### User Routes

* **Register**: **POST /api/v1/users** - Creates a new user account.
* **Login**: **POST /api/v1/users/login** - Authenticates a user and returns a token.
* **Logout**: **PUT /api/v1/users/logout** - Logs out a user, requiring token authentication. (user does get logged out)
* **Fetch User**: **GET /api/v1/users/:id** - Retrieves user details, requires token authentication. (only works with mongo \_id, need to work on that)
* **Delete User**: **DELETE /api/v1/users/:id** - Removes a user account, requires token authentication and admin rights. (doesn’t work but says it was successful, if it was to completely remove the account it did not work if it was only to make the account not active it worked
* **Edit User**: **PUT /api/v1/users/:id** - Updates user details, requires token authentication. (doesn’t work but says it was successful
* **Fetch All Users**: **GET /api/v1/users** - Retrieves all users, requires token authentication and admin rights.

### Category Routes

* **Create Category**: **POST /api/v1/categories** - Adds a new category, requires token authentication and admin rights.
* **Edit Category**: **PUT /api/v1/categories/:id** - Updates category details, requires token authentication and admin rights.
* **Delete Category**: **DELETE /api/v1/categories/:id** - Removes a category, requires token authentication and admin rights.
* **Fetch Category**: **GET /api/v1/categories/:id** - Retrieves a category detail. (it works only with the mongo \_id
* **Fetch All Categories**: **GET /api/v1/categories** - Lists all categories.

### Cuisine Routes

* **Create Cuisine**: **POST /api/v1/cuisines** - Adds a new cuisine, requires token authentication and admin rights. (works if no id is used)
* **Edit Cuisine**: **PUT /api/v1/cuisines/:id** - Updates cuisine details, requires token authentication and admin rights.(works using mongo \_is)
* **Delete Cuisine**: **DELETE /api/v1/cuisines/:id** - Removes a cuisine, requires token authentication and admin rights.
* **Fetch Cuisine**: **GET /api/v1/cuisines/:id** - Retrieves a cuisine detail.
* **Fetch All Cuisines**: **GET /api/v1/cuisines** - Lists all cuisines.

### Fooditem Routes

* **Create Fooditem**: **POST /api/v1/fooditems** - Adds a new fooditem, requires token authentication and admin rights.(did not work)
* **Edit Fooditem**: **PUT /api/v1/fooditems/:id** - Updates fooditem details, requires token authentication and admin rights. (did not work)
* **Delete Fooditem**: **DELETE /api/v1/fooditems/:id** - Removes a fooditem, requires token authentication and admin rights.(did not work)
* **Fetch Fooditem**: **GET /api/v1/fooditems/:id** - Retrieves a fooditem detail.
* **Fetch All Fooditems**: **GET /api/v1/fooditems** - Lists all fooditems.

Restaurant Routes

● **Create Restaurant**: **POST /api/v1/restaurants** - Adds a new restaurant, requires token authentication and admin rights.

* **Edit Restaurant**: **PUT /api/v1/restaurants/:id** - Updates restaurant details, requires token authentication and admin rights.(but did not take the id though still had to use mongo \_id)
* **Delete Restaurant**: **DELETE /api/v1/restaurants/:id** - Removes a restaurant, requires token authentication and admin rights. It works
* **Fetch Restaurant**: **GET /api/v1/restaurants/:id** - Retrieves a restaurant detail.(works only with mongo\_id)
* **Fetch All Restaurants**: **GET /api/v1/restaurants** - Lists all restaurants.
* **Fetch All Restaurants by Cuisine Name**: **GET /api/v1/restaurants?cuisine=<CuisineName> -** Lists all restaurants based on a cuisine.( it says it was a success but no details show)
* **Fetch All Restaurants by Category Name**: **GET /api/v1/restaurants?category=<CategoryName> -** Lists all restaurants based on a category. (it says it worked but no details shown)

### Menu Routes

* **Create Menu**: **POST /api/v1/menus** – Adds a new menu, requires token authentication and admin rights. (Error
* {
* "message": "Error creating menu",
* "error": "Error while creating menu: Menus validation failed: restaurantId: Path `restaurantId` is required., description: Path `description` is required."
* }
* **Edit Menu**: **PUT /api/v1/menus/:id** – Updates menu details, requires token authentication and admin rights.(not working)
* **Delete Menu**: **DELETE /api/v1/menus/:id** – Removes a menu, requires token authentication and admin rights.( completely removed from DB)
* **Fetch Menu Based on RestaurantId**: **GET /api/v1/menus/:restaurantId –** Retrieves a menu detail based on a restaurantId.

### Order Routes

* **Place Order**: **POST /api/v1/orders** - Submits a new order, requires token authentication.
* **Get Order Details**: **GET /api/v1/:userId/:orderId –** Retrieves the order details of a user’s order, requires token authentication.
* **User Orders**: **GET /api/v1/orders/:userId** - Retrieves a user's order history, requires token authentication.
* **All Orders**: **GET /api/v1/orders** - Lists order history of all users, requires token authentication and admin rights.
* **Change Order Status**: **PUT /api/v1/orders/:orderId** - Modifies the status of an order, requires token authentication and admin rights.