

# Sample REST Demo application

## Step-02

### Main Method

```
package com.suji.sample;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SampleDemoFirstApplication {

    public static void main(String[] args) {
        SpringApplication.run(SampleDemoFirstApplication.class, args);
    }
}
```

---

### Employee Model

```
package com.suji.sample.model;

public class Employee {
    private int id;
    private String name;
    private String dept;
    private int salary;

    public Employee() {
    }

    public Employee(int id, String name, String dept, int salary) {
        this.id = id;
        this.name = name;
        this.dept = dept;
        this.salary = salary;
    }
    // Setters and Getters
}
```

---

### TestControllerEasy

```
package com.suji.sample.controller;

import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
```

```

import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.suji.sample.model.Employee;

@RestController
@RequestMapping("/easy")
public class TestControllerEasy {

    // NOTE: No two APIs must not have same METHOD and SAME Mapping.

    // _____GET MAPPING_____//
    // GET Method is used for READ/GET data

    // Say Hello
    @GetMapping("/hi")
    public String sayHi() {
        return "Hello world";
    }

    // Add two numbers
    // "Path variable name" and "Method parameter variable" name must be same.

    @GetMapping("/add/{num1}/{num2}")
    public String add(@PathVariable int num1, @PathVariable int num2) {

        return "Sum is: " + (num1 + num2);
    }

    // Reverse String
    // "Path variable name" and "Method parameter variable" name must be same.

    @GetMapping("/reverse/{inputStr}")
    public String add(@PathVariable String inputStr) {
        return "Reverse String is: " + new StringBuilder(inputStr).reverse();
    }

    // Get Fake employee
    // "Path variable name" and "Method parameter variable" name must be same.

    @GetMapping("/fake/{empId}")
    public Employee getFake(@PathVariable int empId) {

        return new Employee(empId, "FAke name", "Fake Dept", 10000);
    }

    // _____POST MAPPING_____//
    // POST Method is used for CREATE/SAVE data

    // Double Employee salary and id

```

```

// "Request Body" must be provide as JSON Object

@PostMapping("/double")
public Employee doubleIt(@RequestBody Employee e) {
    return new Employee(e.getId() * 2, e.getName(), e.getDept(), e.getSalary() *
2);
}

// Employee Details in String
// "Request Body" must be provide as JSON Object

@PostMapping("/instr")
public String details(@RequestBody Employee e) {
    return e.toString();
}

// _____PUT MAPPING_____//
// PUT Method is used for UPDATE data

// Update Employee
// "Path variable name" and "Method parameter variable" name must be same.
// "Request Body" must be provide as JSON Object

@PutMapping("/update/{empId}")
public Employee updateEmployee(@PathVariable int empId, @RequestBody Employee emp)
{
    Employee newEmp = new Employee(empId, "Updated Name", "Updated Dept", 10000);
    return newEmp;
}

// _____DELETE MAPPING_____//
// DELETE Method is used for DELETE data

// Update Employee
// "Path variable name" and "Method parameter variable" name must be same.

@DeleteMapping("/delete/{empId}")
public String deleteEmployeeById(@PathVariable int empId) {
    return "Employee with id '" + empId + "' has been deleted";
}
}

```

---

## application.properties

```

# Server Port
server.port=8080

# Open API Available at http://localhost:8080/api-docs/
springdoc.api-docs.path=/api-docs

```

```
# Swagger UI Available at http://localhost:8080/swagger.html
springdoc.swagger-ui.path=/swagger.html
```

---

## **pom.xml**

```
<dependencies>

    <dependency>
        <groupId>org.springdoc</groupId>
        <artifactId>springdoc-openapi-ui</artifactId>
        <version>1.6.4</version>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
</dependencies>
```