**WEEK 4 : SPRING REST USING SPRINGBOOT HANDSON**

**SUPERSET ID : 6386707**

**HANDSON 1: CREATE A SPRING WEB PROJECT USING MAVEN**

**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) { System.out.println("Inside main method..."); SpringApplication.run(SpringLearnApplication.class, args);

}

}

**HelloController.java**

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 "Welcome to Spring Boot!";

}

}

**OUTPUT**

Wecome to Spring Boot!

**HANDSON 2 : SPRING CORE LOAD SIMPLEDATEFORMAT FROM SPRING CONFIGURATION XML**

**SpringLearnApplication.java**

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication; import

org.springframework.boot.autoconfigure.SpringBootApplicati on;

import org.springframework.context.ApplicationContext; import

org.springframework.context.support.ClassPathXmlApplicatio nContext;

import java.text.SimpleDateFormat; import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) throws Exception { System.out.println("Inside main method...");

SpringApplication.run(SpringLearnApplication.class, args);

displayDate(); // Call your new method

}

public static void displayDate() throws Exception { ApplicationContext context = new

ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format =

context.getBean("dateFormat", SimpleDateFormat.class); Date parsedDate = format.parse("13/07/2025");

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

}

}

**OUTPUT**

Inside main method

Parsed Date:Mon Mar 12 22:48:00 IST 2025

**HANDSON 3 SPRING CORE – INCORPORATE LOGGING**

**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;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationCon text;

import java.text.SimpleDateFormat;

import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) throws Exception {

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

SpringApplication.run(SpringLearnApplication.class, args);

displayDate();

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

}

public static void displayDate() throws Exception { LOGGER.info("START - displayDate()");

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

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

Date parsedDate = format.parse("31/12/2018"); LOGGER.debug("Parsed Date = {}", parsedDate);

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

}

}

**OUTPUT**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 240712 | 13:45:33. | main | INFO | SpringLe | main | START - |
|  | 123 |  |  | arnAppli |  | main() |
|  |  |  |  | cation |  |  |
| 240712 | 13:45:33. | main | INFO | SpringLe | displayD | START - |
|  | 456 |  |  | arnAppli | ate | displayD |
|  |  |  |  | cation |  | ate() |
| 240712 | 13:45:33. | main | DEBUG | SpringLe | displayD | Parsed |
|  | 789 |  |  | arnAppli | ate | Date = |
|  |  |  |  | cation |  | Mon Dec |
|  |  |  |  |  |  | 31 |
|  |  |  |  |  |  | 00:00:0 |
|  |  |  |  |  |  | 0 IST |
|  |  |  |  |  |  | 2018 |
| 240712 | 13:45:33. | main | INFO | SpringLe | displayD | END - |
|  | 790 |  |  | arnAppli | ate | displayD |
|  |  |  |  | cation |  | ate() |
| 240712 | 13:45:33. | main | INFO | SpringLe | main | END - |
|  | 791 |  |  | arnAppli |  | main() |
|  |  |  |  | cation |  |  |
|  |  |  |  |  |  |  |

**HANDSON 4 LOAD COUNTRY FROM SPRING CONFIGURATION XML**

**Country.xml**

<?xml version="1.0" encoding="UTF-8"?>

<beans

[xmlns="http://www.springframework.org/schema/beans"](http://www.springframework.org/schema/beans)

[xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)

xsi:schemaLoca[tion="http://www.springframework.org/schem](http://www.springframework.org/schem) a/beans

https://[www.springframework.org/schema/beans/spring-](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>

**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("getCode() called."); return code;

}

public void setCode(String code) {

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

this.code = code;

}

public String getName() {

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

return name;

}

public void setName(String name) {

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

this.name = name;

}

@Override

public String toString() {

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

"]";

}

}

**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;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationCon text;

import java.text.SimpleDateFormat;

import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) throws Exception {

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

SpringApplication.run(SpringLearnApplication.class, args);

displayDate();

displayCountry();

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

}

public static void displayDate() throws Exception {

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

ApplicationContext context = new

ClassPathXmlApplicationContext("date-format.xml");

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

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

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

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()");

}

}

**OUTPUT**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 240712 | 14:10:45. | main | INFO | SpringLe | main | START - |
|  | 101 |  |  | arnAppli |  | main() |
|  |  |  |  | cation |  |  |
| 240712 | 14:10:45. | main | INFO | SpringLe | displayD | START - |
|  | 453 |  |  | arnAppli | ate | displayD |
|  |  |  |  | cation |  | ate() |
| 240712 | 14:10:45. | main | DEBUG | SpringLe | displayD | Parsed |
|  | 754 |  |  | arnAppli | ate | Date = |
|  |  |  |  | cation |  | Mon Dec |
|  |  |  |  |  |  | 31 |
|  |  |  |  |  |  | 00:00:0 |
|  |  |  |  |  |  | 0 IST |
|  |  |  |  |  |  | 2018 |
| 240712 | 14:10:45. | main | INFO | SpringLe | displayD | END - |
|  | 754 |  |  | arnAppli | ate | displayD |
|  |  |  |  | cation |  | ate() |
| 240712 | 14:10:45. | main | INFO | SpringLe | displayC | START - |
|  | 755 |  |  | arnAppli | ountry | displayC |
|  |  |  |  | cation |  | ountry() |
| 240712 | 14:10:45. | main | DEBUG | Country | <init> | Inside |
|  | 756 |  |  |  |  | Country |
|  |  |  |  |  |  | Construc |
|  |  |  |  |  |  | tor. |
| 240712 | 14:10:45. | main | DEBUG | Country | setCode | setCode( |
|  | 756 |  |  |  |  | ) called. |
| 240712 | 14:10:45. | main | DEBUG | Country | setName | setName |
|  | 756 |  |  |  |  | () called. |
| 240712 | 14:10:45. | main | DEBUG | Country | toString | Country : |
|  | 757 |  |  |  |  | Country |
|  |  |  |  |  |  | [code=IN |
|  |  |  |  |  |  | , |
|  |  |  |  |  |  | name=In |
|  |  |  |  |  |  | dia] |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 240712 | 14:10:45.  757 | main | INFO | SpringLe arnAppli cation | displayC ountry | END -  displayC ountry() |
| 240712 | 14:10:45. | main | INFO | SpringLe | main | END - |
|  | 757 |  |  | arnAppli |  | main() |
|  |  |  |  | cation |  |  |

**HANDSON 5 : DEMONSTRATION OF SINGLETON SCOPE AND PROTOTYPE SCOPE**

**Country.java**

package com.cognizant.springlearn;

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 + "]";

}

}

**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>

**SpringScopeDemo.java**

package com.cognizant.springlearn;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringScopeDemo {

public static void main(String[] args) {

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

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

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

System.out.println("Country 1: " + country1);

System.out.println("Country 2: " + country2);

System.out.println("Are both references equal? " + (country1 == country2));

}

}

**OUTPUT**

Country constructor called

Country constructor called

Country 1: Country [code=IN, name=India]

Country 2: Country [code=IN, name=India]

Are both references equal? false

**HANDSON 6 : Load list of countries from Spring Configuration XML**

**Country.java**

package com.cognizant.springlearn;

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 + "]";

}

}

**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="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>

</beans>

**SpringLearnApplication.java**

package com.cognizant.springlearn;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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 - displayCountries()");

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

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

for (Country country : countryList) {

LOGGER.debug(country.toString());

}

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

}

}

**OUTPUT**

INFO SpringLearnApplication: START - displayCountries()

DEBUG SpringLearnApplication: Country [code=IN, name=India]

DEBUG SpringLearnApplication: Country [code=US, name=United States]

DEBUG SpringLearnApplication: Country [code=DE, name=Germany]

DEBUG SpringLearnApplication: Country [code=JP, name=Japan]

INFO SpringLearnApplication: END - displayCountries()