# Assignment 4: Testing Patterns

**Introduction:**

```
Restaurant Catalog service that exposes an API for Users, Cart, and Restaurant
view services. Uses gateway routing pattern to pass requests upstream in a
microservice architecture.
```

**API Definitions:**

## */api/restaurants*

*Endpoint that allows users to get restaurants with a basic search or advanced search, allows owners to create restaurants, allows owners to delete restaurants, allows owners to update restaurants.*

**Payload Type:** All Endpoints only take JSON as a payload, any other data type will result in *HTTP 400*

**Supported Methods:**

- ***GET***
    - ***Get a list of restaurants that match the parameters specified***
    - ***Returns****: HTTP 201 on success ; HTTP 4xx on failure*
    - *Input:*
        - *Takes one required argument "zipcode"*
        - *Takes up to 5 additional arguments for 6 total*

Advanced Arguments Table

| Argument | Value |
|---|---|
| *"Zipcode" (required)* | *5 Digit INTEGER* |
| *"Name" (optional)* | *String* |
| *"City" (optional)* | *String* |
| *"Delivery" (optional)* | *INTEGER Boolean Value (0 or 1)* |
| *"Pickup" (optional)* | *INTEGER Boolean Value (0 or 1)* |
| *"Address" (optional)* | *String* |

- ***DELTE***
    - ***Remove a restaurant and all its associated items in the database***

    - ***Returns:*** *HTTP 200 on success ; HTTP 4xx on failure*
    - *Input:*
        - *3 required parameters*

| **Argument** | **Value (datatype)** |
|---|---|
| *“email”* | *string* |
| *“password”* | *string* |
| *“Id” (restaurant unique id)* | *int* |

- ***PUT***
    - ***Add a new restaurant to the database***
    - *Input:*
        - *10 required parameters*

| **Argument** | **Value (datatype)** |
|---|---|
| “name” | string |
| *“address”* | *string* |
| *“city”* | *string* |
| *“zip”* | *5 digit integer* |
| *“state”* | *string* |
| *“Lat” (GPS restaurant latitude)* | *Float* |
| *“Phone” (ex 12825550909)* | *Int (1+areacode+phonenum)* |
| *“Long” (GPS restaurant longitude)* | *float* |
| *“Delivery” (truth value if the restaurant provides delivery)* | *Boolean int (1 = true, 0 = false)* |
| *“Pickup” (truth value if restaurant provides pickup services)* | *Boolean int (1=true, 0=false)* |
| *“Hours” (operating hours)* | *string* |

## */api/users*

*Endpoint that allows members to sign up for an account, authenticate and delete their account.*

**Payload Type:** All Endpoints only take JSON as a payload, any other data type will result in *HTTP 400*

**Supported Methods:**

- ***GET***
  - ***Authenticate a user given their username and password***
  - ***Returns**: HTTP 200 on success ; HTTP 4xx on failure*
    - *True or False boolean value*
  - *Input:*
    - *Two required arguments ("email", "password")*

Arguments Table

| Argument | Value |
|---|---|
| *"email" (required)* | *string* |
| *"Password" (required)* | *string* |

- ***DELETE***
  - ***Delete a user's account (CANNOT BE DONE WHEN THE USER'S ID HAS ACTIVE ORDERS)***
  - ***Returns:** HTTP 200 on success ; HTTP 4xx on failure*
  - *Input:*
    - *3 required parameters*

| **Argument** | **Value (datatype)** |
|---|---|
| *"email"* | *string* |
| *"password"* | *string* |
| *"Id" (user unique id)* | *int* |

- ***PUT***
  - ***Create a new user***
  - *Input:*
    - *9 required parameters*

| **Argument** | **Value (datatype)** |
|---|---|
| "name" | string |
| *"address"* | *string* |
| *"city"* | *string* |

| | |
|---|---|
| *“zip”* | *5 digit integer* |
| *“state”* | *string* |
| *“Lat” (GPS restaurant latitude)* | *Float* |
| *“Long” (GPS restaurant longitude)* | *float* |
| *“Delivery” (truth value if the restaurant provides delivery)* | *Boolean int (1 = true, 0 = false)* |
| *“Pickup” (truth value if restaurant provides pickup services)* | *Boolean int (1=true, 0=false)* |
| *“Hours” (operating hours)* | *string* |

## */api/carts*

*Endpoint that allows users to add and delete from their carts. Also allows users to view the items in their cart. Endpoint also allows users to make orders out of the items in their carts.*

**Payload Type:** All Endpoints only take JSON as a payload, any other data type will result in *HTTP 400*

**Supported Methods:**

- ***GET***
    - ***Get a list of items currently in a user’s cart, with a pre-tax subtotal value***
    - ***Returns****: HTTP 201 on success ; HTTP 4xx on failure*
    - *Input:*
        - *Takes one required argument “email”*

Advanced Arguments Table

| Argument | Value |
|---|---|
| *“email” (required)* | *string* |

- ***DELTE***
    - ***Remove item(s) that are currently in a users cart.***
    - ***Returns:*** *HTTP 200 on success ; HTTP 4xx on failure*
    - *Input:*

- *2 required parameters*

| Argument | Value (datatype) |
|---|---|
| *“email”* | *string* |
| *“items”* | *List of int’s (where int’s are item ID’s)* |

- ***PUT***
  - ***Add new item(s) to a user’s cart***
  - ***Returns:*** *HTTP 200 on success ; HTTP 4xx on failure*
  - *Input:*
    - *2 required parameters*

| Argument | Value (datatype) |
|---|---|
| “email” | string |
| *“items”* | *List of int(s) (where int is a product ID)* |

- ***POST***
  - ***Take all items currently in a user’s cart and create an order with them, then empty the cart.***
  - ***Returns:*** *HTTP 200 on success ; HTTP 4xx on failure*
  - *Input:*
    - *1 required parameter*

| Argument | Value (datatype) |
|---|---|
| “email” | string |