Overview

The Items API provides a way to manage a list of food items, including their names, prices, and descriptions. It supports the basic CRUD (Create, Read, Update, Delete) operations.

Base URL

http://localhost:3000/items

Data Model: Item

Code

```
{
  "Name": "String",
  "Price": "Number",
  "Description": "String"
}
```

Endpoints

1. Get All Items

- **Endpoint:** /items
- **Method:** GET
- **Description:** Retrieves a list of all items in the pizza store.
- **Response:**
  - **Status 200:** An array of items.
  - **Status 500:** Internal server error.

Code

```
[
    "_id": "string",
"Name": "Margherita",
       "Price": 8.99,
       "Description": "Classic pizza with tomato sauce and
mozzarella cheese"
},
]
```

2. Get Item Creation Form

- **Endpoint:** /item/new
- **Method:** GET
- **Description:** Renders a form to create a new item.
- **Response:**
  - **Status 200:** HTML form for creating a new item.
  - **Status 500:** Internal server error.

3. Get Item Edit Form

- **Endpoint:** /items/:id/edit
- **Method:** GET
- **Description:** Renders a form to edit an existing item.
- **URL Parameters:**
  - id (String): The unique identifier of the item to edit.
- **Response:**
  - **Status 200:** HTML form for editing the item.
  - **Status 404:** Item not found.
  - **Status 500:** Internal server error.

4. Create a New Item

- **Endpoint:** /items
- **Method:** POST
- **Description:** Creates a new item and saves it to the database.
- **Request Body:**
    - Name (String): The name of the item (required).
    - Price (Number): The price of the item (required).
    - Description (String): The description of the item (required).
- **Response:**
    - **Status 302:** Redirects to /items after successful creation.
    - **Status 500:** Internal server error.

```
{
  "Name": "Pepperoni",
  "Price": 9.99,
  "Description": "Pizza topped with pepperoni slices"
}
```

5. Update an Item

- **Endpoint:** /items/:id
- **Method:** PUT
- **Description:** Updates an existing item in the database.
- **URL Parameters:**
    - id (String): The unique identifier of the item to update.
- **Request Body:**
    - Name (String): The name of the item (optional).

    - Price (Number): The price of the item (optional).
    - Description (String): The description of the item (optional).
- **Response:**
    - **Status 302:** Redirects to /items after successful update.
    - **Status 404:** Item not found.
    - **Status 500:** Internal server error.

Code

```
{
  "Name": "Vegetarian",
  "Price": 7.99,
  "Description": "Pizza topped with a variety of vegetables"
}
```

6. Delete an Item

- **Endpoint:** /items/:id
- **Method:** DELETE
- **Description:** Deletes an existing item from the database.
- **URL Parameters:**
    - id (String): The unique identifier of the item to delete.
- **Response:**
    - **Status 302:** Redirects to /items after successful deletion.
    - **Status 404:** Item not found.
    - **Status 500:** Internal server error.

Error Responses

For all endpoints, in case of an error, the API will return the following:

- **Status 500:** Internal server error with a message describing the issue.

Code

```
{
   "error": "Internal Server Error",
   "message": "Detailed error message"
}
```