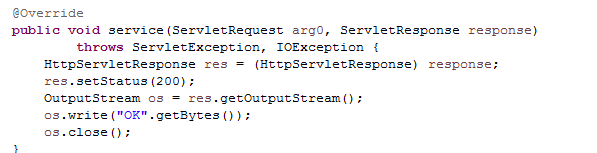
手动部署一个web工程MyServlet

1. 编写一个MyServlet类，需要实现javax.servlet.Servlet接口。

注意：**有没有包**，如果有包，后面需要建立文件夹，如果没有包，直接拷贝到WEB-INF文件夹下即可。

1. **实现Servlet接口**，需要实现5个方法，这里简单起见，只需要实现service方法即可。

**不设置200，默认也是200。打印出OK。**



1. 然后对此类进行编译，得到MyServlet.class文件。
2. 进入webapps文件夹下，创建一个web文件夹，文件夹的名称就是web的名称，如web1；
3. 然后在web1文件夹中创建一个名为WEB-INF的文件夹；
4. 在WEB-INF文件夹下，创建一个classes文件夹，并建立一个web.xml文件，如果web.xml文件的内容不是很懂，可以从其他web工程中复制过来，并做简单的修改即可；
5. 最后把MyServlet.class文件拷贝到classes文件夹中，如果有包，需要创建文件夹即可。
6. 在web.xml文件中，必须做一下配置：

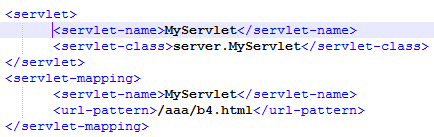
**一个是servlet，一个是servlet-mapping。**

servlet-name可以自己随便起，和下面的servlet-name一致即可。

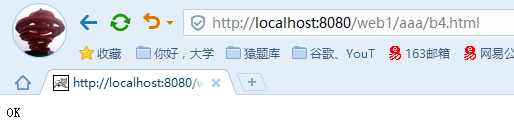
servlet-class必须按照类的全名指定。

servlet-pattern 这个必须以/开头，后面的随意写，浏览器访问时就是这个路径：[http://localhost:8080/web1**/aaa/b4.html**](http://localhost:8080/web1/aaa/b4.html)

**注意：/\*.html 是不允许的，如果\*号前面有/，则后面不能带扩展名。**



1. 配置结束后，tomcat会自动加载，在地址栏输入<http://localhost:8080/web1/aaa/b4.html>，即可进行访问。结果：



1. 如果需要修改java文件，需要重新编译获取.class文件，然后再拷贝到WEB-INF/classes文件夹下，需要在web.xml中稍微改一下，再保存，tomcat就会自动重新加载这个web1。 **tomcat随时监听web.xml文件是否改变。**
2. 以后利用eclipse，就会直接完成web工程的部署，不再需要手动部署，但是掌握这个过程，有助于理解原理。