

<https://my.oschina.net/lujianing/blog/186737>

摘要：非maven项目和 maven项目部署配置的探究

## 1. 前言

虽说IntelliJ IDEA已经用了四个多月，里面大部分功能还是比较熟悉的，但是很多细节地方还是搞不清楚，这几天公司另一个小组也换成了IntelliJ IDEA，帮他们迁移项目的时候，因为是非maven项目，遇到了很多问题，自己又研究了一番，下面整理一下

## 2. 项目配置（Project Structure）

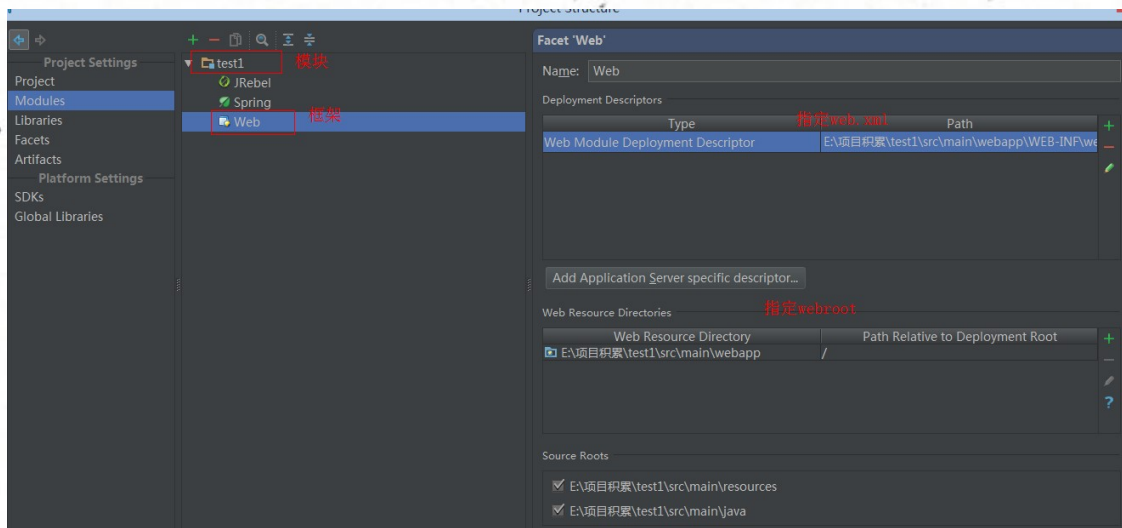
### 2.1 Project

project compiler output：配置的是项目中的默认编译输出总目录

里面英文也说明了，每个模块可以自己设置特殊的输出目录 所以这个基本上没用

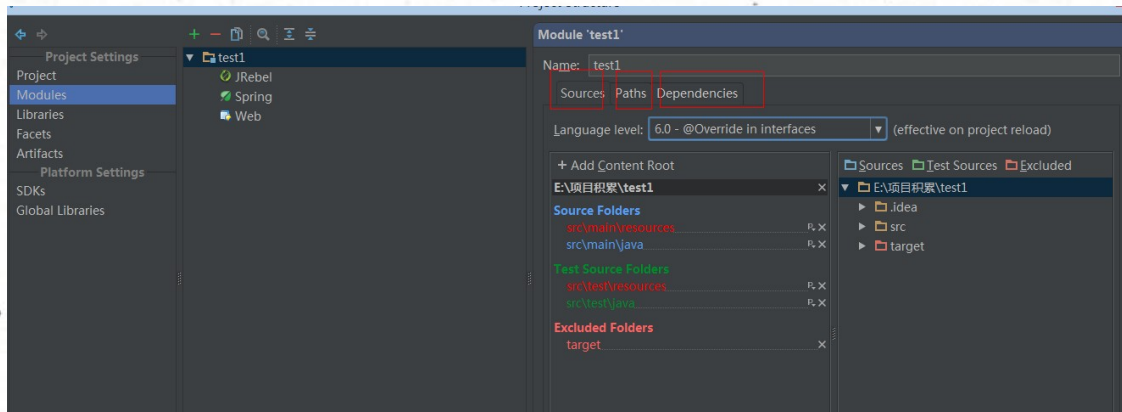
### 2.2 Modules

项目的模块，一个项目中可以有多个子项目，每个子项目相当于一个模块一般我们项目只是单独的一个，所以只需要配置一个模块，我们可以给模块添加框架



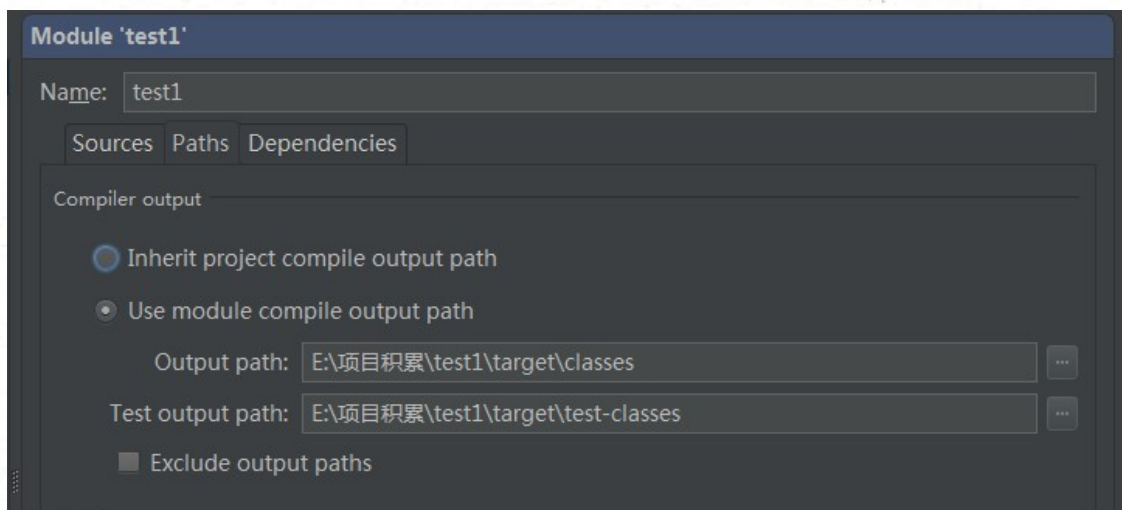
根据我们项目中的使用到东西，添加对应的框架，并在框架里面确定一下相关的配置文件（因为不同IDE下，项目的文件结构可能不同 所以导入的项目我们要注意一下配置）

模块配置根目下 分为了三个标签：Sources—— Paths—— Depedencies



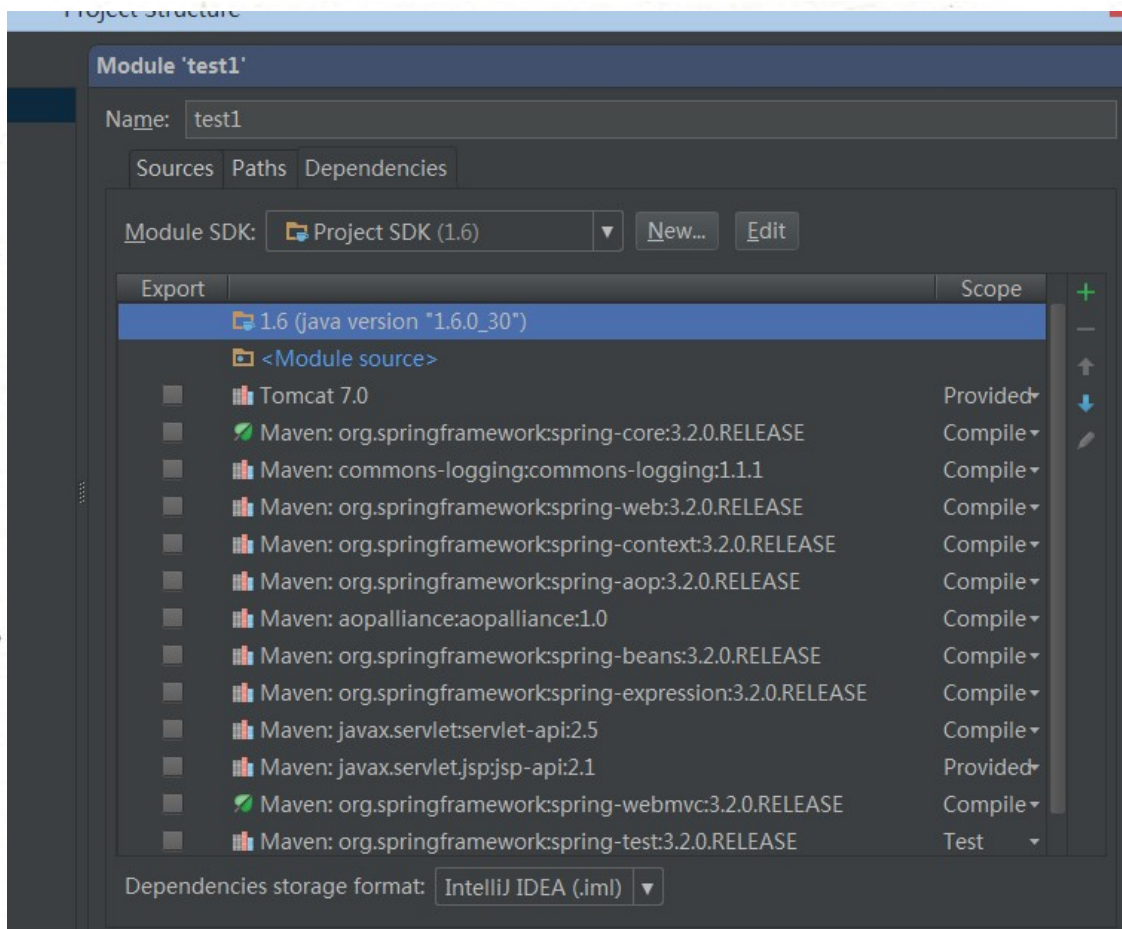
Sources标签下 主要显示项目的目录资源 那些是项目部署的时候需要的目录 有颜色提示  
(比如绿色是测试目录 橘色是target的目录 这些部署的时候都不需要 大红色是我的项目下没有的 蓝色是真正需要部署)

Paths标签下



可以指定项目的编译输出目录 也就是项目类 和 测试类的编译输出地址  
(替换掉了Project的默认输出地址)

Depedencies是项目的依赖



我们可以点击右侧的+号 添加lib 比如tomcat的lib 默认加进来就是 Provided

(和maven的里面的jar依赖差不多)

Provided 也就是项目部署的时候是忽略的 只是再编译项目的时候进行使用

Compile 是跟随着项目部署的 Test也是不部署的

## 2.3 Libraries

里面可以添加项目jar包 并且可以把多个jar放在一个组里面 类似于jar包整理

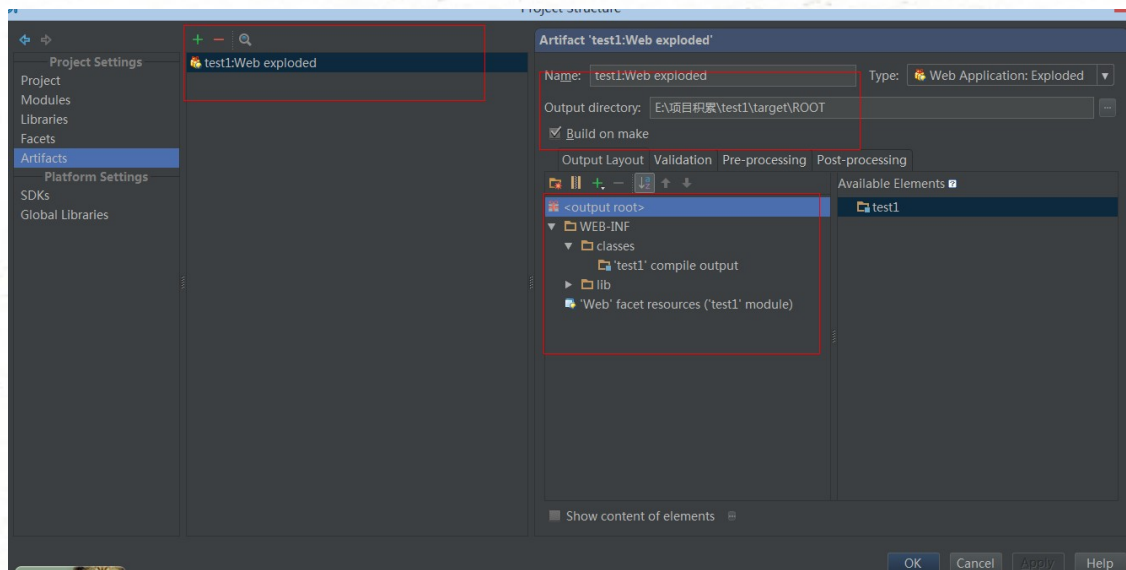
这个下添加的jar 会在Dependencies中显示 (反之不可以)

## 2.4 Facts

项目的门面, 说实话, 真不知道这个是干嘛的, 和modules差不多, 主要是为了确认一下吧

## 2.5 Artifacts

项目的打包部署设置，这个是项目配置里面比较关键的地方



注意红框的地方

第一个是我们需要添加项目的部署包 注意选择是exploded的war包

第二个是项目的输出目录 非maven项目这个可以设置到任何地方

（maven项目的需要特别注意 这个后面再说）

第三个是输出结构（output layout） 参考图中的几个文件

（maven项目如果不使用maven的打包编译 注意图中WEB-INF下的lib要有项目部署的jar）

## 3. 非Maven项目的创建部署

创建File——New Project——Java——Web Application

（注意New Project里面不是选择的web 而是java）

intellij中web项目 默认网页根目录是web

项目配置可参考2中的几个地方 根据自己的需求自己修改

artifact中out directory 可以指定到任何地方 项目打包编译到指定的目录下

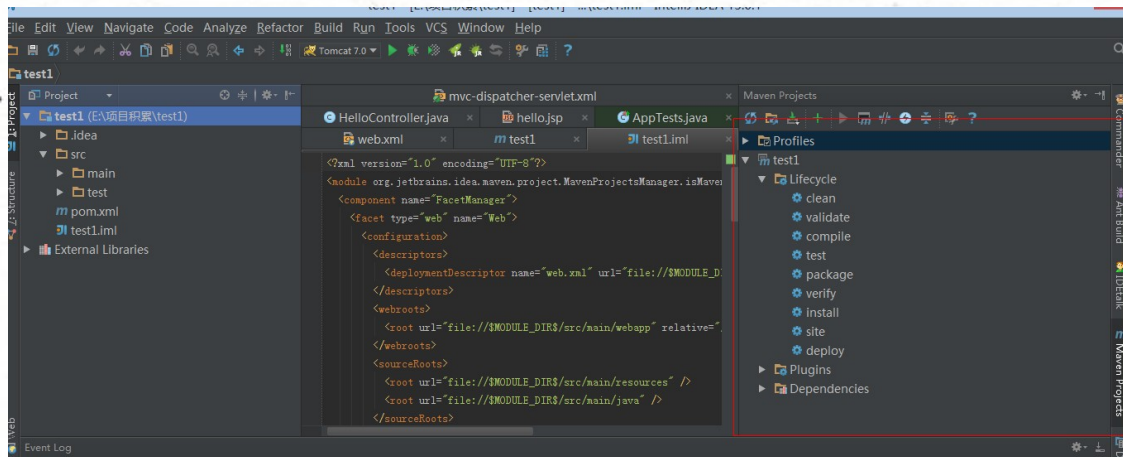
在intellij中启动tomcat 都会以对应war包out directory的设置的目录为工作空间

tomcat配置请参考[http://my.oschina.net/u/140593/blog/177042#OSC\\_h1\\_2](http://my.oschina.net/u/140593/blog/177042#OSC_h1_2) 中的2-7

## 4. Maven项目的部署



在intellij中 maven项目还是可以按照maven的打包编译 参考图中的几个生命周期 手动去执行



因为maven项目的约定 是项目打包编译到 target目录下

所以我们在2中配置都应该去遵循maven的配置约定进行修改

这样我们的项目在文件修改或者热部署的时候 intellij的编译才能和 maven项目保持一致

（如果是在intellij中创建maven项目 那个intellij会给你配置好所有的配置

基本不用修改 我们可以创建一个springmvc的例子进行参考）

需要注意的地方是：

maven项目 artifact中out directory 的目录名

一定要和pom文件中build的finalName一致

另外pom文件中 要配置<packaging>war</packaging>

比如pom中配置的项目bulid finalName叫做R00T

那么我们的out directory为 E:\项目积累\test1\target\R00T

通过maven编译以及打包项目后

在服务器的配置中，可以去掉下图红色区域的两个选项，避免每次启动时都重新编译和打包

