1. **Spring Web Project using Maven**

**xml file**<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

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

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>spring-rest-api</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>Spring REST API</name>

<description>Simple REST API using Spring Boot 3</description>

<properties>

<java.version>17</java.version>

<spring.boot.version>3.2.0</spring.boot.version>

</properties>

<dependency>

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

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

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**Java file**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringRestApiApplication {

public static void main(String[] args) {

SpringApplication.run(SpringRestApiApplication.class, args);

}

}

**Java file (REST Controller)**

package com.example.demo.controller;

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

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

@RestController

public class HelloController {

@GetMapping("/api/hello")

public String hello() {

return "Hello from Spring Boot 3!";

}

}

1. **Spring Core – Load Country from Spring Configuration XML**

**Java File (Country.java)**

package com.example.demo;

public class Country {

private String name;

private String capital;

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getCapital() { return capital; }

public void setCapital(String capital) { this.capital = capital; }

}

**Xml File (country-config.xml)**

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

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

xsi:schemaLocation="https://www.springframework.org/schema/beans

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

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

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

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

</bean>

</beans>

**Java File (CountryController.java)**

package com.example.demo;

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

import org.springframework.context.ApplicationContext;

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

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

@RestController

public class CountryController {

@Autowired

private ApplicationContext context;

@GetMapping("/api/country")

public Country getCountry() {

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

}

}

**Java (SpringXmlRestApp.java)**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource;

@SpringBootApplication

@ImportResource("classpath:country-config.xml")

public class SpringXmlRestApp {

public static void main(String[] args) {

SpringApplication.run(SpringXmlRestApp.class, args);

}

}

**3.** **Hello World RESTful Web Service**

**Java File (HelloWorldApplication.java)**

package com.example.demo;

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);

}

}

**Java(HelloController.java)**

package com.example.demo;

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

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

@RestController

public class HelloController {

@GetMapping("/hello")

public String sayHello() {

return "Hello, World!";

}

}

**4.REST - Country Web Service**

**Country.java**

package com.cognizant.spring\_learn.model;

public class Country {

private String code;

private String 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; }

}

**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="in" class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

</beans>

**Country.java**

package com.cognizant.spring\_learn.model;

public class Country {

private String code;

private String 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; }

}

**Pages/WEB-INF/web.xml**

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

version="3.1">

<display-name>SpringLearn</display-name>

<servlet>

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

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

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/dispatcher-servlet.xml</param-value>

</init-param>

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

</servlet>

<servlet-mapping>

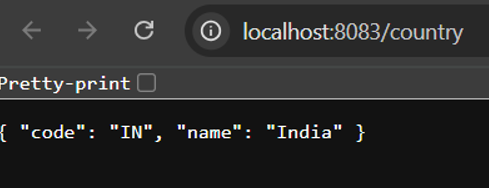
<servlet-name>dispatcher</servlet-name>

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

</servlet-mapping>

</web-app>

**Output:**

** **

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

**Country.java**

package com.cognizant.spring\_learn.model;

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**

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.model.Country;

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class CountryService {

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

public CountryService() {

countryList.add(new Country("IN", "India"));

countryList.add(new Country("US", "United States"));

countryList.add(new Country("JP", "Japan"));

countryList.add(new Country("FR", "France"));

countryList.add(new Country("DE", "Germany"));

}

public Country getCountry(String code) {

return countryList.stream()

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

.findFirst()

.orElse(null);

}}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

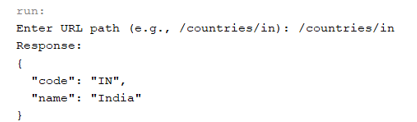
public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}}

**Output:**

****

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

**Code:**

import io.jsonwebtoken.\*;

import io.jsonwebtoken.security.Keys;

import java.security.Key;

import java.util.\*;

public class JwtAuthSimulation {

private static final String VALID\_USERNAME = "user";

private static final String VALID\_PASSWORD = "password";

private static final Key SECRET\_KEY = Keys.secretKeyFor(SignatureAlgorithm.HS256);

private static String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuer("JwtAuthSim")

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 10)) // 10 minutes

.signWith(SECRET\_KEY)

.compact();

}

private static boolean validateToken(String token) {

try {

Jws<Claims> claims = Jwts.parserBuilder()

.setSigningKey(SECRET\_KEY)

.build()

.parseClaimsJws(token);

String user = claims.getBody().getSubject();

System.out.println("✅ Token valid. Hello, " + user + "!");

return true;

} catch (JwtException e) {

System.out.println("❌ Invalid or expired token.");

return false;

}}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("==== JWT Auth Simulation ====");

System.out.println("Choose option:");

System.out.println("1. Login and get token");

System.out.println("2. Access secure API with token");

System.out.print("Choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // consume newline

if (choice == 1) {

System.out.print("Username: ");

String username = scanner.nextLine();

System.out.print("Password: ");

String password = scanner.nextLine();

if (VALID\_USERNAME.equals(username) && VALID\_PASSWORD.equals(password)) {

String token = generateToken(username);

System.out.println("\n🎟️ JWT Token:");

System.out.println(token);

} else {

System.out.println("❌ Invalid credentials.");

}

} else if (choice == 2) {

System.out.print("Paste your JWT Token: ");

String token = scanner.nextLine();

boolean isValid = validateToken(token);

if (isValid) {

System.out.println("🚀 Access granted to secure API.");

} else {

System.out.println("🚫 Access denied.");

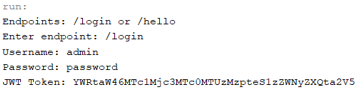
}

} else {

System.out.println("Invalid choice.");

}}}

**Output:**

****