[Online Event Management - Get and Post](https://cognizant.tekstac.com/mod/vpl/view.php?id=9505)

**Grade settings**: Maximum Grade: 100  
**Based on**: [Online Event Management - Get and Post](https://cognizant.tekstac.com/mod/vpl/view.php?id=9505)  
**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 **EventController** for Event Management, we should map the endpoint**/OEMS/addEvent** to the annotation corresponding to the **HTTP POST** request, as it is designed for adding new events to the system.
2. We should map the endpoint **/OEMS/findEventById/4** to the annotation corresponding to the **HTTP GET**request to access event details.
3. The endpoint **/OEMS/findAllEvents** should be mapped to another **HTTP GET** request, allowing for the retrieval of details for all events.

**Online Event Management**

Online Event Management System desires to use a web service for adding new events to their schedule and retrieving details of existing events. Help them automate this process by developing a REST Service using Maven to add and retrieve event details.

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

Request URL --> /OEMS/findEventById/4 : This service should invoke the **findEventById**method of the EventService class and return the event object for the given event ID.

Request URL --> /OEMS/findAllEvents : This service should invoke the **findAllEvents**method of the EventService class and return all the events.

Request URL--> /OEMS/addEvent : This service should invoke the **addEvent** method of the EventService class to add a new event to the list.

**Note:**

* The EventService class is provided with four sample events added to a 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 EventService class, include the required business logic.
* The data returned from the Controller should be JSON.
* Inject the EventService inside the EventController and invoke the appropriate methods.