Spring frame work

<https://www.tutorialspoint.com/spring/spring_hello_world_example.htm>

basic hello world program based on the configuration done in Spring Beans Configuration file.

Step1. Create java project

Step 2 - Add Required Libraries

**Build Path → Configure Build Path->add external jarfiles**

Step 3 – in src, create a new package and and to files Hello worls and main class

package com.tutorialspoint;

public class HelloWorld {

private String message;

public void setMessage(String message){

this.message = message;

}

public void getMessage(){

System.out.println("Your Message : " + message);

}

package com.tutorialspoint;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

1. ApplicationContext context = new ClassPathXmlApplicationContext("Beans.xml");

2.HelloWorld obj = (HelloWorld) context.getBean("helloWorld");

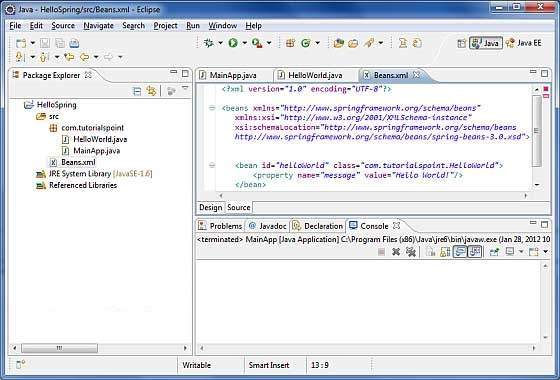
obj.getMessage();

}

}

* The first step is to create an application context where we used framework API **ClassPathXmlApplicationContext()**. This API loads beans configuration file and eventually based on the provided API, it takes care of creating and initializing all the objects, i.e. beans mentioned in the configuration file.
* The second step is used to get the required bean using **getBean()** method of the created context. This method uses bean ID to return a generic object, which finally can be casted to the actual object. Once you have an object, you can use this object to call any class method.

Step 4- Create Bean Configuration File (Xml) in src



In bean.xml we need to provide mappings for all classes

Eg

<beans>

<bean id =”name to be used get bean class” class=”actual class name along with package”>

<property name=”message(attributr name ,must have valid setter to map name and value)” value=” value to be passesd”></property>

</bean>

</beans>

The Beans.xml is used to assign unique IDs to different beans and to control the creation of objects with different values without impacting any of the Spring source files. For example, using the following file you can pass any value for "message" variable and you can print different values of message without impacting HelloWorld.java and MainApp.java files. Let us see how it works –

When Spring application gets loaded into the memory, Framework makes use of the above configuration file to create all the beans defined and assigns them a unique ID as defined in **<bean>** tag. You can use **<property>** tag to pass the values of different variables used at the time of object creation.

Step 5 -run