**Cognizant Deep Nurture 4.0 Hands-on Exercise - Week 4**

Hands on 1

Create a Spring Web Project using MavenFollow steps below to create a project:   
1.Go tohttps://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.SME to walk through the following aspects related to the project created:   
1.src/main/java - Folder with application code   
2.src/main/resources - Folder for application configuration   
3.src/test/java - Folder with code for testing the application   
4.SpringLearnApplication.java - Walkthrough the main() method.   
5.Purpose of @SpringBootApplication annotation6.pom.xml1.Walkthrough all the configuration defined in XML file2.Open 'Dependency Hierarchy' and show the dependency tree.   
   
   
   
   
1.Importing into Eclipse

1. Extract the .zip file to your workspace.
2. In Eclipse:   
    File > Import > Maven > Existing Maven Projects > Browse your extracted folder > Finish.

3. Build the Project with Proxy (Cognizant network)

bash

CopyEdit

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 \

-Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 \

-Dhttp.proxyUser=123456

4. Add Logging in SpringLearnApplication.java

📄 src/main/java/com/cognizant/springlearn/SpringLearnApplication.java

java

CopyEdit

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("Inside main method - Application Started Successfully");

}

}

   
OUTPUT:   
   
 

4df056f807ba75b482a06e6eb28bc186.png

**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.

<?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="dateFormat" class="java.text.SimpleDateFormat">

<constructor-arg value="dd/MM/yyyy" />

</bean>

</beans>

Create new method displayDate() in SpringLearnApplication.java

In displayDate() method create the ApplicationContext. Refer code below:

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

Get the dateFormat using getBean() method. Refer code below.

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

Using the format variable try to parse string '31/12/2018' to Date class and display the result using System.out.println.

Run the application as 'Java Application' and check the result in console log output.

Troubleshooting Tips

If the tomcat port has a conflict and the server is not starting include the below

property in application.properties file in src/main/resources folder.

   
   
1. Create XML Bean Configuration

**Path**: src/main/resources/date-format.xml

xml

CopyEdit

<?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="dateFormat" class="java.text.SimpleDateFormat">

<constructor-arg value="dd/MM/yyyy" />

</bean>

</beans>

📄 2. Modify SpringLearnApplication.java

**Path**: src/main/java/com/cognizant/springlearn/SpringLearnApplication.java

java

CopyEdit

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.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@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 parsedDate = format.parse("31/12/2018");

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

} catch (ParseException e) {

e.printStackTrace();

}

LOGGER.info("END");

}

}

OUTPUT:



   
 