**Hands on 1**

**Create a Spring Web Project using Maven**

Directory:  
spring-learn/

├── pom.xml

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── cognizant/

│ │ │ └── springlearn/

│ │ │ └── SpringLearnApplication.java

│ │ └── resources/

│ │ ├── application.properties

**Code for pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.1.1</version>

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-learn</name>

<description>Spring Boot Web Hands-On</description>

<packaging>jar</packaging>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Code for SpringLearnApplication.java:**

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) {

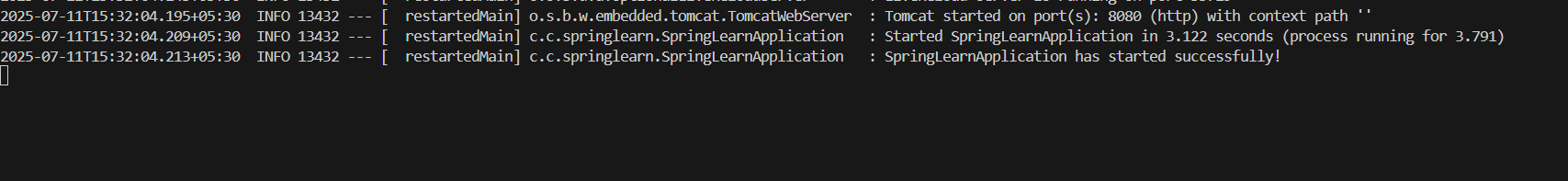
SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("SpringLearnApplication has started successfully!");

}

}

Output:

****

**Hands on 2**

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

Directory:  
spring-learn/

├── pom.xml

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── cognizant/

│ │ │ └── springlearn/

│ │ │ └── SpringLearnApplication.java

│ │ └── resources/

│ │ ├── application.properties

│ │ └── date-format.xml

**Code for 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>

**Code for SpringLearnApplication .java:**

package com.cognizant.springlearn;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.boot.SpringApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

    public static void main(String[] args) {

        SpringApplication.run(SpringLearnApplication.class, args);

       System.out.println("Spring Boot Application Started");

        displayDate();

    }

    public static void displayDate(){

        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) {

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

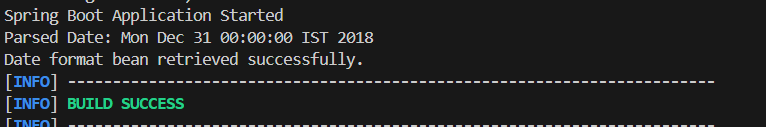
        }

       System.out.println("Date format bean retrieved successfully.");

    }

}

**Output:**

****

**Hands on 3**

**Spring Core - Incorporate Logging** 

Directory:  
spring-learn/

├── pom.xml

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── cognizant/

│ │ │ └── springlearn/

│ │ │ └── SpringLearnApplication.java

│ │ └── resources/

│ │ ├── application.properties

│ │ └── date-format.xml

**Code for 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>

**Code for SpringLearnApplication.java:**

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.boot.SpringApplication;

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) {

        SpringApplication.run(SpringLearnApplication.class, args);

        LOGGER.info("SpringLearnApplication has started successfully.");

        displayDate();

    }

    public static void displayDate(){

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

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

        try {

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

            LOGGER.debug("Parsed Date: {}", parsedDate.toString());

        } catch (ParseException e) {

            LOGGER.error("Parse error: {}", e.getMessage());

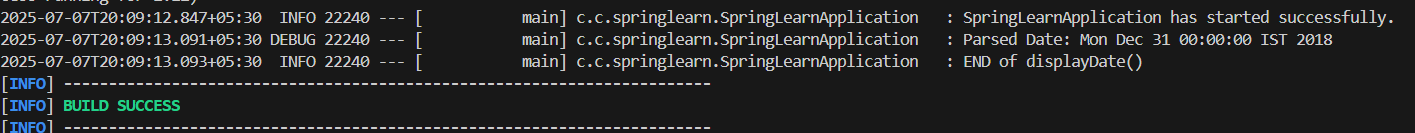
        }

        LOGGER.info("END of displayDate()");

    }

}

**Output:**



**Hands on 4**

**Spring Core – Load Country from Spring Configuration XML**   
Directory:  
spring-learn/

├── pom.xml

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── cognizant/

│ │ │ └── springlearn/

│ │ │ └── SpringLearnApplication.java

│ │ │ └──Country.java

│ │ └── resources/

│ │ ├── application.properties

│ │ └──country.xml

**Code for 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>

**Code for 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("Inside getCode()");

        return code;

    }

    public void setCode(String code) {

        LOGGER.debug("Inside setCode()");

        this.code = code;

    }

    public String getName() {

        LOGGER.debug("Inside getName()");

        return name;

    }

    public void setName(String name) {

        LOGGER.debug("Inside setName()");

        this.name = name;

    }

    @Override

    public String toString() {

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

    }

}

**Code for SpringLearnApplication.java:**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

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 of main()");

        SpringApplication.run(SpringLearnApplication.class, args);

        displayCountry();

        LOGGER.info("END of main()");

    }

     public static void displayCountry() {

        LOGGER.info("START of displayCountry()");

        // Load Spring XML configuration

        ApplicationContext context =

                new ClassPathXmlApplicationContext("country.xml");

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

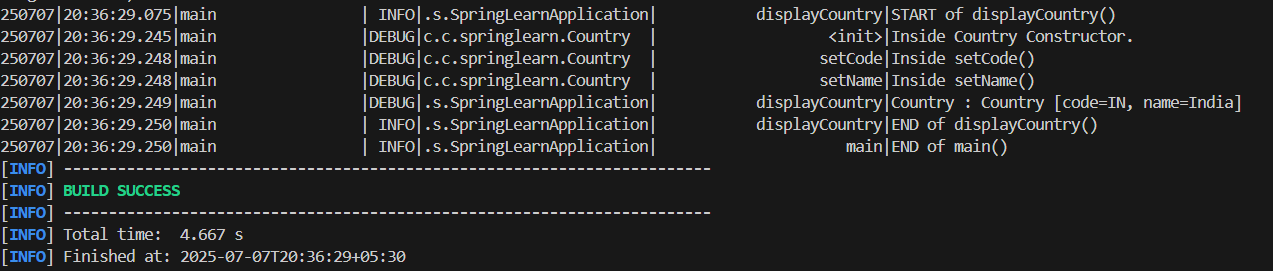
        LOGGER.debug("Country : {}", country);

        LOGGER.info("END of displayCountry()");

    }

}

**Output:**



**Hands on 5**

**Spring Core – Demonstration of Singleton Scope and Prototype Scope**  Directory:  
spring-learn/

├── pom.xml

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── cognizant/

│ │ │ └── springlearn/

│ │ │ └── SpringLearnApplication.java

│ │ │ └──Country.java

│ │ └── resources/

│ │ ├── application.properties

│ │ └──country.xml

**Code for 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" scope=”prototype”>

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

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

    </bean>

</beans>

**Code for 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("Inside getCode()");

        return code;

    }

    public void setCode(String code) {

        LOGGER.debug("Inside setCode()");

        this.code = code;

    }

    public String getName() {

        LOGGER.debug("Inside getName()");

        return name;

    }

    public void setName(String name) {

        LOGGER.debug("Inside setName()");

        this.name = name;

    }

    @Override

    public String toString() {

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

    }

}

**Code for SpringLearnApplication.java:**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

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 of main()");

        SpringApplication.run(SpringLearnApplication.class, args);

        displayCountry();

        LOGGER.info("END of main()");

    }

     public static void displayCountry() {

        LOGGER.info("START of displayCountry()");

        // Load Spring XML configuration

        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("END of displayCountry()");

    }

}

**Output:**

