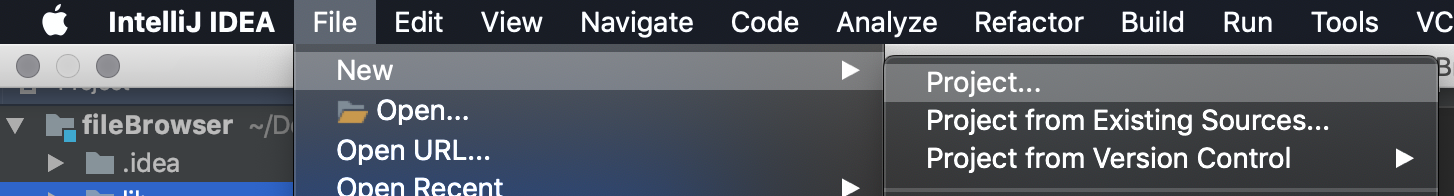
# SpringMVC学习笔记

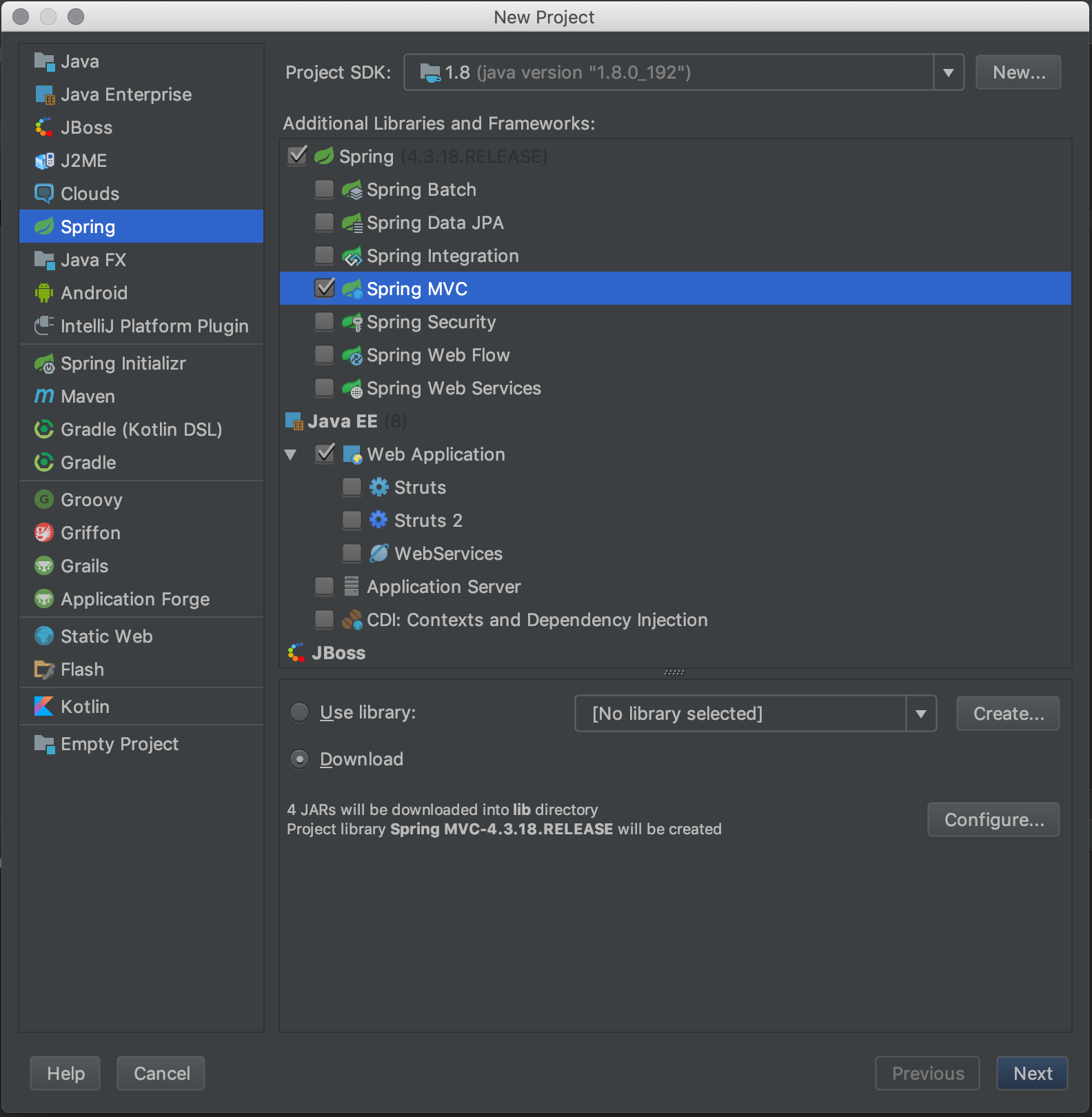
## 第一课 helloworld

### 项目创建

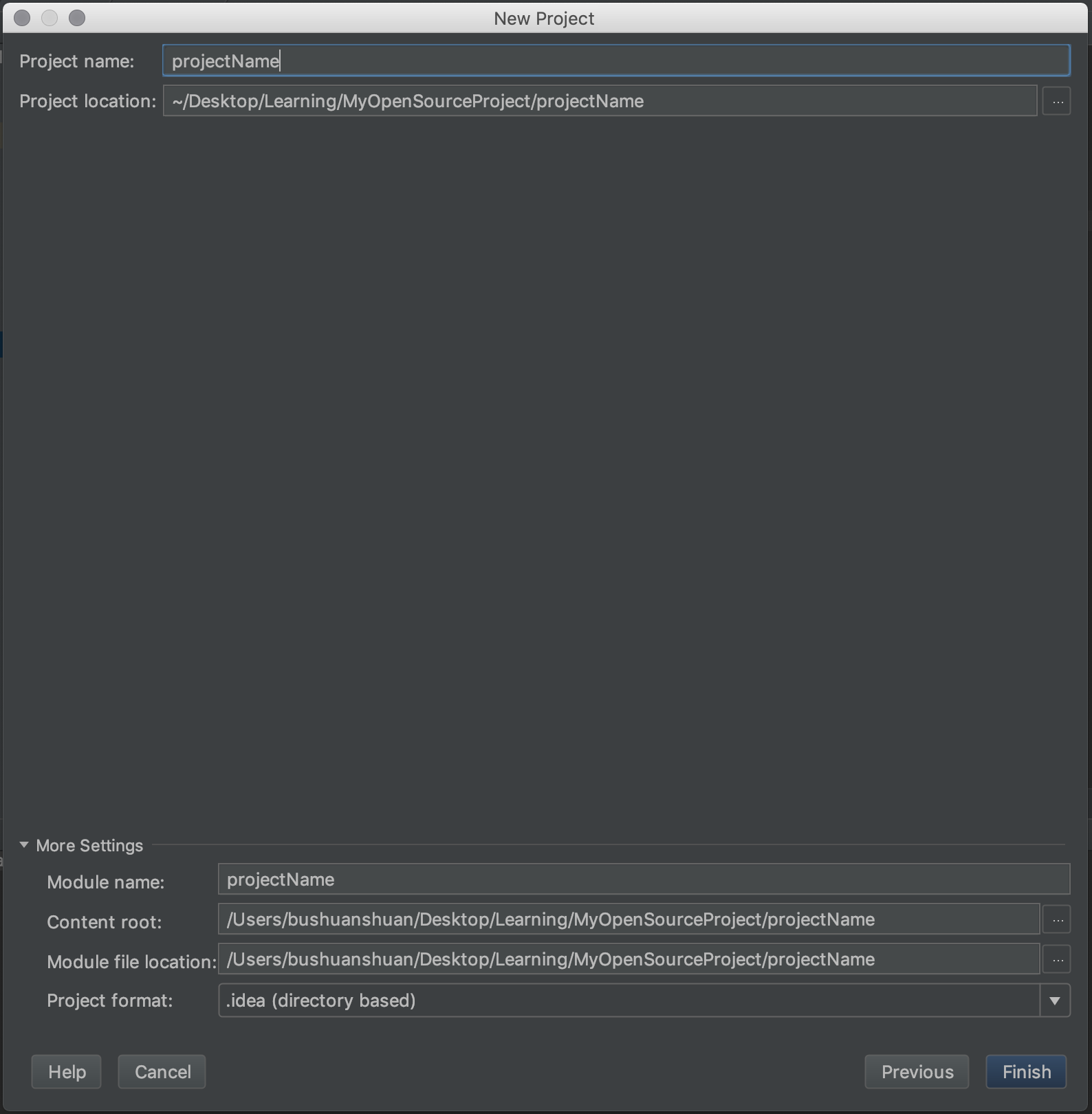
在Intellij中选择：File->New->Project



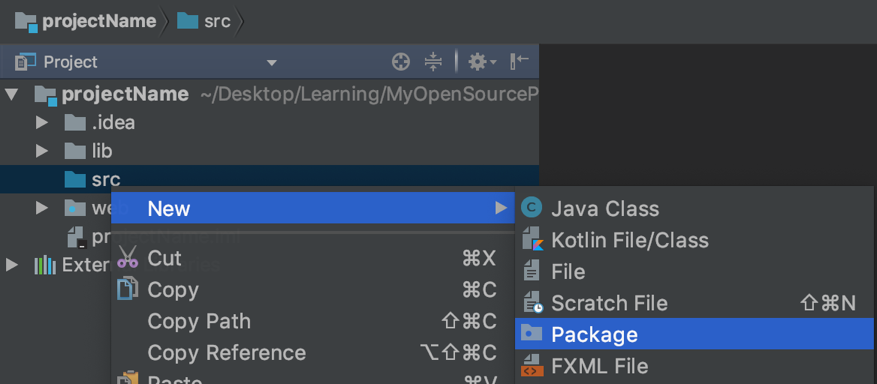
然后选择Spring->Spring，勾选Spring MVC，并勾选下面的WebApplication如下：

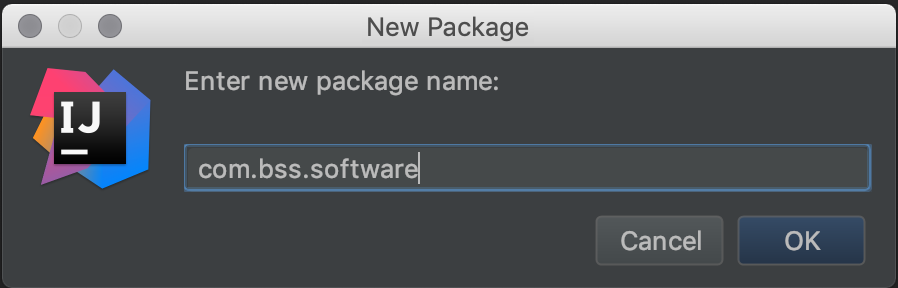


然后输入项目名称，点击Finish即可

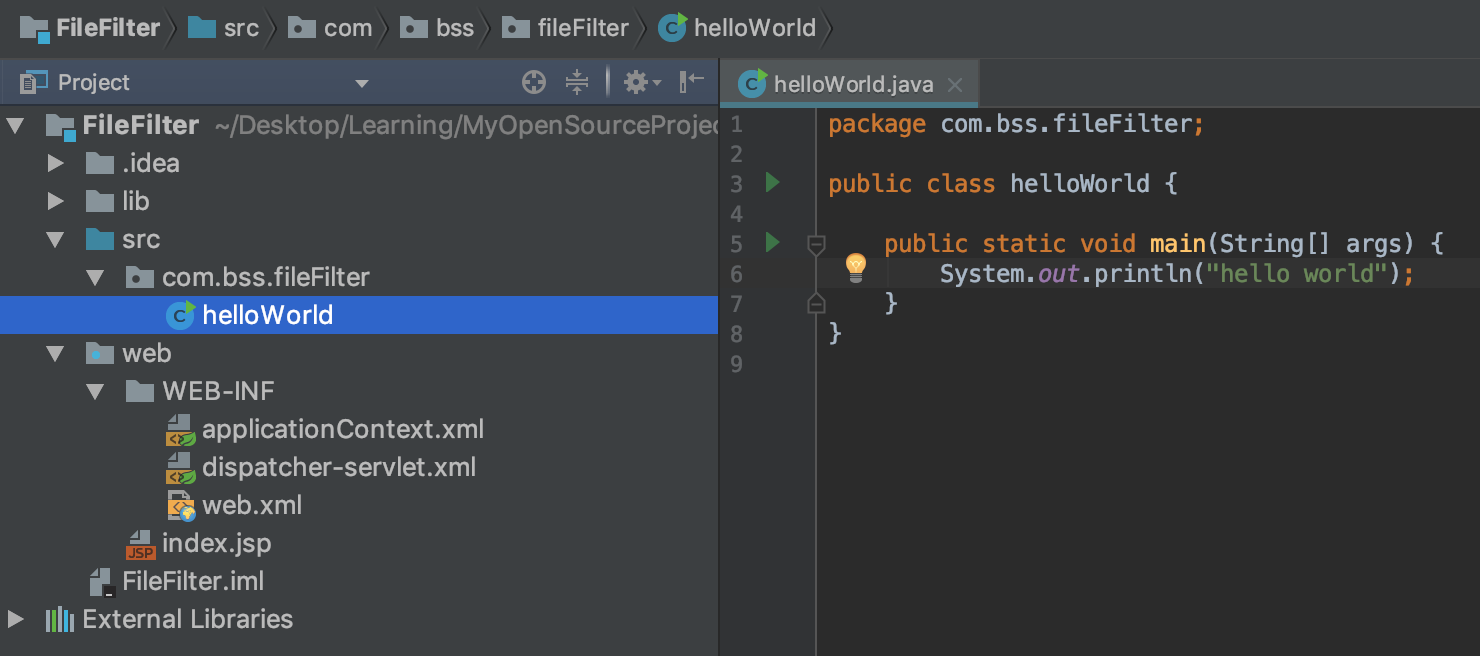


然后在src上点击New->Package，输入项目创建的包名称：

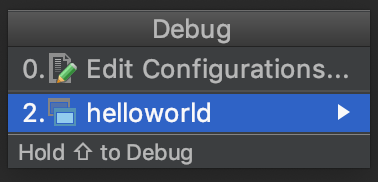




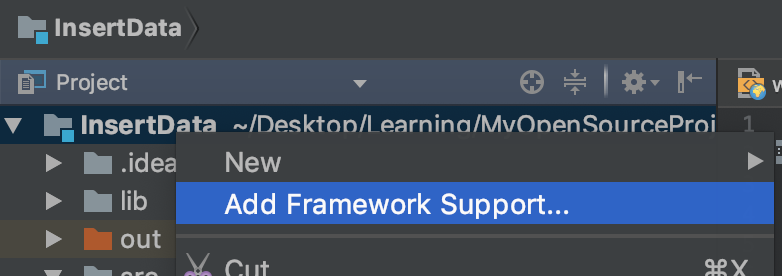
此时，可以看到WEB-INF下Intellij已经为我们创建好了web.xml,dispatcher-servlet.xml和applicationContext.xml文件，我们可以简单地然后创建一个基本的helloworld，并执行：



点击Run->Run，然后选择helloworld

即可看到结果。

### 添加SpringMVC框架支持



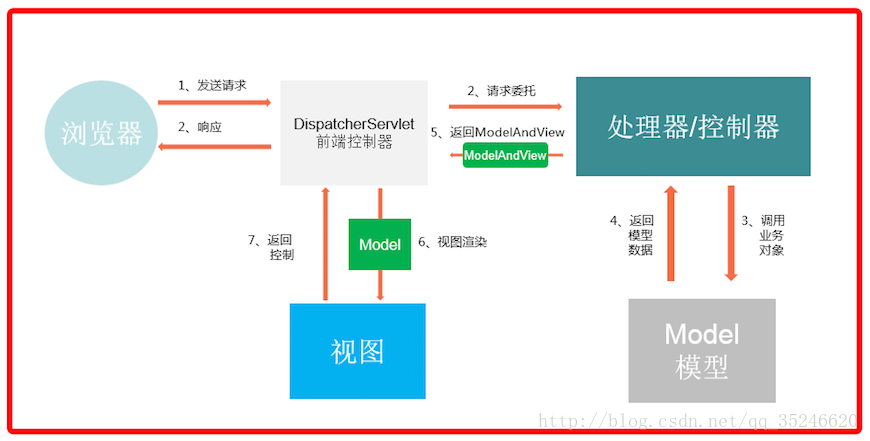
在项目目录上右键，选择Add Framework Support，然后选择Maven即可，缺少依赖包后直接在maven中添加即可。注意如果运行失败，需要重新配置Tomcat中的war，重新拷贝相关包。

#### 依赖的核心包：

1. SpringMVC核心包：spring-web-4.3.18.RELEASE.jar, spring-webmvc-4.3.18.RELEASE.jar
2. AOP核心包：spring-aop.4.3.18.RELEASE.jar,spring-aspects.4.3.18.RELEASE.jar
3. 日志包：commons-logging.jar

#### SpringMVC流程：

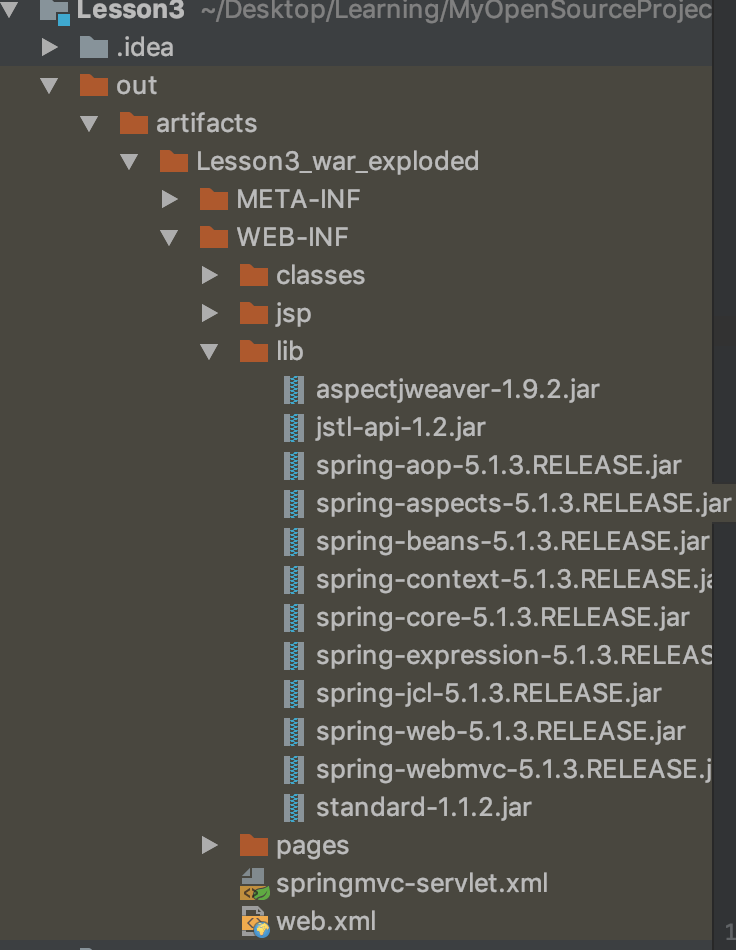
所有请求经过Dispatcher servlet，然后distpatcher servlet再转发给其他类去处理，该servlet在spring中已经实现，只需要在web.xml配置文件中使用上就可以。在Spring MVC框架中，从“Request（请求）”开始，依次进入“DispatcherServlet（核心分发器）” —> “HandlerMapping（处理器映射）” —> “Controller（控制器）” —> “ModelAndView（模型和视图）” —> “ViewResolver（视图解析器）” —> “View（视图）” —> “Response（响应）”结束，其中DispatcherServlet、HandlerMapping和ViewResolver 只需要在XML文件中配置即可，从而大大提高了开发的效率，特别是对于 HandlerMapping 框架为其提供了默认的配置。Spring MVC 框架的结构图如下所示：



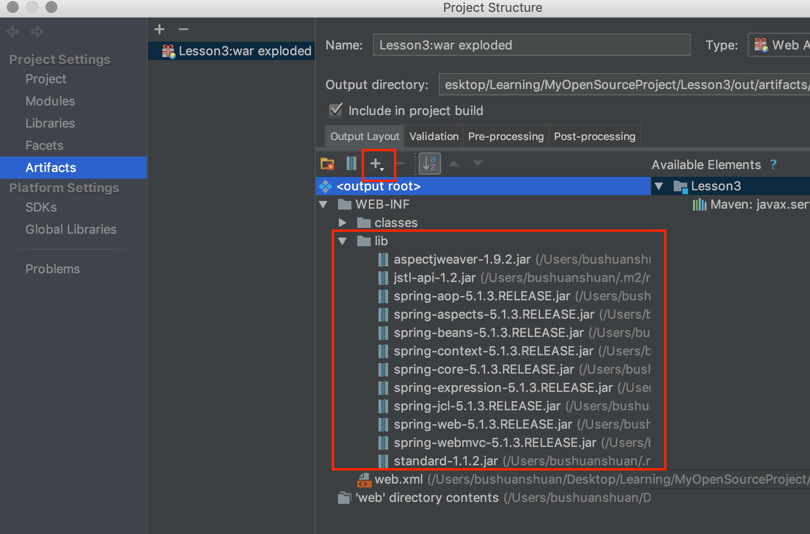
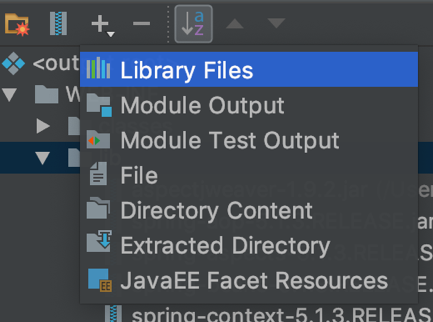
Web.xml中配置使用SpringMVC servlet示例：

错误解决1：

1. 报错异常可在out目录中确认是否缺少相关包或者jsp等文件。



1. 在使用MAVEN进行以上SpringMVC项目管理时，如果新添加了依赖的jar包，则在配置Tomcat生成war时，要重新在lib中增加下载的jar包。点击加号Add copy of按钮，选择Library Files，将下载的库中的jar包拷贝到lib依赖目录。

## 第二课 SpringMVC使用及原理

### 注解说明

#### <mvc: annotation-driven />注解

Spring 3.0.x及以后版本中使用了mvc:annotation-driven后，默认会帮我们注册默认处理请求，参数和返回值的类，即使用注解的方式进行SpringMVC开发。

#### <context:component-scan base-package="xxx"/>注解

启用包扫描生成bean，扫描包路径为base-package路径。

#### 视图解析器注解

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"> <!-- 将视图解析为Web应用的内部资源，一般为JSP -->  
 <property name="prefix" value="/WEB-INF/jsp/"></property>  
 <property name="suffix" value=".jsp"></property>  
</bean>

使用哪个类来进行视图解析。

## JSP 整理

### Spring表单标签

Spring封装了一系列的表单标签，这些标签都可以访问到ModelMap中的内容。使用表单标签，首先需要现在JSP中声明使用表单标签。

<%@ **taglib** prefix="form" uri="http://www.springframework.org/tags/form" %>

#### <meta>标签

meta是用来在HTML文档中模拟HTTP协议的响应头报文。meta 标签用于网页的<head>与</head>中。

#### <input>标签

SpringMVC的input标签会被渲染成一个type为text的普通html input标签。该标签的唯一作用就是它能绑定表单数据。

#### <hidden>标签

hidden标签会被渲染为一个type为hidden的普通Html input标签。用法跟input标签一样，也能绑定表单数据，只是它生成的是一个隐藏域。

#### <checkbox>标签

checkbox标签会被渲染为一个type为checkbox的普通HTML input标签。checkbox标签也是支持绑定数据的。

#### <password>标签

password标签将会被渲染为一个type为password的普通HTML input标签。

#### <select>标签

select标签将会被渲染为一个普通的HTML select标签。

### JSTL标签库

JSTL即是JavaServer Pages Standard Tag Library，JSP标准标签库。JSTL可以用标签的形式来实现循环语句、条件语句、格式化等。

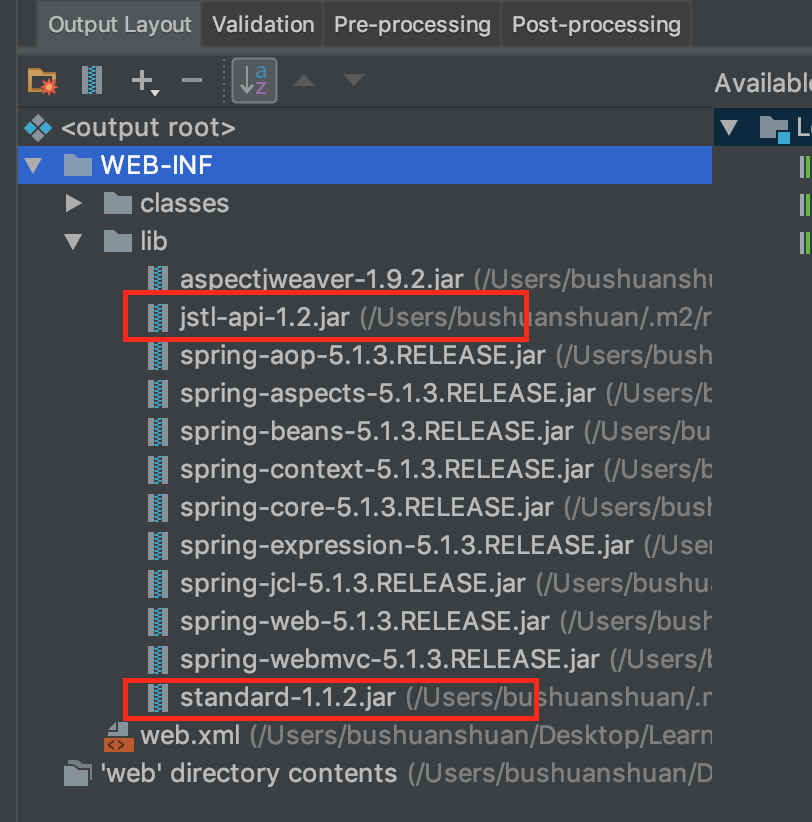
#### JSTL标签库使用

在SpringMVC中使用JSTL标签库需要以下配置：

1. 在Maven中增加以下依赖

<dependency>  
 <groupId>javax.servlet.jsp.jstl</groupId>  
 <artifactId>jstl-api</artifactId>  
 <version>1.2</version>  
</dependency>  
  
<dependency>  
 <groupId>taglibs</groupId>  
 <artifactId>standard</artifactId>  
 <version>1.1.2</version>  
</dependency>

并确保在生成war的地方包含了依赖jar包：



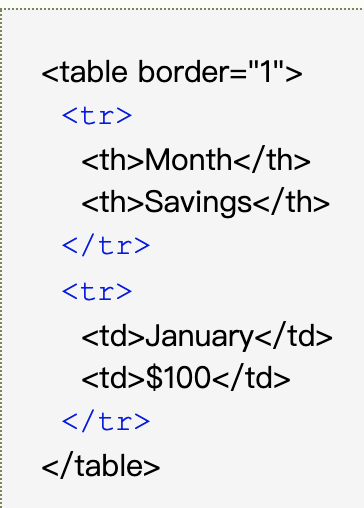
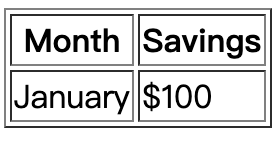
1. 在JSP文件中使用时，在文件起始位置添加了如下行

<%@ **taglib** uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

# HTML整理

## <tr>标签

<tr>标签定义HTML表格中的行【table row】，元素包含一个或多个th或td元素。

 显示为：

## <td> 标签

定义 HTML 表格中的标准单元格。HTML 表格有两类单元格：1) 标准单元 - 包含数据（由 td 元素创建）; 2) 表头单元 - 包含头部信息（由 th 元素创建）。

td 元素中的文本一般显示为正常字体且左对齐。

colspan属性:规定单元格可横跨的列数。

## <form>标签

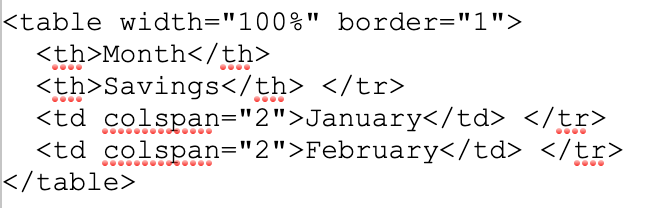
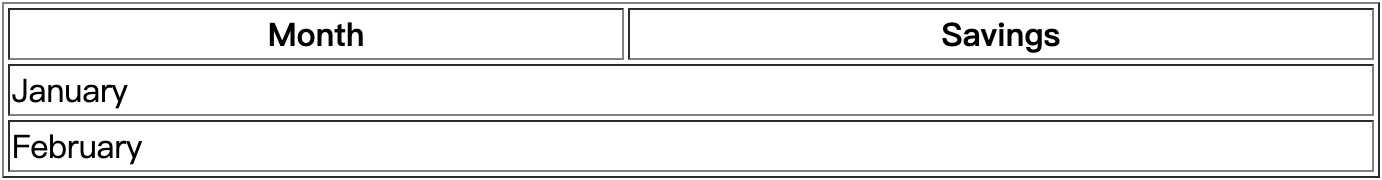
用于为用户输入创建 HTML 表单，表单用于获取用户输入。form标签有两个主要作用：

1）自动绑定来自Model中的属性值到当前form的实体中，类似自动装配。【默认是command属性】

2）支持在提交表单时除了GET和POST外的其他方法进行提交，包括DELETE，PUT等。

## <table> 标签

table标签用于定义HTML表格，表格用于展示数据信息。简单的HTML表格由table元素以及一个或多个tr、th或td元素组成。tr元素定义表格行，th元素定义表头，td元素定义表格单元。使用示例：

 🡺 

## 补充

MAC下查看监听端口和IP

lsof -iTCP -sTCP:LISTEN -nP