

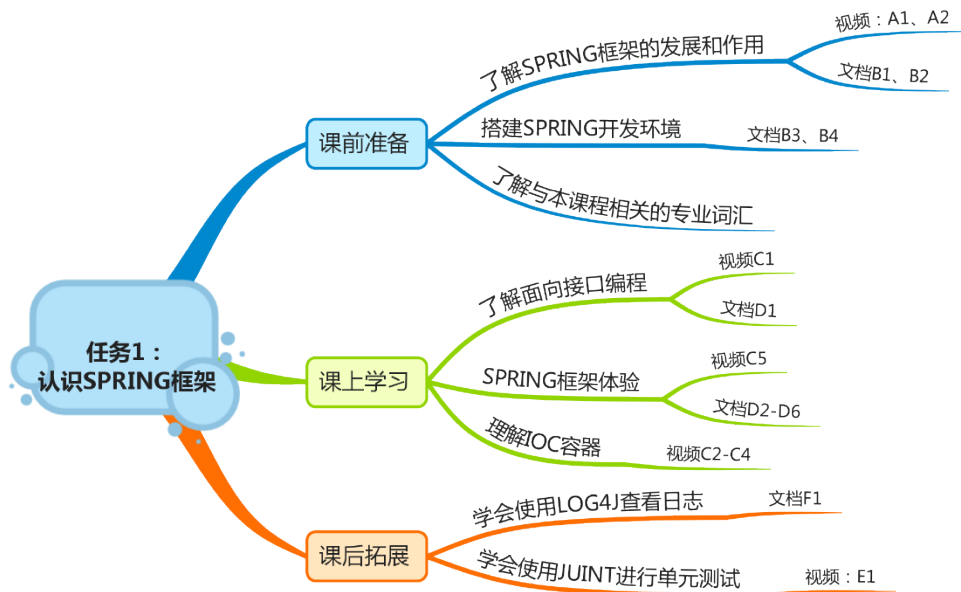
## 单元 3.3.1-搭建 Spring 开发环境

## 课程导入

同学们，在本单元的学习中，我们将会认识一个新的框架——Spring 框架，Spring 是一个轻量级 Java 开发框架，最早有 Rod Johnson 创建，是为了解决企业级应用开发的业务逻辑层和其他各层的耦合问题。它是一个分层的 JavaSE/JavaEE full-stack (一站式) 轻量级开源框架，为开发 Java 应用程序提供全面的基础架构支持。Spring 负责基础架构，因此 Java 开发者可以专注于应用程序的开发。

知识目标	能力目标	素质目标
1.掌握如何构建 Spring 的开发环境。 2.了解 Spring 的体系结构 3.掌握 Spring 入门程序的开发流程以及 Spring 开发环境的构建。 4.理解 spring 的 bean	1.能够熟练利用 IDEA+Maven 构建 spring 的开发环境 2.能够注册一个 bean	1.培养学生的团队意识和团队协作精神，锻炼学生的沟通交流能力; 2.通过项目教学，让学生真切的体验项目分析、设计、管理及实施的全过程;
学习任务	重点难点	突破方法
配置好 spring 开发环境，用 XML 的当时注册 bean 到 IOC 容器，并读取 bean。	1.IOC 容器 2.spring 环境配置 3.注册 bean	采用翻转课堂、项目导入的教学模式，进行分组讨论、演示动画原理。

学  
习  
导  
航



### 搭建 spring 开发环境

#### 第 1 步：安装 Java 开发工具包（JDK）

你可以从 Oracle 的 Java 网站 [Java SE Downloads](#) 下载 JDK 的最新版本。你会在下载的文件中找到教你如何安装 JDK 的说明，按照给出的说明安装和配置 JDK 的设置。最后，设置 PATH 和 JAVAHOME 环境变量，引入包含 java 和 javac 的目录，通常分别为 java install dir/bin 和 java install \_dir。

如果你运行的是 Windows，并在 C:\jdk1.6.0\_15 上安装了 JDK，你就可以把下面这行写入 C:\autoexec.bat 文件中。

```
set PATH=C:\jdk1.6.0_15\bin;%PATH%
set JAVA_HOME=C:\jdk1.6.0_15
```

或者，在 Windows XP/7/8 中，你也可以右键单击“我的电脑”，选择“属性”，然后是“高级”，然后是“环境变量”。接下来，你将更新 PATH 值，并且按下 OK 按钮。

在 Unix(Solaris、Linux 等等)上，如果在 /usr/local/jdk1.6.0\_15 上安装 SDK，并且使用 C shell 命令，你需要把下面的内容添加到 .cshrc 文件中。

```
setenv PATH /usr/local/jdk1.6.0_15/bin:$PATH
setenv JAVA_HOME /usr/local/jdk1.6.0_15
```

或者，如果你使用集成开发环境（IDE），如 Borland JBuilder, Eclipse, IntelliJ IDEA 或者 Sun ONE Studio，编译和运行一个简单的程序，用来确认 IDE 知道你安装了 Java，否则应该根据 IDE 给定的文档做正确的设置。

#### 第 2 步：安装 Apache Commons Logging API

你可以从 <http://commons.apache.org/logging/> 下载 Apache Commons Logging API 的最新版本。一旦你下载完安装包，并且解压二进制的发行版本到一个方便的位置。例如在 windows 上的 C:\commons-

任  
务  
1

logging-1.1.1 中，或在 Linux/Unix 上的 /usr/local/commons-logging-1.1.1 中。该目录将有如下的 jar 文件和其他支持的文件等。

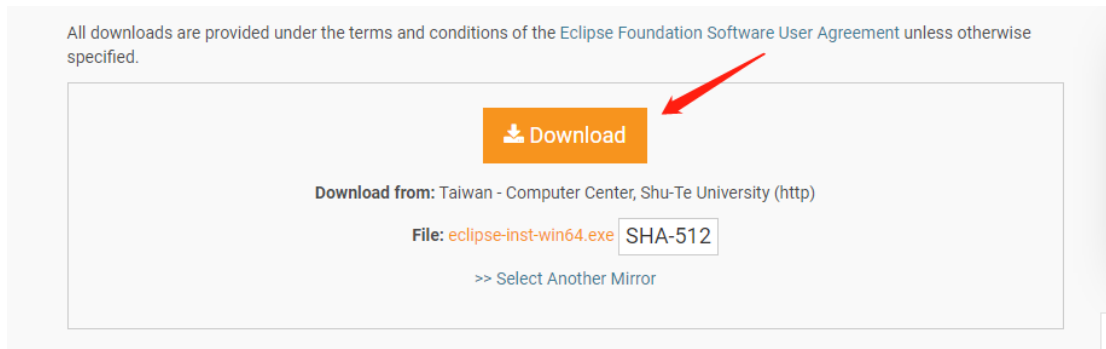
Name	Date modified	Type	Size
site	11/22/2007 12:28 ...	File folder	
commons-logging-1.1.1	11/22/2007 12:28 ...	WinRAR archive	60 KB
commons-logging-1.1.1-javadoc	11/22/2007 12:28 ...	WinRAR archive	139 KB
commons-logging-1.1.1-sources	11/22/2007 12:28 ...	WinRAR archive	74 KB
commons-logging-adapters-1.1.1	11/22/2007 12:28 ...	WinRAR archive	26 KB
commons-logging-api-1.1.1	11/22/2007 12:28 ...	WinRAR archive	52 KB
commons-logging-tests	11/22/2007 12:28 ...	WinRAR archive	109 KB
LICENSE	11/22/2007 12:27 ...	Text Document	12 KB
NOTICE	11/22/2007 12:27 ...	Text Document	1 KB
RELEASE-NOTES	11/22/2007 12:27 ...	Text Document	8 KB

确保你在这个目录上正确的设置 CLASSPATH 变量，否则你将会在运行应用程序时遇到问题。

### 第 3 步：安装 Eclipse IDE

本教程中的所有例子使用 Eclipse IDE 编写。所以我建议你应该在你的机器上安装 Eclipse 的最新版。

为了安装 Eclipse IDE，点开网址 <http://www.eclipse.org/downloads/>，根据操作系统位数（32 位或 64 位）下载相应的版本。

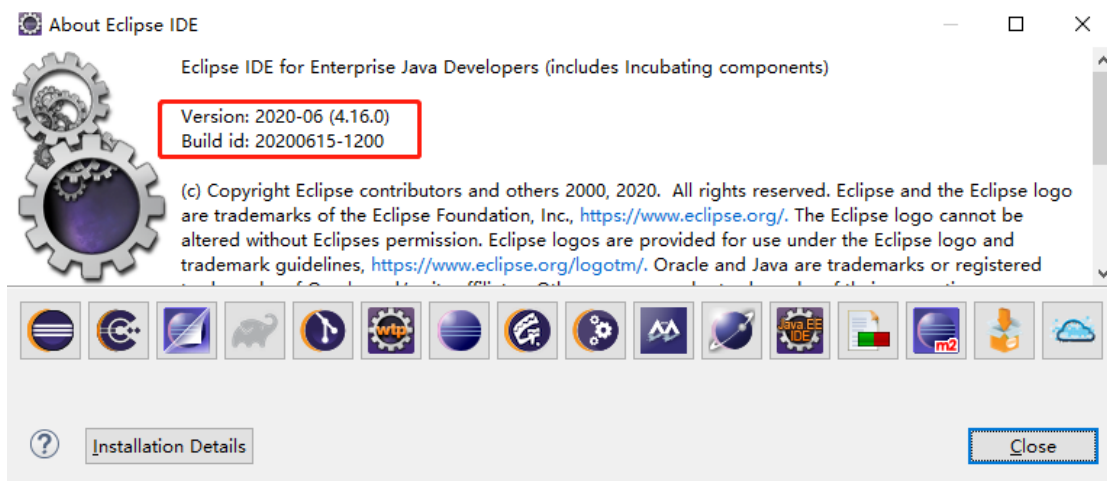


### 第 4 步：安装 Spring 框架库

如果前三个步骤一切正常，你就可以继续设置你的 Spring 框架。下面是在电脑上下载并安装框架的简单步骤。

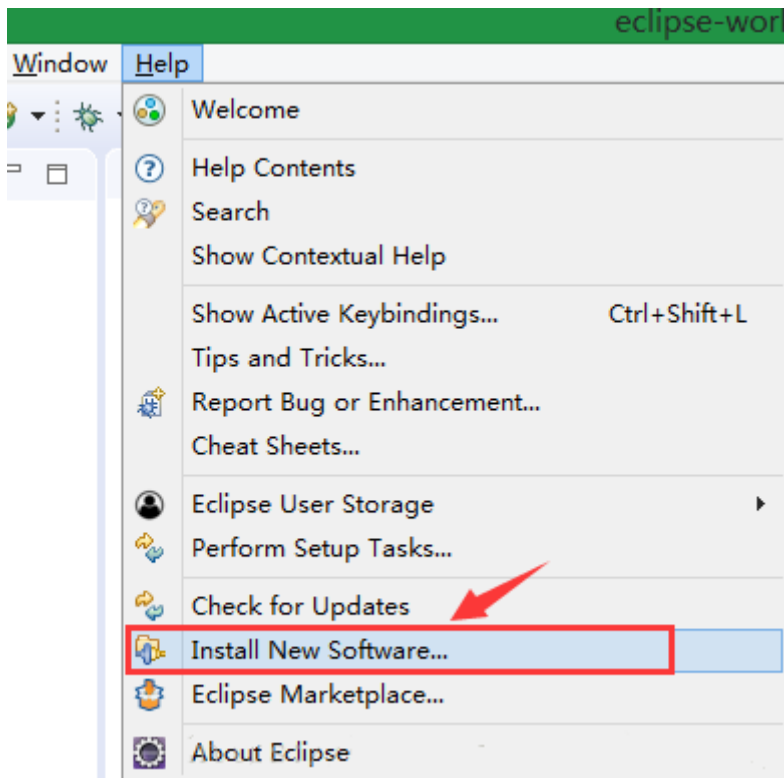
选择是要在 Windows 还是在 UNIX 上安装 Spring，然后继续进行下一个步骤，在 Windows 上下载 .zip 文件，在 Unix 上安装则下载 .tz 文件。

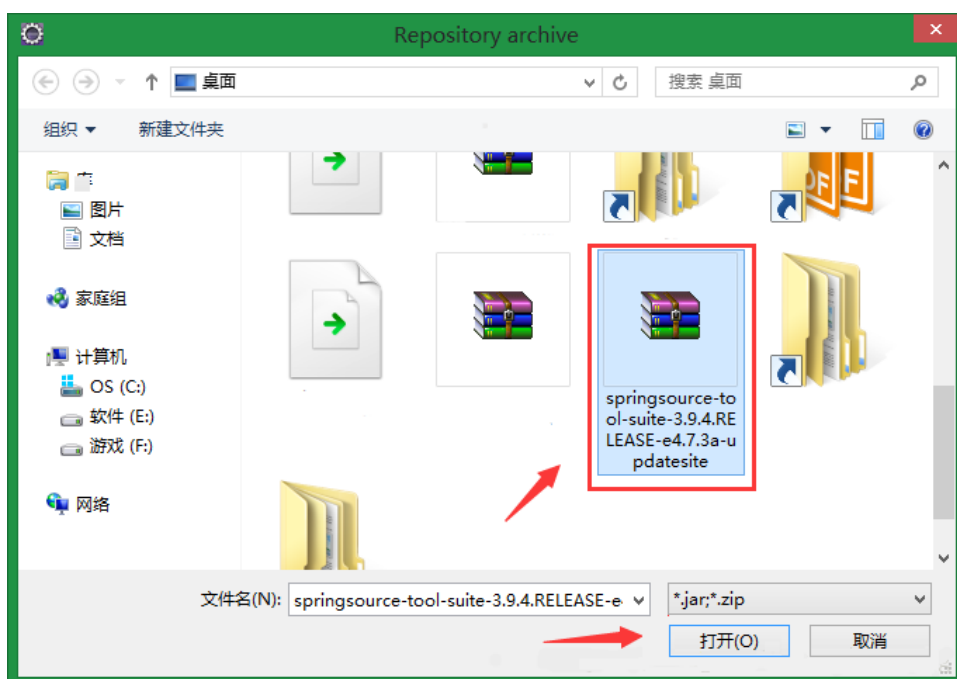
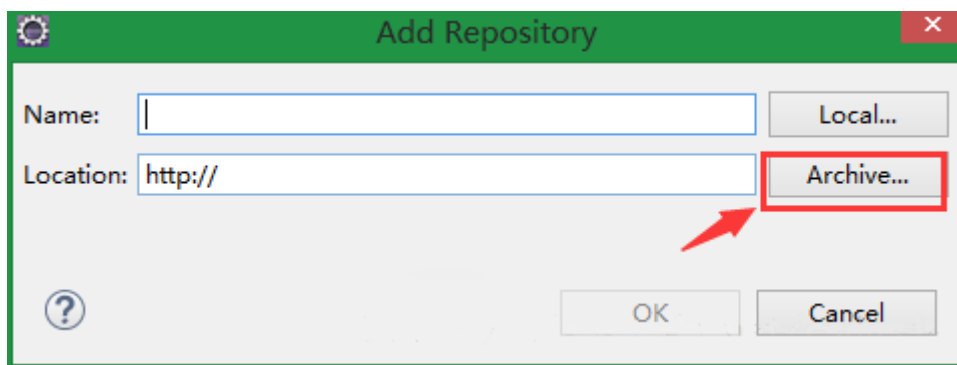
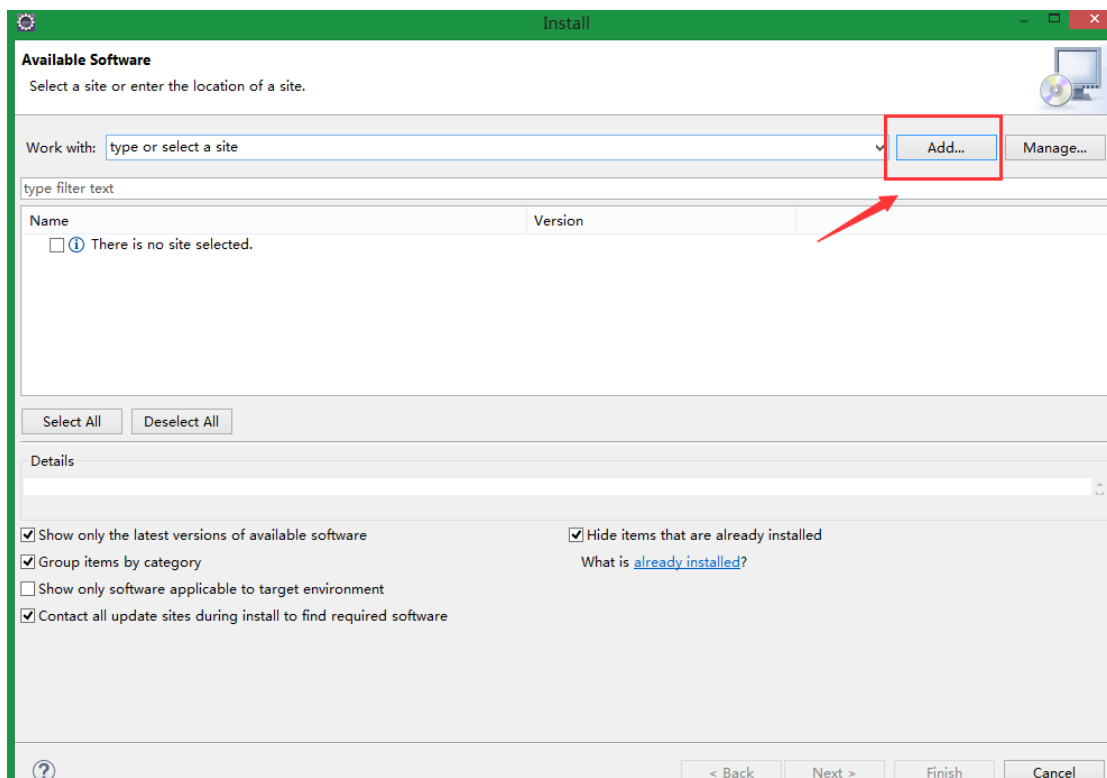
从 Spring 官方下载对应 Eclipse 版本的 springsource-tool-suite。



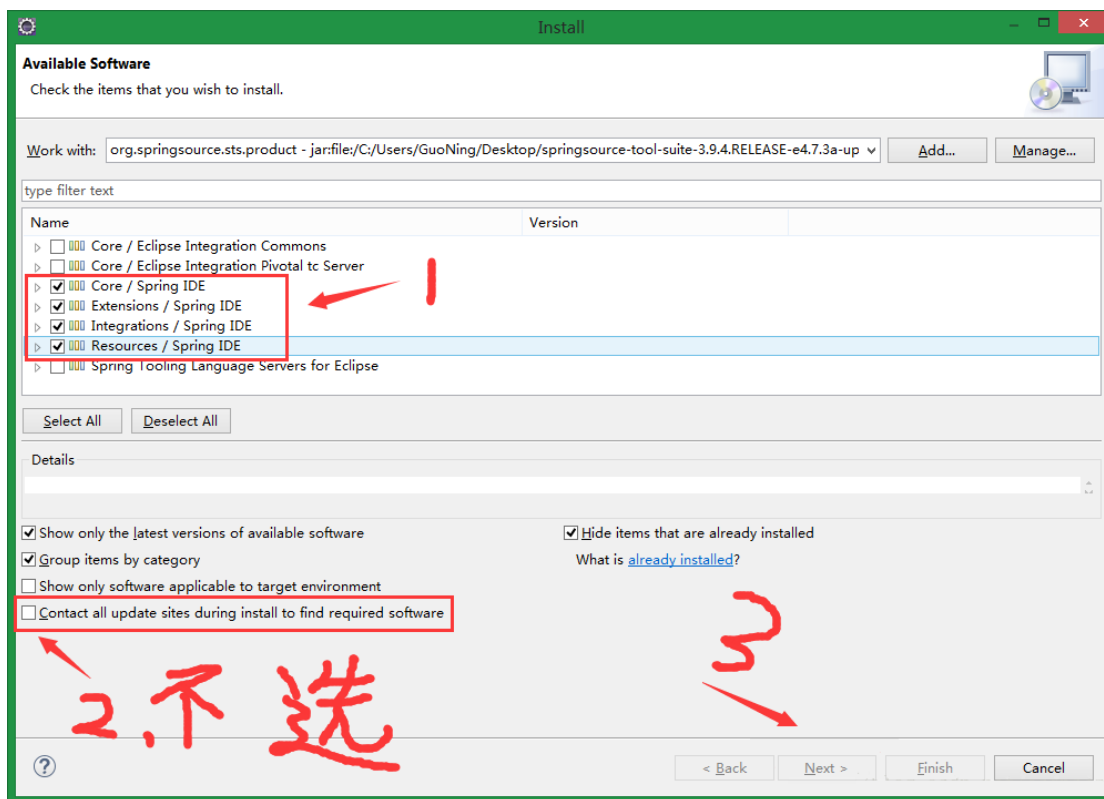
安装 springsource-tool-suite

Eclipse - Help - Install New Software - Add - Archive-- 选择刚才下载好的 springsource-tool-suite 压缩包 --打开 - OK





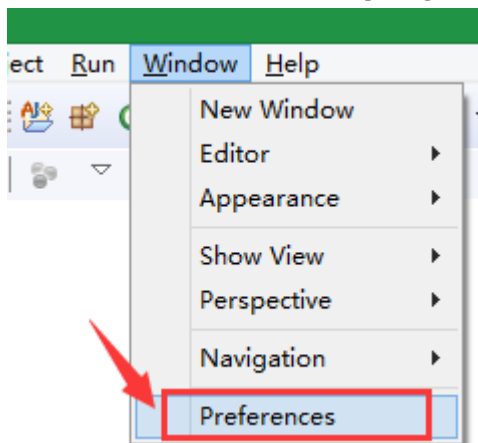
选择要安装的组件：选定 4 个标有“IDE”的组件：



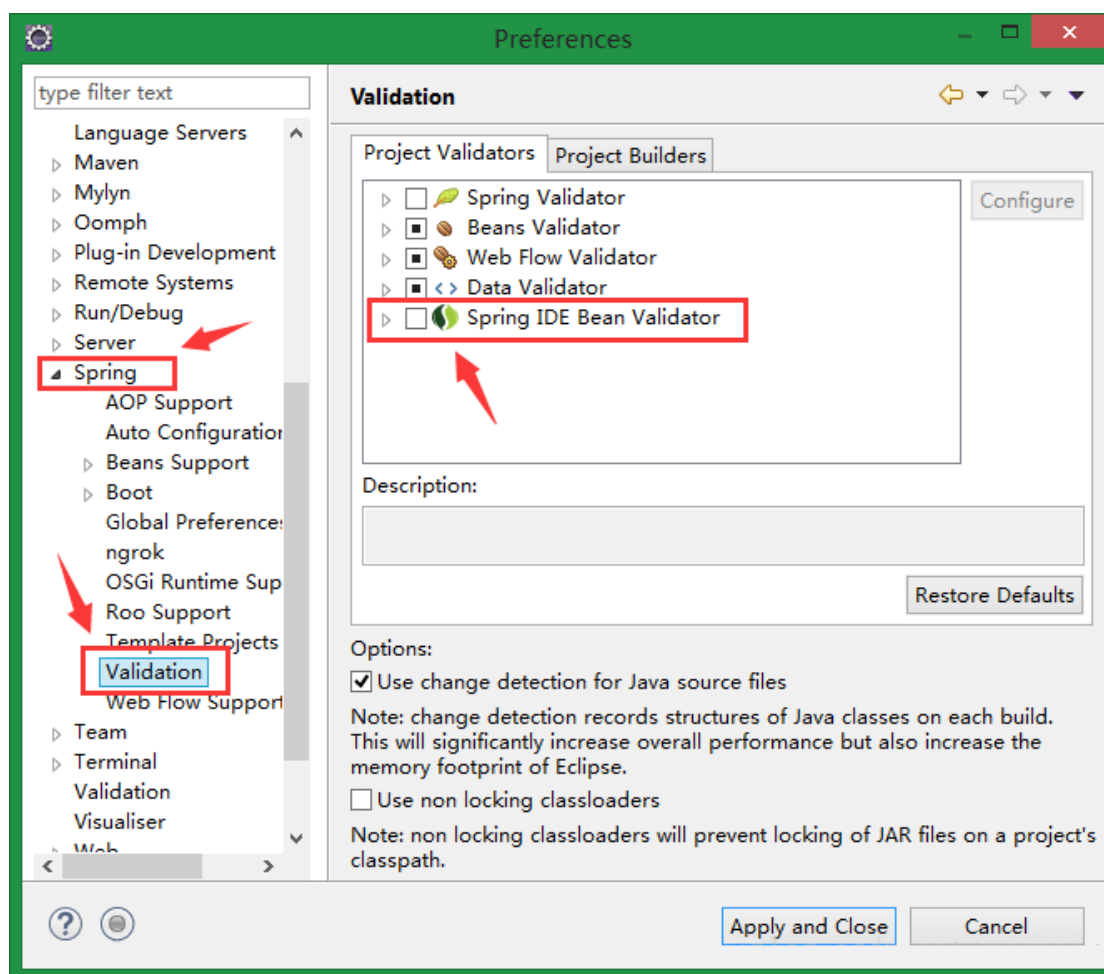
一路点“Next”，最后“Finish”。等待安装。

安装完成后在弹出对话框中点“yes” 重启 eclipse。

顺利的话这时你已经安装好 Spring 插件，点击“Window” --> “Preferences”查看是否安装成功：



看到下图两片绿叶的图标，说明已经安装成功：



任

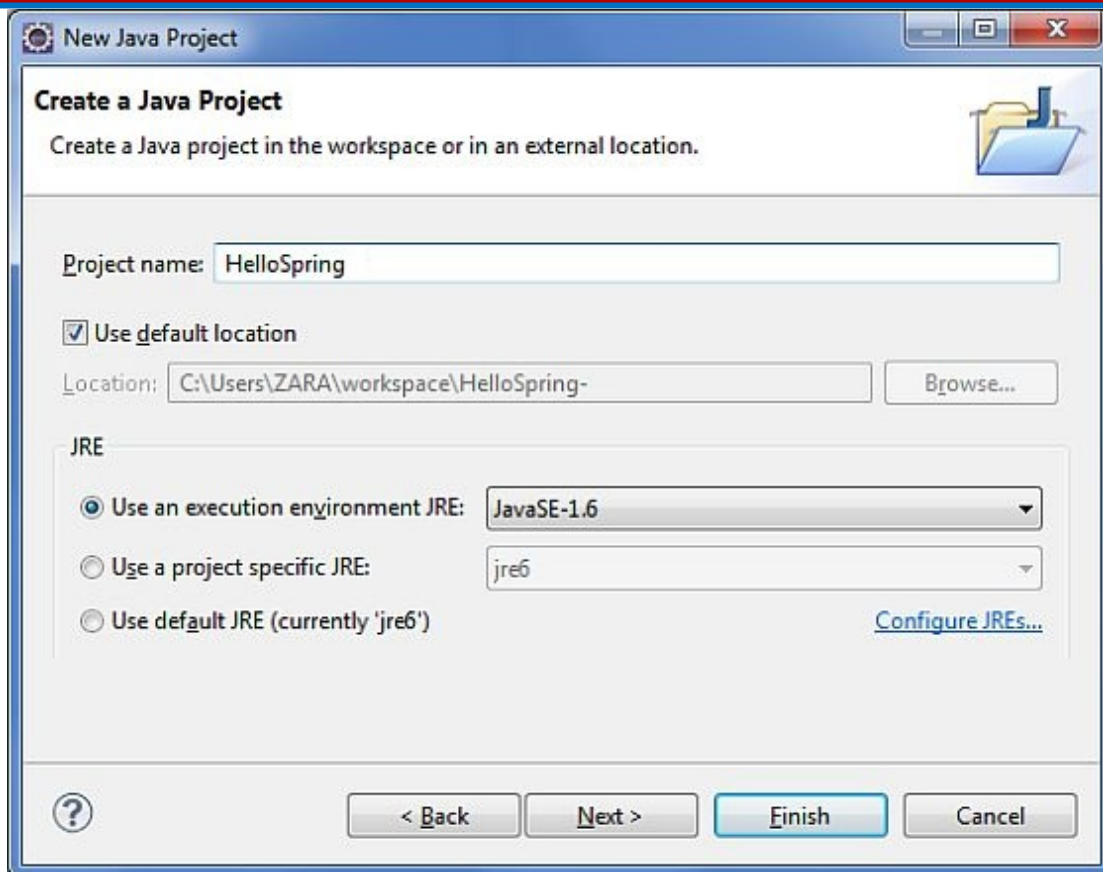
## spring 执行 hello world 实例

务

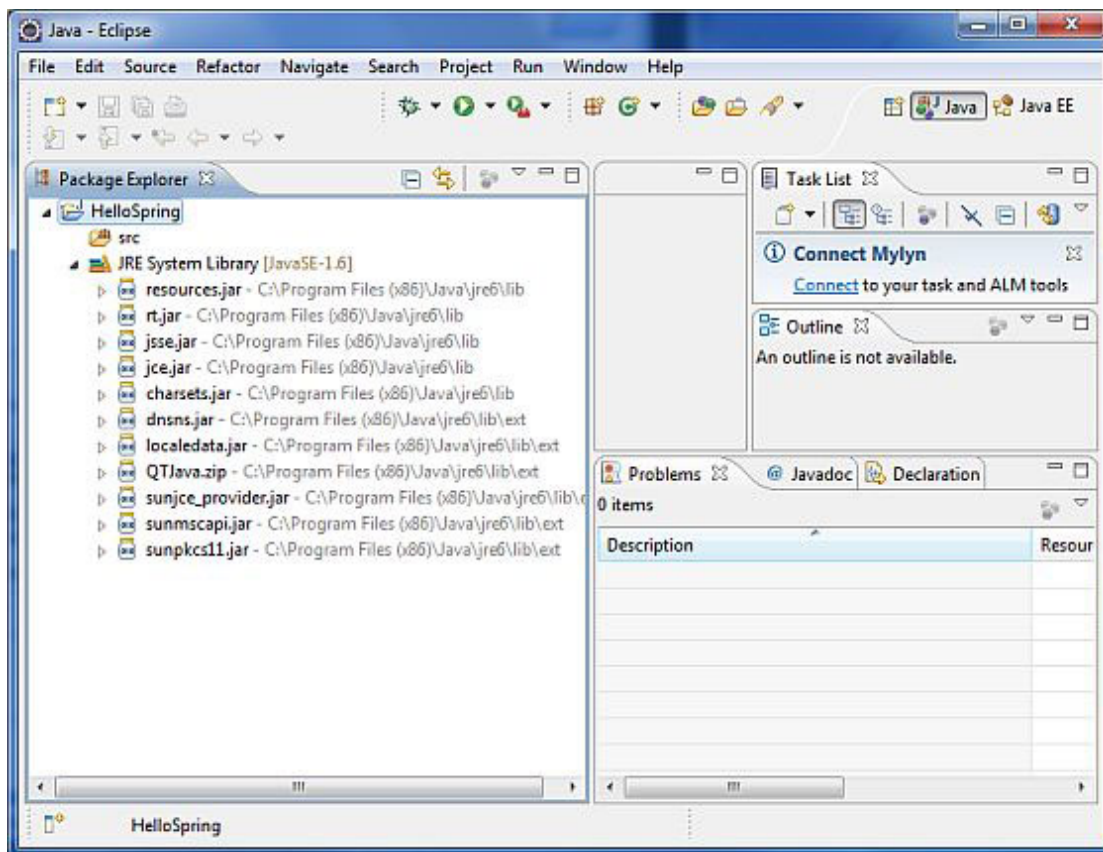
第 1 步：创建 Java 项目

2

第一步是使用 Eclipse IDE 创建一个简单的 Java 项目。按照选项 **File** -> **New** -> **Project**，最后从向导列表中选择 **Java Project** 向导。现在，使用向导窗口将你的项目命名为 **HelloSpring**，如下所示：



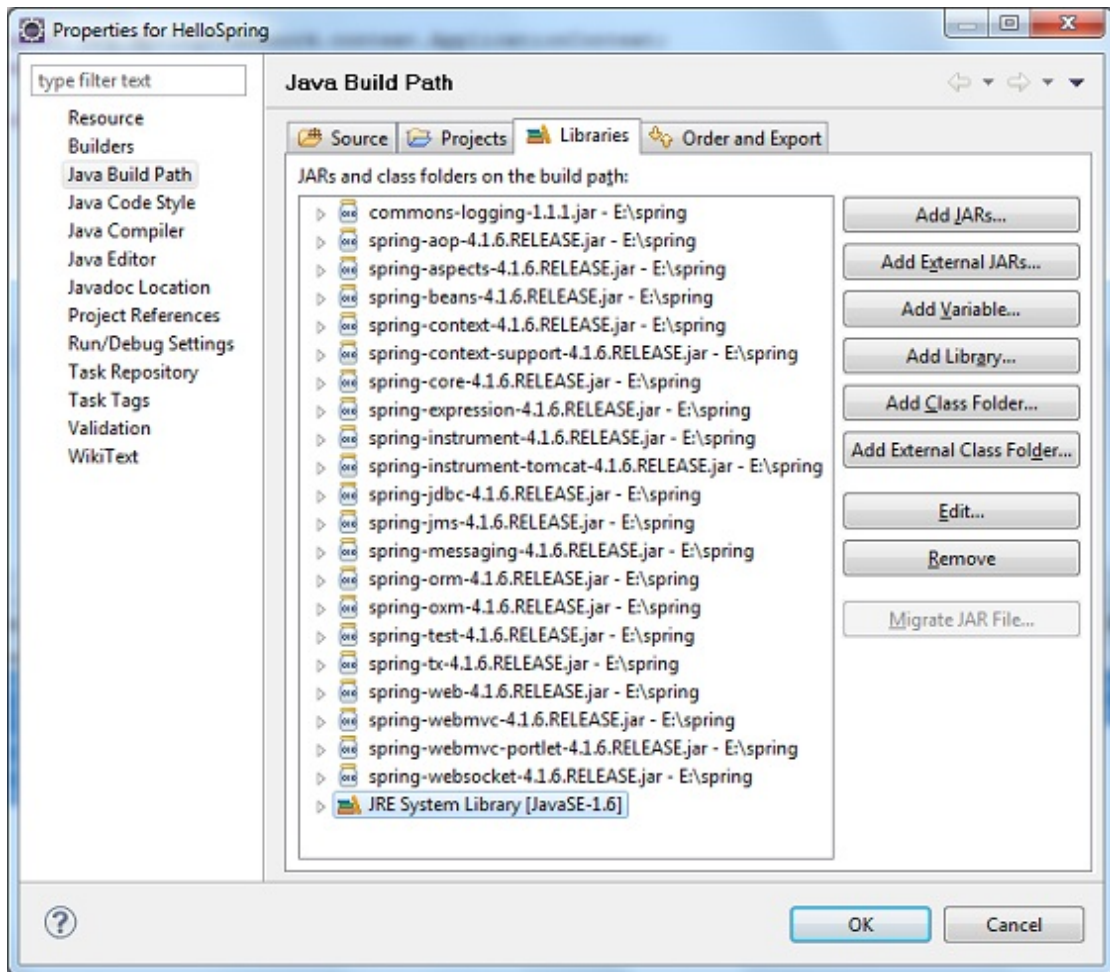
一旦你的项目创建成功后，将在 Project Explorer 看到下面的内容：



第 2 步：添加必需的库



第二步让我们添加 Spring 框架和通用的日志 API 库到我们的项目中。为了做到这个，在你的项目名称 HelloSpring 上单击右键，然后在快捷菜单上按照下面可用的选项：Build Path -> Configure Build Path 显示 Java 构建路径窗口，如下所示：



现在，在 Libraries 标签中使用可用的 Add External JARs 按钮，添加从 Spring 框架和通用日志安装目录下面的核心 JAR 文件：

```
commons-logging-1.1.1
```

```
spring-aop-4.1.6.RELEASE
```

```
spring-aspects-4.1.6.RELEASE
```

```
spring-beans-4.1.6.RELEASE
```

```
spring-context-4.1.6.RELEASE
```

```
spring-context-support-4.1.6.RELEASE
```

```
spring-core-4.1.6.RELEASE
```

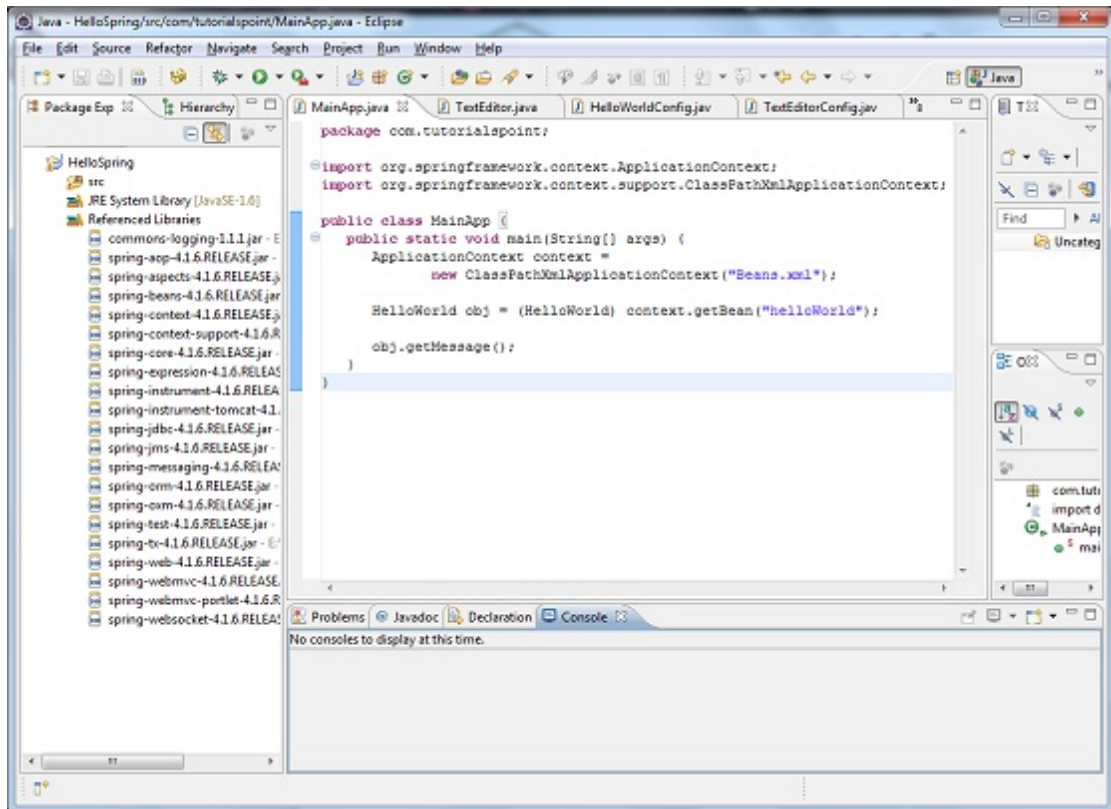
```
spring-expression-4.1.6.RELEASE
```

```
spring-instrument-4.1.6.RELEASE  
spring-instrument-tomcat-4.1.6.RELEASE  
spring-jdbc-4.1.6.RELEASE  
spring-jms-4.1.6.RELEASE  
spring-messaging-4.1.6.RELEASE  
spring-orm-4.1.6.RELEASE  
spring-oxm-4.1.6.RELEASE  
spring-test-4.1.6.RELEASE  
spring-tx-4.1.6.RELEASE  
spring-web-4.1.6.RELEASE  
spring-webmvc-4.1.6.RELEASE  
spring-webmvc-portlet-4.1.6.RELEASE  
spring-websocket-4.1.6.RELEASE
```

### 第 3 步：创建源文件

现在让我们在 **HelloSpring** 项目下创建实际的源文件。首先，我们需要创建一个名为 **com.tutorialspoint** 的包。在 **package explore** 区域中的 **src** 上点击右键，并按照选项：**New -> Package**。

接下来，我们在包 **com.tutorialspoint** 下创建 **HelloWorld.java** 和 **MainApp.java** 文件。



这里是 HelloWorld.java 文件的内容:

```
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);
    }
}
```

下面是第二个文件 MainApp.java 的内容:

```
package com.tutorialspoint;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context =
            new ClassPathXmlApplicationContext("Beans.xml");
        HelloWorld obj = (HelloWorld) context.getBean("helloWorld");
        obj.getMessage();
    }
}
```

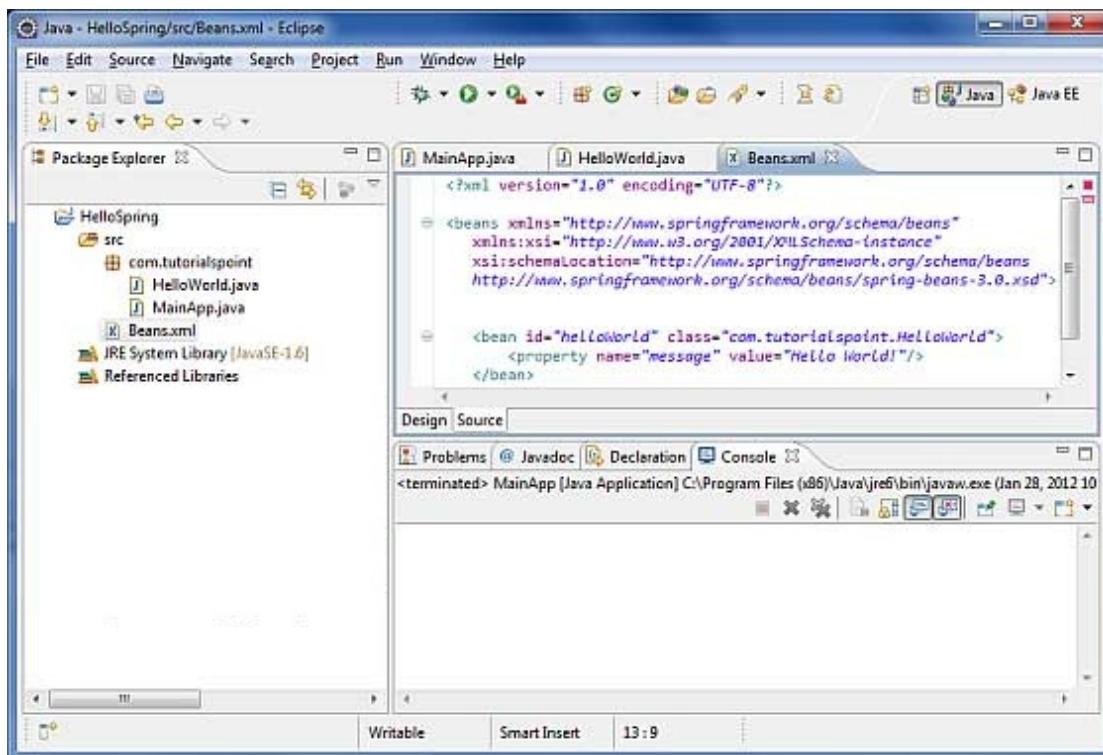
关于主要程序有以下两个要点需要注意：

第一步是我们使用框架 API `ClassPathXmlApplicationContext()` 来创建应用程序的上下文。这个 API 加载 beans 的配置文件并最终基于所提供的 API，它处理创建并初始化所有的对象，即在配置文件中提到的 beans。

第二步是使用已创建的上下文的 `getBean()` 方法来获得所需的 bean。这个方法使用 bean 的 ID 返回一个最终可以转换为实际对象的通用对象。一旦有了对象，你就可以使用这个对象调用任何类的方法。

#### 第 4 步：创建 bean 的配置文件

你需要创建一个 Bean 的配置文件，该文件是一个 XML 文件，并且作为粘合 bean 的粘合剂即类。这个文件需要在 `src` 目录下创建，如下图所示：



通常开发人员保存该文件的名称为 `Beans.xml` 文件，当然你也可以设置成任何你喜欢的名称。但是你必须确保这个文件在 `CLASSPATH` 中是可用的，并在主应用程序中使用相同的名称，而在 `MainApp.java` 文件中创建应用程序的上下文。

`Beans.xml` 用于给不同的 bean 分配唯一的 ID，并且控制不同值的对象的创建，而不会影响 Spring 的任何源文件。例如，使用下面的文件，你可以为“message”变量传递任何值，因此你就可以输出信息的不同值，而不会影响的 `HelloWorld.java` 和 `MainApp.java` 文件。让我们来看看它是如何工作的：

```
<?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-3.0.xsd">

  <bean id="helloWorld" class="com.tutorialspoint.HelloWorld">
    <property name="message" value="Hello World!"/>
  </bean>
</beans>
```

```
</bean>
```

```
</beans>
```

当 Spring 应用程序被加载到内存中时，框架利用了上面的配置文件来创建所有已经定义的 beans，并且按照标签的定义为它们分配一个唯一的 ID。你可以使用标签来传递在创建对象时使用不同变量的值。

### 第 5 步：运行程序

一旦你完成了创建源代码和 bean 的配置文件后，就可以准备编译和运行你的程序了。为了做到这个，请保持 MainApp.java 文件标签是**有效的**，并且在 Eclipse IDE 中使用可用的 Run 选项，或使用 **Ctrl + F11** 编译并运行你的应用程序 **MainApp**。如果你的应用程序一切都正常，将在 Eclipse IDE 控制台打印以下信息：

```
Your Message : Hello World!
```

恭喜你，你已经成功地创建了你的第一个 Spring 应用程序。通过更改 “message” 属性的值并且保持两个源文件不变，你可以看到上述 Spring 应用程序的灵活性。下一步，我们开始在接下来的几个章节中做一些更有趣的事情。