**WEEK 4**

**spring-rest-handson, JWT-handson**

**Spring-rest-handson:**

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

Create a SpringLearn Application and build it:

Follow steps below to create a project: 

1. Go to <https://start.spring.io/>
2. Change Group as “com.cognizant”
3. Change Artifact Id as “spring-learn”
4. Select Spring Boot DevTools and Spring Web
5. Create and download the project as zip
6. Extract the zip in root folder to Eclipse Workspace
7. Build the project using ‘mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456’ command in command line
8. Import the project in Eclipse "File > Import > Maven > Existing Maven Projects > Click Browse and select extracted folder > Finish"
9. Include logs to verify if main() method of SpringLearnApplication.
10. Run the SpringLearnApplication class.

A screenshot of a computer

AI-generated content may be incorrect.

Add Logging to the Application:

A screenshot of a computer

AI-generated content may be incorrect.

Dependency Hierarchy: A screenshot of a computer

AI-generated content may be incorrect.

Output: A screenshot of a computer

AI-generated content may be incorrect.

1. **Spring Core – Load CountryModel from Spring Configuration XML:**

An airlines website is going to support booking on four countries. There will be a drop down on the home page of this website to select the respective country. It is also important to store the two-character ISO code of each country. 

|  |  |
| --- | --- |
| **Code** | **Name** |
| US | United States |
| DE | Germany |
| IN | India |
| JP | Japan |

Above data has to be stored in spring configuration file. Write program to read this configuration file and display the details.

**Answer:-**

**Create CountryModel.java in src/main/java :**

package com.charan.spring\_app;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class CountryModel {

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

private String code;

private String name;

public CountryModel() {

LOGGER.debug("Inside CountryModel 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 "CountryModel{" +

"code='" + code + '\'' +

", name='" + name + '\'' +

'}';

}

}

**Create a CountryModel.xml in src/main/resources:**

<?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="country" class="com.cognizant.springlearn.CountryModel">

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

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

</bean>

</beans>

**Change SpringRestJwtApp code:**

package com.charan.spring\_app;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ImportResource;

@SpringBootApplication

@ImportResource("classpath:countries-config.xml") // load the XML bean into Spring Boot context

public class SpringRestJwtApp {

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

public static void main(String[] args) {

LOGGER.info("START");

ApplicationContext context = SpringApplication.run(SpringRestJwtApp.class, args);

displayCountry(context);

LOGGER.info("END");

}

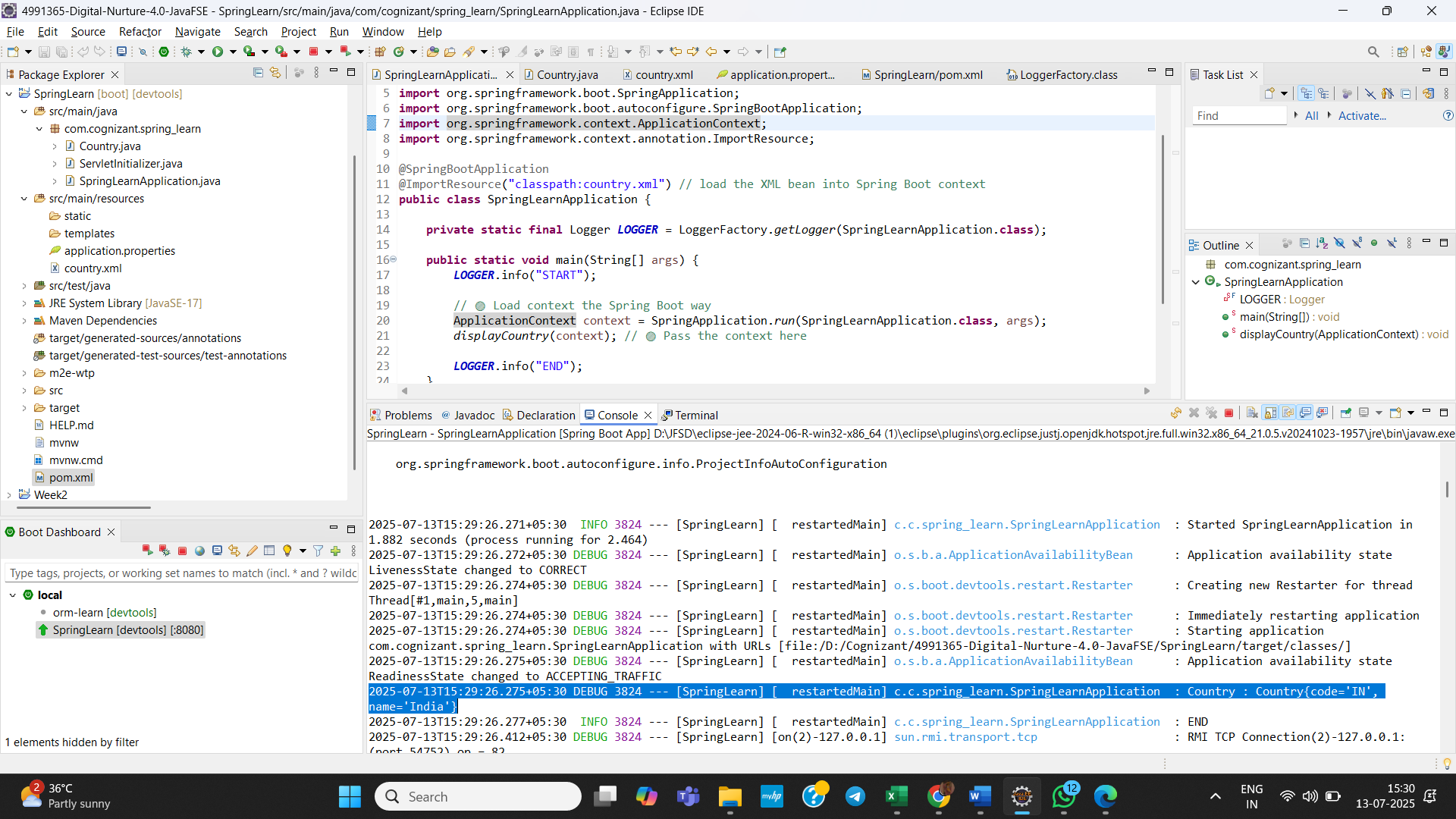
public static void displayCountry(ApplicationContext context) {

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

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

}

}

**Output**:

1. **Hello World RESTful Web Service:**

Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework. Refer details below:  
  
Method: GET  
URL: /hello  
Controller: com.cognizant.spring-learn.controller.HelloController  
Method Signature: public String sayHello()  
Method Implementation: return hard coded string "Hello World!!"  
Sample Request: http://localhost:8083/hello  
Sample Response: Hello World!! 

**Answer:-**

**Create a com.charan.spring\_app.controller package**

**Create a GreetingController.java :**  
package com.charan.spring\_app.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

@RestController

public class GreetingController {

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

@GetMapping("/hello")

public String sayHello() {

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

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

return "Hello World!!";

}

}

Change port to 8083 in application.properties using :

server.port=8083

Output:

A screenshot of a computer

AI-generated content may be incorrect.

1. **REST - CountryModel Web Service:**

Write a REST service that returns India country details in the earlier created spring learn application.  
  
**URL**: /country  
**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation**: @RequestMapping  
**Method Name**: getCountryIndia()  
**Method Implementation**: Load India bean from spring xml configuration and return  
**Sample Request**: http://localhost:8083/country  
**Sample Response**:

1. {
2. "code": "IN",
3. "name": "India"
4. }

**Answer:-**

**Create CountryApiController.java:**

package com.charan.spring\_app.controller;

import com.charan.spring\_app.CountryModel;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.\*;

import org.springframework.context.ApplicationContext;

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

@RestController

public class CountryApiController {

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

@Autowired

private ApplicationContext context;

@RequestMapping("/country")

public CountryModel getCountryIndia() {

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

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

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

return country;

}

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

1. **REST - Get country based on country code:**

Write a REST service that returns a specific country based on country code. The country code should be case insensitive.  
  
Controller: com.cognizant.spring-learn.controller.CountryController  
Method Annotation: @GetMapping("/countries/{code}")  
Method Name: getCountry(String code)  
Method Implemetation: Invoke countryService.getCountry(code)   
Service Method: com.cognizant.spring-learn.service.CountryService.getCountry(String code)

**Answer:-**

**Update countries-config.xml with a list of countries:**

<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="country" class="com.charan.spring\_app.CountryModel">

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

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

</bean>

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

<constructor-arg>

<list>

<bean class="com.charan.spring\_app.CountryModel">

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

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

</bean>

<bean class="com.charan.spring\_app.CountryModel">

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

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

</bean>

<bean class="com.charan.spring\_app.CountryModel">

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

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

</bean>

<bean class="com.charan.spring\_app.CountryModel">

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

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

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**Create a CountryServiceImpl in a service package:**

package com.charan.spring\_app.service;

import com.charan.spring\_app.CountryModel;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryServiceImpl {

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

@Autowired

private ApplicationContext context;

public CountryModel getCountry(String code) {

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

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

// Lambda-based search (case-insensitive)

CountryModel match = countryList.stream()

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

.findFirst()

.orElse(null);

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

return match;

}

}

**Update CountryApiController to get CountryModel by CountryModel Code:**

package com.charan.spring\_app.controller;

import com.charan.spring\_app.CountryModel;

import com.charan.spring\_app.service.CountryServiceImpl;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class CountryApiController {

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

@Autowired

private CountryServiceImpl countryService;

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

public CountryModel getCountry(@PathVariable String code) {

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

CountryModel country = countryService.getCountry(code);

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

return country;

}

}

**Output:** A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**JWT-handson:**

1. **Create authentication service that returns JWT:**

**Add Spring Security dependency:**

<dependency>

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

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

</dependency>

**Create JwtAuthController.java:**

package com.charan.spring\_app.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import java.util.HashMap;

import java.util.Map;

@RestController

public class JwtAuthController {

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

@GetMapping("/authenticate")

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

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

LOGGER.debug("Authorization Header: {}", authHeader);

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

map.put("token", ""); // JWT will be filled in later

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

return map;

}

}

**Create a com.charan.spring\_app.config and in that WebSecurityConfig.java:**

package com.charan.spring\_app.config;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.context.annotation.Bean;

@Configuration

public class WebSecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

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

.authorizeHttpRequests(authz -> authz

.requestMatchers("/countries").hasRole("USER")

.requestMatchers("/authenticate").hasAnyRole("USER", "ADMIN")

.anyRequest().authenticated()

)

.httpBasic();

return http.build();

}

}

**Create a UserDetailsConfig:**

package com.charan.spring\_app.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.core.userdetails.\*;

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration

public class UserDetailsConfig {

@Bean

public InMemoryUserDetailsManager userDetailsService(PasswordEncoder passwordEncoder) {

UserDetails user = User.builder()

.username("user")

.password(passwordEncoder.encode("pwd"))

.roles("USER")

.build();

UserDetails admin = User.builder()

.username("admin")

.password(passwordEncoder.encode("admin"))

.roles("ADMIN")

.build();

return new InMemoryUserDetailsManager(user, admin);

}

@Bean

public PasswordEncoder passwordEncoder() {

return new BCryptPasswordEncoder();

}

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.