## 开发第一个SERVLET

**Servlet 本身不能独立运行，需要在一个web应用中运行的**

**而一个web应用是部署在tomcat中的**

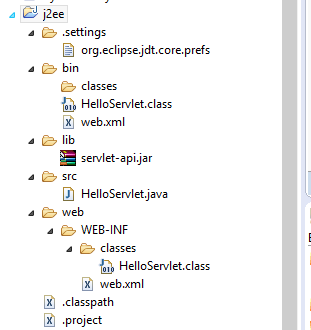
**所以开发一个servlet需要如下几个步骤**

**创建web应用项目**

**编写servlet代码**

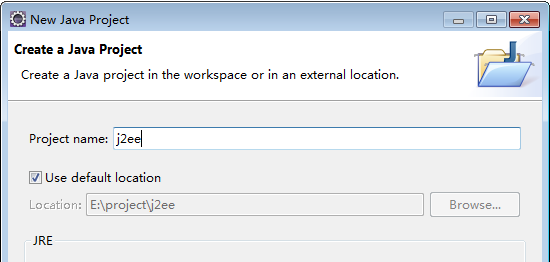
**部署到tomcat中**

### 0、整体项目目录结构



### 创建一个java项目j2ee

File->New->Other->Java->Java Project  
输入名称j2ee  
这样就创建一个单纯的基于java的项目  
注意： 必须 放在****e:\project\j2ee****目录下，后续配置是基于这个目录进行  
注意：****不要使用如下方式****  
File->New->Other-Web->Dynamic Web Project  
这是eclipse ee创建项目的方式，它会把各种tomcat,相关配置文件的生成都自动做掉了。 而这些细节，恰恰就是你需要学习和掌握的。 所以对于初学者不建议使用这种方式。



### 导入必须的servlet-api.jar包

因为是开发Servlet所以需要用到 servlet-api.jar 这个包里的相关类。  
  
servlet-api.jar 包位于 D:\tomcat\lib\servlet-api.jar  
  
或者从本网页的右侧下载   
  
导入办法:(复制到在lib下，再add to build path)  
  
右键点击项目 -> properties -> Java Build Path ->Libraries -> Add External JARs

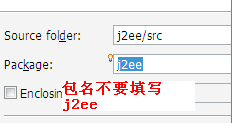
### 编写HelloServlet

创建 HelloServlet  
****注意：****eclipse创建的时候会自带一个****包名j2ee****，****不要使用这个包名，让包名置空****  
HelloServlet继承了 HttpServlet  
并且提供了一个doGet方法  
  
后续测试该类的时候，会直接在浏览器输入地址

<http://127.0.01/hello> 或

http://127.0.01:8080/hello

在浏览器中输入地址提交数据的方式是GET ([参考GET和POST的区别](http://how2j.cn/k/html/html-form/190.html" \l "step1235))  
所以该Servlet需要提供一个对应的doGet方法  
  
在doGet中，输出html  
分别是 标题元素h1 和 当前时间的字符串



import java.io.IOException;

import java.util.Date;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class HelloServlet extends HttpServlet{

public void doGet(HttpServletRequest request, HttpServletResponse response){

try {

response.getWriter().println("<h1>Hello Servlet!</h1>");

response.getWriter().println(new Date().toLocaleString());

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

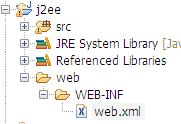
}

}

}

### 4、配置web.xml

首先在j2ee下创建目录web  
接着再创建目录web/WEB-INF  
然后在WEB-INF目录中创建 ****web.xml****  
  
web.xml提供路径与servlet的映射关系  
把/hello这个路径，映射到 HelloServlet这个类上  
  
<servlet> 标签下的 <servlet-name>  
与 <servlet-mapping> 标签下的 <servlet-name> 必须一样  
  
<servlet-name>与<servlet-class>可以不一样，但是为了便于理解与维护，一般都会写的一样。 一目了然



<?xml version="1.0" encoding="UTF-8"?>

<web-app>

<servlet>

<servlet-name>HelloServlet</servlet-name>

<servlet-class>HelloServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HelloServlet</servlet-name>

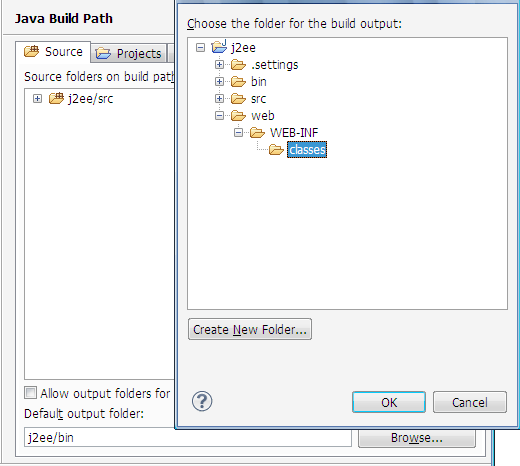
<url-pattern>/hello</url-pattern>

</servlet-mapping>

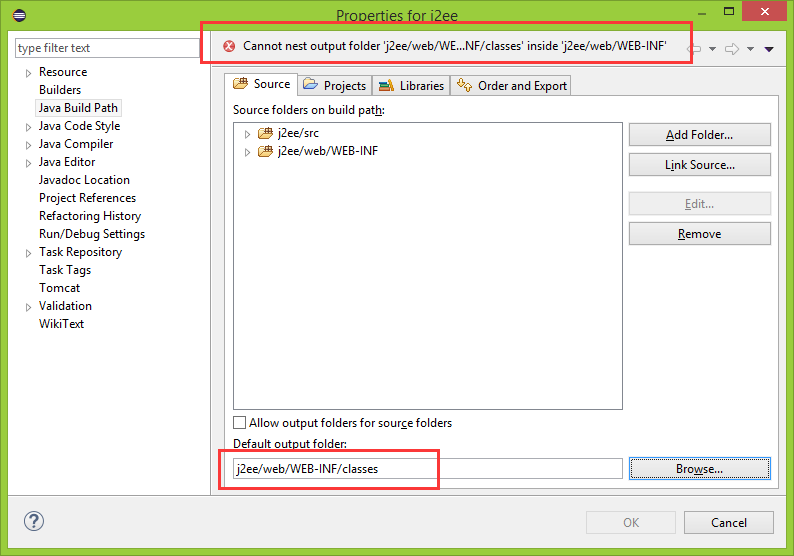
</web-app>

### 5、指定项目输出到classes目录

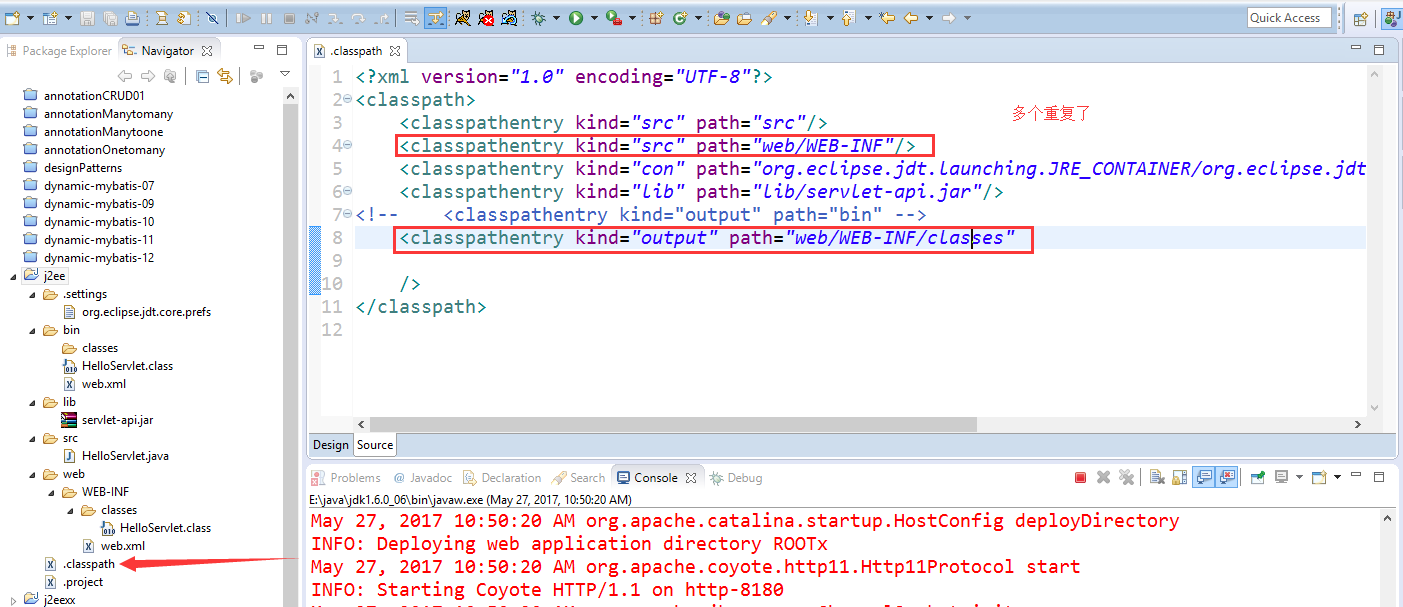
首先在WEB-INF下创建classes目录  
  
把项目的class文件输出由原来的 j2ee/bin 设置到 j2ee/web/WEB-INF/classes下  
  
步骤:   
项目右键->properties->Java Build Path->Source->右下角的 Brower-> 指定位置是 j2ee/web/WEB-INF/classes。  
  
****为什么要有这一步？**** 在ecilpse中默认输出的class是在bin目录下，但是tomcat启动之后，在默认情况下，不会去bin目录找这些class文件，而是到WEB-INF/classes这个目录下去寻找。 所以通过这一步的配置，使得eclipse的class文件输出到WEB-INF/classes目录下，那么这样就和tomcat兼容了。



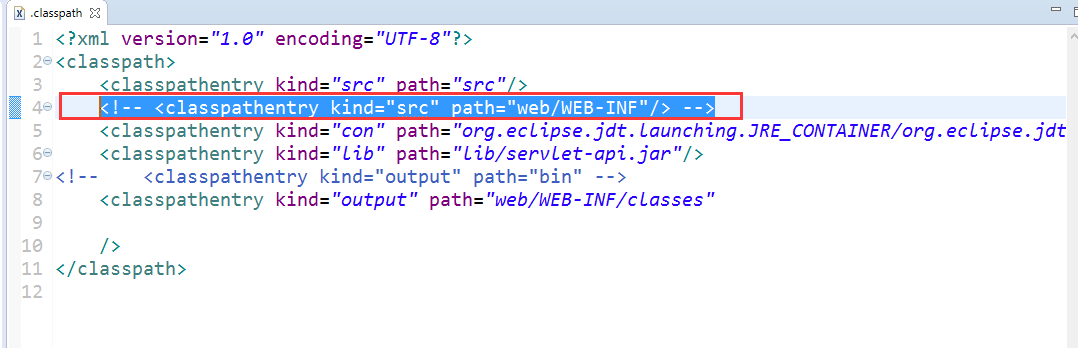
如果：报错，不能指定



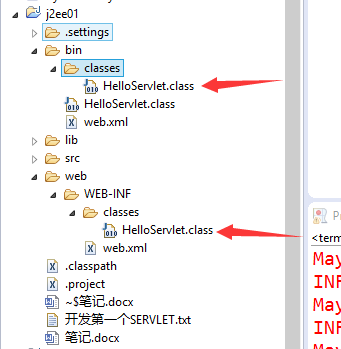
则：是因为：



注释掉一个就OK了：



或把 bin->classes下的xxx.class文件剪切到web->WEB-INF->classes下：



### 6、配置tomcat的server.xml 中的<context

Tomcat 请使用 [80端口的，纯净版tomcat](http://how2j.cn/k/tomcat/tomcat-download/1130.html" \l "step4455)

不是80端口的，默认是8080，此时访问时不能省略8080端口号  
  
为了把 j2ee/web 这个目录部署到tomcat中，进行如下操作：  
  
打开tomcat/conf/sever.xml  
  
在 <Host name="localhost" 。。。 下面加一句

<Context path="/" docBase="e:\\project\\j2ee\\web" debug="0" reloadable="false" />

path="/" 就表示直接通过 http://127.0.0.1/hello 就可以访问网页了。  
  
如果设置为

<Context path="/j2ee" docBase="e:\\project\\j2ee\\web" debug="0" reloadable="false" />

那么表示需要通过 http://127.0.0.1/****j2ee****/hello才能访问呢  
  
  
如果这一步老是失败，请直接在右侧下载修改好的server.xml

<Context path="/" docBase="e:\\project\\j2ee\\web" debug="0" reloadable="false" />

### 7、删除tomcat webapps下的ROOT目录

在上一步部署web的时候server.xml中的path 配置为 "/" 与 webapps下的ROOT目录冲突了  
  
所以要****删除ROOT目录****

### 8、重启tomcat，访问http://127.0.0.1/hello

关闭tomcat后，启动startup.bat 访问下面地址就可以看到servlet的输出

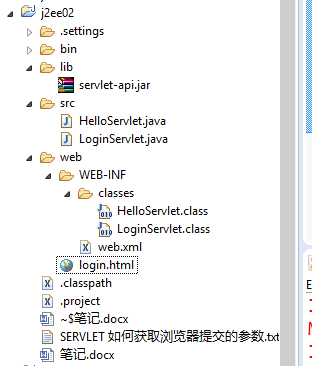
http://127.0.0.1/hello

并且每一次刷新，都能看到输出新的时间

## 获取参数

通过登录行为，演示servlet如何获取从浏览器提交的账号密码

### 0、整体项目目录结构



### 1、创建 login.html

在web上右键 -> New ->File  
创建一个 login.html文件  
然后添加一个form元素  
action="login" 标题会提交到login路径，login路径在后续步骤会映射到LoginServlet  
method="post" post方式表示提交的密码信息在浏览器地址栏看不到  
  
接着准备账号和密码的input元素  
因为要提交两个数据，在servlet端为了区分哪个是账号，哪个是密码，要给这两个input元素的name属性分别叫做name和password。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>登录页面</title>

</head>

<body>

<form action="login" method="post">

账号: <input type="text" name="name"> <br>

密码: <input type="password" name="password"> <br>

<input type="submit" value="登录">

</form>

</body>

</html>

### 创建 LoginServlet

创建一个LoginServlet  
  
因为浏览器中的****form的method是post****,所以LoginServlet需要提供一个****doPost****方法  
  
在doPost方法中，通过****request.getParamter****根据name取出对应的账号和密码  
  
然后用System.out.println() 打印在控制台  
  
****注**** 这里是打印在控制台，并没有在网页上输出

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String name = request.getParameter("name");

String password = request.getParameter("password");

System.out.println("name:" + name);

System.out.println("password:" + password);

}

}

### 映射LoginServlet到路径login

在web.xml中新增映射

<?xml version="1.0" encoding="UTF-8"?>

<web-app>

<servlet>

<servlet-name>HelloServlet</servlet-name>

<servlet-class>HelloServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HelloServlet</servlet-name>

<url-pattern>/hello</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>LoginServlet</servlet-name>

<servlet-class>LoginServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>LoginServlet</servlet-name>

<url-pattern>/login</url-pattern>

</servlet-mapping>

</web-app>

### 在页面提交数据

****首先重启tomcat****  
然后访问页面

http://127.0.0.1/login.html

输入账号密码，提交  
然后****在tomcat的窗口****，就可以看到提交的账号和密码了。

### 出现的问题：HTTP Status 404 (The requested resource is not available)

原来字母前面有一个空格符。试着去掉**空格**，然后运行，正常了！

一个英文页面里面提到的解决方法就是反复强调检查拼写。

注意检查拼写，还有文件名。切记！切记！

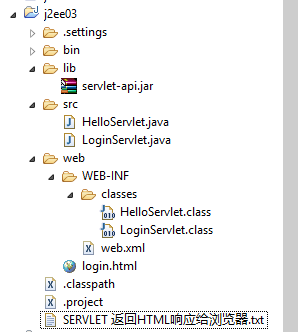
## 三.返回响应

根据浏览器提交的账号密码返回登录成功或者失败

这一步本来应该通过访问数据库来实现，这里简化一下，直接在内存中进行校验

如果账号是 admin,密码是123, 就返回登录成功，否则返回登录失败

### 整体项目目录结构



### 1、LoginServlet

判断账号密码是否为 admin 123，如果是就打印  
success 否则就打印 fail  
  
根据账号密码，创建对应的html字符串。  
  
然后通过response.getWriter().println() 发送到浏览器。  
  
  
****注:**** 比较的时候把常量字符串"admin" "123"放前面，因为用户可能没有输入账号密码就提交，servlet会获取到null。 这样就规避了空指针异常的问题。  
  
****注:**** 这里输出的响应是success和fail，不是中文。 如果使用中文浏览器有可能看到乱码。  
  
关于中文乱码问题，统一在[中文问题](http://how2j.cn/k/servlet/servlet-gbk/554.html)章节解决，咱们一个知识点一个知识点的掌握

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String name = request.getParameter("name");

String password = request.getParameter("password");

String html = null;

if ("admin".equals(name) && "123".equals(password))

html = "<div style='color:green'>success</div>";

else

html = "<div style='color:red'>fail</div>";

PrintWriter pw = response.getWriter();

pw.println(html);

}

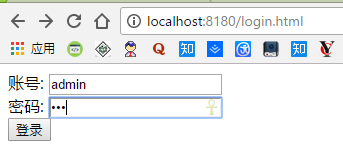
}

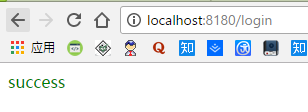
### 重启tomcat，提交数据

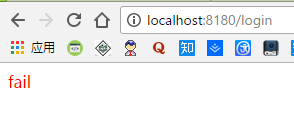
访问页面

http://127.0.0.1/login.html

http://localhost:8180/login.html





  
  
账号输入admin  
密码输入123  
提交可以看到success