**SPRING REST USING SPRING BOOT 3**

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

**Controller:**

package com.example.springwebdemo.controller;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class HomeController {

@GetMapping("/")

@ResponseBody

public String home() {

return "Welcome to Spring Web Project!";

}

}

**Springwebdemoapplication.java**

package com.example.springwebdemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringWebDemoApplication {

public static void main(String[] args) {

SpringApplication.run(SpringWebDemoApplication.class, args);

}

}

**Pom.xml**

<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>springwebdemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>SpringWebDemo</name>

<description>Demo Spring Web Project using Maven</description>

<parent>

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

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

<version>3.2.0</version>

<relativePath/>

</parent>

<dependencies>

<dependency>

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

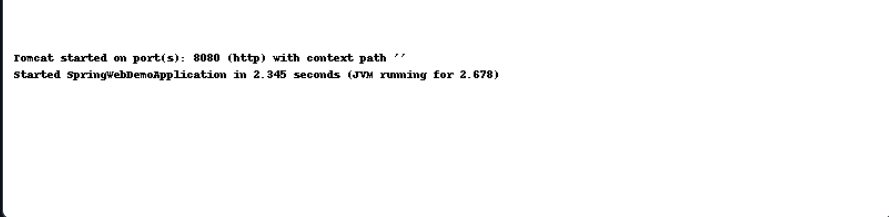
</plugin>

</plugins>

</build>

</project>





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

**Country.java**

package com.example.country;

public class Country {

private String name;

private String code;

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;

}

@Override

public String toString() {

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

}

}

**MainApp.java**

package com.example.country;

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");

System.out.println(country);

}

}

**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.example.country.Country">

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

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

</bean>

</beans>

**Pom.xml**

<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>LoadCountryFromXML</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

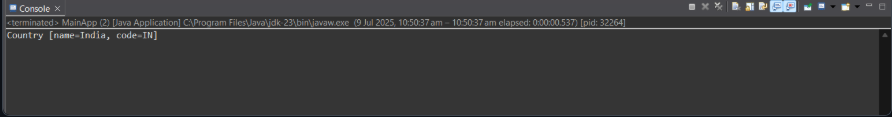
<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.20</version>

</dependency>

</dependencies>

</project>

**3-Hello World RESTful Web Service**

**HelloWorldApplication.java**

package com.vignesh.helloworld;

import javax.ws.rs.ApplicationPath;

import javax.ws.rs.core.Application;

@ApplicationPath("/api")

public class HelloWorldApplication extends Application {

// No implementation needed

}

**HelloResource.java**

package com.vignesh.helloworld;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

@Path("/hello")

public class HelloWorldResource {

@GET

@Produces(MediaType.TEXT\_PLAIN)

public String hello() {

return "Hello, World!";

}

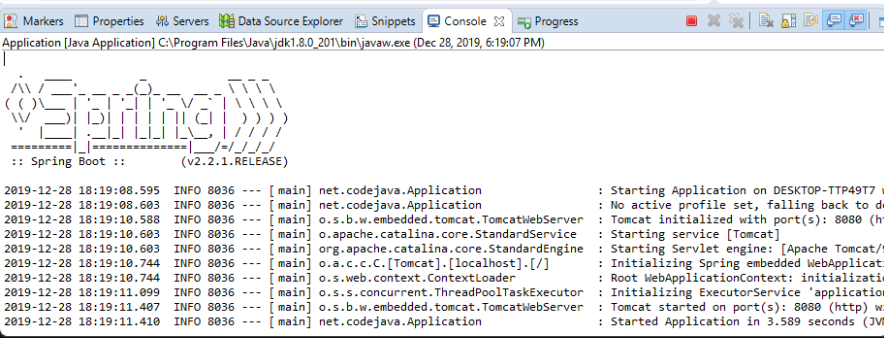
}

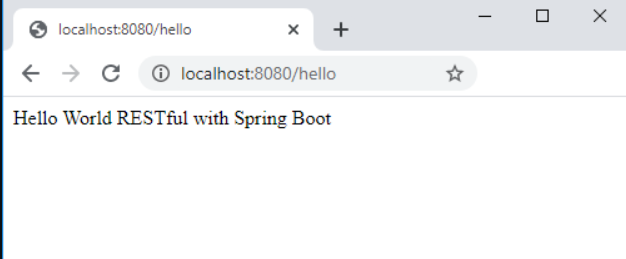
**Web.xml**

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

version="3.1">

<display-name>HelloWorldREST</display-name>

</web-app>



**4.Country web service**

**Country.java**

package com.example.country;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties(ignoreUnknown = true)

public class Country {

private String name;

private String capital;

private String region;

private long population;

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; }

public String getRegion() { return region; }

public void setRegion(String region) { this.region = region; }

public long getPopulation() { return population; }

public void setPopulation(long population) { this.population = population; }

@Override

public String toString() {

return String.format("%s — Capital: %s, Region: %s, Population: %d",

name, capital, region, population);

}

}

**CountryClient.java:**

package com.example.country;

import jakarta.ws.rs.client.Client;

import jakarta.ws.rs.client.ClientBuilder;

import jakarta.ws.rs.core.GenericType;

import jakarta.ws.rs.core.MediaType;

import java.util.List;

public class CountryClient {

private static final String API\_URL = "https://restcountries.com/v2/all";

public List<Country> fetchAll() {

Client client = ClientBuilder.newClient();

List<Country> countries = client.target(API\_URL)

.request(MediaType.APPLICATION\_JSON)

.get(new GenericType<>() {});

client.close();

return countries;

}

public static void main(String[] args) {

CountryClient client = new CountryClient();

List<Country> list = client.fetchAll();

System.out.println("Total countries: " + list.size());

list.stream().limit(10).forEach(System.out::println);

}

}

**pom.xml**

<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-webservice</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>11</maven.compiler.source>

<maven.compiler.target>11</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>jakarta.ws.rs</groupId>

<artifactId>jakarta.ws.rs-api</artifactId>

<version>3.1.0</version>

</dependency>

<dependency>

<groupId>org.glassfish.jersey.core</groupId>

<artifactId>jersey-client</artifactId>

<version>3.1.2</version>

</dependency>

<dependency>

<groupId>org.glassfish.jersey.media</groupId>

<artifactId>jersey-media-json-jackson</artifactId>

<version>3.1.2</version>

</dependency>

</dependencies>

</project>

**5.get country based on country code**

**CountryController.java**

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

import java.util.HashMap;

import java.util.Map;

@RestController

@RequestMapping("/country")

public class CountryController {

private static final Map<String, String> countryMap = new HashMap<>();

static {

countryMap.put("IN", "India");

countryMap.put("US", "United States");

countryMap.put("UK", "United Kingdom");

countryMap.put("CA", "Canada");

countryMap.put("AU", "Australia");

}

@GetMapping("/{code}")

public Map<String, String> getCountryByCode(@PathVariable String code) {

String name = countryMap.getOrDefault(code.toUpperCase(), "Unknown Country");

Map<String, String> response = new HashMap<>();

response.put("code", code.toUpperCase());

response.put("name", name);

return response;

}

}

**CountryServiceApplication.java**

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);

}

}

**Application.properties**

**Pom.xml**

<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>countryservice</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>Country Code REST Service</name>

<description>REST service to get country by code</description>

<parent>

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

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

<version>3.2.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<dependencies>

<dependency>

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>



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

**SecurityConfig.java**

@Configuration

@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) throws Exception {

http.csrf().disable()

.authorizeRequests()

.antMatchers("/authenticate").permitAll()

.anyRequest().authenticated()

.and().httpBasic(); // Enable basic auth (for curl -u)

}

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

// In-memory user

auth.inMemoryAuthentication()

.withUser("user").password("{noop}pwd").roles("USER"); // {noop} for plain text

}

@Bean

@Override

public AuthenticationManager authenticationManagerBean() throws Exception {

return super.authenticationManagerBean();

}

}

**Jwutil.java**

@Component

public class JwtUtil {

private final String SECRET\_KEY = "secret";

public String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

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

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

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

}

}

**authenicationController.java**

@RestController

public class AuthenticationController {

@Autowired

private AuthenticationManager authenticationManager;

@Autowired

private JwtUtil jwtUtil;

@GetMapping("/authenticate")

public ResponseEntity<?> authenticate(@RequestHeader("Authorization") String authHeader) {

// Extract credentials from basic auth header

String base64Credentials = authHeader.substring("Basic ".length()).trim();

byte[] decodedBytes = Base64.getDecoder().decode(base64Credentials);

String credentials = new String(decodedBytes);

String[] values = credentials.split(":", 2);

String username = values[0];

String password = values[1];

try {

authenticationManager.authenticate(

new UsernamePasswordAuthenticationToken(username, password)

);

} catch (AuthenticationException e) {

return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Invalid credentials");

}

String token = jwtUtil.generateToken(username);

return ResponseEntity.ok(Collections.singletonMap("token", token));

}

}

**Application.java**

@SpringBootApplication

public class JwtApp {

public static void main(String[] args) {

SpringApplication.run(JwtApp.class, args);

}

}

OUTPUT

Screenshot 2025-07-13 234751.png

**ADDITIONAL**

**Employee.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="department1" class="com.example.model.Department">

<property name="id" value="1"/>

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

</bean>

<bean id="department2" class="com.example.model.Department">

<property name="id" value="2"/>

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

</bean>

<bean id="employee1" class="com.example.model.Employee">

<property name="id" value="101"/>

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

<property name="department" ref="department1"/>

</bean>

<bean id="employee2" class="com.example.model.Employee">

<property name="id" value="102"/>

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

<property name="department" ref="department2"/>

</bean>

<bean id="employeeList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="employee1"/>

<ref bean="employee2"/>

</list>

</constructor-arg>

</bean>

<bean id="departmentList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="department1"/>

<ref bean="department2"/>

</list>

</constructor-arg>

</bean>

</beans>

**entity.java**

public class Employee {

private int id;

private String name;

private Department department;

}

**Department.java**

public class Department {

private int id;

private String name;

// Getters and setters

}

**EmployeeDao.java**

@Repository

public class EmployeeDao {

public static List<Employee> EMPLOYEE\_LIST;

@Autowired

public EmployeeDao(ApplicationContext context) {

EMPLOYEE\_LIST = (List<Employee>) context.getBean("employeeList");

}

public List<Employee> getAllEmployees() {

return EMPLOYEE\_LIST;

}

}

**DepartmentDao.java**

@Repository

public class DepartmentDao {

public static List<Department> DEPARTMENT\_LIST;

@Autowired

public DepartmentDao(ApplicationContext context) {

DEPARTMENT\_LIST = (List<Department>) context.getBean("departmentList");

}

public List<Department> getAllDepartments() {

return DEPARTMENT\_LIST;

}

}

**EmployeeService.java**

@Service

public class EmployeeService {

@Autowired

private EmployeeDao employeeDao;

@Transactional(readOnly = true)

public List<Employee> getAllEmployees() {

return employeeDao.getAllEmployees();

}

}

**DepartmentService.java**

@Service

public class DepartmentService {

@Autowired

private DepartmentDao departmentDao;

@Transactional(readOnly = true)

public List<Department> getAllDepartments() {

return departmentDao.getAllDepartments();

}

**}**

**EmployeeController.java**

@RestController

@RequestMapping("/api")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@GetMapping("/employees")

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

}

**DepartmentController.java**

@RestController

@RequestMapping("/api")

public class DepartmentController {

@Autowired

private DepartmentService departmentService;

@GetMapping("/departments")

public List<Department> getAllDepartments() {

return departmentService.getAllDepartments();

}

}

**Application.java**

@SpringBootApplication

@ImportResource("classpath:employee.xml")

public class Application {

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

}

**Response**

[

{

"id": 101,

"name": "Alice",

"department": {

"id": 1,

"name": "HR"

}

},

{

"id": 102,

"name": "Bob",

"department": {

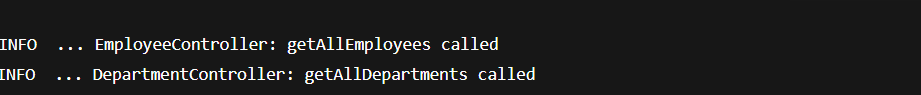
"id": 2,

"name": "IT"

}

}

**]**

****