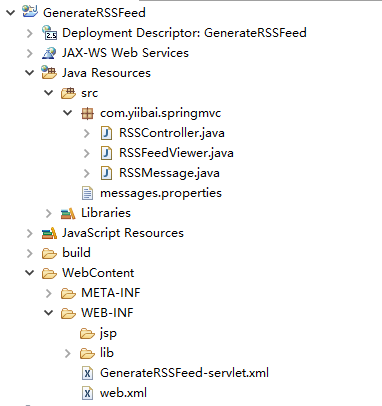
下面的示例演示如何使用**Spring Web MVC**框架生成RSS源。 首先使用Eclipse IDE，并按照以下步骤使用**Spring Web Framework**开发基于动态表单的Web应用程序：

1. 创建一个名称为 **GenerateRSSFeed** 的动态WEB项目。
2. 在 com.yiibai.springmvc 包下创建三个Java类：RSSMessage, RSSFeedViewer 和 RSSController。
3. 从相同的maven存储库页面下载 [Rome](http://mvnrepository.com/artifact/com.rometools/rome/1.5.0) 库及其依赖项[rome-utils](http://mvnrepository.com/artifact/com.rometools/rome-utils/1.5.0)，[jdom](http://mvnrepository.com/artifact/org.jdom/jdom/2.0.2)和[slf4j](http://mvnrepository.com/artifact/org.slf4j/slf4j-api/1.7.7)。和所需的依赖关系存并将它们放在CLASSPATH中。
4. 在src文件夹下创建属性文件messages.properties。
5. 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 -



**RSSMessage.java** 的代码如下所示 -

package com.yiibai.springmvc;

import java.util.Date;

public class RSSMessage {

String title;

String url;

String summary;

Date createdDate;

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

public String getSummary() {

return summary;

}

public void setSummary(String summary) {

this.summary = summary;

}

public Date getCreatedDate() {

return createdDate;

}

public void setCreatedDate(Date createdDate) {

this.createdDate = createdDate;

}

}

Java

**RSSFeedViewer.java** 的代码如下所示 -

package com.yiibai.springmvc;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.view.feed.AbstractRssFeedView;

import com.rometools.rome.feed.rss.Channel;

import com.rometools.rome.feed.rss.Content;

import com.rometools.rome.feed.rss.Item;

public class RSSFeedViewer extends AbstractRssFeedView {

@Override

protected void buildFeedMetadata(Map<String, Object> model, Channel feed,

HttpServletRequest request) {

feed.setTitle("YiibaiPoint Dot Com");

feed.setDescription("Java Yiibai and Examples");

feed.setLink("http://www.yiibai.com");

super.buildFeedMetadata(model, feed, request);

}

@Override

protected List<Item> buildFeedItems(Map<String, Object> model,

HttpServletRequest request, HttpServletResponse response) throws Exception {

List<RSSMessage> listContent = (List<RSSMessage>) model.get("feedContent");

List<Item> items = new ArrayList<Item>(listContent.size());

for(RSSMessage tempContent : listContent ){

Item item = new Item();

Content content = new Content();

content.setValue(tempContent.getSummary());

item.setContent(content);

item.setTitle(tempContent.getTitle());

item.setLink(tempContent.getUrl());

item.setPubDate(tempContent.getCreatedDate());

items.add(item);

}

return items;

}

}

Java

**RSSController.java** 文件中的代码如下所示 -

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import org.springframework.stereotype.Controller;

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

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

import org.springframework.web.servlet.ModelAndView;

@Controller

public class RSSController {

@RequestMapping(value="/rssfeed", method = RequestMethod.GET)

public ModelAndView getFeedInRss() {

List<RSSMessage> items = new ArrayList<RSSMessage>();

RSSMessage content = new RSSMessage();

content.setTitle("Spring教程");

content.setUrl("http://www.yiibai/spring");

content.setSummary("Spring tutorial summary...");

content.setCreatedDate(new Date());

items.add(content);

RSSMessage content2 = new RSSMessage();

content2.setTitle("Spring MVC教程");

content2.setUrl("http://www.yiibai/spring\_mvc");

content2.setSummary("Spring MVC tutorial summary...");

content2.setCreatedDate(new Date());

items.add(content2);

ModelAndView mav = new ModelAndView();

mav.setViewName("rssViewer");

mav.addObject("feedContent", items);

return mav;

}

}

Java

**GenerateRSSFeed-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"

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.springmvc" />

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

<bean id="rssViewer" class="com.yiibai.springmvc.RSSFeedViewer" />

</beans>

XML

上面的代码中创建了一个**RSS feed POJO RSSMessage**和一个**RSS Message Viewer**，它扩展了AbstractRssFeedView并覆盖了它的方法。 在RSSController中，生成了一个示例RSS Feed。

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

