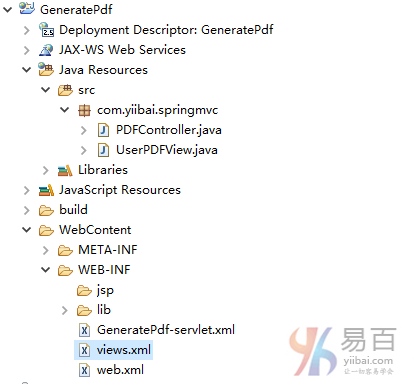
以下示例演示如何使用Spring Web MVC框架生成PDF格式的文件。首先使用Eclipse IDE，并按照以下步骤使用**Spring Web Framework**开发基于动态表单的Web应用程序：

1. 创建一个名称为 **GeneratePdf** 的动态WEB项目。
2. 在 com.yiibai.springmvc 包下创建三个Java类：UserPDFView 和 PDFController。
3. 从[maven](http://www.yiibai.com/maven/)存储库页面下载Apache iText库：[Apache iText](http://mvnrepository.com/artifact/com.lowagie/itext/2.1.7)。 把它放在CLASSPATH中。
4. 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 -



**PDFController.java** 的代码如下所示 -

package com.yiibai.springmvc;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.ModelAndView;

import org.springframework.web.servlet.mvc.AbstractController;

public class PDFController extends AbstractController {

@Override

protected ModelAndView handleRequestInternal(HttpServletRequest request,

HttpServletResponse response) throws Exception {

//user data

Map<String,String> userData = new HashMap<String,String>();

userData.put("100", "Xiao.Lu");

userData.put("102", "User 102");

userData.put("301", "User 301");

userData.put("400", "User 400");

return new ModelAndView("UserSummary","userData",userData);

}

}

Java

**UserPDFView.java** 的代码如下所示 -

package com.yiibai.springmvc;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.view.document.AbstractPdfView;

import com.lowagie.text.Document;

import com.lowagie.text.Table;

import com.lowagie.text.pdf.PdfWriter;

public class UserPDFView extends AbstractPdfView {

protected void buildPdfDocument(Map<String, Object> model, Document document,

PdfWriter pdfWriter, HttpServletRequest request, HttpServletResponse response)

throws Exception {

Map<String,String> userData = (Map<String,String>) model.get("userData");

Table table = new Table(2);

table.addCell("No.");

table.addCell("User Name");

for (Map.Entry<String, String> entry : userData.entrySet()) {

table.addCell(entry.getKey());

table.addCell(entry.getValue());

}

document.add(table);

}

}

Java

**GeneratePdf-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">

<bean

class="org.springframework.web.servlet.mvc.support.ControllerClassNameHandlerMapping" />

<bean class="com.yiibai.springmvc.PDFController" />

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

<property name="location">

<value>/WEB-INF/views.xml</value>

</property>

</bean>

</beans>

XML

**views.xml** 文件中的配置如下所示 -

<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"

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">

<bean id="UserSummary" class="com.yiibai.springmvc.UserPDFView"></bean>

</beans>

XML

在上面的代码中，创建了一个PDFController和UserPDFView类。iText库是用来处理PDF文件格式，并将数据转换为PDF文档。

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/GeneratePdf/pdf](http://localhost:8080/GeneratePdf/pdf" \t "_blank) ， 如果**Spring Web**应用程序没有问题，应该看到以下结果：

