**The Complete Spring Tutorial**

In this tutorial I will show you how you can integrate struts, spring and hibernate in your web application.Spring framework is developed to simplify the developed of enterprise applications in Java technologies. It is an open source framework begin developed by Spring source company. Spring framework is also available for .NET framework (Spring .NET).

The Spring is light weight, non-invasive IoC Container and AOP framework. It provides support for JPA, Hibernate, Web services, Schedulers, Ajax, Struts, JSF and many other frameworks. The Spring MVC components can be used to developed MVC based web applications. Spring framework provides many features that makes the development of enterprise application easy work.In this Spring tutorial series we will learn Spring framework with the help of many example codes.

In this Spring 3.0 tutorial you will learn Spring 3.0 with the help of example code. The Spring 3.0 tutorial explains you different modules of the Spring 3.0 framework with the help of easy to learn examples. You can also download the example code in the zip format. Then run the code in  Eclipse IDE or any other IDE of your choice.The Spring 3.0 framework has been released with major enhancements and support for the Java 5. It also includes the the support for REST Web services and the expression language. We will learn all the features of the Spring 3.0 with example code.With the help of Spring Framework you can simplify the development of Java based enterprise applications. The Spring Framework provides the support for Hibernate, JPA, EJB 3, JSF, Struts 2 etc. It makes the development of Enterprise application must easier.

In this section we will understand the features of Spring 3.0 Framework. The Spring 3.0 Framework is released with the support of Java 5. So, you can use all the latest features of Java 5 with Spring 3 framework.The first version of Spring Framework was released in the year 2002. And it was developed and released to simplify the development of Enterprise Java applications. Since then it have undergone the major releases Spring 2.0,  Spring 2.5 and now Spring 3.0. The Spring Framework is used by many frameworks. You can use the existing frameworks such as JSF, Struts 2, Flex with Spring DS to quickly develop and deploy large scale enterprise applications. The Spring framework simplifies the development of complex enterprise applications.The Spring Framework simplifies the development, testing, deployment of the application much easier.

**Spring 3 what's new**

Let's see what all new features are available in the Spring 3 framework.

**Features of Spring 3.0 Framework:**

* **Java 5 Support:**The core API of Spring 3.0 framework is using JDK 5, so JDK 5 or above is required to run Spring 3.0 based applications. The Spring 3.0 framework provides annotation-based configuration support. JDK 5 features such as generics, annotations and varargs can used in Spring 3.0 Framework based applications. The Spring 3.0 framework is based on JDK 1.5 and it fully supports JDK 6. The Spring 3.0 is fully compatible with the JEE1.4 and EE5.
* **Expression language - Spring Expression Language (SpEL):** In this release of Spring framework Expression language support is present. The Spring Expression Language  can be used while defining the XML and Annotation based bean definition.
* **Support for Comprehensive REST Webservices:**Spring 3.0 framework supports REST web services.
* **Java EE 6 Support:**Spring 3.0 supports many features of JEE 6 such as JAP 2.0 and JSF 2.0.
* **Annotation based formatting support:**Spring comes with Annotation based formatting support. For example bean fields can be automatically formatted and converted using different annotations. For example you can use the  @DateFimeFormat(iso=ISO.DATE) and @NumberFormat(style=Style.CURRENCY) annotations to convert the date  and currency formats.
* **New module organization and build system:   
  The modules in the Spring 3.0 framework have been revised and it managed separately with one source-tree per module jar:  
    
  \* org.springframework.aop  
  \* org.springframework.beans  
  \* org.springframework.context  
  \* org.springframework.context.support  
  \* org.springframework.expression  
  \* org.springframework.instrument  
  \* org.springframework.jdbc  
  \* org.springframework.jms  
  \* org.springframework.orm  
  \* org.springframework.oxm  
  \* org.springframework.test  
  \* org.springframework.transaction  
  \* org.springframework.web  
  \* org.springframework.web.portlet  
  \* org.springframework.web.servlet  
  \* org.springframework.web.struts**

In the next section we will download, create a new project in Eclipse IDE, install the library files and then develop a "Hello World" example program.

**Spring 3 Hello World Example**

In this section we will download spring, create new project in Eclipse IDE and then write simple Hello World application. We will finally run the application in the Eclipse IDE.

In this section you will learn how to download, create a new project in Eclipse IDE, add Spring 3.0 library files, the write simple program. Finally we will test the example code in Eclipse IDE.

**Let's start developing "Hello World" example code**

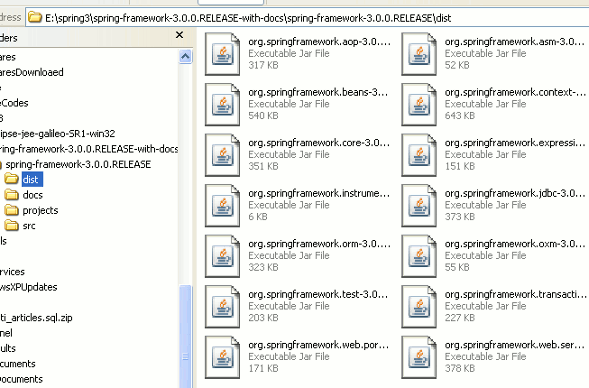
**Step 1:**

The Spring 3.0 at least requires JDK 5. So, make sure you have JDK 5 or above. Open dos prompt if you are using windows and type java -version. This will display the version of Java installed on your machine as shown below:

|  |
| --- |
| C:\>java -version java version "1.6.0\_17" Java(TM) SE Runtime Environment (build 1.6.0\_17-b04) Java HotSpot(TM) Client VM (build 14.3-b01, mixed mode, sharing)  C:\>  **Step 2:**  Download Eclipse IDE from Eclipse download site  at [**http://www.eclipse.org/downloads/**](http://www.eclipse.org/downloads/). Installing Eclipse IDE is easy task, just extract the downloaded file and you will find the eclipse.exe in the extracted folder. To run the IDE, double click on the eclipse.exe file. |

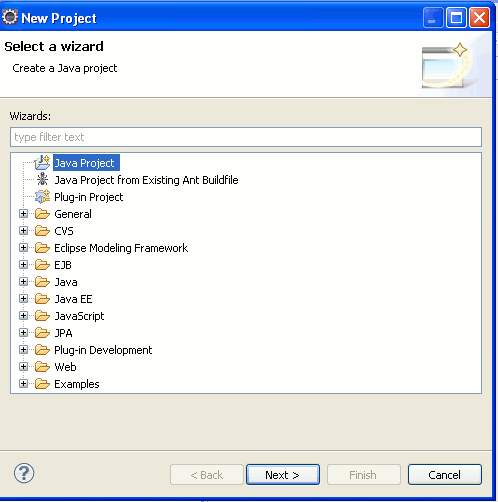
**Step 3:**

Download the latest version of Spring 3 from [**http://www.springsource.org/download**](http://www.springsource.org/download). For this tutorial we have downloaded **spring-framework-3.0.0.RELEASE-with-docs.zip,**which contains the documentation also. After extracting the file we got the following directories:

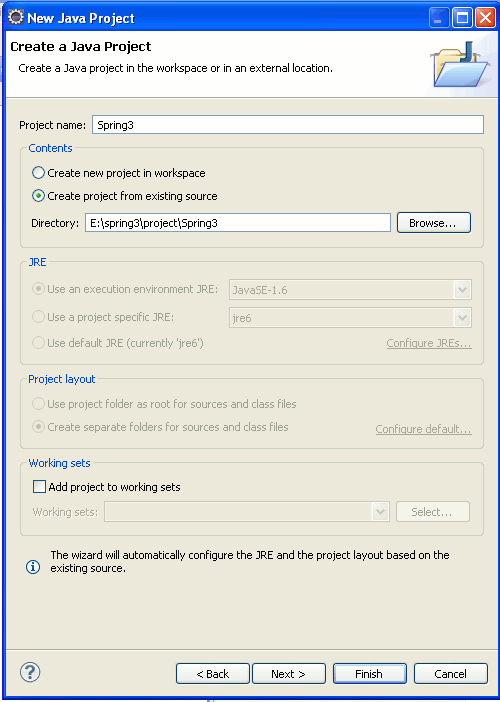


**Step 4:**

Now we will create a new project in Eclipse IDE and then add the library files.



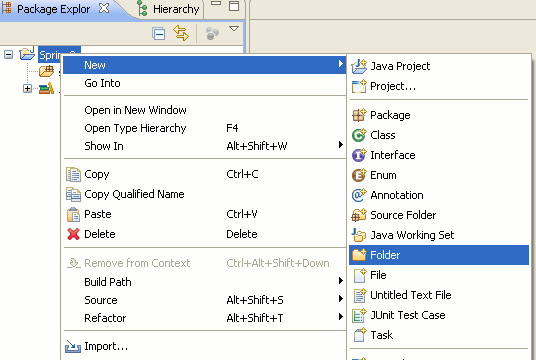
Click next. Give any name to the project, say "Spring 3" and then click on the "Finish" button.



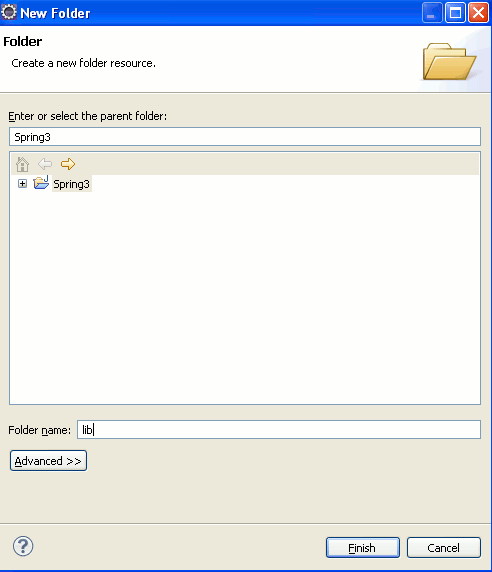
Eclipse will create a new project.

**Step 5:**

Create a new folder "lib" in the project space to hold the Spring 3.0 libraries. Right click on the "Spring3" in the project explorer and then select new folder option as shown below:

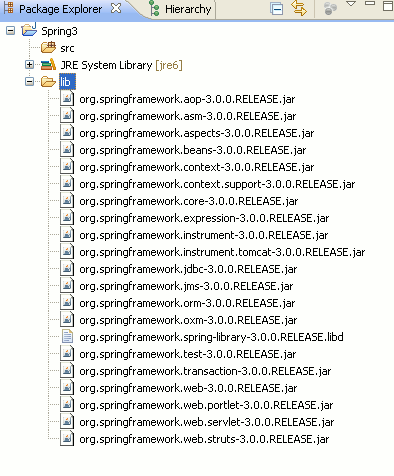


Then in the next screen enter "lib" next to the "Folder name" text field and click on the "Finish" button.



**Step 6:**

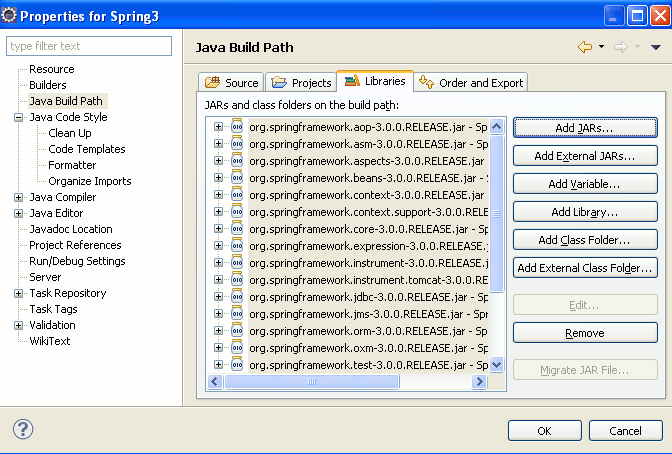
Now we will add the Spring 3. libraries to the project. Extract the "spring-framework-3.0.0.RELEASE-with-docs.zip" file if you have not extracted. Now go to the "dist" directory of the and then copy all the jar files (Ctrl+C) and paste on the **lib** directory (of our project) in the Eclipse IDE.



Then find the commons-logging.jar from extracted folder and also copy this file into Eclipse IDE. You will find this library into spring-framework-3.0.0.RELEASE-with-docs\spring-framework-3.0.0.RELEASE\projects\spring-build\lib\ivy folder.

**Step 7:**

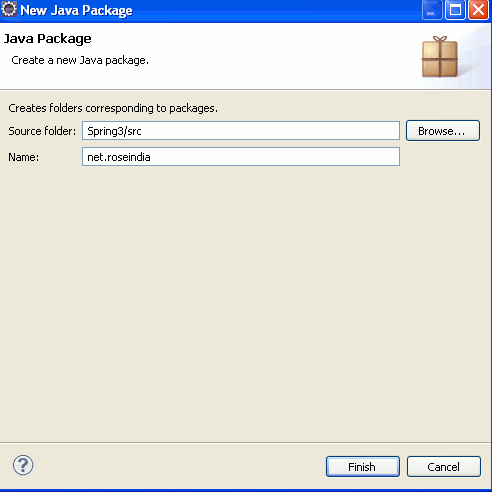
Now add all the libraries to "Java Build Path". Right click on the "Spring3" in project explorer and then select properties. Then select "Java Build Path" --> Libraries and then click on the "Add JARs" button. And add all the libraries to Java Build Path.



Then click on the "OK" button. This will add all the libraries to the project. Now we can proceed with our Spring 3 Hello World example.

**Step 8:**

Create a new package net.roseindia to hold the java files. Right click on the "Spring3" and then select New --> Package. Then provide the package name as net.roseindia and click on the "Finish" button.



**Step 9:**

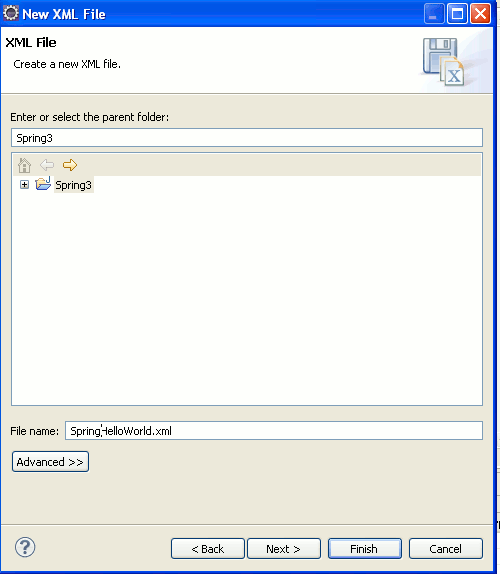
Create a new Java file **Spring3HelloWorld.java** under the package **net.roseindia** and add the following code:

package net.roseindia;  
  
public class Spring3HelloWorld {  
public void sayHello(){  
System.out.println("Hello Spring 3.0");  
}  
}

In the above class we have created a method sayHello() which prints the "Hello Spring 3.0" on the console. In this section we will use the Spring framework to manage the Spring3HelloWorld bean, and then get the bean from the Spring runtime environment (Spring context) and the call the sayHello() method. In the next step we will xml file which will be used as Metadata to configure the bean.

**Step 10:**

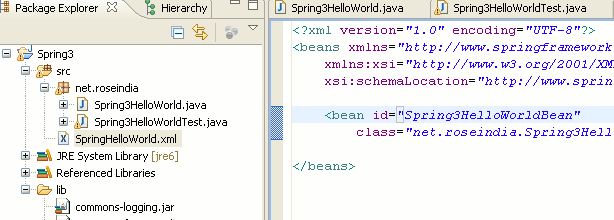
Now create a new xml (SpringHelloWorld.xml) file using Eclipse IDE.



Add the following code to the xml file:

<?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="net.roseindia.Spring3HelloWorld" />  
  
</beans>

The above xml file declares the spring bean "Spring3HelloWorldBean" of the class**net.roseindia.Spring3HelloWorld.**The <bean .../> tag is used to declare a bean in the xml file. Spring uses the xml file to configure the spring run-time environment. Spring framework manages the beans in our program. In the next sections we will learn Spring core components in detail. **Note: You should move the SpringHelloWorld.xml file into src directory of the project. Jut use mouse to drag and dop in the src folder.**



**Step 11:**

Now create a java file (Spring3HelloWorldTest.java) into net.roseindia package and add the following code:

package net.roseindia;  
  
import java.util.Map;  
import org.springframework.beans.factory.xml.XmlBeanFactory;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.util.Assert;  
  
public class Spring3HelloWorldTest {  
  
public static void main(String[] args) {  
  
XmlBeanFactory beanFactory = new XmlBeanFactory(new ClassPathResource(  
"SpringHelloWorld.xml"));  
  
Spring3HelloWorld myBean = (Spring3HelloWorld) beanFactory  
.getBean("Spring3HelloWorldBean");  
myBean.sayHello();  
}  
}

In the above code we have created the instance of **XmlBeanFactory**and the retrieved the "Spring3HelloWorldBean". Then we can call the sayHello() method on the bean. The **XmlBeanFactory** class is extension of DefaultListableBeanFactory that reads bean definitions from an XML document. In our case it reads the bean definitions from SpringHelloWorld.xml file.

**Step 12:**

To run the code in Eclipse open Spring3HelloWorldTest.java in the editor and then right click and select Run as --> Java Application. This execute the Spring3HelloWorldTest.java file and following output will be displayed in the console.

Jan 1, 2010 6:49:57 PM org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions

INFO: Loading XML bean definitions from class path resource [SpringHelloWorld.xml]

Hello Spring 3.0

In this section we developed Hello World example that uses the Spring 3.0. In the next section we will learn about @configuration, and see how this annotation can used in the Spring 3.0 application.

# @configuration annotation in Spring 3

In the last section we developed the .xml file to configure the IOC (Spring Container). In this section we will learn another method for configuring the Spring 3 IOC, which is through the Java configuration annotation @configuration. The @configuration annotation is code centric and can be used in place of xml configuration file.

**Two way to configure Spring IOC container**

1. **xml file:**Developer can define the bean configuration in xml file and then instruct the Spring to use the xml file to configure the beans. The xml file configuration method is targeted to infrastructure providers.
2. **Annotation:** The @configuration spring annotation can also be used to configure the Spring IOC container. Here programmer add the @configuration to the Java class and this class is considered as special configuration class. The @configuration annotation method is targeted the developers.  This is pure-java approach to configure the Spring IoC Container. The @Bean  tag is used to define the bean, and the Spring framework executes the method and then register the object returned. By default the name of the method is used as the bean name.

**Step 1:**

Download the asm-3.2 from [**http://forge.ow2.org/project/showfiles.php?group\_id=23&release\_id=3334**](http://forge.ow2.org/project/showfiles.php?group_id=23&release_id=3334) and the add the jar file into project workspace.

**Step 2:**

Download cglib from [**http://cglib.sourceforge.net/**](http://cglib.sourceforge.net/) and then add the jar file into project workspace. Above jar files are required to work with  @configuration annotation. These are runtime code generation library used by the Spring 3.0 framework.

**Step 3:**

Creating the configuration file: Create a new java file (**Spring3HelloWorldConfig.java**) in net.roseindia package. This file is used to programmatically configure the beans in the application. In our example we will configure **Spring3HelloWorld** bean. Here is the code of **Spring3HelloWorldConfig.java:**

**package** net.roseindia;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

@Configuration

**public** **class** Spring3HelloWorldConfig {

**public** @Bean Spring3HelloWorld spring3HelloWorld() {

**return** **new** Spring3HelloWorld();

}

}

The @configuration spring annotation configures as special class to be used by the Spring IoC to configure beans. The @Bean annotation instruct the IoC container to execute the function**spring3HelloWorld()** function and register the returned class with the name **spring3HelloWorld**. Now we will see how to configure the IoC container and retrieve the **spring3HelloWorld**  bean.

**Step 4:**

Now create a new java class (**Spring3HelloWorldConfigTest.java**) and add the following code:

**package** net.roseindia;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**public** **class** Spring3HelloWorldConfigTest {

**public** **static** **void** main(String[] args) {

//Initialize IoC Container

AnnotationConfigApplicationContext context = **new** AnnotationConfigApplicationContext(

Spring3HelloWorldConfig.**class**);

System.*out*.println("Calling Bean method: sayHello()");

//Retrieve the bean from Container

Spring3HelloWorld myBean = (Spring3HelloWorld) context

.getBean("spring3HelloWorld");

myBean.sayHello();

}

}

Following code initializes the Spring IoC and configures the beans defined in the**Spring3HelloWorldConfig** class.

//Initialize IoC Container

AnnotationConfigApplicationContext context = **new** AnnotationConfigApplicationContext(

Spring3HelloWorldConfig.**class**);

To retrieve the bean from context we can use the following code:

Spring3HelloWorld myBean = (Spring3HelloWorld) context

.getBean("spring3HelloWorld");

**Step 5:**

To run the example open Spring3HelloWorldConfigTest.java in Eclipse editor window, right click and then select Run As -> Java application. This will execute the code and following output will be displayed on the console.

Jan 2, 2010 2:50:22 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh

INFO: Refreshing[**org.springframework.context.annotation.AnnotationConfigApplicationContext@c1b531**](mailto:org.springframework.context.annotation.AnnotationConfigApplicationContext@c1b531):   
startup date [Sat Jan 02 14:50:22 IST 2010]; root of context hierarchy

Jan 2, 2010 2:50:22 PM org.springframework.context.annotation.ConfigurationClassEnhancer enhance

INFO: Successfully enhanced net.roseindia.Spring3HelloWorldConfig; enhanced class name is:   
net.roseindia.Spring3HelloWorldConfig$$EnhancerByCGLIB$$196e06e4

Jan 2, 2010 2:50:22 PM org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons

INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@1888759: defining beans [org.springframework.context.annotation.internalConfigurationAnnotationProcessor,  
org.springframework.context.annotation.internalAutowiredAnnotationProcessor,  
org.springframework.context.annotation.internalRequiredAnnotationProcessor,  
org.springframework.context.annotation.internalCommonAnnotationProcessor,  
spring3HelloWorldConfig,spring3HelloWorld]; root of factory hierarchy

Calling Bean method: sayHello()

Hello Spring 3.0