**Inventory Management System - Get and Post**

**Grade settings**: Maximum Grade: 100  
**Disable external file upload, paste and drop external content**: Yes  
**Based on**: [Inventory Management System - Get and Post](https://cognizant.tekstac.com/mod/vpl/view.php?id=9508)  
**Run**: Yes **Save**: Yes **Evaluate**: Yes  
**Automatic grade**: Yes **Maximum execution time**: 120 s **Maximum memory used**: 1 GiB

**Objective:**

To work with @RestController, http methods - GET and POST

**Concept Explanation:**

1. The **HTTP POST** method is like the web's 'create' button, pushing new data to the server. It's essential for tasks like submitting forms or registering new users, perfect for scenarios where new information needs to be securely stored.
2. The**HTTP GET** method is the go-to choice for safely retrieving data. It's like the 'read' function of the web, pulling information without altering any server data, perfect for things like viewing web pages or getting search results.

**Concept Implementation:**

1. In the REST Controller, named **OrderController** for Inventory Management, we should map the endpoint**/BHO/addOrder** to the annotation corresponding to the **HTTP POST** request, as it is designed for adding new orders to the system.
2. We should map the endpoint /**BHO/findOrderById/101** to the annotation corresponding to the **HTTP GET**request to access order details.
3. The endpoint **/BHO/viewAllOrders** should be mapped to another **HTTP GET** request, allowing for the retrieval of details for all orders.

**Inventory Management System**

[**Click here to download the code skeleton**](https://cognizant.tekstac.com/mod/vpl/viewfile.php/21050/mod_vpl/intro/InventoryManagementSystem%20%281%29.zip)

A local bookstore, Book Haven, aims to streamline its inventory management system by incorporating a web service for handling book orders, updating book availability, cancelling orders, and retrieving order details. Help Book Haven to automate the above process by developing Rest Service using Maven to add and retrieve the order details.

As a first step, the **OrderController** which is RestController should be created with the following services:

**Request URL -->/BHO/findOrderById/101**: This service should invoke the **findOrderById()** method of the OrderService class and return the Order object for the given order ID.

**Request URL -->/BHO/viewAllOrders:** This service should invoke the **viewAllOrders()** method of the OrderService class and return all the orders.

**Request URL -->/BHO/addOrder:**This service should invoke the **addOrder()**method of the OrderService class and add the order details to the list.

**Note:**

* The **OrderService** class is provided with two sample order details and added to the list as part of the skeleton. Do not alter the same.
* Partial maven solution for the same is provided. Do not alter the className/packageName/MethodName.
* You can add new methods/attributes/classes if required.
* In the **OrderService** class , include the required business logic.
* **The data returned from the Controller should be a JSON.**
* Inject the **OrderService** inside the **OrderController** and invoke the appropriate methods.

[Next activity](https://cognizant.tekstac.com/mod/vpl/view.php?id=20586&forceview=1)**[Inventory Management System - Put and Delete](https://cognizant.tekstac.com/mod/vpl/view.php?id=20586&forceview=1)**

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