EXPERIMENT NO. 09: TESTING RESTFUL WEB SERVICES IN SPRING BOOT

Objective

To create a Java application using Spring Boot that demonstrates testing of RESTful web services.

Theory

Testing RESTful web services ensures they function as intended under different scenarios. Spring Boot provides built-in support for testing REST APIs using JUnit and MockMvc, enabling simulation of HTTP requests and assertion of expected responses.

Create the Employee Entity

```
@Entity

public class Employee {
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   private Long id;
   private String name;
   private String department;
}
```

Create the Employee Controller

```
@RestController
@RequestMapping("/api/employees")
public class EmployeeController {
  }
```

Write Integration Tests Using JUnit and MockMvc

```
@RunWith(SpringRunner.class)

@SpringBootTest
@AutoConfigureMockMvc
public class EmployeeControllerIntegrationTest {

   @Autowired
   private MockMvc mockMvc;
```

```
    @Test
    public void testGetAllEmployees() throws Exception {
        mockMvc.perform(get("/api/employees"))
                .andExpect(status().isOk())
                .andExpect(jsonPath("$", hasSize(0)));
    }

}
```

How to Run the Tests

Use your IDE (IntelliJ, Eclipse, or VS Code) to run the `EmployeeControllerIntegrationTest` class as a JUnit test. Ensure the Spring Boot context loads correctly and all endpoint assertions pass.

Explanation

1. Employee entity is defined similarly to earlier experiments.
2. REST Controller is set up to manage employee data.
3. MockMvc simulates REST calls to the controller for testing without needing to deploy the application.
4. JUnit asserts expected HTTP status and JSON structure in the response.

Important Note

Always include tests for valid, invalid, and edge case scenarios in your test class. Ensure that the APIs return proper status codes and handle exceptions gracefully.

Conclusion

This experiment illustrates the importance of testing REST APIs using JUnit and MockMvc. Proper testing ensures application reliability, especially in production-ready enterprise systems.