[Online Recipe Management - Exception Handling](https://cognizant.tekstac.com/mod/vpl/view.php?id=9510)

**Grade settings**: Maximum Grade: 100  
**Disable external file upload, paste and drop external content**: Yes  
**Based on**: [Online Recipe Management - Exception Handling](https://cognizant.tekstac.com/mod/vpl/view.php?id=9510)  
**Run**: Yes **Save**: Yes **Evaluate**: Yes  
**Automatic grade**: Yes **Maximum execution time**: 120 s **Maximum memory used**: 1 GiB

**Objective:**

To work with @RestController, @RequestMapping annotation and Exception Handling (User-defined Exception)

**Concept Explanation:**

1. User-defined exceptions allow for specific error handling in a REST API, enabling developers to return meaningful error messages and appropriate HTTP status codes like 404 Not Found when resources are unavailable.

**Concept Implementation:**

1. In our recipe management system, when querying the endpoint **/recipe/find/123** via the GET method, we need to handle cases where the recipe data isn't found.
2. To effectively manage this, we'll implement a custom exception that, when a recipe ID doesn't match any records, will trigger a specific HTTP status code 404 signaling that the resource is unavailable.

**Online Recipe Management System**

FoodieX is a leading culinary platform that offers a wide range of recipes to its users. They have recently implemented a Spring REST solution for managing recipes. Now, FoodieX wants to enhance their system by developing a REST service to retrieve recipe details using Maven.

In the RecipeController, create the following service:

**Request URL --> /recipe/find/123** : This service should invoke the **findRecipe()** method of the RecipeService class and return the recipe object for the given recipe ID.

If the recipe for the given ID is not found, a custom exception named **RecipeNotFoundException**should be thrown with the message **"Recipe not found for the provided ID"** in the controller class. Additionally, instead of returning a 500 error, the status code should be changed to 404 to indicate that the resource (recipe) was not found.

**Note:**

* **RecipeService** and **Recipe**classes are already provided as part of the code skeleton.  Do not alter the same.
* Inject the **RecipeService**inside the **RecipeController**and invoke the appropriate method.
* The data returned from the controller should be in JSON format.
* Partial Maven Solution for the same is provided. Do not alter the className/packageName/MethodName.
* You can add new methods, attributes, or classes if necessary.

[Next activity](https://cognizant.tekstac.com/mod/vpl/view.php?id=20588&forceview=1)**[Movie Ticket Booking REST Controller App](https://cognizant.tekstac.com/mod/vpl/view.php?id=20588&forceview=1)**

**Kudos!**You have earned some XP points.