WEEK 4

ADDITIONAL HANDS ON

**Hands-on 2:**

**Load SimpleDateFormat from Spring XML**

**src/main/resources/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>

**Method in SpringLearnApplication.java**

public 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);

}

LOGGER.info("END");

}

**src/main/resources/application.properties**

logging.level.org.springframework=info

logging.level.com.cognizant=debug

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

**Logger in SpringLearnApplication.java**

full

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

**Hands-on 3**

**Spring Core – Incorporate Logging**

Add Logging Configuration

# Log level configuration

logging.level.org.springframework=info

logging.level.com.cognizant=debug

# Console log format

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

SpringLearnApplication.java

package com.cognizant;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication {

// Initialize Logger

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(SpringLearnApplication.class, args);

SpringLearnApplication app = new SpringLearnApplication();

app.displayDate();

LOGGER.info("END");

}

public 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);

}

LOGGER.info("END");

}

}

Spring XML Configuration for Date Format

<?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">

<!-- Define a reusable date format bean -->

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

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

</bean>

</beans>

Output

240712|18:32:01.200|main | INFO|SpringLearnApplication | main|START

240712|18:32:01.314|main | INFO|SpringLearnApplication | displayDate|START

240712|18:32:01.322|main |DEBUG|SpringLearnApplication | displayDate|Parsed Date: Mon Dec 31 00:00:00 IST 2018

240712|18:32:01.322|main | INFO|SpringLearnApplication | displayDate|END

240712|18:32:01.322|main | INFO|SpringLearnApplication | main|END

**Hands-on 4:**

**Load Country from XML**

**Country.java in com.cognizant.springlearn**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

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

private String code;

private String name;

public Country() {

LOGGER.debug("Inside Country Constructor.");

}

public String getCode() {

LOGGER.debug("Getting code");

return code;

}

public void setCode(String code) {

LOGGER.debug("Setting code");

this.code = code;

}

public String getName() {

LOGGER.debug("Getting name");

return name;

}

public void setName(String name) {

LOGGER.debug("Setting name");

this.name = name;

}

@Override

public String toString() {

return "Country{" + "code='" + code + '\'' + ", name='" + name + '\'' + '}';

}

}

src/main/resources/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

https://www.springframework.org/schema/beans/spring-beans.xsd">

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

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

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

</bean>

</beans>

**Method in SpringLearnApplication.java**

public 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");

}

**Hands-on 5**

*Spring Core: Demonstration of Singleton Scope and Prototype Scope*

**Country.java**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

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

private String code;

private String name;

public Country() {

LOGGER.debug("Inside Country Constructor.");

}

public String getCode() {

LOGGER.debug("Getting code");

return code;

}

public void setCode(String code) {

LOGGER.debug("Setting code");

this.code = code;

}

public String getName() {

LOGGER.debug("Getting name");

return name;

}

public void setName(String name) {

LOGGER.debug("Setting name");

this.name = name;

}

@Override

public String toString() {

return "Country{" +

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

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

'}';

}

}

**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

https://www.springframework.org/schema/beans/spring-beans.xsd">

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

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

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

</bean>

</beans>

**For Prototype Scope**

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

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

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

</bean>

**application.properties**

logging.level.org.springframework=info

logging.level.com.cognizant=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;

import com.cognizant.springlearn.Country;

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) {

LOGGER.info("START");

SpringLearnApplication app = new SpringLearnApplication();

app.displayCountry();

LOGGER.info("END");

}

public 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);

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

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

LOGGER.info("END");

}

}

**Output Explanation**

DEBUG|Inside Country Constructor.

DEBUG|Country 1 : Country{code='IN', name='India'}

DEBUG|Country 2 : Country{code='IN', name='India'}

INFO |country1 == country2 ? true

DEBUG|Inside Country Constructor.

DEBUG|Inside Country Constructor.

DEBUG|Country 1 : ...

DEBUG|Country 2 : ...

INFO |country1 == country2 ? false

**Hands-on 6:**

**List of Countries**

**country.xml**

<bean id="in" class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean id="us" class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean id="de" class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean id="jp" class="com.cognizant.springlearn.Country">

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

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

</bean>

<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>

**SpringLearnApplication.java**

public void displayCountries() {

LOGGER.info("START");

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

List<Country> countries = context.getBean("countryList", ArrayList.class);

LOGGER.debug("Countries : {}", countries);

LOGGER.info("END");

}