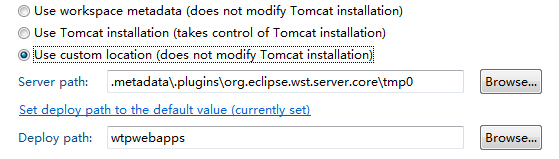
Ide会将编译好的文件存放在制定目录，

Intellij发布在${dir.to.idea.project}\YourWebApp\out\artifacts

Eclipse



然后web server（tomcat）会根据此访问该制定目录

JSP在请求的时候，会先转化成Servlet（其实就是个java类），然后生成class文件

对应目录下的class删除也可以正常访问，是因为server会缓存运行过的class，需要重启server

实现**热部署**就可以不重启server修改类

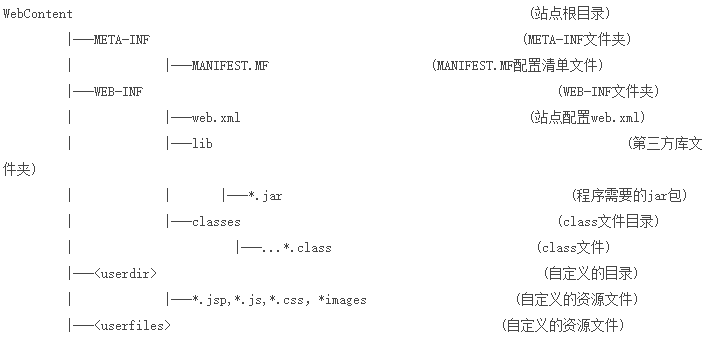
Jsp也会生成在对应目录

Webapp的实现是根据javaee的web容器实现

有相应的目录结构来解析

Eclipse中的webroot，intellij里的web都是实际发布到服务器上的内容，是web容器的缺省结构：

一般的目录结构如下：



1.WebRoot(WebContent)下是发布到服务器上的内容。

2.META-INF是工程自身相关的一些信息，元文件信息，通常由开发工具，环境自动生成。

3．文件web.xml：完成servlet在web容器的注册。web.xml是Web应用程序的部署描述文件，是用来给Web服务器解析并获取Web应用程序相关描述的。

不按照sun公司的规范做应用web程序的结构，web容器找不到，比如，xml文件写错了，启动tomcat的时候会报错

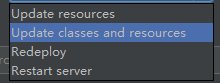
4．凡是客户端能访问的资源(\*.html,\*.jpg)必须跟WEB-INF在同一目录。即放在Web根目录下的资源，从客户端是可以通过URL地址直接访问。

5. 切忌：凡是WEB-INF里面的文件都不能被客户端直接访问(比如隐藏的信息)。WEB-INF目录下的资源对用户来说是不可见的，而对Web服务器来说则没有这样的限制。

例如在WEB-INF下的index.htm，客户端无法与对待其他文件夹内的信息一样，通过http://yourserver/yourwebapp/WEB-INF/index.htm访问。WEB-INF文件夹是禁止通过URL访问的。  
6. 在WEB-INF目录的classes及lib子目录下，都可以存放[Java](http://lib.csdn.net/base/javase)类文件。在运行时，Servlet容器的类加载器先加载classes目录下的类，再加载lib目录下的JAR文件（Java类库的打包文件）中的类，jar包是许多class文件的集合。因此，如果两个目录下存在同名的类，classes目录下的类具有优先权。

|  |  |
| --- | --- |
| Tomcat固定的目录结构 | |
| /bin | 存放在Windows平台以及[Linux](http://lib.csdn.net/base/linux)平台上启动和关闭Tomat的脚本文件 |
| /conf | 存放Tomat服务器的各种配置文件， 其中最重要的文件是Server.xml |
| /server | 包含3个子目录：classes,lib和webapps |
| /server/lib | 存放Tomat服务器所需的Jar文件 |
| /server/webapps | 存放Tomat自带的两个Web应用：admin应用和manager应用 |
| /common/lib | 存放Tomat服务器以及所有Web应用都可以访问的Jar应用 |
| /share/lib | 存放所有Web应用都可以访问的Jar文件 |
| /logs | 存放Tomat的日志文件 |
| /webapps | 当发布Web应用时，默认情况下把Web应用文件放于此目录下 |
| /work | Tomcat把由Jsp生成的Servlet放于此目录下 |

war模式—-将WEB工程以包的形式上传到服务器   
war exploded模式—-将WEB工程以当前文件夹的位置关系上传到服务器

war exploded可选择update classes and resources进行热加载

这里只是ide发布到服务器时更新了class，但服务器本身没有热加载！