LIBRARY MANAGEMENT SYSTEM

Higher National Diploma in Software Engineering

EAD-2 Project Documentation

24.1F

National Institute of Business Management
Kandy Regional Center
No 2, Asgiriya Road,
Kandy

Declaration

“We certify that this project does not incorporate without acknowledgement, any material previously submitted for a Diploma in any institution n and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself expect where due reference is made in the text. We also hereby give consent for our project report, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organizations”

Project Title: Restaurant Management System

Group Members:

Name	**Index Number**
M.F.M. Nuzki	KAHDSE231F-025
W.H.D.Yeshani	KAHDSE231F-067
S.A.Jonathan	KAHDSE231F-024
M.F.M.Nabeel	KAHDSE231F-027

Contents

Overview

This document outlines the REST API endpoints for managing items and payments in a Restaurant Management System. The system allows for basic CRUD (Create, Read, Update, and Delete) operations on restaurant items and payments. The following are the available endpoints categorized into two sections: **Items** and **Payments**.

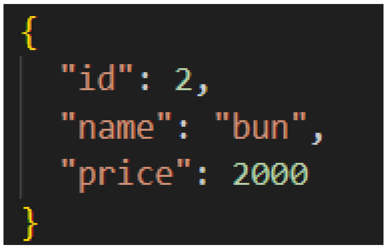
---

1. Items API Endpoints

These endpoints are used for managing restaurant menu items such as food, beverages, or other products available in the restaurant.

GET /api/items

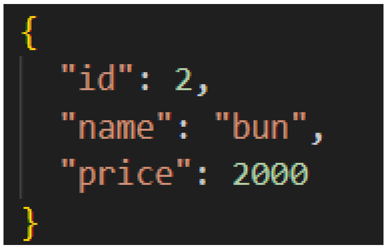
- **Description**: Retrieves all items in the restaurant.
- **Method**: `GET`
- **Response**:



```
{
  "id": 2,
  "name": "bun",
  "price": 2000
}
```

GET /api/items/{id}

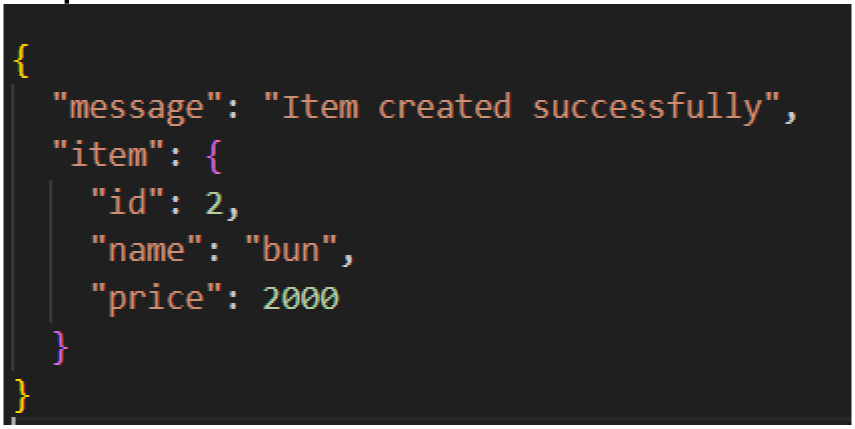
- **Description**: Retrieves a specific item by its unique ID.
- **Method**: `GET`
- **Path Parameter**:
    - `id`: The unique identifier of the item to retrieve.
- **Response**:



```
{
  "id": 2,
  "name": "bun",
  "price": 2000
}
```

POST /api/items

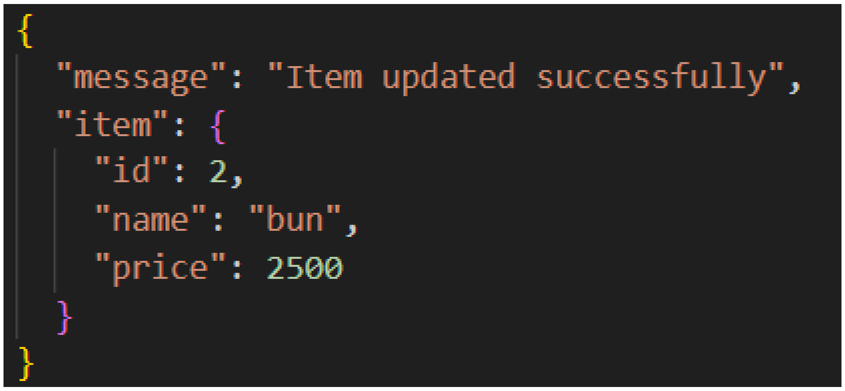
- **Description**: Creates a new item in the restaurant menu.
- **Method**: `POST`
- **Response**:



```
{
  "message": "Item created successfully",
  "item": {
    "id": 2,
    "name": "bun",
    "price": 2000
  }
}
```

PUT /api/items/{id}

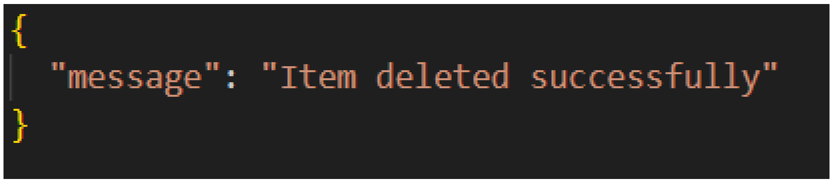
- **Description**: Updates an existing item by its unique ID.
- **Method**: `PUT`
- **Path Parameter**:
  - `id`: The unique identifier of the item to update.
- **Response**:



```
{
  "message": "Item updated successfully",
  "item": {
    "id": 2,
    "name": "bun",
    "price": 2500
  }
}
```

DELETE /api/items/{id}

- **Description**: Deletes an item from the restaurant menu by its unique ID.
- **Method**: `DELETE`
- **Path Parameter**:
  - `id`: The unique identifier of the item to delete.
- **Response**:



```
{
  "message": "Item deleted successfully"
}
```

---

2. Payments API Endpoints

These endpoints are used for managing payments in the restaurant system, including customer payments for orders.

GET /api/payments

- **Description**: Retrieves all payment records made in the restaurant.
- **Method**: `GET`
- **Response**: A list of all payment records with details

GET /api/payments/{id}

- **Description**: Retrieves a specific payment record by its unique ID.
- **Method**: `GET`
- **Path Parameter**:
  - `id`: The unique identifier of the payment to retrieve.
- **Response**: The details of the requested payment record.

POST /api/payments

- **Description**: Creates a new payment record.
- **Method**: `POST`
- **Response**: A confirmation message along with the created payment record's details.

PUT /api/payments/{id}

- **Description**: Updates an existing payment record by its unique ID.
- **Method**: `PUT`
- **Path Parameter**:
  - `id`: The unique identifier of the payment to update.
- **Response**: A confirmation message with the updated payment details.

DELETE /api/payments/{id}

- **Description**: Deletes a payment record by its unique ID.
- **Method**: `DELETE`
- **Path Parameter**:
  - `id`: The unique identifier of the payment to delete.
- **Response**: A confirmation message indicating successful deletion.

Summary

The Restaurant Management System API provides two primary sets of functionalities:

1. **Item Management**: Operations for creating, retrieving, updating, and deleting restaurant items.
2. **Payment Management**: Operations for managing customer payments.

Each endpoint is designed to be straightforward, with standard HTTP methods and simple URL structures, ensuring easy integration with front-end systems or external applications for managing restaurant operations.