**SPRING REST USING SPRING BOOT 3**

1. **Spring -rest-handson**

* **Hands on 1**

**Create a Spring Web Project using Maven**

1. Spring Initializr

* Project: Maven
* Language: Java
* Spring Boot: 3.5.3
* Group: com.cognizant
* Artifact: spring-learn
* Name: spring-learn
* Description: spring boot web application
* Package name: com.cognizant.springlearn
* Packaging: jar
* Java: 17

Dependencies to add:

* Spring Boot Dev Tools
* Spring Web

-Generate ZIP file and extract it.

1. Import into eclipse

* Copy the extracted file path.
* Open Eclipse IDE
* File -> Import -> Maven -> Existing Maven Projects
* Browse or select your extracted folder
* Finish

1. Build the project

* In eclipse -> windows -> show view -> terminal
* Click Open Terminal
* Navigate to your project directory and run:

mvn clean package (or)

* In command line run

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

1. Project Structure

* Src/main/java -> application code
* Src/main/resources -> application.properties
* Src/test/java -> unit testing
* Pom.xml -> maven configuration
* SpringLearnApplication.java -> main class (main() method)

1. Src/main/java -> com.cognizant.springlearn -> SpringLearnApplication.java

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.class, args);

}

}

1. pom.xml

<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>

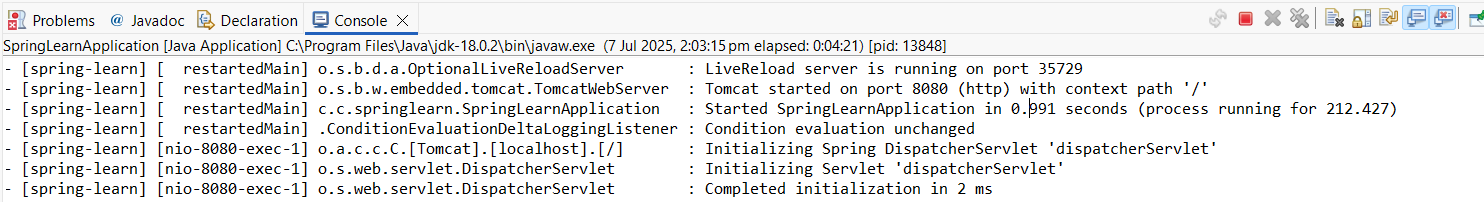
<scope>test</scope>

</dependency>

</dependencies>

1. Right click SpringLearnApplication.java -> Run As -> Java Application

**OUTPUT:**



* To see output in browser:
* Create controller REST class
* Src/main/java -> com.cognizant.springlearn -> HelloController

package com.cognizant.springlearn;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

@GetMapping("/")

public String sayHello() {

return "Hello, Spring Boot!";

}

@GetMapping("/greet")

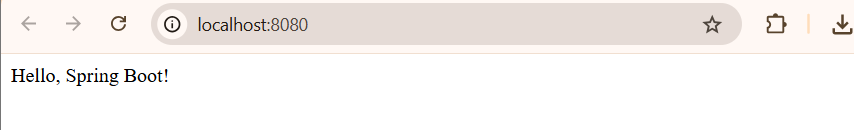
public String greet() {

return "Greetings from the Spring Boot Controller to Hamsini!";

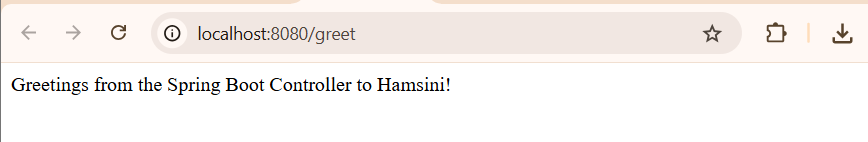
}

}

* Src/main/java -> com.cognizant.springlearn -> SpringLearnApplication.java
* <http://localhost:8080/>



* <http://localhost:8080/greet/>



* **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.

1. date-format.xml

* src/main/resources -> New -> File
* filename: 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

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

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

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

</bean>

</beans>

1. Update SpringLearnApplication.java or create new class

* New class: src/main/java -> com.cognizant.springlearn -> SpringLearnApplication1

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication1 {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate();

}

public static void displayDate() {

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

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

try {

Date date = format.parse("28/12/2003");

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

} catch (ParseException e) {

e.printStackTrace();

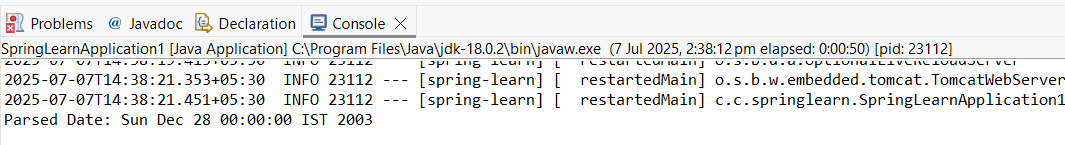
}

}

}

1. Right click SpringLearnApplication1 -> Run As -> Java Application

**OUTPUT**: Last line of the picture shows the parsed date.



* **Hands on 3**

**Spring Core - Incorporate Logging**

1. application.properties

* src/main/resources -> application.properties (file)

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

1. SpringLearnApplication2.java

* Src/main/java -> com.cognizant.springlearn -> SpringLearnApplication2

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication2 {

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate(); // call the method

}

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("07/09/2003");

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

} catch (ParseException e) {

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

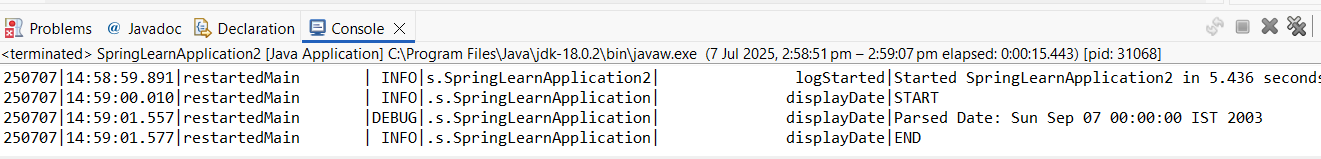
}

LOGGER.info("END");

}

}

**OUTPUT:**



* **Hands on 4**

**Spring Core – Load Country from Spring Configuration XML**   
  
An airlines website is going to support booking on four countries. There will be a drop down on the home page of this website to select the respective country. It is also important to store the two-character ISO code of each country. 

|  |  |
| --- | --- |
| **Code** | **Name** |
| US | United States |
| DE | Germany |
| IN | India |
| JP | Japan |

We can continue the hands on 4 separately by creating a new class or can continue in hands on 2 main application also.

1. Src/main/resources -> New -> File

* 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.springlearn.Country">

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

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

</bean>

</beans>

1. Src/main/java -> com.xognizant.springlearn -> Country (class)

* 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: {}", code);

return code;

}

public void setCode(String code) {

LOGGER.debug("Setting code: {}", code);

this.code = code;

}

public String getName() {

LOGGER.debug("Getting name: {}", name);

return name;

}

public void setName(String name) {

LOGGER.debug("Setting name: {}", name);

this.name = name;

}

@Override

public String toString() {

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

} }

1. In hands on 2 or 3 class (SpringLearnApplication2) add displayCountry method

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication2{

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate(); // Hands-on 2 & 3

**displayCountry(); // Hands-on 4**

}

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

}

LOGGER.info("END");

}

**public static void displayCountry() {**

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

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

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

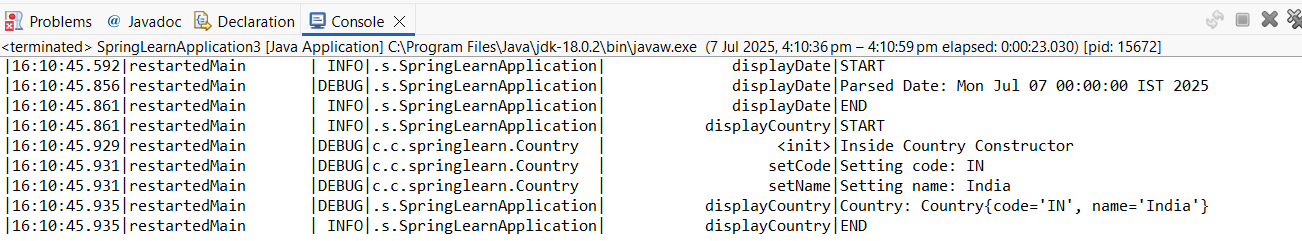
**LOGGER.debug("Country: {}", country.toString());**

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

**}**

}

**OUTPUT:** displayDate() along with displayCountry()



**Or**

1. Create new SpringLearnApplication3.java (we won’t get displayDate() method )

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication3 {

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayCountry();

}

public static void displayCountry() {

LOGGER.info("START");

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

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

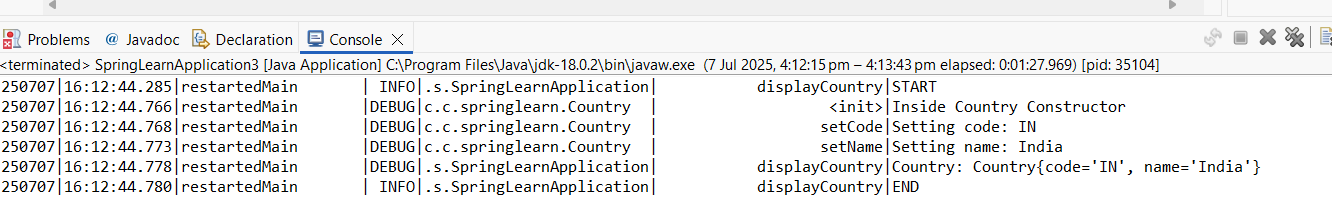
LOGGER.debug("Country: {}", country.toString());

LOGGER.info("END");

}

}

**OUTPUT:**



* **Hands on 5**

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

1. Update displayCountry() in StringLearnApplication3.java

public static void displayCountry() {

LOGGER.info("START");

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

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

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

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

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

LOGGER.info("END");

}

1. **Singleton Scope**

* Src/main/resources -> new -> file -> country.xml

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

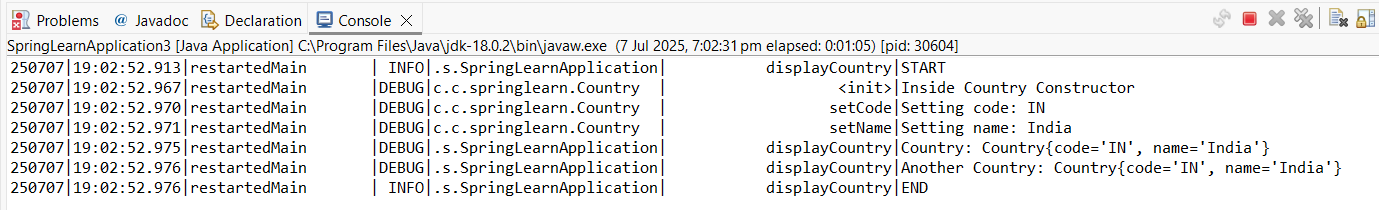
<property name="code" value="IN" />

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

</bean>

1. StringLearnApplication3 -> Run As -> Java Application

**OUTPUT:**



1. **Prototype Scope**

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

scope="prototype">

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

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

</bean>

1. StringLearnApplication3 -> Run As -> Java Application

**OUTPUT:**



* **Hands on 6**

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

1. Update 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>">

<!-- Single Country Bean (Hands-on 4 & 5) -->

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

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

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

</bean>

<!-- Individual Beans for Hands-on 6 -->

<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>

<!-- List of Countries (Hands-on 6) -->

<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>

1. Add displayCountries() method in SpringLearnApplication3.java

public static void displayCountries() {

LOGGER.info("START");

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

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

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

LOGGER.info("END");

}

1. Call DisplayCountries() from main()

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate(); // From Hands-on 2 & 3

displayCountry(); // From Hands-on 4

**displayCountries();**

}

**OUTPUT:** output for displayCountries() method

