**Objectives**

* Demonstrate creation of Spring Boot Application
  + Spring initializr, https://start.spring.io, @SpringBootApplication, SpringApplication.run()
    - Ref - https://start.spring.io
* Explain the need and benefits of Spring Boot
  + Makes Java development easy, avoids tedious development steps, reduces development time, avoids writing boilerplate code, provides embedded tomcat server, avoid XML configuration
    - Ref - https://www.journaldev.com/7969/spring-boot-tutorial
* Demonstrate loading bean from spring configuration file
  + Spring configuration xml, spring xml schema spring-beans.xsd, <bean>, id, class, <constructor-arg>, <property>, name, value, ClassPathXmlApplicationContext, ApplicationContext, context.getBean(), singleton scope, prototype scope
    - Ref - https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html
    - IoC Container - https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html#beans
    - Scopes - https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html#beans-factory-scopes
    - Constructor Injection - https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html#beans-constructor-injection
    - Setter method injection - https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html#beans-setter-injection
* Demonstrate inclusion of logging in Spring Boot Application
  + application.properties, logging.level, logging.pattern, server.port, LoggerFactory, Logger, log levels (trace, debug, info, warn, error)
    - Ref - https://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-logging.html

**Hands on 1**

**Create a Spring Web Project using Maven**   
  
A screenshot of a computer program

AI-generated content may be incorrect.

**Hands on 2**

**Spring Core – Load SimpleDateFormat from Spring Configuration XML** 

**SpringLearnApplication.java** -

package com.cognizant.spring\_learn;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

public static void main(String[] args) {

*displayDate*(); // call the method here

}

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("Error parsing date: " + e.getMessage());

}

}

}

**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>

OUTPUT-

A screenshot of a computer

AI-generated content may be incorrect.

**Hands on 3**

**Spring Core - Incorporate Logging**   
  
**SpringLearnApplication.java** –

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

**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;

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

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

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

***LOGGER***.info("Main method START");

*displayDate*();

***LOGGER***.info("Main method END");

}

**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("Error parsing date", e);

}

***LOGGER***.info("END");

}

}

**Application.properties-**

spring.application.name=spring-learn

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

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

**Hands on 4**

**Spring Core – Load Country from Spring Configuration XML**   
  
**SpringLearnApplication.java** –

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

**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;

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

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

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

***LOGGER***.info("START - main()");

*displayDate*();

*displayCountry*();

***LOGGER***.info("END - main()");

}

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

***LOGGER***.info("START - displayDate()");

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("Error parsing date", e);

}

***LOGGER***.info("END - displayDate()");

}

**public** **static** **void** displayCountry() {

***LOGGER***.info("START - displayCountry()");

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

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

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

***LOGGER***.info("END - displayCountry()");

}

}

**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.spring\_learn.Country">

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

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

</bean>

</beans>

**County.java-**

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

**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("getCode() called");

**return** code;

}

**public** **void** setCode(String code) {

***LOGGER***.debug("setCode() called with value: {}", code);

**this**.code = code;

}

**public** String getName() {

***LOGGER***.debug("getName() called");

**return** name;

}

**public** **void** setName(String name) {

***LOGGER***.debug("setName() called with value: {}", name);

**this**.name = name;

}

@Override

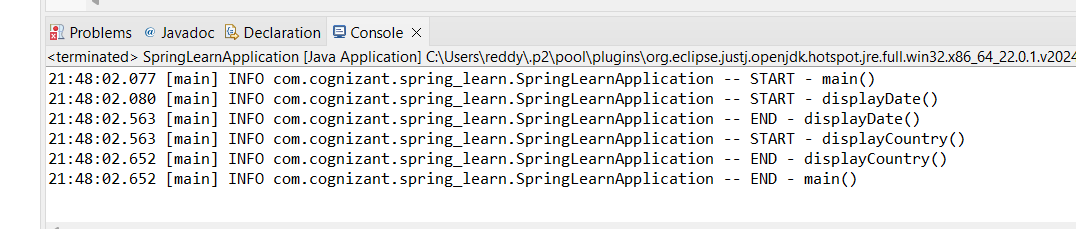
**public** String toString() {

**return** "Country [code=" + code + ", name=" + name + "]";

}

}

OUTPUT –



**Hands on 5**

**Spring Core – Demonstration of Singleton Scope and Prototype Scope**   
  
**SpringLearnApplication.java** –

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

**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;

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

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

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

***LOGGER***.info("START - main()");

*displayDate*();

*displayCountry*();

***LOGGER***.info("END - main()");

}

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

***LOGGER***.info("START - displayDate()");

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("Error parsing date", e);

}

***LOGGER***.info("END - displayDate()");

}

**public** **static** **void** displayCountry() {

***LOGGER***.info("START - displayCountry()");

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

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

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

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

***LOGGER***.debug("Another Country : {}", anotherCountry);

***LOGGER***.info("Are both country objects the same? {}", country == anotherCountry);

***LOGGER***.info("END - displayCountry()");

}

}

**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.spring\_learn.Country" scope="prototype">

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

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

</bean>

</beans>

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

**Hands on 6**

**Spring Core – Load list of countries from Spring Configuration XML** 

**SpringLearnApplication.java** –

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

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**import** java.util.List;

**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) **throws** ParseException {

***LOGGER***.info("START - main()");

*displayDate*();

*displayCountry*();

*displayCountries*();

***LOGGER***.info("END - main()");

}

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

***LOGGER***.info("START - displayDate()");

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

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

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

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

***LOGGER***.info("END - displayDate()");

}

**public** **static** **void** displayCountry() {

***LOGGER***.info("START - displayCountry()");

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

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

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

***LOGGER***.info("Are both country objects the same? {}", country == anotherCountry);

***LOGGER***.info("END - displayCountry()");

}

**public** **static** **void** displayCountries() {

***LOGGER***.info("START - displayCountries()");

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

@SuppressWarnings("unchecked")

List<Country> countryList = (List<Country>) context.getBean("countryList");

**for** (Country country : countryList) {

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

}

***LOGGER***.info("END - displayCountries()");

}

}

**Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {

System.*out*.println("Country constructor called");

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}