**Spring hello world example**

Create a new Java file **Spring3HelloWorld.java** under the package **org.arpit.javapostsforlearning** and add the following code:

package org.arpit.javapostsforlearning;

public class Spring3HelloWorld {

    String name;

    public void printHello()

    {

        System.out.println("Hello World from "+name);

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

}

now create a Xml file called Spring3HelloWorld in src folder.Add following content to it

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

<beans xmlns="http://www.springframework.org/schema/beans"

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

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="Spring3HelloWorldBean"

class="org.arpit.javapostsforlearning.Spring3HelloWorld">

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

</bean>

</beans>

Here tag is for defining multiples beans.All beans of our program are defined in tag  
tag is for defining a single bean.

“id” is for providing unique identification to bean.  
“class” is fully qualified name of class  
“property” is used for defining attribute of bean class.

The above xml file declares the spring bean “Spring3HelloWorldBean” of the class **org.arpit.javapostsforlearningSpring3HelloWorld.**

Create another java file named “Spring3HelloWorldMain.java” under package **org.arpit.javapostsforlearning.**Copy following content in “Spring3HelloWorldMain.java” file

package org.arpit.javapostsforlearning;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.core.io.ClassPathResource;

public class Spring3HelloWorldMain {

       public static void main(String[] args) {

              ApplicationContext beanFactory = new ClassPathXmlApplicationContext("spring3HelloWorld.xml");

              Spring3HelloWorld myBean = (Spring3HelloWorld) beanFactory.getBean("Spring3HelloWorldBean");

              myBean.printHello();

    }

}

In the above code, we have created the instance of **ApplicationContext** and the retrieved the “Spring3HelloWorldBean”. Then we called the printHello() method on the bean.

Spring provides @configuaration and @Bean annotation for java based configuration.

### Create Bean class

Create a bean class called country.java in package **org.arpit.java2blog.model**.

package org.arpit.java2blog.model;

public class Country {

    String countryName;

    public Country(String countryName) {

  this.countryName=countryName;

}

public String getCountryName() {

        return countryName;

    }

    public void setCountryName(String countryName) {

        this.countryName = countryName;

    }

}

### Create application configuration class

This class will have @Configuaration and @Bean annotation .

Create class called ApplicationConfiguration.java in package **org.arpit.java2blog.config**

package org.arpit.java2blog.config;

import org.arpit.java2blog.model.Country;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

@Configuration

public class ApplicationConfiguration {

@Bean(name="countryObj")

public Country getCountry()

{

  return new Country("India");

}

}

Above file is equivalent to below spring configuration 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"

    xmlns:context="http://www.springframework.org/schema/context"

    xsi:schemaLocation="http://www.springframework.org/schema/beans

    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

    http://www.springframework.org/schema/context

    http://www.springframework.org/schema/context/spring-context-3.0.xsd"<

<context:annotation-config/<

<bean id="countryObj" class="org.arpit.java2blog.Country" <

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

</bean<

</beans<

### Create main class to run the program

Create class called SpringJavaConfigMain.java

package org.arpit.java2blog.main;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.arpit.java2blog.config.ApplicationConfiguration;

import org.arpit.java2blog.model.Country;;

public class SpringJavaConfigMain {

public static void main(String[] args) {

  @SuppressWarnings("resource")

  ApplicationContext appContext = new AnnotationConfigApplicationContext(ApplicationConfiguration.class);

  Country countryObj = (Country) appContext.getBean("countryObj");

  String countryName=countryObj.getCountryName();

  System.out.println("Country name: "+ countryName);

}

}

# Dependency Injection via Setter method in spring

As the name implies, using setter method spring container will inject the dependencies.This technique is considered as the best approach for dependency injection.

**Country.java:**

This is simple pojo class having some attributes so here country has name and object of Capital class.

Create Country.java under package **org.arpit.javapostsforlearning**.Copy following content into Country.java.

|  |
| --- |
| package org.arpit.javapostsforlearning;    public class Country {        String countryName;      Capital capital;      public String getCountryName() {          return countryName;      }      public void setCountryName(String countryName) {          this.countryName = countryName;      }      public Capital getCapital() {          return capital;      }      public void setCapital(Capital capital) {          this.capital = capital;      }  } |

**2.Capital.java**

This is also simple pojo class having one attribute called “capitalName”.

Create Capital.java under package **org.arpit.javapostsforlearning**.java.Above Country class contains object of this class.Copy following content into Capital.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | package org.arpit.javapostsforlearning;    public class Capital {        String capitalName;        public String getCapitalName() {          return capitalName;      }        public void setCapitalName(String capitalName) {          this.capitalName = capitalName;      }  } |

**3.ApplicationContext.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:aop="http://www.springframework.org/schema/aop"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">      <bean id="CountryBean" class="org.arpit.javapostsforlearning.Country">        <property name="countryName" value="India"/>        <property name="capital" ref="CapitalBean"/>    </bean>    <bean id="CapitalBean" class="org.arpit.javapostsforlearning.Capital">        <property name="capitalName" value="Delhi"/>    </bean>  </beans> |

**SetterMehtodMain.java**

This class contains main function.Create SetterMethodMain.java under package **org.arpit.javapostsforlearning**.Copy following content into SetterMethodMain.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | package org.arpit.javapostsforlearning;    import org.springframework.beans.factory.xml.XmlBeanFactory;  import org.springframework.context.ApplicationContext;  import org.springframework.context.support.ClassPathXmlApplicationContext;  import org.springframework.core.io.ClassPathResource;    public class SetterInjectionMain {        public static void main(String[] args) {          ApplicationContext appContext = new ClassPathXmlApplicationContext("ApplicationContext.xml");            Country countryObj = (Country) appContext.getBean("CountryBean");          String countryName=countryObj.getCountryName();          String capitalName=countryObj.getCapital().getCapitalName();          System.out.println(capitalName+" is capital of "+countryName);        }  } |

You can note here that we have used ClassPathXmlApplicationContext for getting bean here.There are various ways for getting beans.In [**hello world example**](https://www.java2blog.com/2012/08/spring-hello-world-example-in-eclipse.html) we have used XmlBeanFactory for getting beans.

## Dependency Injection via Constructor

**Country.java:**

We will create a simple class which has countryName and capital as attributes.

Create Country.java under package **org.arpit.java2blog**.Copy following content into Country.java.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | package org.arpit.java2blog;    public class Country {        String countryName;      Capital capital;        public Country(String countryName, Capital capital) {          super();          this.countryName = countryName;          this.capital = capital;      }      public String getCountryName() {          return countryName;      }        public Capital getCapital() {          return capital;      }    } |

**Capital.java**

Create Capital.java under package **org.arpit.java2blog**.Above Country class contains object of this class.Copy following content into Capital.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | package org.arpit.java2blog;    public class Capital {        String capitalName;        public String getCapitalName() {          return capitalName;      }        public void setCapitalName(String capitalName) {          this.capitalName = capitalName;      }  } |

**3.ApplicationContext.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:aop="http://www.springframework.org/schema/aop"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">    <bean id="CountryBean" class="org.arpit.java2blog.Country">    <constructor-arg index="0" type="java.lang.String" value="India" />    <constructor-arg index="1" ref="CaptialBean" />  </bean>  <bean id="CaptialBean" class="org.arpit.java2blog.Capital">    <property name="capitalName" value="Delhi" />  </bean>  </beans> |

Here We have declared two beans with corresponding ids.  
1.Class Country with id as “CountryBean”  
2.Class Capital with id as “CapitalBean”  
constructor-arg tag is used for providing argument to bean’ s constructor.type is for declaring data types and index defines position in constructor’s argument.  
In above xml,Two arguments are passed.  
1. India as string  
2.CapitalBean ‘s reference  
Property’s **value**tagis for assigning value to corresponding attribute. so In above xml file,we have assigned capitalName attribute of Capital class with value as Delhi

|  |  |
| --- | --- |
| 1  2  3 | <property name="Name Of Attribute" value="Value Of attribute to be assigned"/> |

Property’s **ref** tagis used for assigning reference to the corresponding attribute. so In above xml file, we have assigned reference of Capital class to capital attribute of Country class.

|  |  |
| --- | --- |
| 1  2  3 | <property name="Name Of Attribute" value="id of referencing bean"/> |

**4.ConstructorDIMain.java**

This class contains main function.Create ConstructorDIMain.java under package **org.arpit.java2blog**.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | package org.arpit.java2blog;    import org.springframework.beans.factory.xml.XmlBeanFactory;  import org.springframework.context.ApplicationContext;  import org.springframework.context.support.ClassPathXmlApplicationContext;  import org.springframework.core.io.ClassPathResource;    public class ConstructorDIMain{        public static void main(String[] args) {          ApplicationContext appContext = new ClassPathXmlApplicationContext("ApplicationContext.xml");            Country countryObj = (Country) appContext.getBean("CountryBean");          String countryName=countryObj.getCountryName();          String capitalName=countryObj.getCapital().getCapitalName();          System.out.println(capitalName+" is capital of "+countryName);        }  } |

You can note here that we have used ClassPathXmlApplicationContext for getting bean here.There are various ways for getting beans.In [**hello world example**](https://www.java2blog.com/2012/08/spring-hello-world-example-in-eclipse.html) we have used XmlBeanFactory for getting beans.