

The screenshot shows the Postman interface. At the top, a GET request is defined for the URL `http://localhost:8081/?input=breakfast 1,2,3`. The 'Params' tab is active, displaying a table of query parameters.

	KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/>	input	breakfast 1,2,3			
	Key	Value	Description		

Below the request setup, the 'Body' tab is active, showing the response in 'Pretty' format. The status is 200 OK, with a response time of 93 ms and a size of 264 B. The JSON response is as follows:

```

1 {
2   "serviceStatus": {
3     "statusCode": "200",
4     "success": true,
5     "errorMessage": ""
6   },
7   "meal": "Eggs, Toast, Coffee"
8 }

```

For this RESTful API project, I use SpringBoot which is designed in MVC.

Port number: 8081

Model: I create a class called Meal and 3 of its children are Breakfast, Lunch and Dinner

View: Since this one has no UI so I would mention Postman

Controller: MealController.java

For the testing, I use both Unit Test and Postman

I have 3 unit tests (breakfast, lunch and dinner) in the EviveApplicationTests file. This includes all the test cases mentioned.

1. An order consists of a meal and collection of comma separated item Ids.
2. The system should return the name of the items ordered
3. The system should always return items in the following order: meal, side, drink
4. If multiple items are ordered, the number of items should be indicated
5. Each order must contain a main and a side

6. If no drink is ordered, water should be returned
7. At breakfast, multiple cups of coffee can be ordered
8. At lunch, multiple sides can be ordered
9. At dinner, dessert must be ordered
10. At dinner, water is always provided

There is some unclear description in the rules so I will state all my assumption:

Input: string int,int,int,...

There is a space between string and int,int,int,...

There is no space and only “,” between numbers.

Breakfast: Main: 1
 Side: 1
 Drink: >= 0
 Desert: 0

Lunch: Main: 1
 Side: >= 1
 Drink: [0,1]
 Desert: 0

Dinner: Main: 1
 Side: 1
 Drink: [0,1]
 Desert: 1

If the type of meal is not breakfast, lunch or dinner. It will print out the Invalid Type Error Message.

For the Postman, I also add more info to the response by using ServiceStatus.

- Status code: 200 is ok, and 500 means Internal Server Error.
- Success: success status of the request
- Error Message: this will show up if the input is invalid
- Meal: the output that we will get if the input is valid.

The reason why I add is because I think it will give us more detail of what is going wrong with the request, and easier for us to debug when we want to scale up the project.

Here is some of the input/output using Postman:

Input	Output
-------	--------

breakfast 1,2,3	<pre>{ "serviceStatus": { "statusCode": "200", "success": true, "errorMessage": "" }, "meal": "Eggs, Toast, Coffee" }</pre>
breakfast 1,2,3,3,3	<pre>{ "serviceStatus": { "statusCode": "200", "success": true, "errorMessage": "" }, "meal": "Eggs, Toast, Coffee(3)" }</pre>
breakfast 1	<pre>{ "serviceStatus": { "statusCode": "500", "success": false, "errorMessage": "Unable to process: Side is missing" }, "meal": "" }</pre>
breakfast 1,2,3,4	<pre>{ "serviceStatus": { "statusCode": "500", "success": false, "errorMessage": "Unable to process: Desert is not allowed to be ordered for breakfast" }, "meal": "" }</pre>
lunch 1,2	<pre>{ "serviceStatus": { "statusCode": "200",</pre>

	<pre> "success": true, "errorMessage": "" }, "meal": "Salad, Chips, Water" }</pre>
lunch 1,1,2,3	<pre> { "serviceStatus": { "statusCode": "500", "success": false, "errorMessage": "Unable to process: Salad cannot be ordered more than once" }, "meal": "" }</pre>
lunch	<pre> { "serviceStatus": { "statusCode": "500", "success": false, "errorMessage": "Main is missing, side is missing" }, "meal": "" }</pre>
dinner 1,2,3,4	<pre> { "serviceStatus": { "statusCode": "200", "success": true, "errorMessage": "" }, "meal": "Steak, Potatoes, Wine, Water, Cake" }</pre>
dinner 1,2,3	<pre> { "serviceStatus": { "statusCode": "500", "success": false,</pre>

	<pre> "errorMessage": "Unable to process: Desert is missing" }, "meal": "" } </pre>
dinner 1,1,2,2,3,3,4,4	<pre> { "serviceStatus": { "statusCode": "500", "success": false, "errorMessage": "Unable to process: Steak cannot be ordered more than once, Potatoes cannot be ordered more than once, Wine cannot be ordered more than once, Cake cannot be ordered more than once" }, "meal": "" } </pre>
brunch	<pre> { "serviceStatus": { "statusCode": "500", "success": false, "errorMessage": "Main is missing, side is missing" }, "meal": "" } </pre>
brunch 1,2	<pre> { "serviceStatus": { "statusCode": "500", "success": false, "errorMessage": "INVALID TYPE OF MEAL" }, "meal": "" } </pre>