***API Documentation for Mobile Validation and Inventory System:***

1. ***Overview***

This system provides two main functionalities:

1. Mobile number validation using the NumVerify API , which will be used as a microservice for our project
2. Basic inventory management with Categories and Items using the JpaRepository library, which makes our work much easier and simpler.
3. ***Prerequisites***

* Java 17 or higher.
* Maven.
* PostgreSQL (or any other database you know)
* NumVerify API key (which is essential for the mobile-validation to workk).

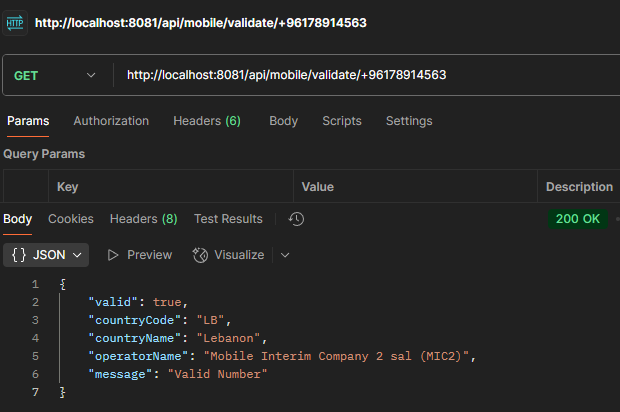
1. ***API Endpoints***
2. ***Mobile Validation Service:***

**Base URL:** <http://localhost:8081/api/mobile> , which is the url that the get validation request will be sent to.

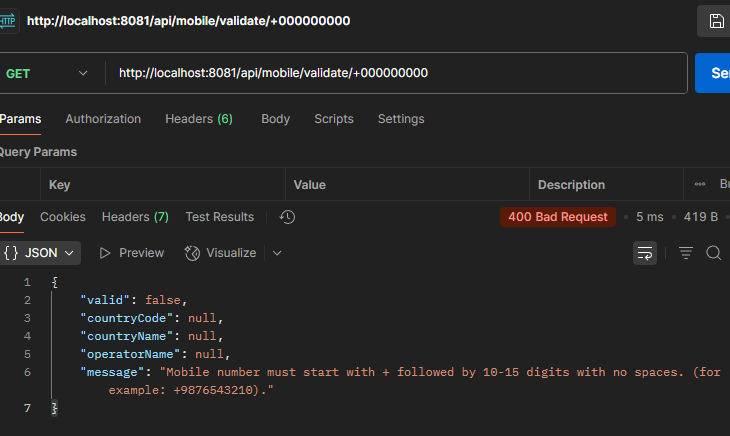
1. ***Validate Mobile Number***

* **Endpoint:** [http://localhost:8081/api/mobile/validate/{mobileNumber}](http://localhost:8081/api/mobile/validate/%7bmobileNumber%7d)
* **Method:** GET
* **Description:** Validates a mobile number using NumVerify microservicve API.
* **Request Parameters:**
  + mobileNumber (path parameter): Mobile number in the following format (for example: +96178914563) – this will be written at the end of URL
* **Response:**
  + 200 OK: Valid request
  + 400 Bad Request: spelling mistake or invalid number

**Example Response (Success):**



**Example Response (Error):**



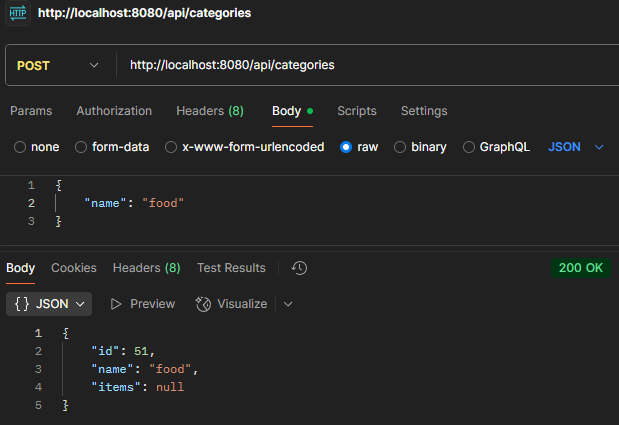
**Inventory Management System**

**Base URL:** <http://localhost:8080/api> , which is the URL where all our requests will be sent to.

**Categories**

<http://localhost:8080/api//categories>

* **Endpoint:** /categories
  + **GET:** Get all categories
  + **POST:** Create a new category
    - **Request Body:**

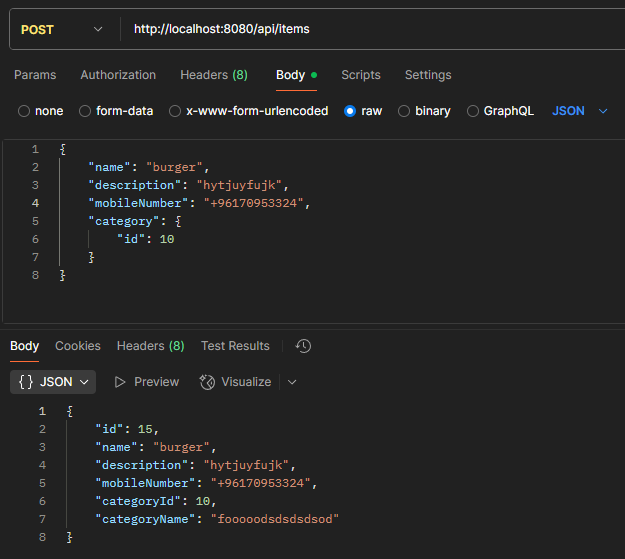


* **Endpoint:** /categories/{id}
  + **GET:** Get category by ID
  + **PUT:** Update category
  + **DELETE:** Delete category

**Items**

<http://localhost:8080/api/items>

* **Endpoint:** /items
  + **GET:** Get all items
  + **POST:** Create a new item
    - **Request Body:**

****

* **Endpoint:** /items/{id}
  + **GET:** Get item by ID
  + **PUT:** Update item
  + **DELETE:** Delete item

**Very Important Notes**

1. The mobile number must be in the following format in order to be checked if it is valid or invalid:
   * Starts with +
   * Followed by 10-15 digits
   * No spaces or other characters
2. When creating or updating items:
   * Mobile number is o0bligatory
   * Mobile number will be validated using the validation service, we don’t use the frontend for validation.
   * Category must be specified
3. The system doesn’t allow cascade deleting, that is:
   * You cannot delete a category that has associated items.
   * Every item must belong to a category, otherwise you get an error.
4. DTOs:

I used DTOs for items, since I was having a lot of bugs in my code, especially when displaying the category of the item created , in the frontend. Where it was always showing “No Category”, when it wasn’t the case.