# SPRING REST USING SPRING BOOT

Hands on 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

Welcome to Spring Boot!

Hands on 2-Spring Core – Load SimpleDateFormat from Spring Conflguration 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("31/12/2018");

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

}

}

# output

Inside main method...

Parsed Date: Mon Dec 31 00:00:00 IST 2018

Hands on 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 |  |  |

Hands on 4-Spring Core – Load Country from Spring Conflguration 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 |  |  |

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("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 + "]";

}

}

# 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/bea> ns

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>

# SpringLearnApplication.java

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext; import

org.springframework.context.support.ClassPathXmlApplicationCon text;

public class SpringLearnApplication {

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

public static void main(String[] args) { LOGGER.info("START - main()");

displayCountry();

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

}

public static void displayCountry() { LOGGER.info("START - displayCountry()");

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

Country country = context.getBean("country", Country.class); LOGGER.debug("Country 1 : {}", country.toString());

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

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

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

}

}

# output

## Singleton Scope (one constructor call):

DEBUG

DEBUG

DEBUG

main

main

main

<init>

toString

toString

Inside Country Constructor.

Country [code=IN, name=India]

Country [code=IN, name=India]

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Prototype Scope (two constructor calls):

|  |  |  |  |
| --- | --- | --- | --- |
| DEBUG | main | <init> | Inside Country Constructor. |
| DEBUG | main | <init> | Inside Country Constructor. |
| DEBUG | main | toString | Country [code=IN, name=India] |
| DEBUG | main | toString | Country [code=IN, name=India] |

Hands on 6-Spring Core – Load list of countries from Spring Conflguration XML

## 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/bea> ns

https://[www.springframework.org/schema/beans/spring-](http://www.springframework.org/schema/beans/spring-) beans.xsd">

<!-- Individual country beans -->

<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 country beans -->

<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 org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext; import

org.springframework.context.support.ClassPathXmlApplicationCon text;

import java.util.List;

public class SpringLearnApplication {

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

public static void main(String[] args) { LOGGER.info("START - main()");

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

}

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.toString());

}

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

}

}

output

|  |  |  |  |
| --- | --- | --- | --- |
| DEBUG | main | <init> | Inside Country Constructor. |
| DEBUG | main | <init> | Inside Country Constructor. |
| DEBUG | main | <init> | Inside Country Constructor. |
| DEBUG | main | <init> | Inside Country Constructor. |
| DEBUG | main | toString | Country [code=IN, name=India] |
| DEBUG | main | toString | Country [code=US, name=United States] |
| DEBUG | main | toString | Country [code=DE, name=Germa ny] |
| DEBUG | main | toString | Country [code=JP, name=Japan] |