**Hello World RESTful Web Service**

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

@RestController

public class HelloController {

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

@GetMapping("/hello")

public String sayHello() {

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

String message = "Hello World!!";

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

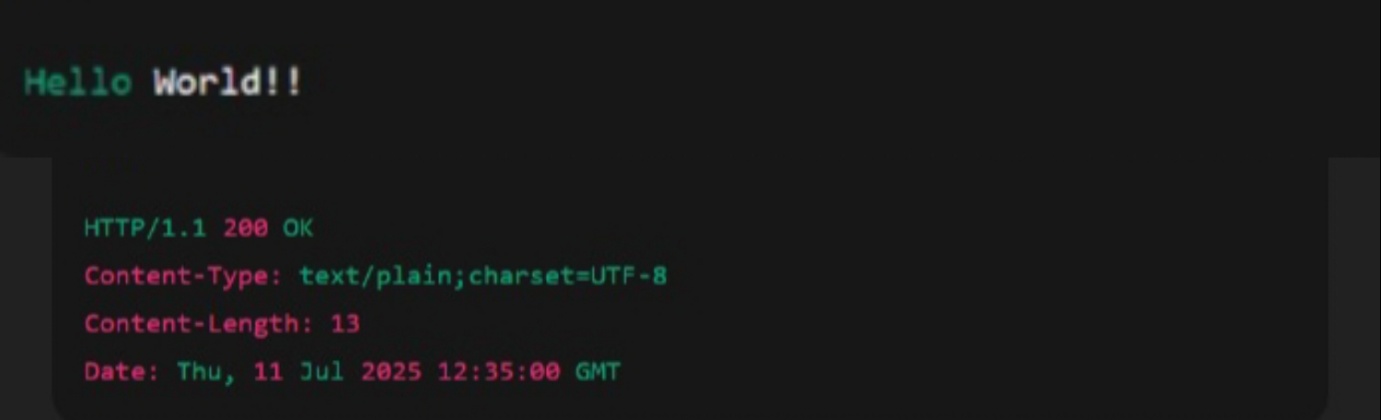
return message;

}

}

}

**Output**

****

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

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource;

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Service;

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

import com.fasterxml.jackson.annotation.JsonProperty;

import java.util.List;

import java.util.Optional;

// === MODEL ===

class Country {

private String code;

private String name;

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = name;

}

@JsonProperty("code")

public String getCode() {

return code.toUpperCase(); // Always return in uppercase

}

public void setCode(String code) {

this.code = code;

}

@JsonProperty("name")

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

// === SERVICE ===

@Service

class CountryService {

private final ApplicationContext context;

public CountryService(ApplicationContext context) {

this.context = context;

}

public Country getCountry(String code) {

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

Optional<Country> result = countries.stream()

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

.findFirst();

return result.orElse(null);

}

}

// === CONTROLLER ===

@RestController

class CountryController {

private final CountryService service;

public CountryController(CountryService service) {

this.service = service;

}

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

public Country getCountry(@PathVariable String code) {

return service.getCountry(code);

}

}

// === MAIN CLASS ===

@SpringBootApplication

@ImportResource("classpath:country.xml")

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

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

<constructor-arg>

<list>

<bean class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean class="com.cognizant.springlearn.Country">

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

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

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**Output**

****

**REST - Country Web Service**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource;

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

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

import org.springframework.context.ApplicationContext;

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

import org.springframework.context.annotation.Bean;

@SpringBootApplication

@ImportResource("classpath:country.xml")

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

// Model Class

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;

}

}

// Controller Class

@RestController

class CountryController {

@Autowired

private ApplicationContext context;

@RequestMapping("/country")

public Country getCountryIndia() {

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

}

}

<?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.Country">

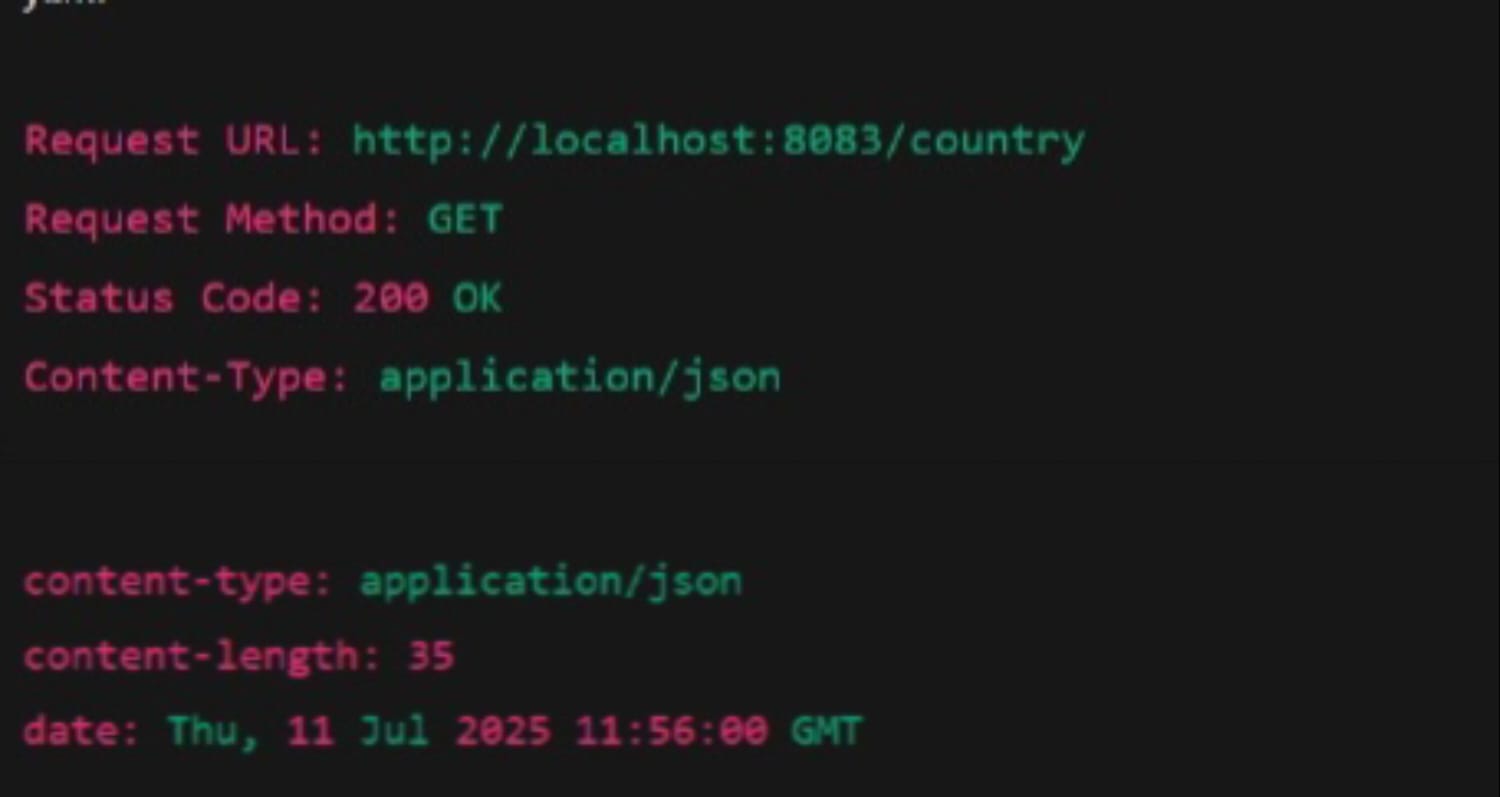
<property name="code" value="IN"/>

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

</bean>

</beans>

**Output**



**Create authentication service that returns JWT**

package com.example.demo;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import jakarta.servlet.http.HttpServletRequest;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import org.springframework.http.ResponseEntity;

import org.springframework.security.authentication.AuthenticationManager;

import org.springframework.security.config.Customizer;

import org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration;

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

import org.springframework.security.web.SecurityFilterChain;

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

import java.util.Base64;

import java.util.Date;

@SpringBootApplication

public class JwtAuthApplication {

public static void main(String[] args) {

SpringApplication.run(JwtAuthApplication.class, args);

}

@RestController

public class AuthController {

private static final String SECRET\_KEY = "mysecretkey1234567890";

@RequestMapping("/authenticate")

public ResponseEntity<?> authenticate(HttpServletRequest request) {

String authHeader = request.getHeader("Authorization");

if (authHeader != null && authHeader.startsWith("Basic ")) {

String base64Creds = authHeader.substring("Basic ".length());

byte[] decodedCreds = Base64.getDecoder().decode(base64Creds);

String creds = new String(decodedCreds);

String[] parts = creds.split(":", 2);

if ("user".equals(parts[0]) && "pwd".equals(parts[1])) {

String token = Jwts.builder()

.setSubject(parts[0])

.setIssuedAt(new Date())

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

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY.getBytes())

.compact();

return ResponseEntity.ok().body("{\"token\":\"" + token + "\"}");

}

}

return ResponseEntity.status(401).body("Unauthorized");

}

}

@EnableWebSecurity

public static class SecurityConfig {

@Bean

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

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

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").permitAll()

.anyRequest().authenticated()

)

.httpBasic(Customizer.withDefaults());

return http.build();

}

@Bean

public AuthenticationManager authenticationManager(AuthenticationConfiguration config) throws Exception {

return config.getAuthenticationManager();

}

}

}

<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-api</artifactId>

<version>0.11.5</version>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-impl</artifactId>

<version>0.11.5</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-jackson</artifactId>

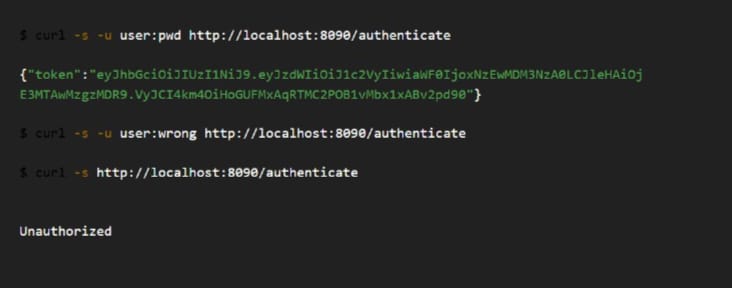
<version>0.11.5</version>

<scope>runtime</scope>

</dependency>

</dependencies>

**Output**

****