Technical Document for Online Restaurant Ordering System API

12.08.2024

Created By

- Mason Scarbro
- Vincent Medina
- Jordan Koontz
- Elijah Collins

Overview

The Online Restaurant Ordering System API is a service built with FastAPI to facilitate the management of restaurant-related operations. The API is designed for scalability, efficiency, and ease of integration, providing a reliable backend solution for ordering systems. It

supports essential operations like customer management, menu creation, order processing, and review handling, making it adaptable for small to mid-sized restaurants.

Architecture Overview

Framework and Libraries:

1. Framework: FastAPI
2. Database: PostgreSQL with SQLAlchemy ORM
3. Web Server: Uvicorn

Project Structure

```
GroupProject/
├── api/
│   ├── controllers/
│   ├── dependencies/
│   ├── models/
│   ├── routers/
│   ├── schemas/
│   └── tests/
├── Class Diagrams
├── Documentation
├── evn
├── main.py
├── README.md
├── database.py
└── requirements.txt
```

Endpoint Documentation

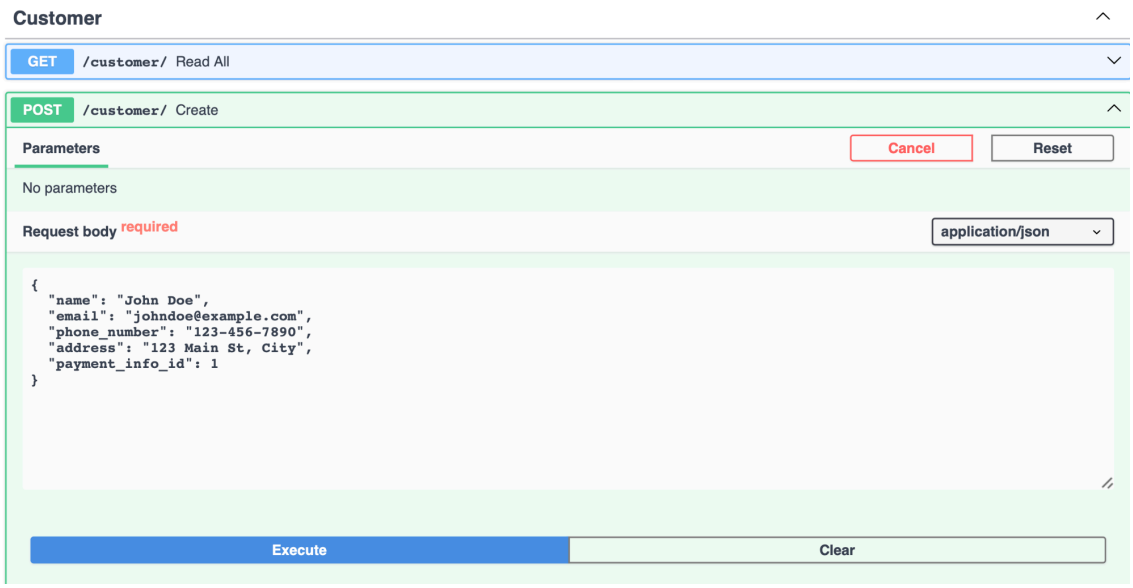
Customer Endpoints

Methods	Endpoint	Description
POST	**/customers/**	**Create a new customer**
DELETE	**/customers/{id}**	**Delete a customer by ID**

PUT	/customers/{id}	Update a customer by ID

Create Customer Example

Request:



Response:

Menu Endpoints

Methods	Endpoint	Description
GET	/menus/	Get all menus
POST	/menus/	Create a new menu

PUT	/menus/{menu_id}	Update a menu by ID
DELETE	/menus/{menu_id}	Delete a menu by ID

Code Examples

Create Customer Function

```
def create(db: Session, request):
    new_item = model.Customer(
        name =request.name,
        email =request.email,
        phone_number =request.phone_number,
        address =request.address,
        payment_info_id =request.payment_info_id
    )
    try:
        db.add(new_item)
        db.commit()
        db.refresh(new_item)
except SQLAlchemyError as e:
        raise HTTPException(status_code=400, detail=str(e.__dict__['orig']))
return new_item
```

Development Environment

Setup Instructions

1. **Clone the Repository:**

   ```
   git clone https://github.com/MasonScarbro/GroupProject-ITSC-3155.git
   ```

2. **Install Dependencies:**

Requirements: fastapi, uvicorn, sqlalchemy, pymysql, pytest, pytest-mock, httpx, and cryptography

Necessary packages:

pip install fastapi

pip install "uvicorn[standard]"

pip install sqlalchemy

pip install pymysql

pip install pytest

pip install pytest-mock

pip install httpx

pip install cryptography

Set Up the Database:

Configure your database settings in config.py or .env file.

Run the API Server:

uvicorn main:app --reload

Access the API Documentation:

Navigate to http://127.0.0.1:8000/docs for the interactive Swagger UI.