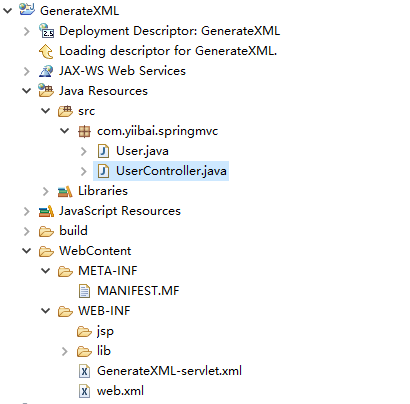
以下示例演示如何使用Spring Web MVC框架生成XML。首先使用Eclipse IDE，并按照以下步骤使用**Spring Web Framework**开发基于动态表单的Web应用程序：

1. 创建一个名称为 **GenerateXML** 的动态WEB项目。
2. 在 com.yiibai.springmvc 包下创建三个Java类：User 和 UserController。
3. 最后一步是创建所有源和配置文件的内容并运行应用程序，详细如下所述。

完整的项目文件目录结构如下所示 -



**User.java** 的代码如下所示 -

package com.yiibai.springmvc;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "user")

public class User {

private String name;

private int id;

public String getName() {

return name;

}

@XmlElement

public void setName(String name) {

this.name = name;

}

public int getId() {

return id;

}

@XmlElement

public void setId(int id) {

this.id = id;

}

}

Java

**UserController.java** 的代码如下所示 -

package com.yiibai.springmvc;

import org.springframework.stereotype.Controller;

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

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

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

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

@Controller

@RequestMapping("/user")

public class UserController {

@RequestMapping(value="{name}", method = RequestMethod.GET)

public @ResponseBody User getUser(@PathVariable String name) {

User user = new User();

user.setName(name);

user.setId(1);

return user;

}

}

Java

**GenerateXML-servlet.xml** 配置如下所示 -

<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"

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

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

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

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

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

<mvc:annotation-driven />

</beans>

XML

上面的代码中创建了一个XML映射的POJO用户，在UserController中返回了User对象。 Spring基于RequestMapping自动处理XML转换。

完成创建源和配置文件后，发布应用程序到Tomcat服务器。

现在启动Tomcat服务器，当访问URL => <http://localhost:8080/GenerateXML/user/yiibai> ， 如果**Spring Web**应用程序没有问题，应该看到以下结果：

