Cognizant - DN 4.0 I Deep Skilling

Week-4

Create a Spring Web Project using Maven

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example.demo</groupId>

<artifactId>spring-web-maven-project</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>war</packaging>

<properties>

<java.version>1.8</java.version>

<spring.version>5.3.32</spring.version>

</properties>

<dependencies>

<!-- Spring Web -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- JSTL -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- Servlet API -->

<dependency>

<groupId>javax.servlet</groupId>

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

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!-- Logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.7.30</version>

</dependency>

</dependencies>

<build>

<finalName>spring-web-maven-project</finalName>

</build>

</project>

package com.example.demo.controller;

import org.springframework.stereotype.Controller;

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

import org.springframework.ui.Model;

@Controller

public class HomeController {

@GetMapping("/")

public String home(Model model) {

model.addAttribute("message", "Welcome to Spring Web MVC with Maven!");

return "home";

}

}

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}

Spring Core – Load Country from Spring Configuration XML

package com.example.model;

public class Country {

private String name;

private String code;

// Getters and Setters

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

// toString method for easy display

@Override

public String toString() {

return "Country{name='" + name + "', code='" + code + "'}";

}

}

<?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">

<!-- Define Country bean -->

<bean id="country" class="com.example.model.Country">

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

<property name="code" value="IN"/>

</bean>

</beans>

package com.example.main;

import com.example.model.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

// Load Spring configuration

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

// Retrieve Country bean

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

// Display country details

System.out.println("Loaded Country: " + country);

}

}

Hello World RESTful Web Service

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>hello-world-rest</artifactId>

<version>1.0.0</version>

<packaging>jar</packaging>

<properties>

<java.version>1.8</java.version>

<spring.boot.version>2.7.18</spring.boot.version>

</properties>

<dependencies>

<!-- Spring Boot Starter Web -->

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Optional: For testing -->

<dependency>

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

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Spring Boot Maven Plugin -->

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

package com.example.helloworld;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class HelloWorldApplication {

public static void main(String[] args) {

SpringApplication.run(HelloWorldApplication.class, args);

}

}

package com.example.helloworld;

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

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

@RestController

public class HelloWorldController {

@GetMapping("/hello")

public String helloWorld() {

return "Hello, World!";

}

}

REST - Country Web Service

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>country-rest-api</artifactId>

<version>1.0.0</version>

<dependencies>

<!-- Spring Web -->

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Optional: Test -->

<dependency>

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

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Spring Boot Maven Plugin -->

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

package com.example.country.model;

public class Country {

private String name;

private String code;

public Country() {

}

public Country(String name, String code) {

this.name = name;

this.code = code;

}

// Getters and setters

public String getName() {

return name;

}

public String getCode() {

return code;

}

public void setName(String name) {

this.name = name;

}

public void setCode(String code) {

this.code = code;

}

}

package com.example.country.controller;

import com.example.country.model.Country;

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

import java.util.Arrays;

import java.util.List;

@RestController

@RequestMapping("/api")

public class CountryController {

// GET /api/country

@GetMapping("/country")

public Country getDefaultCountry() {

return new Country("India", "IN");

}

// GET /api/countries

@GetMapping("/countries")

public List<Country> getAllCountries() {

return Arrays.asList(

new Country("India", "IN"),

new Country("United States", "US"),

new Country("Germany", "DE")

);

}

// GET /api/country/{code}

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

public Country getCountryByCode(@PathVariable String code) {

// Normally this would be a DB call, but using hardcoded values here

switch (code.toUpperCase()) {

case "IN": return new Country("India", "IN");

case "US": return new Country("United States", "US");

case "DE": return new Country("Germany", "DE");

default: return new Country("Unknown", code);

}

}

}

package com.example.country;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryRestApiApplication {

public static void main(String[] args) {

SpringApplication.run(CountryRestApiApplication.class, args);

}

}

REST - Get country based on country code

package com.example.country.model;

public class Country {

private String name;

private String code;

public Country() {}

public Country(String name, String code) {

this.name = name;

this.code = code;

}

// Getters and setters

public String getName() {

return name;

}

public String getCode() {

return code;

}

public void setName(String name) {

this.name = name;

}

public void setCode(String code) {

this.code = code;

}

}

package com.example.country.controller;

import com.example.country.model.Country;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

import java.util.Map;

@RestController

@RequestMapping("/api")

public class CountryController {

// Simulated "database"

private static final Map<String, Country> countries = Map.of(

"IN", new Country("India", "IN"),

"US", new Country("United States", "US"),

"UK", new Country("United Kingdom", "UK"),

"DE", new Country("Germany", "DE")

);

// GET /api/country/{code}

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

public ResponseEntity<?> getCountryByCode(@PathVariable String code) {

Country country = countries.get(code.toUpperCase());

if (country != null) {

return ResponseEntity.ok(country);

} else {

return ResponseEntity

.status(HttpStatus.NOT\_FOUND)

.body("Country with code '" + code + "' not found.");

}

}

}

package com.example.country;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryApiApplication {

public static void main(String[] args) {

SpringApplication.run(CountryApiApplication.class, args);

}

}

Create authentication service that returns JWT

<dependencies>

<!-- Spring Web -->

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Security -->

<dependency>

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

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<!-- JWT -->

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

</dependencies>

package com.example.jwt.model;

public class AuthRequest {

private String username;

private String password;

// Getters and 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; }

}

package com.example.jwt.service;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.stereotype.Service;

import java.util.Date;

@Service

public class JwtService {

private final String SECRET\_KEY = "my-secret-key"; // Use env var in real projects

public String generateToken(String username) {

long expirationTime = 1000 \* 60 \* 60; // 1 hour

return Jwts.builder()

.setSubject(username)

.setIssuer("MyAuthService")

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + expirationTime))

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

}

}

package com.example.jwt.controller;

import com.example.jwt.model.AuthRequest;

import com.example.jwt.service.JwtService;

import org.springframework.http.ResponseEntity;

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

@RestController

@RequestMapping("/api/auth")

public class AuthController {

private final JwtService jwtService;

public AuthController(JwtService jwtService) {

this.jwtService = jwtService;

}

@PostMapping("/login")

public ResponseEntity<?> login(@RequestBody AuthRequest authRequest) {

// Fake hardcoded validation (replace with DB/service later)

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

String token = jwtService.generateToken(authRequest.getUsername());

return ResponseEntity.ok().body("JWT Token: " + token);

} else {

return ResponseEntity.status(401).body("Invalid credentials");

}

}

}

package com.example.jwt;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class JwtAuthApplication {

public static void main(String[] args) {

SpringApplication.run(JwtAuthApplication.class, args);

}

}