**WEEK 4**

**HANDS ON 1 : CREATE A SPRING WEB PROJECT USING MAVEN**

**PROGRAM**

**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) {

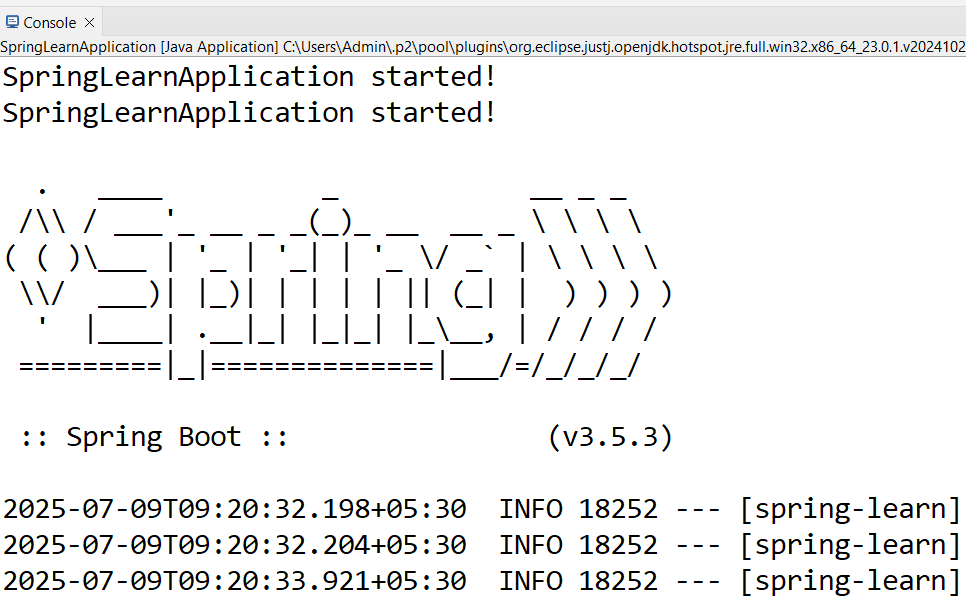
System.out.println("SpringLearnApplication started!");

SpringApplication.run(SpringLearnApplication.class, args);

}

}

**OUTPUT**

****

**HANDS ON 4 : LOAD COUNTRY FROM SPRING CONFIGURATION XML**

**PROGRAM**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

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

public static void main(String[] args) {

*LOGGER*.debug("Before calling displayCountry()");

*displayCountry*();

*LOGGER*.debug("After calling displayCountry()");

}

public static void displayCountry() {

try {

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

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

System.*out*.println("Country: " + country.toString());

} catch (Exception e) {

System.***out***.println("Exception caught:");

e.printStackTrace();

}

}

}

**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 void setCode(String code) {

*LOGGER*.debug("Inside setCode()");

this.code = code;

}

public void setName(String name) {

*LOGGER*.debug("Inside setName()");

this.name = name;

}

public String getCode() {

*LOGGER*.debug("Inside getCode()");

return code;

}

public String getName() {

*LOGGER*.debug("Inside getName()");

return name;

}

@Override

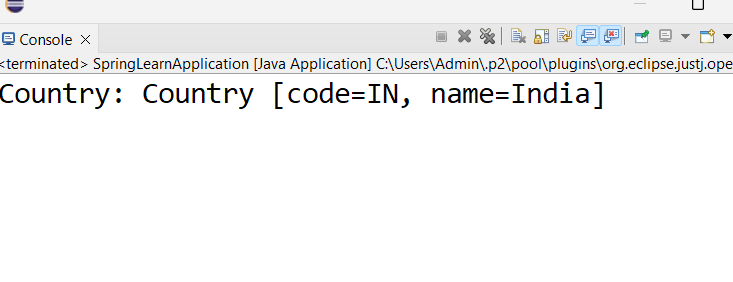
public String toString() {

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

}

}

**OUTPUT**

****

**EXERCISE 3**

**HELLO WORLD RESTFUL WEB SERVICE**

**PROGRAM**

**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.info("START sayHello()");

String response = "Hello World!!";

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

return response;

}

}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication {

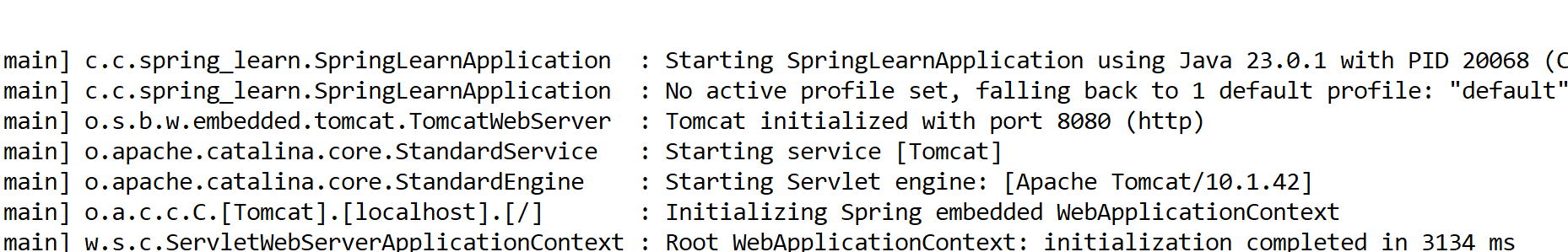
public static void main(String[] args) {

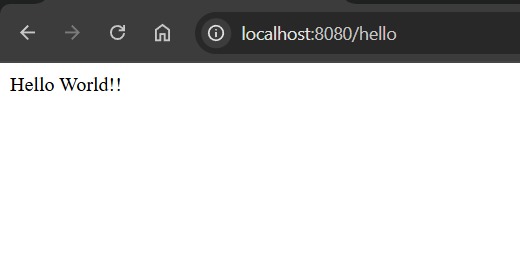
SpringApplication.run(SpringLearnApplication.class, args);

}

}

**OUTPUT**

****

****

**EXERCISE 4**

**REST - COUNTRY WEB SERVICE**

**PROGRAM**

**Country.java**

package com.cognizant.spring\_learn.model;

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.model.Country;

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 {

@RequestMapping("/country")

public Country getCountryIndia() {

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

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

return country; // Automatically converted to JSON

}

}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

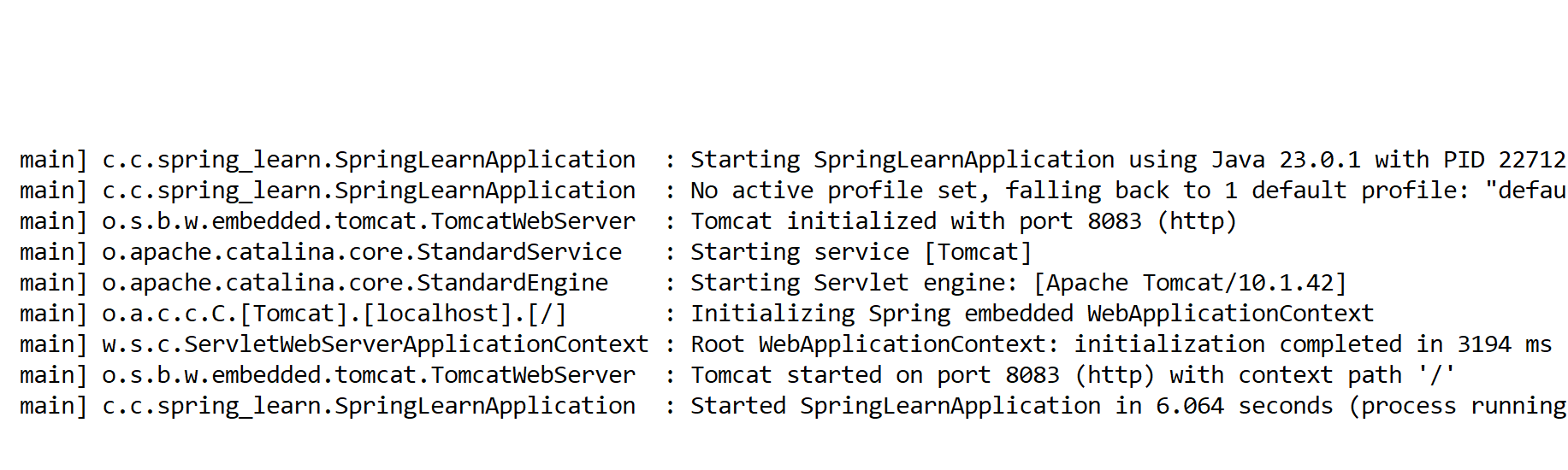
public static void main(String[] args) {

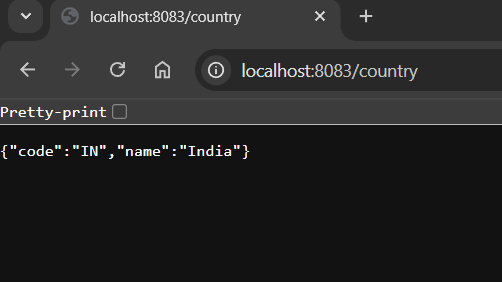
SpringApplication.run(SpringLearnApplication.class, args);

}

}

**OUTPUT**

****



**EXERCISE 5**

**REST - GET COUNTRY BASED ON COUNTRY CODE**

**PROGRAM**

**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

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

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

<constructor-arg>

<list>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**CountryService.java**

package com.cognizant.spring\_learn.service;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) {

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

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

return countryList.stream()

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

.findFirst()

.orElse(null);

}

}

**CountryController.java**

package com.cognizant.spring\_learn.controller;

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

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

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

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

@RestController

public class CountryController {

@Autowired

private CountryService countryService;

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

public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

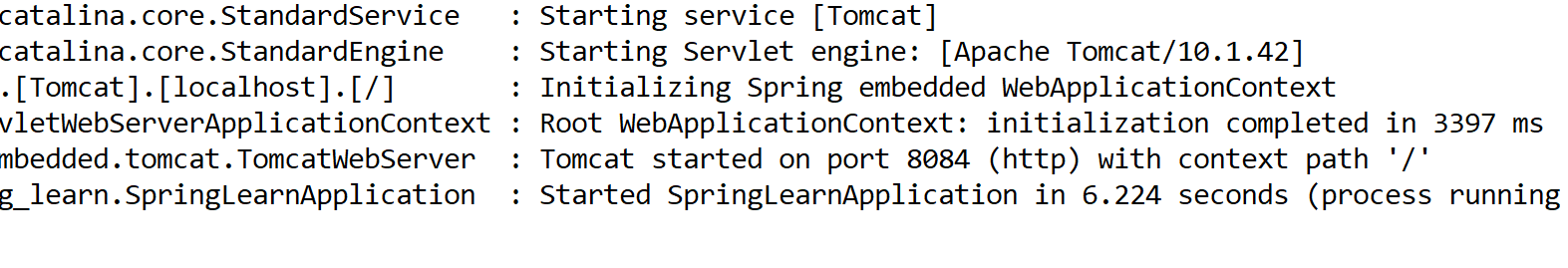
public static void main(String[] args) {

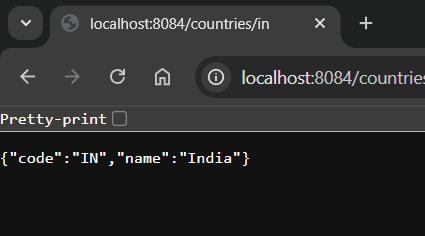
SpringApplication.run(SpringLearnApplication.class, args);

}

}

**OUTPUT**

****

****

**EXERCISE 6**

**CREATE AUTHENTICATION SERVICE THAT RETURNS JWT**

**PROGRAM**

**AuthenticationController.java**

package com.cognizant.spring\_learn.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import jakarta.servlet.http.HttpServletRequest; // ✅ Jakarta import for Spring Boot 3.x

import org.springframework.http.ResponseEntity;

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

import java.util.Base64;

import java.util.Date;

@RestController

public class AuthenticationController {

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

@RequestMapping(value = "/authenticate", method = RequestMethod.*GET*)

public ResponseEntity<?> authenticate(HttpServletRequest request) {

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

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

// Decode base64 encoded credentials

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

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

String credentials = new String(credDecoded);

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

String username = values[0];

String password = values[1];

// Validate credentials

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

String token = Jwts.*builder*()

.setSubject(username)

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

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

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

.compact();

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

}

}

return ResponseEntity.*status*(401).body("{\"error\": \"Invalid Credentials\"}");

}

}

**SecurityConfig.java**

package com.cognizant.spring\_learn.security;

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;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

.csrf(csrf -> csrf.disable()) // Disable CSRF for testing

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").permitAll() // Allow access to /authenticate

.anyRequest().authenticated()

)

.httpBasic(); // Enable Basic Auth (so curl -u works)

return http.build();

}

}

**Application.properties**

server.port=8098

spring.security.user.name=user

spring.security.user.password=pwd

**SpringLearnApplication.java**

**package** com.cognizant.spring\_learn;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

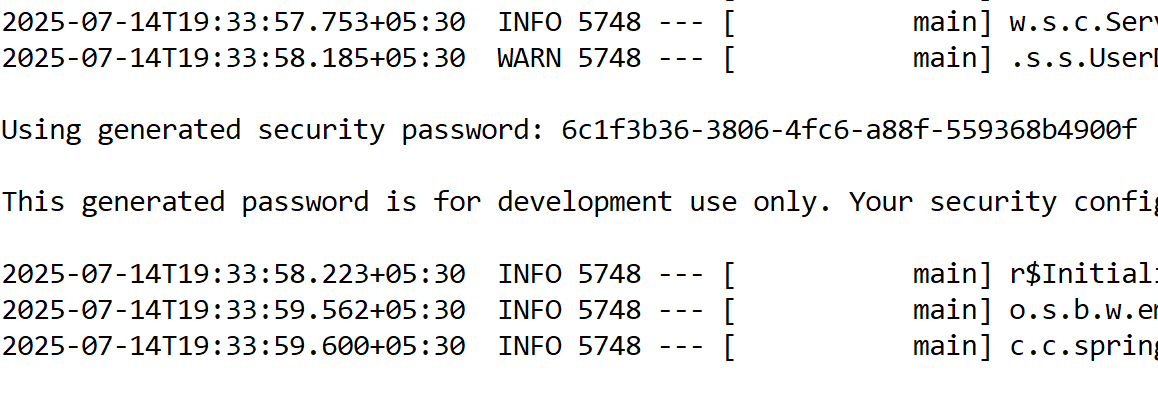
**public** **static** **void** main(String[] args) {

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

}

}

**OUTPUT**

****