**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.

**Solution**

**date-format.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  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="dateFormat" class="java.text.SimpleDateFormat">  
 <constructor-arg value="dd/MM/yyyy" />  
 </bean>  
  
</beans>

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.text.SimpleDateFormat;

import java.util.Date;

public class SpringLearnApplication {

public static void main(String[] args) {

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

SimpleDateFormat sdf = (SimpleDateFormat) context.getBean("dateFormat");

try {

Date date = sdf.parse("12/07/2025");

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

} catch (Exception e) {

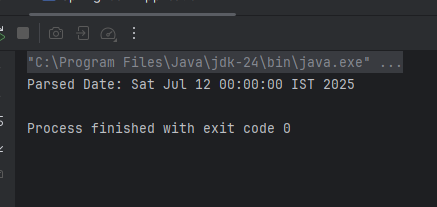
System.out.println("Something went wrong while parsing date: " + e.getMessage());

}

}

}

**Output**



**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

logging.level.org.springframework=info

logging.level.com.cognizant.springlearn=debug

logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n

* In SpringLearnApplication.java include the following imports:

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

* Include the below static variable in SpringLearnApplication.java:

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

* Include info log on start and end of method. Debug log for displaying the date (refer code below)

public void displayDate() {

LOGGER.info(“START”);

//..

LOGGER.debug(date);

//..

LOGGER.info(“END”);

}

**IMPORTANT NOTE:** Going forward all methods should incorporate logging as specified above. **Never** use System.out.println().

**Solution**

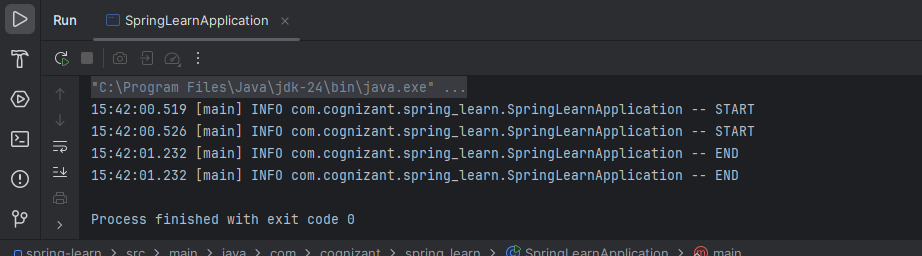
**application.properties**

logging.level.org.springframework=info  
logging.level.com.cognizant.spring\_learn=debug  
logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n

**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;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
public class SpringLearnApplication {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 LOGGER.info("START");  
 displayDate();  
 LOGGER.info("END");  
 }  
  
 public static void displayDate() {  
 LOGGER.info("START");  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat sdf = (SimpleDateFormat) context.getBean("dateFormat");  
 try {  
 Date date = sdf.parse("12/07/2025");  
 LOGGER.debug("Parsed Date: {}", date);  
 } catch (Exception e) {  
 LOGGER.error("Error while parsing date", e);  
 }  
 LOGGER.info("END");  
 }  
}

**Output**



**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.  
   
**Follow steps below to demonstrate Singleton Scope**

* Include a line in displayCountry() to get country bean reference one more time from the same application context. Only the third line of the below code snippet should be copied and pasted.

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

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

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

* The constructor will be called only once, which means that only one instance of Country bean is created

**Solution**

**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="country" class="com.cognizant.spring\_learn.Country" />

</beans>

**SpringLearnerApplication.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) {

displayCountry();

}

public static void displayCountry() {

LOGGER.info("START");

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

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

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

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

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

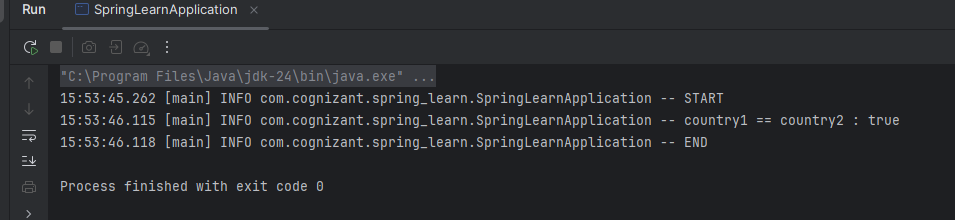
LOGGER.info("country1 == country2 : {}", country1 == country2);

LOGGER.info("END");

}

}

**Output**



**Follow steps below to demonstrate Prototype Scope**

* Include scope="prototype" attribute in bean definition xml.

<bean id="country" class="com.cognizant.springlearn.Country" scope="prototype">

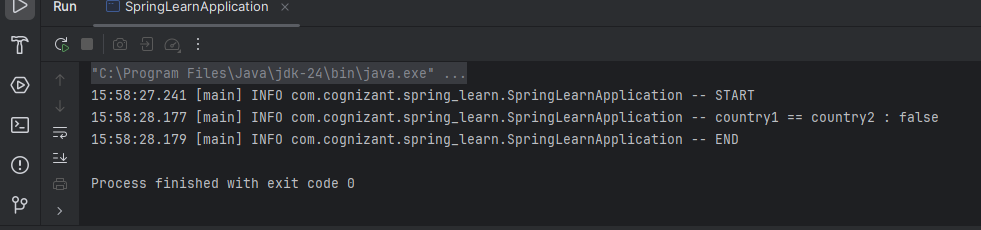
* Run the application
* Constructor will be called twice, which means that two instances of country is created.

**Solution**

**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="country" class="com.cognizant.spring\_learn.Country" scope="prototype" />  
  
  
</beans>

**Output**



**Hands on 6**

**Spring Core – Load list of countries from Spring Configuration XML**   
   
Our main objective was to retrieve the list of four countries for the airlines website. Refer steps below to get this incorporated.

* Create a separate bean for each of the four country in country.xml.
* Create an ArrayList of Country in country.xml. Refer code below.

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

<constructor-arg>

<list>

<ref bean="in"></ref>

<ref bean="us"></ref>

<ref bean="de"></ref>

<ref bean="jp"></ref>

</list>

</constructor-arg>

</bean>

* Include new method displayCountries() in SpringLearnApplication.java
* In displayCountries() read the country list created above
* Display the list of countries as debug log.

SME to provide detailing on below aspects:

* <list>
* <ref>
* bean attribute

**IMPORTANT NOTE**: Do not forget to include the start and end logs in this new method.

**Solution**

**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">  
  
 <!-- Individual country beans -->  
 <bean id="in" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
 <bean id="us" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
  
 <bean id="de" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
  
 <bean id="jp" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
  
 <!-- List of countries -->  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="in"/>  
 <ref bean="us"/>  
 <ref bean="de"/>  
 <ref bean="jp"/>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

**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;  
  
import java.util.List;  
  
public class SpringLearnApplication {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 displayCountries();  
 }  
  
 public static void displayCountries() {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
  
 List<Country> countryList = context.getBean("countryList", List.class);  
  
 LOGGER.debug("Country List: {}", countryList);  
  
 LOGGER.info("END");  
 }  
}

**Output**

