**1.Create a Spring Web Project using Maven** 

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);

public static void main(String[] args) {

LOGGER.info("STARTING SPRING APPLICATION...");

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("SPRING APPLICATION STARTED");

}

}

**2. Hello World RESTful Web Service**

**src/main/java/com/cognizant/springlearn/controller/HelloController.java**

package com.cognizant.springlearn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class HelloController {

private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);

@GetMapping("/hello")

public String sayHello() {

LOGGER.info("START - sayHello()");

String message = "Hello World!!";

LOGGER.info("END - sayHello()");

return message;

}

}

**3.REST - Country Web Service**

1. **Country.java**

package com.cognizant.spring-learn.model;

public class Country {

private String code;

private String name;

// Getters and Setters

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

1. **beans.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="in" class="com.cognizant.spring-learn.model.Country">

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

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

</bean>

</beans>

1. **CountryController.java**

package com.cognizant.spring-learn.controller;

import com.cognizant.spring-learn.model.Country;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.core.io.ClassPathResource;

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

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

@SuppressWarnings("deprecation")

@RestController

public class CountryController {

@RequestMapping("/country")

public Country getCountryIndia() {

XmlBeanFactory factory = new XmlBeanFactory(new ClassPathResource("beans.xml"));

Country country = (Country) factory.getBean("in");

return country;

}

}

**4.REST - Get country based on country code**

package com.cognizant.spring-learn.controller;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ImportResource;

import org.springframework.stereotype.Service;

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

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

import java.util.List;

// ----- MODEL -----

class Country {

private String code;

private String name;

// Getters and Setters

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code.toUpperCase(); // Normalize

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

// ----- SERVICE -----

@Service

class CountryService {

@Autowired

private ApplicationContext context;

public Country getCountry(String code) {

List<Country> countryList = context.getBean("countryList", List.class);

return countryList.stream()

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

.findFirst()

.orElse(null);

}

}

// ----- CONTROLLER -----

@RestController

class CountryController {

@Autowired

private CountryService countryService;

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

public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

}

// ----- MAIN APP -----

@SpringBootApplication

@ImportResource("classpath:country.xml")

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

**country.xml**

<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="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.cognizant.spring-learn.controller.Country">

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

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

</bean>

<bean class="com.cognizant.spring-learn.controller.Country">

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

<property name="name" value="United States"/>

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**5.Create authentication service that returns JWT**

package com.cognizant.springsecurity;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.\*;

import org.springframework.http.ResponseEntity;

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

import org.springframework.security.config.Customizer;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.userdetails.\*;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

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

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import java.util.\*;

@SpringBootApplication

public class JwtAuthApplication {

public static void main(String[] args) {

SpringApplication.run(JwtAuthApplication.class, args);

}

// ===== Security Configuration =====

@Configuration

@EnableWebSecurity

public class SecurityConfig {

@Bean

public SecurityFilterChain securityFilterChain(org.springframework.security.config.annotation.web.builders.HttpSecurity http) throws Exception {

return http

.csrf(csrf -> csrf.disable())

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").authenticated()

.anyRequest().permitAll()

)

.httpBasic(Customizer.withDefaults())

.sessionManagement(sess -> sess.sessionCreationPolicy(org.springframework.security.config.http.SessionCreationPolicy.STATELESS))

.build();

}

@Bean

public UserDetailsService userDetailsService(PasswordEncoder passwordEncoder) {

return new InMemoryUserDetailsManager(

User.withUsername("user")

.password(passwordEncoder.encode("pwd"))

.roles("USER")

.build()

);

}

@Bean

public PasswordEncoder passwordEncoder() {

return new BCryptPasswordEncoder();

}

}

// ===== JWT Utility =====

@Component

public class JwtUtil {

private final String secretKey = "mySecretKey123456";

private final long expirationTime = 1000 \* 60 \* 10; // 10 minutes

public String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date(System.currentTimeMillis()))

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

.signWith(SignatureAlgorithm.HS256, secretKey)

.compact();

}

}

// ===== Authentication Controller =====

@RestController

public class AuthenticationController {

private final JwtUtil jwtUtil;

public AuthenticationController(JwtUtil jwtUtil) {

this.jwtUtil = jwtUtil;

}

@GetMapping("/authenticate")

public ResponseEntity<Map<String, String>> authenticate(Authentication authentication) {

String username = authentication.getName();

String token = jwtUtil.generateToken(username);

return ResponseEntity.ok(Map.of("token", token));

}

}

}

**pom.xml**

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

</dependencies>