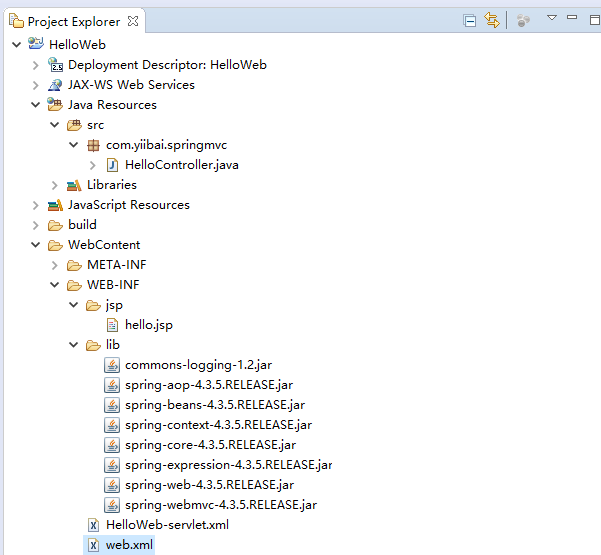
以下示例演示如何使用**Spring MVC**框架编写一个简单的基于**Web**的Hello World应用程序。首先使用Eclipse IDE，并按照以下步骤使用**Spring Web Framework**开发一个动态Web应用程序：

1. 创建一个名为HelloWeb的动态**Web**项目，并在创建的项目中的src文件夹下创建一个包com.yiibai.springmvc。
2. 将下面提到的**Spring**和其他库拖放到文件夹WebContent/WEB-INF/lib中。
3. 在com.yiibai.springmvc包下创建一个Java类HelloController。
4. 在WebContent/WEB-INF文件夹下创建**Spring**配置文件web.xml和HelloWeb-servlet.xml。
5. 在WebContent/WEB-INF文件夹下创建一个名为jsp的子文件夹。在此子文件夹下创建视图文件hello.jsp。
6. 最后一步是创建所有源和配置文件的内容并导出应用程序或直接在**Eclipse**中运行，如下所述。

创建完成后，整个工程的目录结构如下图所示 -



**HelloController.java**

package com.yiibai.springmvc;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.ui.ModelMap;

@Controller

@RequestMapping("/hello")

public class HelloController{

@RequestMapping(method = RequestMethod.GET)

public String printHello(ModelMap model) {

model.addAttribute("message", "Hello Spring MVC Framework!");

return "hello";

}

}

Java

**web.xml**

<web-app id="WebApp\_ID" version="2.4"

xmlns="http://java.sun.com/xml/ns/j2ee"

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

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee

http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd">

<display-name>Spring MVC Application</display-name>

<servlet>

<servlet-name>HelloWeb</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>HelloWeb</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

XML

**HelloWeb-servlet.xml**

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

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

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

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

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

<context:component-scan base-package="com.yiibai" />

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix" value="/WEB-INF/jsp/" />

<property name="suffix" value=".jsp" />

</bean>

</beans>

XML

**hello.jsp**

<%@ page contentType="text/html; charset=UTF-8" %>

<html>

<head>

<title>Hello World</title>

</head>

<body>

<h2>Hello, ${message}</h2>

</body>

</html>

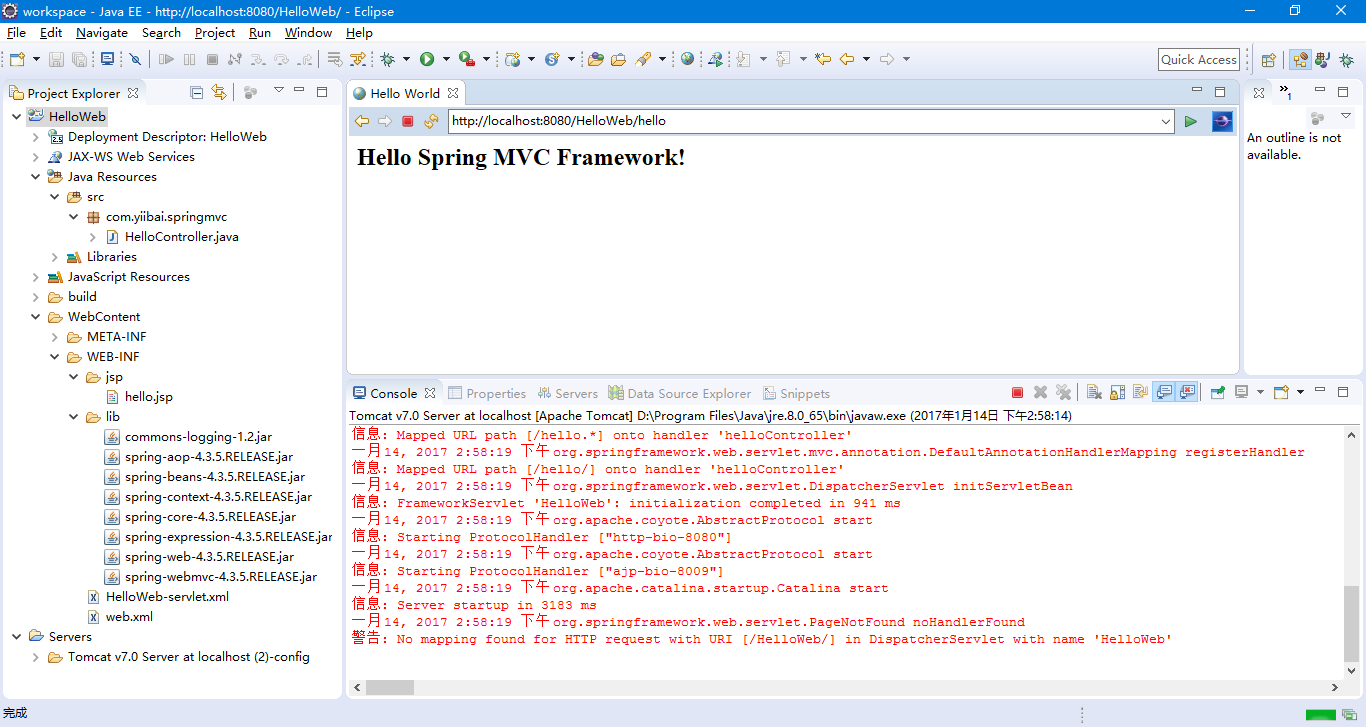
HTML

最后，以下是要包括在Web应用程序中的Spring和其他库的列表。只需拖动这些文件并将其放在WebContent/WEB-INF/lib文件夹中。

* servlet-api-x.y.z.jar
* commons-logging-x.y.z.jar
* spring-aop-x.y.z.jar
* spring-beans-x.y.z.jar
* spring-context-x.y.z.jar
* spring-core-x.y.z.jar
* spring-expression-x.y.z.jar
* spring-webmvc-x.y.z.jar
* spring-web-x.y.z.jar

完成创建源和配置文件后，导出应用程序。右键单击您的应用程序，并使用**导出> WAR文件**选项，并将 HelloWeb.war 文件保存在Tomcat的webapps文件夹中。

现在启动 Tomcat 服务器，并确保能够使用标准浏览器访问到 webapps 文件夹的其他网页。 现在尝试访问URL => http://localhost:8080/HelloWeb/hello ，如果一切都没有问题，**Spring Web**应用程序，应该看到以下结果：



应该要注意，在给定的URL中，HelloWeb是应用程序名称，hello是在控制器中使用[@RequestMapping](https://github.com/RequestMapping" \o "@RequestMapping" \t "_blank)(“/hello”)提到的虚拟子文件夹。在使用[@RequestMapping](https://github.com/RequestMapping" \o "@RequestMapping" \t "_blank)(“/”)映射到URL时，可以直接使用根(“/“)，在这种情况下，可以使用短URL => http://localhost:8080/HelloWeb/ 访问同一页面，但建议使用不同的文件夹。

注意：如果没有在Eclipse上安装配置过Tomcat服务器，在 **Window -> Preference -> Server -> Runtime Environments** 设置。