**Spring REST using Spring Boot 3**

**(WEEK 4)**

SUPERSET ID: 6410152

EX 1:Create a Spring Web Project using Maven:

<dependencies>

<!-- Spring Web -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.32</version>

</dependency>

<!-- Servlet API (provided scope) -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!-- JSTL -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

</dependencies>

**Add Project Structure:**

Manually create these folders inside src/main:

src

└── main

├── java

├── resources

└── webapp

└── WEB-INF

Inside WEB-INF, create a file:  
 web.xml

Create web.xml:

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

version="3.1">

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

Add Spring Config:

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:component-scan base-package="com.springdemo"/>

<mvc:annotation-driven/>

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix" value="/WEB-INF/views/"/>

<property name="suffix" value=".jsp"/>

</bean>

</beans>

Add Controller:

package com.springdemo;

import org.springframework.stereotype.Controller;

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

import org.springframework.web.servlet.ModelAndView;

@Controller

public class HomeController {

@RequestMapping("/")

public ModelAndView home() {

ModelAndView mv = new ModelAndView("home");

mv.addObject("msg", "Welcome to Spring MVC Web Project with Maven!");

return mv;

}

}

Add JSP Page:

<%@ page contentType="text/html;charset=UTF-8" %>

<html>

<head>

<title>Home Page</title>

</head>

<body>

<h2>${msg}</h2>

</body>

</html>

**- Run on Apache Tomcat**

**EX 2: Spring Core – Load Country from Spring Configuration XML**

**1. Create a New Java Project in Eclipse**

* Open Eclipse
* Go to File > New > Java Project
* Name it: SpringCountryApp
* Click **Finish**

Create the Country Class:

package com.springdemo;

public class Country {

private String name;

private String capital;

public void setName(String name) {

this.name = name;

}

public void setCapital(String capital) {

this.capital = capital;

}

public void display() {

System.out.println("Country Name: " + name);

System.out.println("Capital City: " + capital);

}

}

**Create spring-config.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="countryBean" class="com.haru.springdemo.Country">

<property name="name" value="India"/>

<property name="capital" value="New Delhi"/>

</bean>

</beans>

Create the Main App:

package com.springdemo;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("spring-config.xml");

Country country = (Country) context.getBean("countryBean");

country.display();

}

}

**Output:**

Country Name: India

Capital City: New Delhi

EX3: Hello World RESTful Web Service

1. **Create a Spring Boot Project:**

**Option A: Using** [**Spring Initializr**](https://start.spring.io/)

* Go to <https://start.spring.io>
* Project: Maven
* Language: Java
* Spring Boot Version: 3.x.x (latest stable)
* Group: com.rest
* Artifact: hello-rest
* Dependencies:
  + **Spring Web**
* Click **Generate**

Create a REST Controller:

package com.rest;

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

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

@RestController

public class HelloController {

@GetMapping("/hello")

public String hello() {

return "Hello World from Spring Boot REST API!";

}

}

Verify Main Class:

package com.rest;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class HelloRestApplication {

public static void main(String[] args) {

SpringApplication.run(HelloRestApplication.class, args);

}

}

**-Run the Application**

[**http://localhost:8080/hello**](http://localhost:8080/hello)

**OUTPUT:**

Hello World from Spring Boot REST API**!**

EX4: **REST - Country Web Service**

**1. Create Spring Boot Project**

Use [Spring Initializr](https://start.spring.io):

* Group: com.haru.rest
* Artifact: country-service
* Dependencies:
  + **Spring Web**

**2.Create Country Model**

package com.rest.model;

public class Country {

private String name;

private String capital;

private long population;

// Constructor

public Country(String name, String capital, long population) {

this.name = name;

this.capital = capital;

this.population = population;

}

// Getters

public String getName() { return name; }

public String getCapital() { return capital; }

public long getPopulation() { return population; }

}

1. **Create Controller:**

package com.rest.controller;

import com.rest.model.Country;

import org.springframework.web.bind.annotation.\*;

import java.util.\*;

@RestController

@RequestMapping("/api/countries")

public class CountryController {

private List<Country> countries = new ArrayList<>(Arrays.asList(

new Country("India", "New Delhi", 1393409038),

new Country("Japan", "Tokyo", 125800000),

new Country("France", "Paris", 67390000)

));

@GetMapping

public List<Country> getAllCountries() {

return countries;

}

@GetMapping("/{name}")

public Country getCountryByName(@PathVariable String name) {

return countries.stream()

.filter(c -> c.getName().equalsIgnoreCase(name))

.findFirst()

.orElse(null);

}

}

**Main Application Class:**

package com.rest;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryServiceApplication {

public static void main(String[] args) {

SpringApplication.run(CountryServiceApplication.class, args);

}

}

OUTPUT:

{

"name": "India",

"capital": "New Delhi",

"population": 1393409038

}

EX5: REST - Get country based on country code

1. **Model: Add Country Code:**

package com.rest.model;

public class Country {

private String code;

private String name;

private String capital;

private long population;

public Country(String code, String name, String capital, long population) {

this.code = code;

this.name = name;

this.capital = capital;

this.population = population;

}

public String getCode() { return code; }

public String getName() { return name; }

public String getCapital() { return capital; }

public long getPopulation() { return population; }

}

1. **Controller: Fetch by Country Code:**

package com.haru.rest.controller;

import com.haru.rest.model.Country;

import org.springframework.web.bind.annotation.\*;

import java.util.\*;

@RestController

@RequestMapping("/api/countries")

public class CountryController {

private List<Country> countries = new ArrayList<>(Arrays.asList(

new Country("IN", "India", "New Delhi", 1393409038),

new Country("JP", "Japan", "Tokyo", 125800000),

new Country("FR", "France", "Paris", 67390000)

));

@GetMapping

public List<Country> getAllCountries() {

return countries;

}

@GetMapping("/code/{code}")

public Country getCountryByCode(@PathVariable String code) {

return countries.stream()

.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null);

}

}

1. **Main Application:**

package com.rest;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryServiceApplication {

public static void main(String[] args) {

SpringApplication.run(CountryServiceApplication.class, args);

}

}

Output:

"code": "IN",

"name": "India",

"capital": "New Delhi",

"population": 1393409038

EX 6: Create authentication service that returns JWT:

**1. Spring Boot Project Setup**

Use [Spring Initializr](https://start.spring.io):

* Group: com.jwt
* Artifact: jwt-auth-service
* Dependencies:
  + **Spring Web**
  + **Spring Security**
  + **Spring Boot DevTools**
  + **Spring Boot Starter Validation**
  + *(Optional: Spring Data JPA, H2 if using DB)*

**Add jjwt in pom.xml:**

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-api</artifactId>

<version>0.11.5</version>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-impl</artifactId>

<version>0.11.5</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-jackson</artifactId>

<version>0.11.5</version>

<scope>runtime</scope>

</dependency>

**JWT Utility Class:**

package com.jwt.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import java.security.Key;

import java.util.Date;

import org.springframework.stereotype.Component;

@Component

public class JwtUtil {

private final Key key = Keys.secretKeyFor(SignatureAlgorithm.HS256);

public String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuer("haru-auth-service")

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 60)) // 1 hour

.signWith(key)

.compact();

}

}

**User DTO:**

package com.jwt.dto;

public class AuthRequest {

private String username;

private String password;

// Getters & Setters

public String getUsername() { return username; }

public void setUsername(String username) { this.username = username; }

public String getPassword() { return password; }

public void setPassword(String password) { this.password = password; }

}

Auth Controller:

package com.jwt.controller;

import com.jwt.dto.AuthRequest;

import com.jwt.util.JwtUtil;

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

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/api/auth")

public class AuthController {

@Autowired

private JwtUtil jwtUtil;

// Dummy authentication

@PostMapping("/login")

public String login(@RequestBody AuthRequest request) {

if ("admin".equals(request.getUsername()) && "password".equals(request.getPassword())) {

return jwtUtil.generateToken(request.getUsername());

} else {

throw new RuntimeException("Invalid Credentials 😓");

}

}

}

**Security Config (Allow /api/auth/login):**

package com.jwt.config;

import org.springframework.context.annotation.Bean;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

@EnableWebSecurity

public class SecurityConfig {

@Bean

public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

http.csrf().disable()

.authorizeHttpRequests(auth -> auth

.requestMatchers("/api/auth/login").permitAll()

.anyRequest().authenticated()

);

return http.build();

}

}

**Test with Postman**

* **Method**: POST
* **URL**: http://localhost:8080/api/auth/login
* **Body (raw/JSON)**:

"username": "admin",

"password": "password"

**You’ll get :**

"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbiIsImlzcyI6ImhhcnUtYXV0aC1zZXJ2aWNl..."