**Spring REST using Spring Boot 3**

**Create a Spring Web Project using Maven**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

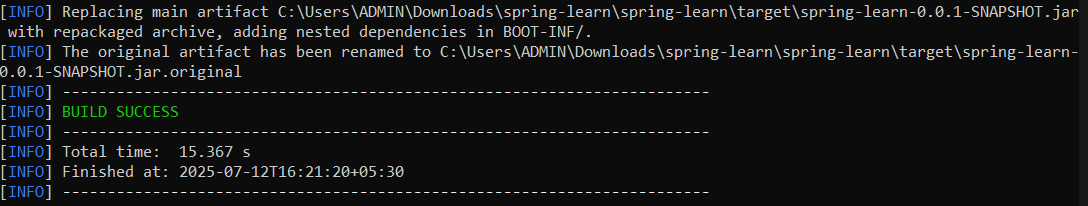
public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.class, args);

}

}

**Output**

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

Country.java

package com.cognizant.spring\_learn;

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

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.cognizant.spring\_learn.Country;

*@SpringBootApplication*(exclude = {

org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration.class

})

public class SpringLearnApplication {

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

public static void main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.class, args);

*displayCountry*();

}

public static void displayCountry() {

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

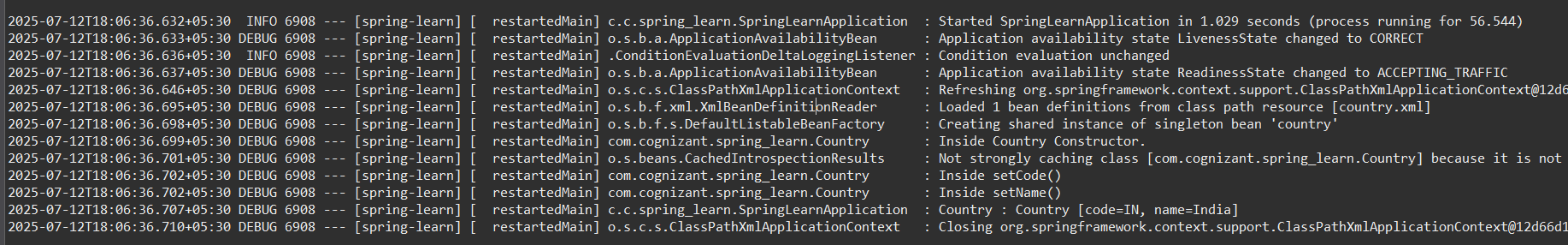
Country country = context.getBean("country", Country.class);

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

context.close();

}

}

Output  


**Hello World RESTful Web Service**

HelloController.java

package com.cognizant.spring\_learn.controller;

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***.debug("Start: sayHello()");

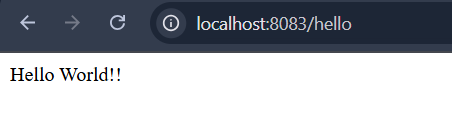
String response = "Hello World!!";

***LOGGER***.debug("End: sayHello()");

return response;

}

}



**REST - Country Web Service**

Country.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

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

<bean id="country" class="com.cognizant.spring\_learn.Country">

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

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

</bean>

</beans>

CountryController.java

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

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;

*@RestController*

public class CountryController {

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

*@RequestMapping*("/country")

public Country getCountryIndia() {

***LOGGER***.debug("START: getCountryIndia()");

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

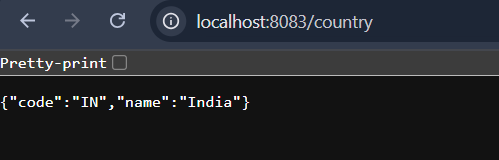
Country country = context.getBean("country", Country.class);

***LOGGER***.debug("END: getCountryIndia()");

return country;

}

}



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

Country.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

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

<bean id="country1" class="com.cognizant.spring\_learn.Country">

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

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

</bean>

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

<constructor-arg>

<list>

<ref bean="country1"/>

</list>

</constructor-arg>

</bean>

</beans>

Country.java

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {

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

}

public String getCode() {

System.***out***.println("Inside getCode()");

return code;

}

public void setCode(String code) {

System.***out***.println("Inside setCode()");

this.code = code;

}

public String getName() {

System.***out***.println("Inside getName()");

return name;

}

public void setName(String name) {

System.***out***.println("Inside setName()");

this.name = name;

}

*@Override*

public String toString() {

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

}

}

CountryController.java

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import com.cognizant.spring\_learn.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

*@RestController*

public class CountryController {

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

*@Autowired*

private CountryService countryService;

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

public Country getCountry(*@PathVariable* String code) {

***LOGGER***.debug("START: getCountry() with code = {}", code);

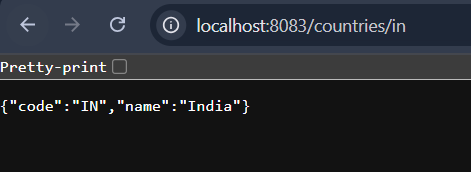
Country country = countryService.getCountry(code);

***LOGGER***.debug("END: getCountry() -> {}", country);

return country;

}

}



**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.web.SecurityFilterChain;

import org.springframework.security.core.userdetails.User;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.core.userdetails.UserDetailsService;

*@*Configuration

public class SecurityConfig {

*@*SuppressWarnings("removal")

*@*Bean

public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

http.csrf().disable()

.authorizeHttpRequests()

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

.anyRequest().authenticated()

.and()

.httpBasic(); // Enable Basic Auth

return http.build();

}

*@*Bean

public UserDetailsService userDetailsService() {

UserDetails user = User.withUsername("user")

.password("{noop}pwd") // No encoding

.roles("USER")

.build();

return new InMemoryUserDetailsManager(user);

}

}

AuthenticationController.java

package com.cognizant.spring\_learn.controller;

import java.nio.charset.StandardCharsets;

import java.util.Base64;

import java.util.Map;

import org.springframework.http.HttpStatus;

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

import org.springframework.web.server.ResponseStatusException;

*@RestController*

public class AuthenticationController {

*@RequestMapping*("/authenticate")

public Map<String, String> authenticate(*@RequestHeader*("Authorization") String authHeader) {

if (!authHeader.startsWith("Basic ")) {

throw new ResponseStatusException(*HttpStatus*.***UNAUTHORIZED***, "Invalid Authorization header");

}

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

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

String credentials = new String(credDecoded, StandardCharsets.***UTF\_8***);

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

String username = values[0];

String password = values[1];

if ("user".equals(username) && "pwd".equals(password)) {

return Map.*of*("token", "mock-jwt-token-for-" + username);

} else {

throw new ResponseStatusException(*HttpStatus*.***UNAUTHORIZED***, "Invalid credentials");

}

}

}

JwtUtils.java

package com.cognizant.spring\_learn.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.stereotype.Component;

import java.util.Date;

*@Component*

public class JwtUtil {

private String secret = "secret-key"; // Use a secure key in production

public String generateToken(String username) {

return Jwts.*builder*()

.setSubject(username)

.setIssuedAt(new Date())

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

.signWith(*SignatureAlgorithm*.***HS256***, secret)

.compact();

}

}

**Output**

