**WEEK-04 HANDS ON**

**SPRING REST USING SPRING BOOT**

**Hands on 1**

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

**Source code with output:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Hands on 2**

**Spring Core – Load SimpleDateFormat from Spring Configuration XML**   
  
SimpleDateFormat with the pattern ‘dd/MM/yyyy’ is created in multiple places of an application. To avoid creation of SimpleDateFormat in multiple places, define a bean in Spring XML Configuration file and retrieve the date.  
  
Follow steps below to implement:

* Create spring configuration file date-format.xml in src/main/resources folder of 'spring-learn' project
* Open https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html#beans-factory-metadata
* Copy the XML defined in the section of previous step URL and paste it into date-format.xml
* Define bean tag in the XML with for date format. Refer code below.

**Source code with Output:**

package com.cognizant.springlearn;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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);

displayDate();

}

public static void displayDate() {

ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

try {

Date date = format.parse("31/12/2018");

System.out.println("Parsed Date: " + date);

} catch (ParseException e) {

System.out.println("Date parsing failed: " + e.getMessage());

}

}

}

**Output:**

**A close-up of a date

AI-generated content may be incorrect.**

**Hands on 3**

**Spring Core - Incorporate Logging   
  
Incorporate logging in the Spring Boot project created in previous hands on. Refer steps below:**

* **Create application.properties if not yet created in src/main/resources folder**
* **Add below lines in application.properties**

**Source code :**

package com.cognizant.springlearn;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

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.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate();

}

public static void displayDate() {

LOGGER.info("START");

ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

try {

Date date = format.parse("31/12/2018");

LOGGER.debug("Parsed Date: {}", date);

} catch (ParseException e) {

LOGGER.error("Date parsing failed: {}", e.getMessage());

}

LOGGER.info("END");

}

}

**Output**:

A close-up of a computer screen

AI-generated content may be incorrect.

**Hands on 4**

**Spring Core – Load Country 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.

**Source code:**

package com.cognizant.springlearn;

import java.text.SimpleDateFormat;

import java.util.Date;

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.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate();

displayCountry();

}

public static void displayDate() {

LOGGER.info("START");

ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

try {

Date date = format.parse("31/12/2018");

LOGGER.debug("Parsed Date: {}", date);

} catch (Exception e) {

LOGGER.error("Error parsing date", e);

}

LOGGER.info("END");

}

public static void displayCountry() {

LOGGER.info("START");

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

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

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

LOGGER.info("END");

}

}

**Output**:

A screenshot of a computer code

AI-generated content may be incorrect.

**Hands on 5**

**Spring Core – Demonstration of Singleton Scope and Prototype Scope**   
  
The Country bean done in the previous hands on will be used to demonstrate the scopes in Spring. Implement the steps below.  
**Source code:**

package com.cognizant.springlearn;

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.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayCountry(); // Singleton / Prototype demonstration

}

public static void displayCountry() {

LOGGER.info("START");

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

Country country1 = context.getBean("country", Country.class);

LOGGER.debug("Country 1: {}", country1.toString());

Country country2 = context.getBean("country", Country.class);

LOGGER.debug("Country 2: {}", country2.toString());

LOGGER.info("Are both countries the same instance? {}", (country1 == country2));

LOGGER.info("END");

}

}

**Output**:

**For Singleton Scope:**

A text on a white background

AI-generated content may be incorrect.

**For Prototype Scope:**

A computer code with black text

AI-generated content may be incorrect.