ReST

Representational State Transfer

SOAP webservice

Simple Object Access Protocol

Disadvantage of SOAP

Only supports XML format

Entire web service is represented by one URL

ReSTful webservices REST API

Supports multiple formats including XML, JSON, text, HTML

For every method, one URL

Every rest api method call is independent (no conversation)

No state management

http methods

GET

POST

PUT

DELETE

PATCH

Demo:

Create a spring boot rest api now.

1. Create a spring boot application using <https://start.spring.io/>
2. Download (generate) and unzip.
3. Copy the folder to your eclipse workspace
4. Import the project (existing maven projects)
5. Run the project (go to Application.java that has main method) ctrl+F11
6. If port number is not available, change the port (server.port=8081) or kill the process that uses 8080
7. Create a class under base package (package that contains Application.java)

package com.ust.rest;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class MyController {

@GetMapping

public String home()

{

return "Hi this is my first restful webservice";

}

}

1. Run/ restart the project is already run.
2. Go to <http://localhost:8080/>

RestController = Controller + ResponseBody

package com.ust.rest;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ResponseBody;

@Controller

public class MyController {

@GetMapping

@ResponseBody

public String home()

{

return "Hi this is my first restful webservice....still working fine";

}

}

package com.ust.rest;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class MyController {

@GetMapping

public String home()

{

return "Hi this is my first restful webservice....still working fine";

}

}

In MVC, we had Model, View and Controller

In Rest api, we have only Model and Controller (no view)

Model

Data definition

Data access

Steps:

1. Add the following starter dependency to pom.xml

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.17</version>

</dependency>

1. In application.properties:

spring.datasource.url=jdbc:mysql://localhost:3306/ust1

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

spring.datasource.username=root

spring.datasource.password=

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.show\_sql=true

1. Create entity class

We have already created Employee class

In that class, we need to mark that class is an entity class

Reminder: lets understand annotations of jpa

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer id;

private String name;

private String department;

1. Create an interface that extends JpaRepository

package com.ust.rest.model;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer>

{

}

1. Controller

package com.ust.rest;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;

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.ust.rest.model.Employee;

import com.ust.rest.model.EmployeeRepository;

@RestController

@RequestMapping("employee")

public class EmployeeController {

@Autowired

private EmployeeRepository er;

@GetMapping

public List<Employee> retrieveAllEmployees()

{

return er.findAll();

}

//find Employee by id

@GetMapping("/{id}")

public Employee findEmployeeById(@PathVariable("id") Integer id)

{

Employee emp=null;

Optional<Employee> temp = er.findById(id);

if(temp.isPresent())

emp=temp.get();

return emp;

}

@PostMapping

public Employee addEmployee(@RequestBody Employee employee)

{

return er.save(employee);

}

@PutMapping

public Employee updateEmployee(@RequestBody Employee employee)

{//we will improvise this later

return er.save(employee);

}

@DeleteMapping("/{id}")

public Employee deleteEmployee(@PathVariable("id") Integer id)

{

Employee emp = findEmployeeById(id);

if(emp!=null)

{

er.delete(emp);

}

return emp;

}

}