

📦 Grocery List API – RESTful CRUD Backend

Built with FastAPI, Pydantic, and Docker

COMP 351 – Advanced Website Programming | UFV | Jan-Apr 2025

📋 Project Overview

This Grocery List API was developed as a backend project for UFV's COMP 351 course. I built a file-based RESTful API using FastAPI that allows users to manage grocery items – supporting creation, updates, deletion, filtering, and data validation.

This project was focused on clean backend architecture, input validation with Pydantic, and deploying with Docker.

🔑 Key Features

- 📦 **CRUD functionality** – Create, read, update, and delete grocery items
 - 📦 **Pydantic validation** – Enforces correct data types, required fields, and formats
 - 📦 **Search & filtering** – Filter groceries by name, category, or price range
 - 📦 **Data persistence** – All items saved to a local JSON file
 - 📦 **Docker-ready** – Containerized app for consistent deployment
 - 📦 **Custom error handling** – Friendly error messages for invalid input
-

🛠️ Tech Stack

- **Backend:** FastAPI
 - **Validation:** Pydantic
 - **Data Storage:** JSON file-based persistence
 - **Deployment:** Docker
-

📖 Setup Instructions (Local)

1. Clone the repository:

```
git clone https://github.com/DJNicoco/grocery-list-api.git
cd grocery-list-api
```

2. Create and activate a virtual environment:

```
python -m venv venv
source venv/bin/activate
```

3. Install dependencies:

```
pip install -r requirements.txt
```

4. Run the app:

```
uvicorn main:app --reload
```

Sample API Endpoints

Method	Endpoint	Description
GET	/items	Get all grocery items
GET	/items/{id}	Get a specific item by ID
POST	/items	Add a new grocery item
PUT	/items/{id}	Update an existing item
DELETE	/items/{id}	Delete an item by ID
GET	/items/search	Search/filter items by query

Example Input

```
{
  "name": "Bananas",
  "category": "Fruit",
  "price": 2.49
}
```

All inputs are validated using Pydantic schemas to ensure consistent and secure data structure.

Docker Deployment

To build and run the app using Docker:

```
docker build -t grocery-api .
docker run -p 8000:8000 grocery-api
```

Reflection

This project helped me build confidence in backend development by focusing purely on core API design principles. I learned how to validate input with Pydantic, handle structured error responses, and keep logic clean and testable. It was a great stepping stone in learning to ship fast, test endpoints with curl and Postman, and containerize the app with Docker.

References

- [FastAPI Official Docs](#)
- [Pydantic Docs](#)
- [Docker + FastAPI Guide](#)

Disclaimer

This project was originally developed for COMP 351 at the University of the Fraser Valley. No original source code from the course is included in this repository. README content is written independently for portfolio and demonstration purposes only.