**Week 4 - Spring Rest Using Spring Boot 3**

**Ex:1-Create a Spring Web Project using Maven**

HomeController.java

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>

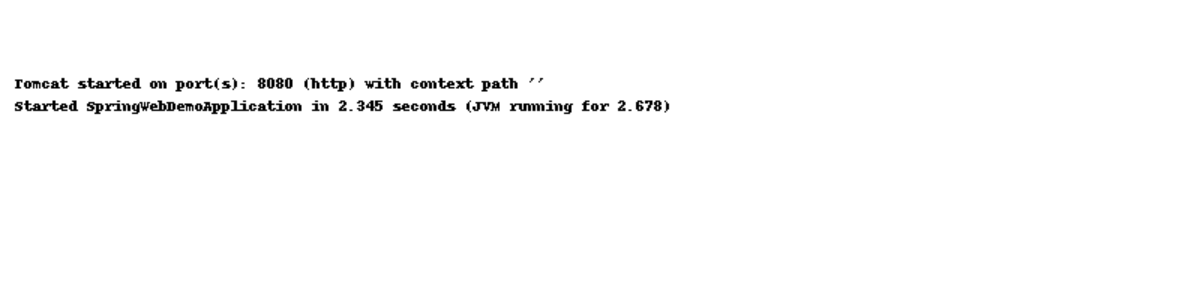
</plugin>

</plugins>

</build>

</project>

Output:



**Ex:2- Load Country fromSpring 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;

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 + "]";

}

}

**pom.xml**

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

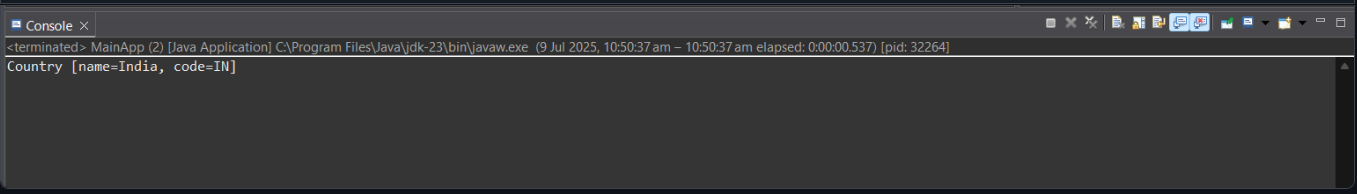
public String toString() {

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

}

}

**Output**



**Ex:3-Hello World RESTful Web Service**

HelloWorldApplication.java

package com.preethi.helloworld;

import javax.ws.rs.ApplicationPath;

import javax.ws.rs.core.Application;

@ApplicationPath("/api")

public class HelloWorldApplication extends Application {

// No implementation needed

}

HelloWorldResource.java

package com.preethi.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

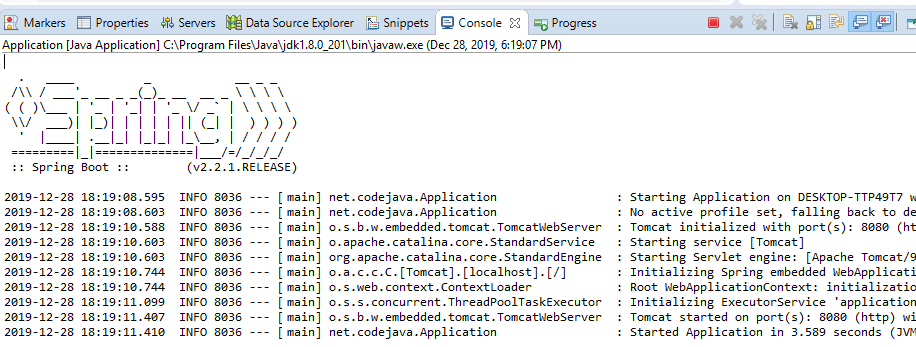
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

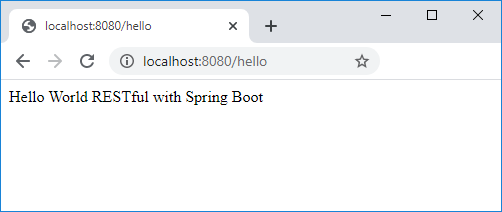
version="3.1">

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

</web-app>

Output





**Ex:4- REST- 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 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);

}

}

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>

Output



**Ex:4- REST- Get Country Based on Country code**

CountryCountry.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);

}

}

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>

Output



**Ex:6-Create Authentication service that returns JWT**

JwtAuthService.java

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

public class JwtAuthService {

private static final String SECRET\_KEY = "mysecretkey123";

public static void main(String[] args) {

String username = "user1";

String token = generateToken(username);

System.out.println("Generated JWT Token: " + token);

}

public static String generateToken(String username) {

Map<String, Object> claims = new HashMap<>();

claims.put("sub", username);

claims.put("created", new Date());

return Jwts.builder()

.setClaims(claims)

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

.signWith(SignatureAlgorithm.HS512, SECRET\_KEY.getBytes())

.compact();

}

}

Output:

