# day17 Servlet

## 一.Web服务器(容器)

Tomcat、GlassFish、WebLogic、JBoss、WebSphere、Jetty、JRun等.

以上这些服务器都是”容器”,是用来装载Web程序的,这些服务器给我们写的Web程序提供了一个运行的环境.

## 二.Tomcat环境搭建

### 1.安装Tomcat服务器:

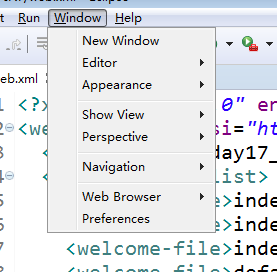
Tomcat服务器分为安装版和绿色版.

注意安装版Tomcat服务器端口号需要是8080.

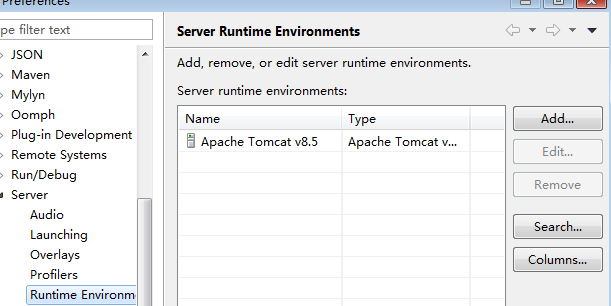
### 2.将Eclipse与Tomcat关联在一起:

配置步骤:

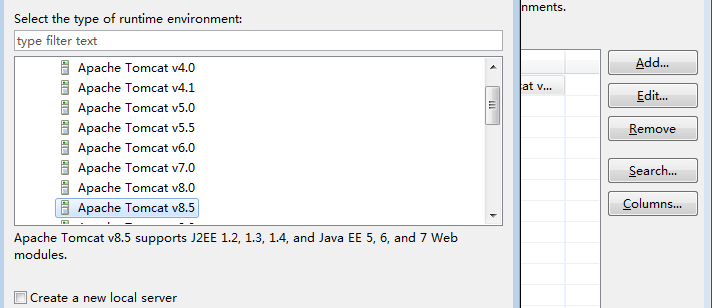
#### .Window-Preferences:

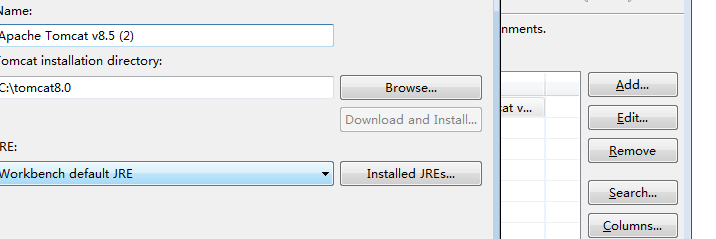


#### .Server-Runtime Environment:



#### .Add一个服务器,选择一个合适的Tomcat版本.





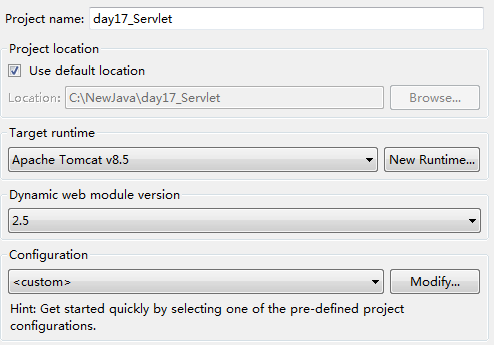
然后一路下一步,这样Tomcat服务器与Eclipse就建立了关联.

## 三.项目创建

### 1.创建Web Dynamic Project:

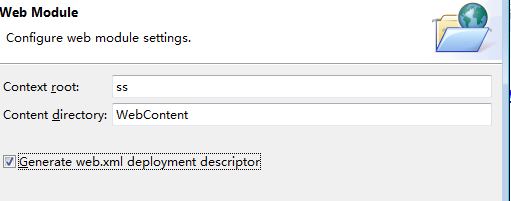
**重点注意:**

Dynamic Web model Version: 2.5,不是3.1

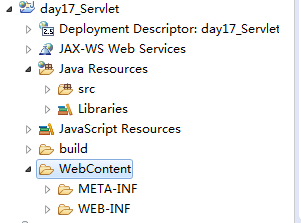


### 2.一路下一步,在该页面注意:

勾选checkbox,Content路径名称为WebContent.



### 3.最终形成的目录结构:



### 4.整个项目组成:

Java Resources:存放Java代码的文件夹;

WebContent:存放前端和lib类库的文件夹;

build:存放编译的classes字节码的文件夹;

JavaScript Resources:js类库

## 四.Servlet

### 1.概念:

Java Servlet就是一个Java类,它是运行在Web服务器上的一个类,它可以被认为是Web浏览器与Web服务器之间的中间层.

Servlet可以浏览器传递过来的请求交给Tomcat服务器;

同时还可以把Tomcat服务器返回的响应给浏览器.

### 2.优势:

①.可以实现跨平台-----Java jvm;

②.依赖于服务器所在的进程,Servlet自己并没有单独开进程,这样就比较节省资源;

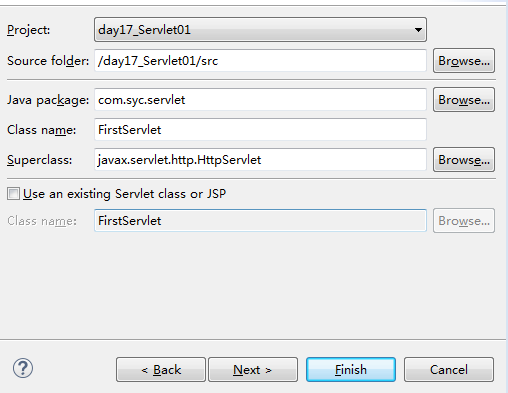
1. .性能更好,相对于CGI(通用网关接口)-----B/S;
2. .Java的所有类库,在Servlet基本都能用.

### 3.Servlet依赖的包

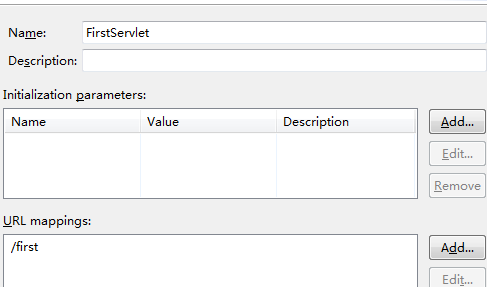
javax.servlet;

javax.servlet.http;

### 4.Servlet创建



编辑Servlet的url:



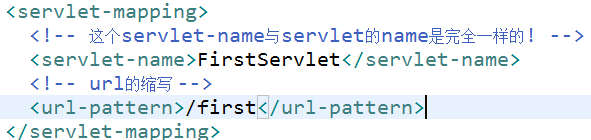
### 5.Servlet执行流程

当创建了一个Servlet,在浏览器中进行访问该Servlet.整个的流程:

1. . <http://localhost:8080/day17_Servlet01/first>

首先浏览器根据http请求协议,调用浏览器引擎,解析整个url.解析出来localhost,这个localhost代表了Tomcat服务器的地址,8080是Tomcat的端口号.接下来就会去Tomcat服务器内部去找day17\_Servlet01项目;一旦找到了项目,接着就会去匹配first这个资源.

1. .根据这个first资源名称,Tomcat服务器就会加载,就会执行web.xml文件.在这个web.xml文件中会遍历所有的servlet-mapping,看看有没有url-pattern是first.



1. .一旦找到了first这个url-pattern,然后就会解析对应的servlet-name.------🡪FirstServlet.
2. .此时Tomcat服务器会根据找到的这个FirstServlet,去遍历所有的servlet节点.



要找到name是FirstServlet的servlet节点.

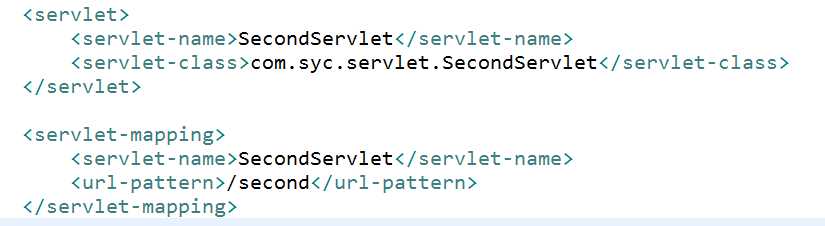
1. .如果找到了,Tomcat服务器就解析出来servlet-name对应的servlet-class的值.此时的值为:com.syc.servlet.FirstServlet.
2. .Tomcat服务器再根据这个class字符串,利用反射的

Class clazz=Class.forName(“com.syc.servlet.FirstServlet”);

得到类的字节码执行,执行一些列的后续方法.

比如执行service()方法---🡪service会根据请求的具体类型,调用对应的get()或者post()或者其他方法.

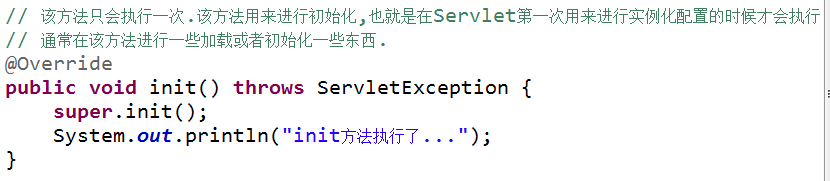
### 6.web.xml文件中servlet的命名规则



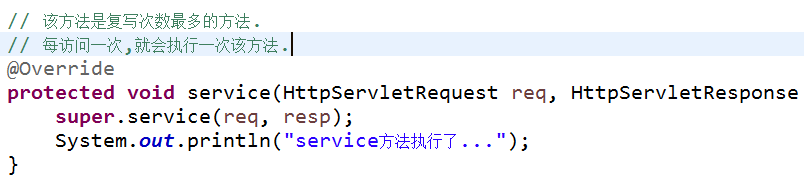
1. .有一个servlet就有一个对应的servlet-mapping节点.
2. .servlet与servlet-mapping中的servlet-name必须一致!
3. .servlet-name的值默认是类名.该名称是可以随便写的.
4. .url-pattern中的值,前面要带一个/.

## 五.Servlet生命周期

### 1.init()



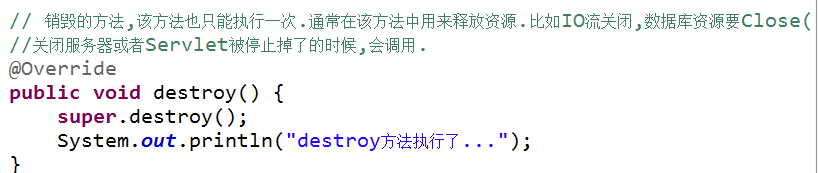
### 2.service(request,response)



service()方法是执行实际业务的方法.

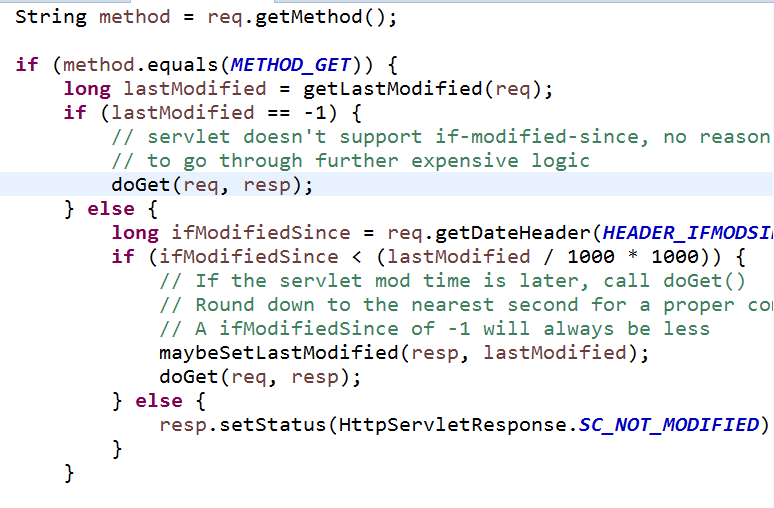
service()方法会根据具体的请求类型,去调用具体的处理方法.最常用的doGet()和doPost().

### 3.destroy()



建议:没事看看简单的Java源码.

Service()方法的源码实现:



## 六.Servlet中web.xml的常用节点

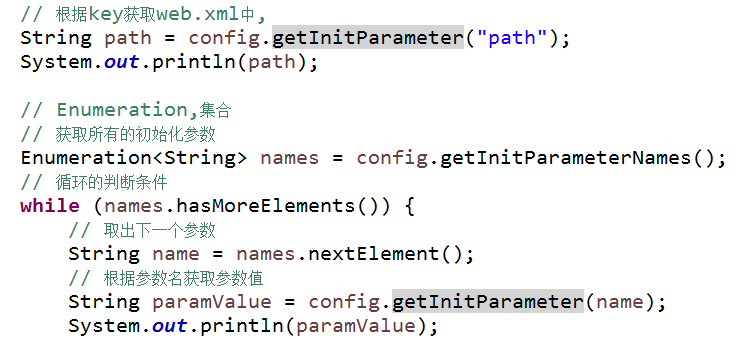
### 1.init-param:主要是用来配置一些初始化的参数.



### 2.load-on-startup:配置Servlet的启动顺序.



### 3.ServletConfig类



## 七.Servlet的线程安全问题

案例:统计网页访问次数

### 1.Servlet的线程

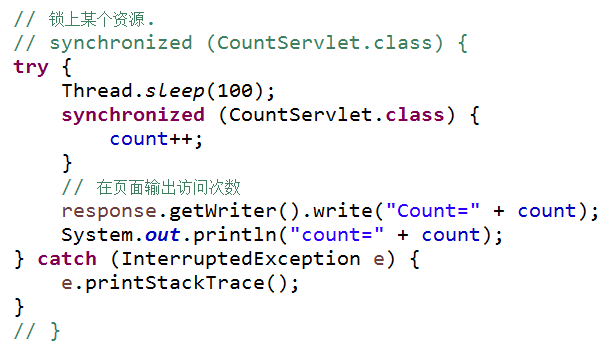
在Servlet中,每访问一次Servlet,就相当于是开了一个线程.

service()每执行一次,就相当于是开了一个线程.

也就相当于doGet()每执行一次,就开了一个线程.

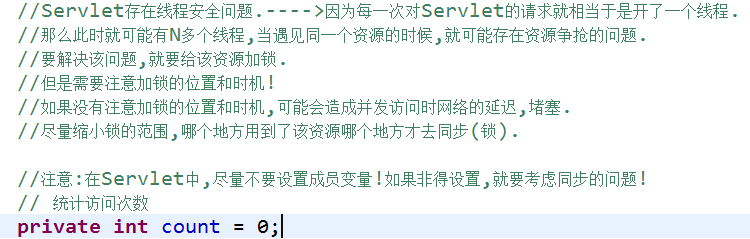
所以,在Servlet的并发访问中就可能存在资源争抢的问题.

### 2.解决办法:给被争抢的资源加锁.



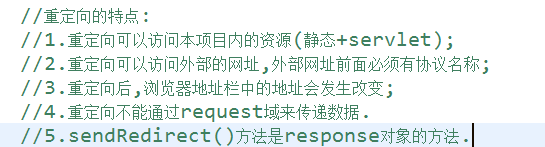
但要注意同步锁的范围,尽可能的缩小范围!

### 3.Servlet线程总结



## 八.Servlet的跳转

### 1.重定向



具体代码:



### 2.请求转发



具体代码:



### 3.如何传递和获取信息:

存储信息:



获取信息:



### 4.response对象的输出内容

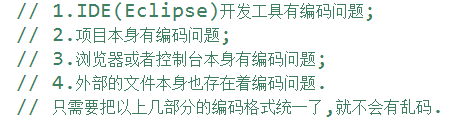
response对象不仅可以输出字符串,还可以输出各种HTML标签.

## 九.解决Servlet中文乱码问题

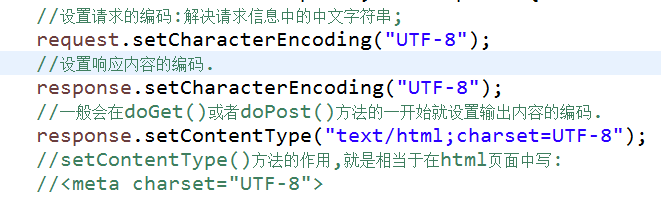
### 1.乱码出现的原因



### 2.涉及到编码的几个地方



### 3.解决方法



作业:

模拟一个登陆页面:

用户名;

密码;

验证用户名密码是否正确?

将用户名+密码信息封装---🡪User