inventory items**

GET /api/inventory-items/

* **Create a new inventory item**

POST /api/inventory-items/

Content-Type: application/json

{

"name": "Test Item",

"description": "This is a test item",

"price": 10.99,

"suppliers": [1, 2]

}

* **Retrieve a single inventory item**

GET /api/inventory-items/{id}/

* **Update an inventory item**

PATCH /api/inventory-items/{id}/

Content-Type: application/json

{

"name": "Updated Test Item"

}

* **Delete an inventory item**

DELETE /api/inventory-items/{id}/

**Suppliers**

* **List all suppliers**

GET /api/suppliers/

* **Create a new supplier**

POST /api/suppliers/

Content-Type: application/json

{

"name": "Test Supplier",

"contact\_info": "123 Test St, Test City",

"items": [1]

}

* **Retrieve a single supplier**

GET /api/suppliers/{id}/

* **Update a supplier**

PATCH /api/suppliers/{id}/

Content-Type: application/json

{

"name": "Updated Test Supplier"

}

* **Delete a supplier**

DELETE /api/suppliers/{id}/

**Usage Examples**

**Create an Inventory Item**

curl -X POST http://127.0.0.1:8000/api/inventory-items/ \

-H "Content-Type: application/json" \

-d '{

"name": "Test Item",

"description": "This is a test item",

"price": 10.99,

"suppliers": [1, 2]

}'

**List Inventory Items**

curl -X GET http://127.0.0.1:8000/api/inventory-items/

**Update an Inventory Item**

curl -X PATCH http://127.0.0.1:8000/api/inventory-items/1/ \

-H "Content-Type: application/json" \

-d '{

"name": "Updated Test Item"

}'

**Delete an Inventory Item**

curl -X DELETE http://127.0.0.1:8000/api/inventory-items/1/

**Create a Supplier**

curl -X POST http://127.0.0.1:8000/api/suppliers/ \

-H "Content-Type: application/json" \

-d '{

"name": "Test Supplier",

"contact\_info": "123 Test St, Test City",

"items": [1]

}'

**List Suppliers**

curl -X GET http://127.0.0.1:8000/api/suppliers/

**Update a Supplier**

curl -X PATCH http://127.0.0.1:8000/api/suppliers/1/ \

-H "Content-Type: application/json" \

-d '{

"name": "Updated Test Supplier"

}'

**Delete a Supplier**

curl -X DELETE http://127.0.0.1:8000/api/suppliers/1/

**Notes**

* Make sure to replace 127.0.0.1:8000 with your server's URL if you are not running the server locally.
* IDs used in the URLs and data payloads should be replaced with actual IDs from your database.

This documentation provides the essential steps to set up the project and interact with the API endpoints for inventory items and suppliers.