[Patient Management - Exception Handling](https://cognizant.tekstac.com/mod/vpl/view.php?id=9507)

**Grade settings**: Maximum Grade: 100  
**Based on**: [Patient Management - Exception Handling](https://cognizant.tekstac.com/mod/vpl/view.php?id=9507)  
**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 patient management system, when querying the endpoint **/pms/find/101** via the GET method, we need to handle cases where the patient data isn't found.
2. To effectively manage this, we'll implement a custom exception that, when a patient ID doesn't match any records, will trigger a specific HTTP status code 404 signaling that the resource is unavailable.

**PatientManagementSystem**

ZEE Health Solutions has already implemented a Spring REST solution for managing patient records. The client has a new requirement. Help Zee-Health to enhance the process by developing a Rest Service using Maven.

In the **PatientController**, create the below service:

Request URL --> /pms/find/101:

This service should invoke the **searchPatient**() method of the PatientService class and return the patient object for the given patient Id.

If the patient for the given patient ID is not found, then a user-defined exception, **PatientNotFoundException**, with the message **"No such patient id"** has to be thrown in the Controller class.

Also, instead of a 500 error, the status code should be changed to 404. [**Hint:**Use the appropriate annotation above the PatientNotFoundException class with the appropriate status code]

**Note:**

* PatientService and Patient classes are already provided as part of the code skeleton. Do not alter the same.
* Inject the PatientService inside the PatientController and invoke the appropriate method.
* The data returned from the Controller should be a JSON.
* A partial Maven solution for the same is provided. Do not alter the className/packageName/MethodName.
* You can add new methods/attributes/classes if required.

[Next activity](https://cognizant.tekstac.com/mod/vpl/view.php?id=20585&forceview=1)**[Inventory Management System - Get and Post](https://cognizant.tekstac.com/mod/vpl/view.php?id=20585&forceview=1)**

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