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

**Country.java**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

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

private String code;

private String name;

public Country() {

LOGGER.debug("Inside Country Constructor.");

}

public String getCode() {

LOGGER.debug("Inside getCode()");

return code;

}

public void setCode(String code) {

LOGGER.debug("Inside setCode()");

this.code = code; }

public String getName() {

LOGGER.debug("Inside getName()");

return name;

}

public void setName(String name) {

LOGGER.debug("Inside setName()");

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}}

**SpringLearnApplication.java**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.SpringApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.debug("Inside main()");

displayCountry();

}

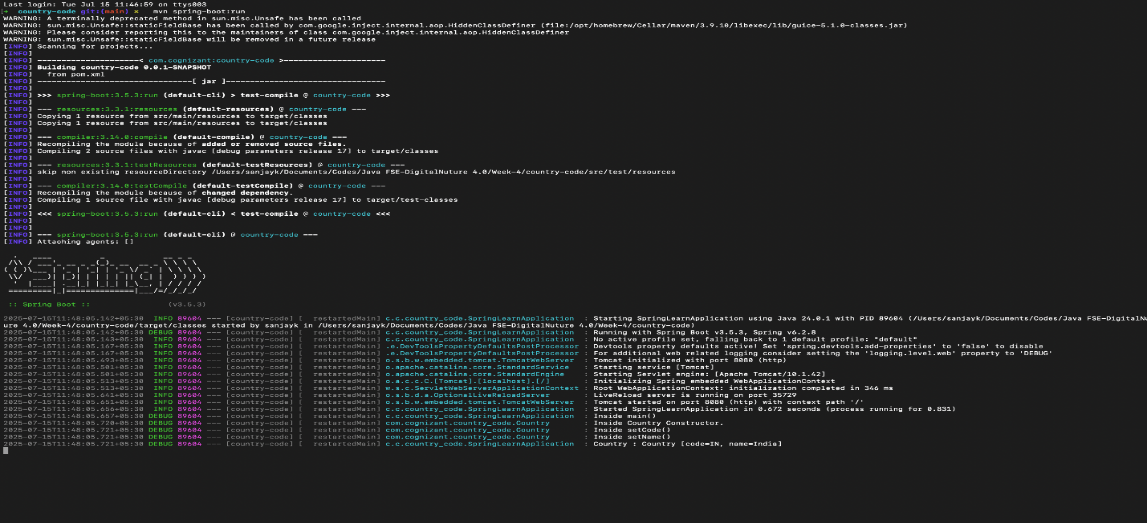
public static void displayCountry() {

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

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

LOGGER.debug("Country : {}", country.toString());

}}



**2. Hello World RESTful Web Service**

**HelloController.java**

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;

}}

**RestApplication.java**

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class RestApplication {

public static void main(String[] args) {

SpringApplication.run(RestApplication.class, args);

}}

**3. REST - Country Web Service**

**CountryController.java**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

import com.cognizant.countryweb.model.Country;

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("Start getCountryIndia()");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("in", Country.class);

LOGGER.info("End getCountryIndia()");

return country;

}}

**Country.java**

public class Country {

private String code;

private String name;

public Country() {

System.out.println("Inside Country Constructor");

}

public String getCode() {

System.out.println("Getting country code");

return code;

}

public void setCode(String code) {

System.out.println("Setting country code");

this.code = code;

}

public String getName() {

System.out.println("Getting country name");

return name;

}

public void setName(String name) {

System.out.println("Setting country name");

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}}

**CountrywebApplication.java**

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountrywebApplication {

public static void main(String[] args) {

SpringApplication.run(CountrywebApplication.class, args);

}}

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

**CountryController.java**

import com.cognizant.countrycode.model.Country;

import com.cognizant.countrycode.service.CountryService;

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

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

@RestController

public class CountryController {

@Autowired

private CountryService countryService;

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

public Country getCountry(@PathVariable String code) throws Exception {

return countryService.getCountry(code);

}}

**Country.java**

public class Country {

private String code;

private String name;

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = name;

}

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;

}}

**CountryService.java**

import com.cognizant.countrycode.model.Country;

import org.springframework.stereotype.Service;

import org.w3c.dom.\*;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import java.io.InputStream;

import java.util.ArrayList;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) throws Exception {

List<Country> countries = loadCountriesFromXML();

for (Country c : countries) {

if (c.getCode().equalsIgnoreCase(code)) {

return c;

} }

throw new Exception("Country not found: " + code);

}

private List<Country> loadCountriesFromXML() throws Exception {

List<Country> countryList = new ArrayList<>();

InputStream is = getClass().getClassLoader().getResourceAsStream("country.xml");

if (is == null) {

throw new Exception("country.xml not found in classpath");

}

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

DocumentBuilder db = dbf.newDocumentBuilder();

Document doc = db.parse(is);

doc.getDocumentElement().normalize();

NodeList nodeList = doc.getElementsByTagName("country");

for (int i = 0; i < nodeList.getLength(); i++) {

Node node = nodeList.item(i);

if (node.getNodeType() == Node.ELEMENT\_NODE) {

Element element = (Element) node;

String code = element.getElementsByTagName("code").item(0).getTextContent();

String name = element.getElementsByTagName("name").item(0).getTextContent();

countryList.add(new Country(code, name));

} }

return countryList;

}}

**CountrycodeApplication.java**

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountrycodeApplication {

public static void main(String[] args) {

SpringApplication.run(CountrycodeApplication.class, args);

}}

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

**SecurityConfig.java**

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

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

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

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

@Bean

public UserDetailsService users() {

UserDetails user = User.builder()

.username("user")

.password("{noop}pwd") // {noop} disables password encoding

.roles("USER")

.build();

return new InMemoryUserDetailsManager(user);

}

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

.csrf().disable()

.authorizeHttpRequests()

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

.anyRequest().permitAll()

.and()

.httpBasic();

return http.build();

}}

**AuthController.java**

import com.cognizant.jwt\_auth\_service.util.JwtUtil;

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

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

import java.security.Principal;

import java.util.Map;

@RestController

public class AuthController {

@Autowired

private JwtUtil jwtUtil;

@GetMapping("/authenticate")

public Map<String, String> authenticate(Principal principal) {

String token = jwtUtil.generateToken(principal.getName());

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

}}

**JwtUtil.java**

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.springframework.stereotype.Component;

import java.security.Key;

import java.util.Date;

@Component

public class JwtUtil {

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

public String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 3600000)) // 1 hour

.signWith(key)

.compact();

}}

**JwtAuthServiceApplication.java**

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class JwtAuthServiceApplication {

public static void main(String[] args) {

SpringApplication.run(JwtAuthServiceApplication.class, args);

}}