**Week4\_Managing SimpleDateFormat using Spring XML Configuration**

#### Step 1: Project Setup

* The exercise was conducted within the existing spring-learn Maven project in the Eclipse IDE.

#### Step 2: Creating the Spring XML Configuration File

A new Spring configuration file was created at the following location, which is the standard classpath root for resources:

* File: date-format.xml
* Location: src/main/resources/

#### Step 3: Defining the SimpleDateFormat Bean

* The date-format.xml file was configured to instruct the Spring container to create and manage a SimpleDateFormat object.

**date-format.xml:**

<beans>

<bean id="dateFormat" class="java.text.SimpleDateFormat">

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

</bean>

</beans>

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "https://aistudio.google.com/prompts/_blank).Xml

* The <bean> tag defines a manageable object.
* id="dateFormat" provides a unique name for this object.
* The <constructor-arg> tag injects the date pattern "dd/MM/yyyy" into the object upon its creation.

#### Step 4: Implementing the Java Application Logic

The main application class was modified to load the Spring context and retrieve the configured bean.

**SpringLearnApplication.java:**

* **Load the Spring Container:**  
  An ApplicationContext was created by loading the XML file from the classpath.

// THIS LINE IS NOW CORRECTED TO MATCH YOUR FOLDER STRUCTURE

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

// Import statements required for this exercise

**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.autoconfigure.SpringBootApplication;

@SpringBootApplication

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

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

// We are calling our new method directly to demonstrate the core Spring container.

*displayDate*();

}

/\*\*

\* This method demonstrates loading a Spring XML configuration file,

\* getting a bean from the Spring container, and using it.

\*/

**public** **static** **void** displayDate() {

System.***out***.println("STARTING: displayDate method");

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

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

**try** {

String dateString = "31/12/2018";

Date date = format.parse(dateString);

// This is the line that will print the final result to the console.

System.***out***.println(date);

} **catch** (ParseException e) {

e.printStackTrace();

}

System.***out***.println("FINISHED: displayDate method");

((ClassPathXmlApplicationContext) context).close();

}

}

* **Retrieve the Bean:**  
  The pre-configured SimpleDateFormat object was fetched from the container using its unique ID.

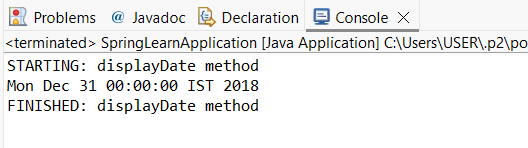
SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

* **Utilize the Bean:**  
  The retrieved format object was used to parse a date string.

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

System.out.println(date);

**Output**

****