

### 1. Servlet简介

Servlet是Sun公司提供的一门动态资源开发的技术。本质就是一个java文件。将编译后的文件放入servlet容器中方可使用servlet提供服务。

servlet容器：存储并运行servlet的环境，称之为servlet容器。-- tomcat

web容器：存储并运行web资源的环境，称之为web容器。-- tomcat

### 2. 手写servlet：

- a. 编写一个类，让这个类实现servlet接口，那么这个类就会变为一个servlet文件
- b. 编译写好的文件，并且将其加入到servlet容器中去运行。
- c. 将servlet文件添加到任意一个web应用的classes目录中。

### 3. 配置servlet映射：

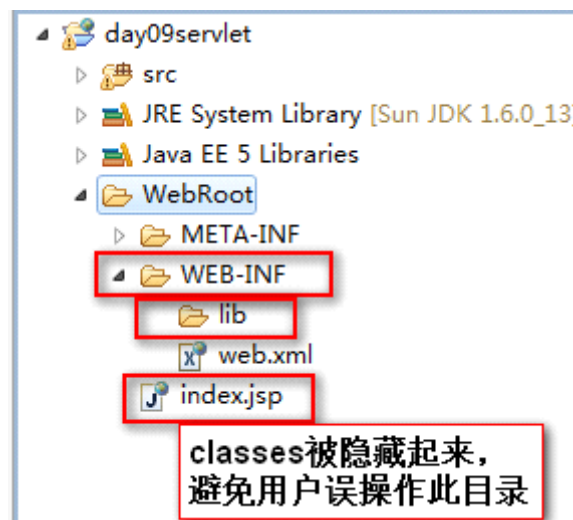
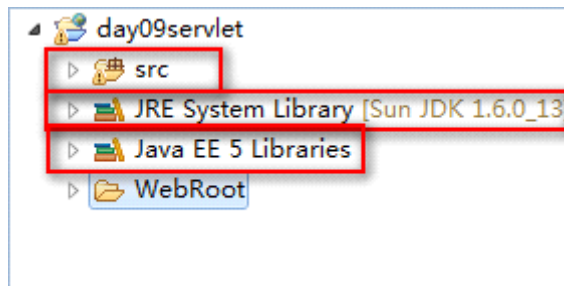
- a. 在添加servlet文件的web应用中配置web.xml文件，内容如下：

```
<servlet>
  <servlet-name>FirstServlet</servlet-name>
  <servlet-class>cn.tedu.servlet.FirstServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>FirstServlet</servlet-name>
  <url-pattern>/servlet/FirstServlet</url-pattern>
</servlet-mapping>
```

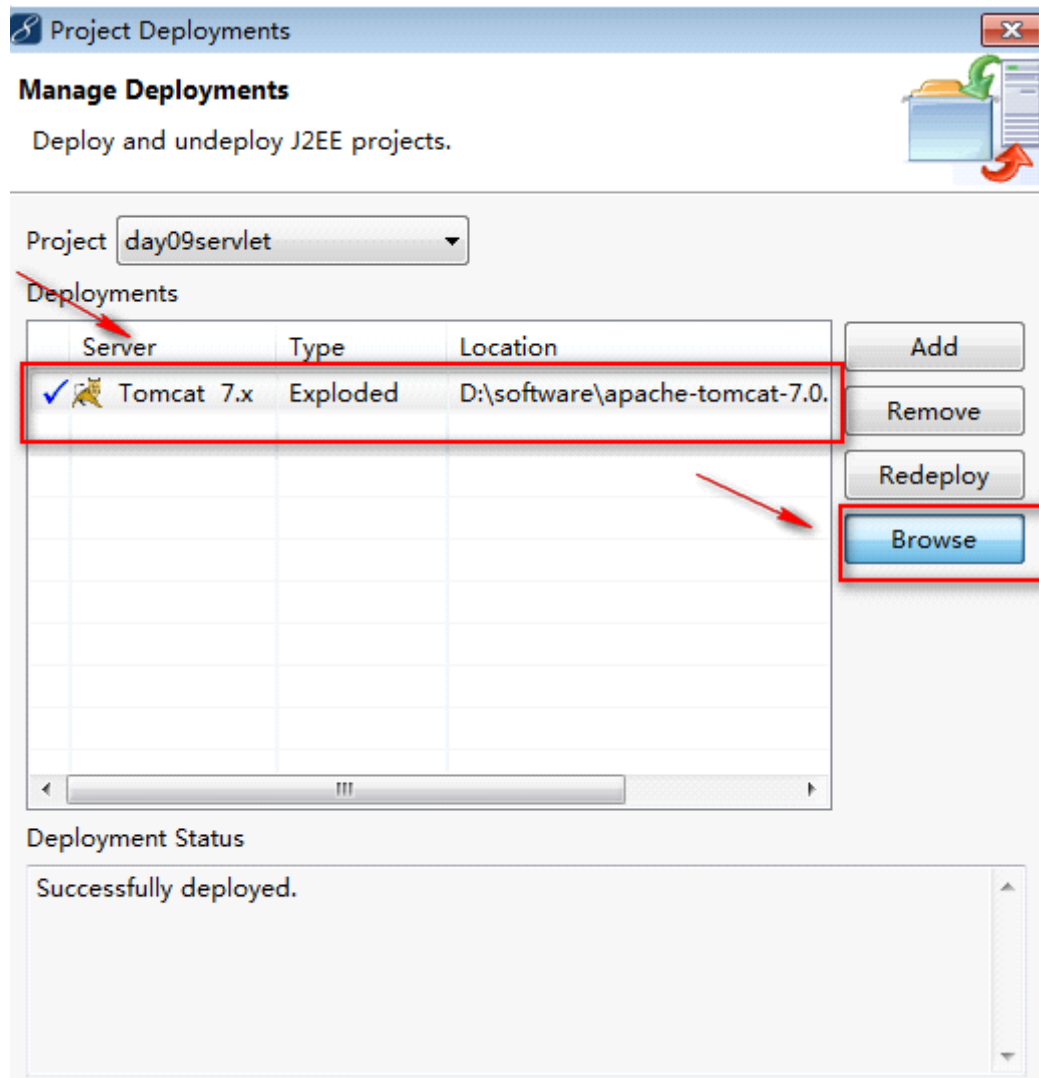
# web应用目录结构

2019年5月15日 星期三 15:02

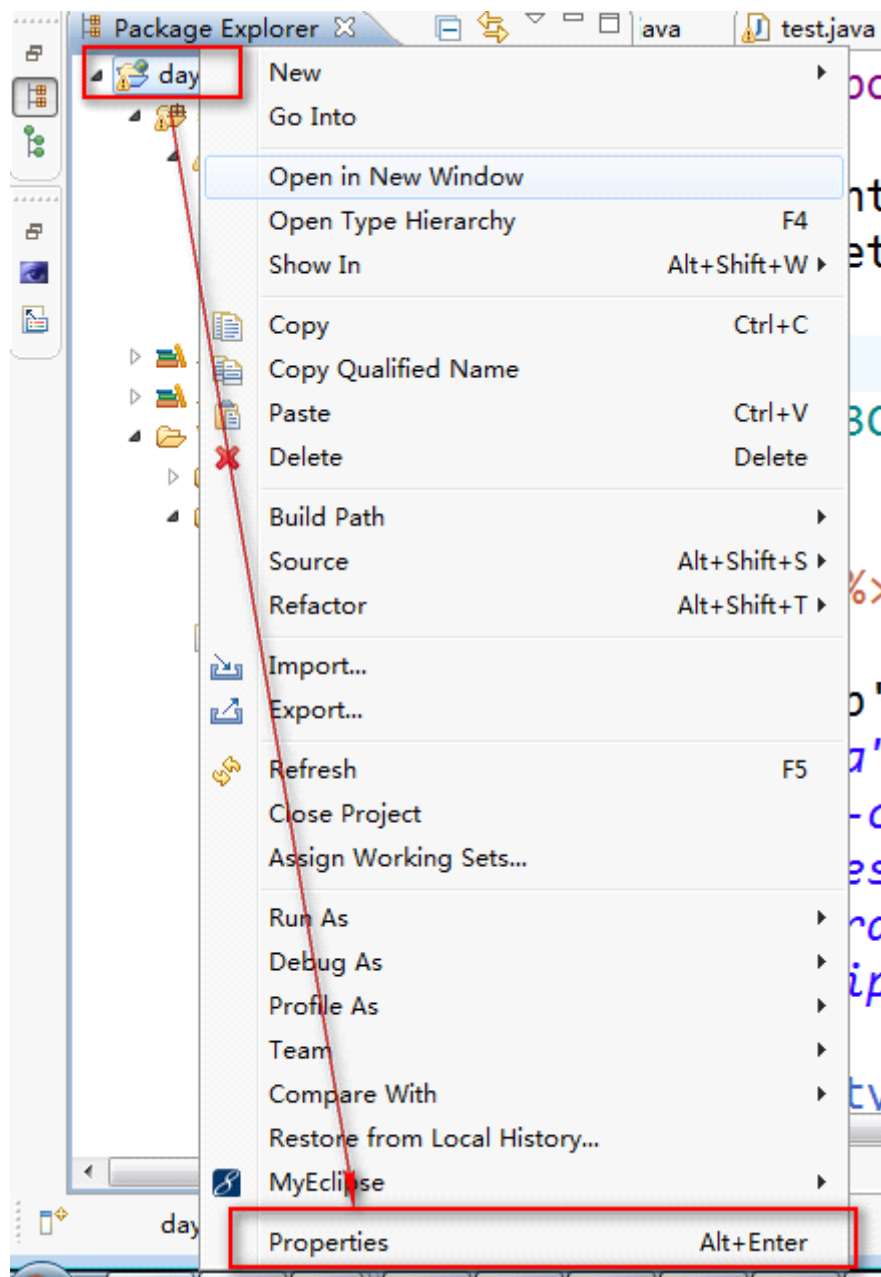
## 1. 目录结构

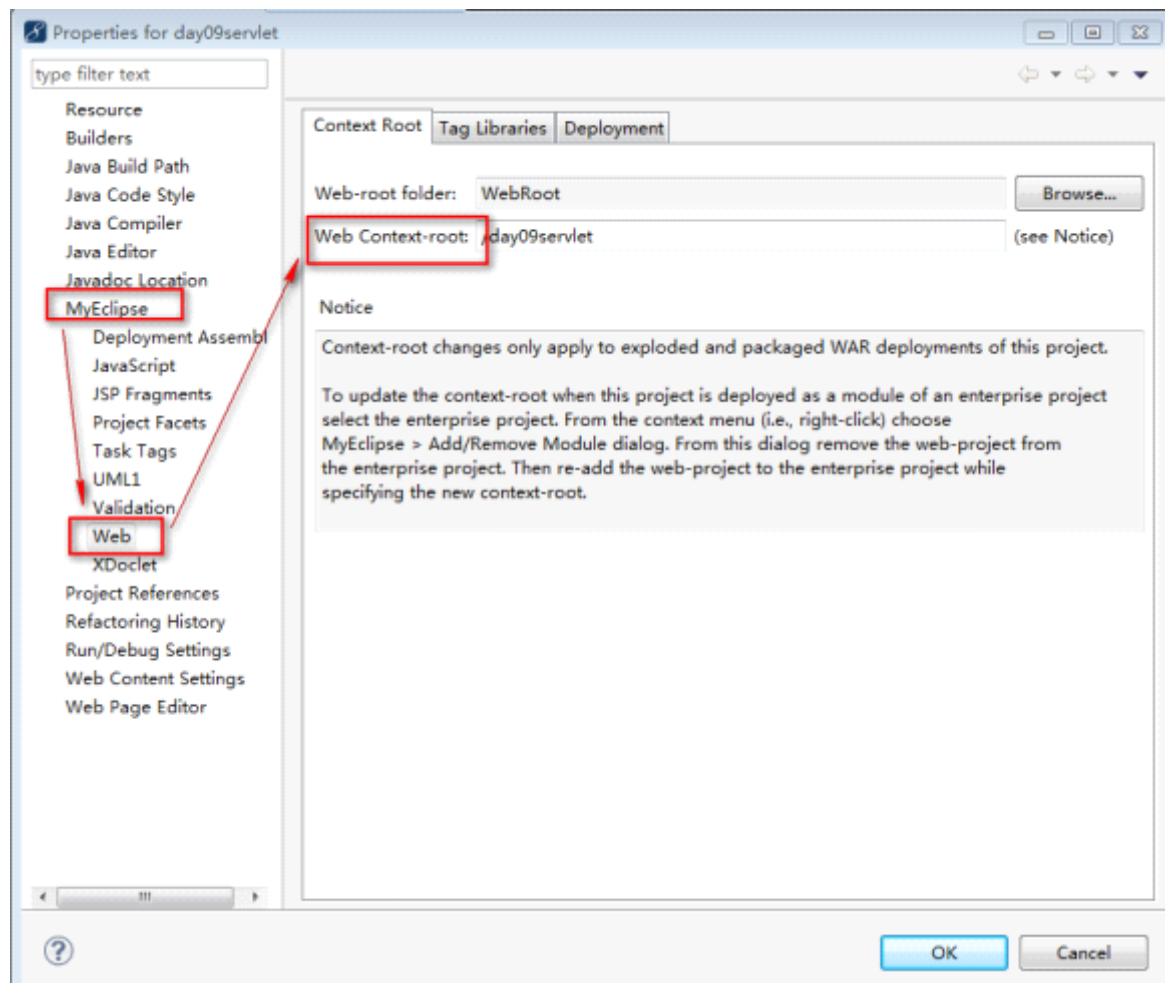


## 2. classes文件存储位置



### 3. 发布时的web应用虚拟路径设置

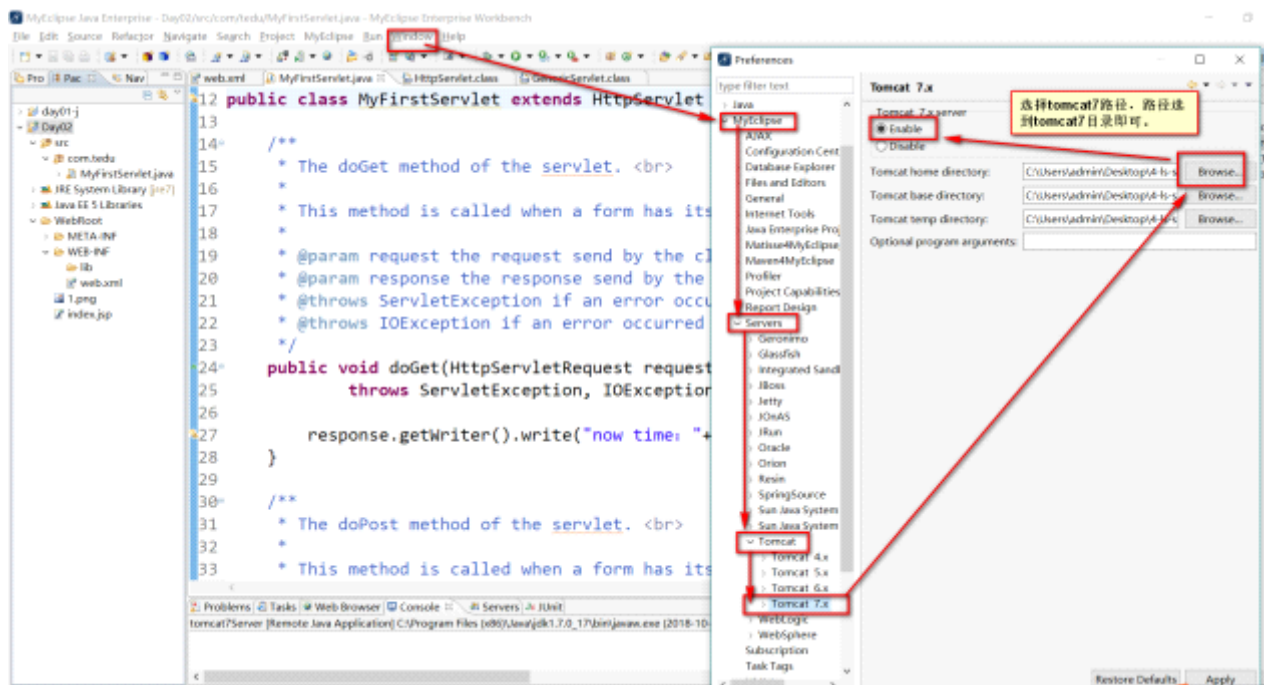




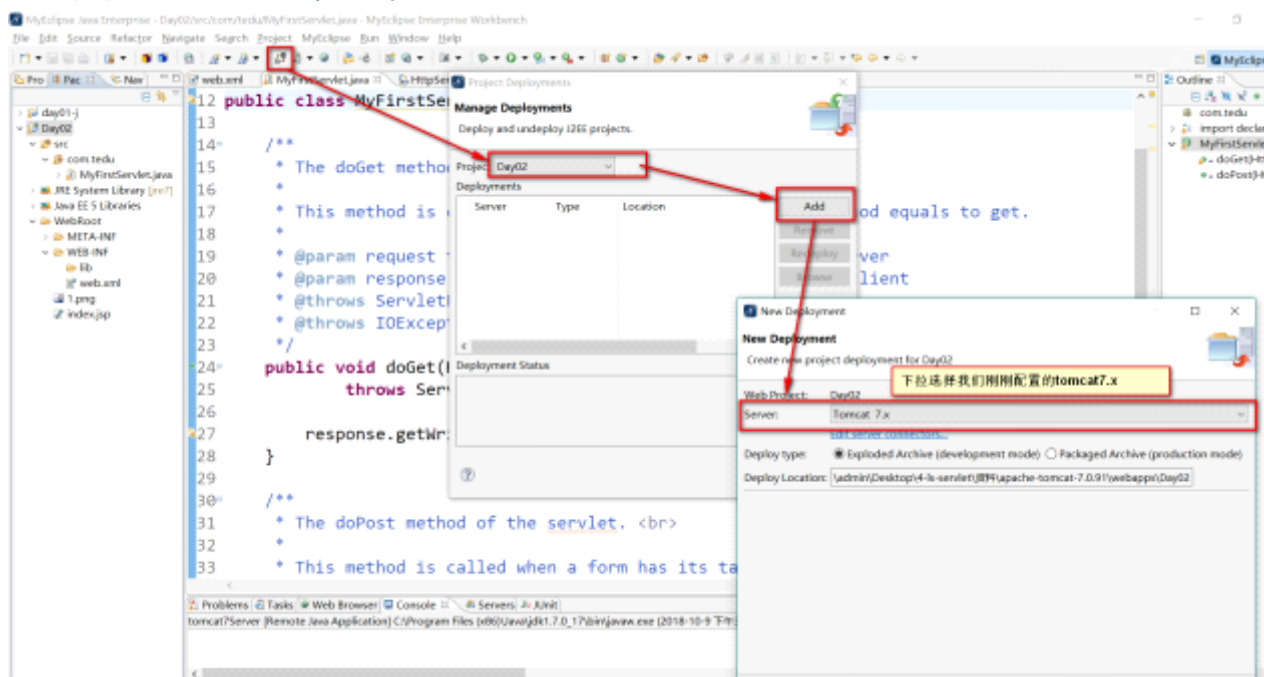
## tomcat配置与web应用的部署

2019年5月15日 星期三 17:13

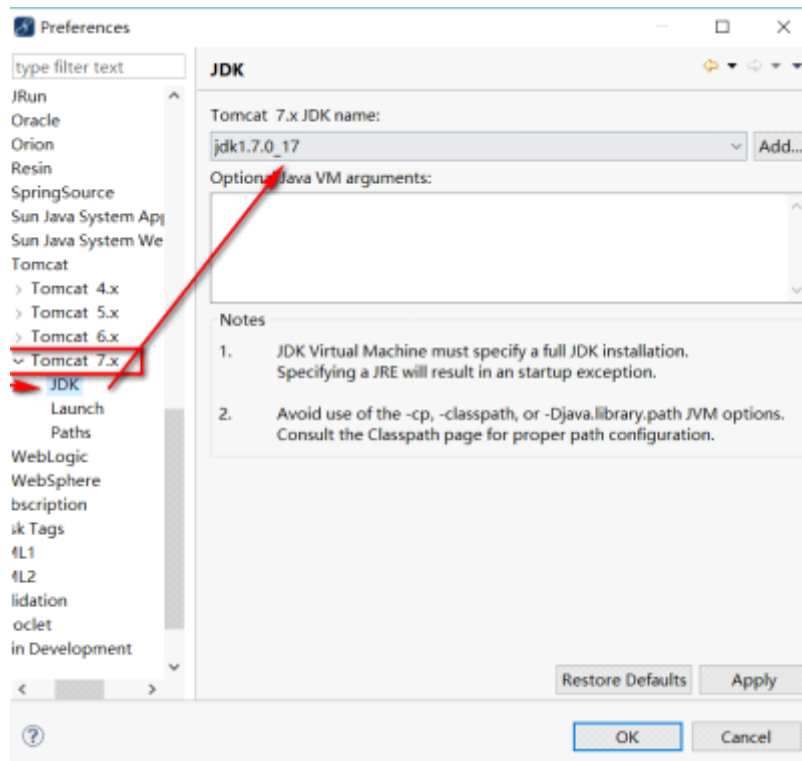
### a. 在MyEclipse中配置tomcat7



### b. 部署web应用至Tomcat服务器中

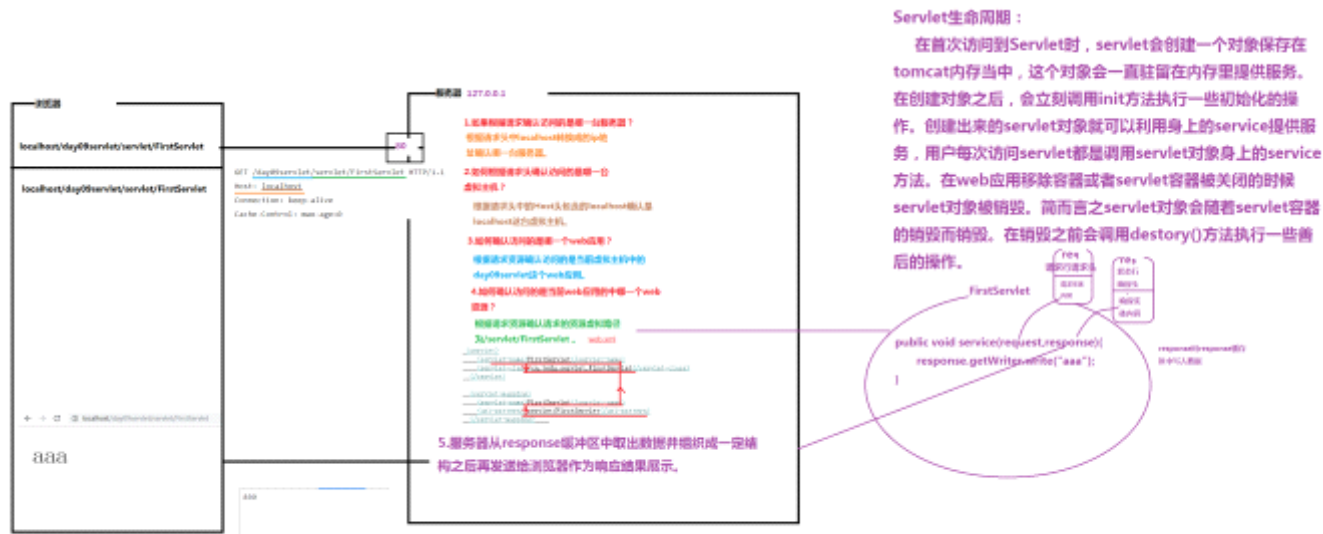


如果此时无法完成部署，检查jdk，使用1.6版本即可。



1. Servlet调用过程和生命周期

- a. 确认ip地址对应的服务器。
- b. 确认服务器中的虚拟主机名称
- c. 确认虚拟主机中对应的web应用。
- d. 确认web应用中的资源虚拟路径。
- e. 根据web.xml文件寻找对应的真实Servlet。
- f. servlet执行生命周期的过程。
- g. servlet中的service方法提供服务，将所有数据写入response缓冲区。
- h. 服务器将response缓冲区中的数据组织成一定结构后响应给浏览器。



1. Servlet继承结构：

Servlet --- 包含五个最基本的方法 init() destroy() service()  
|  
|---GenericServlet --- service() 通用servlet  
|  
|--- HttpServlet --- 在通用Servlet基础上，添加了包含大量和HTTP协议相关的方法。

2. 图片如何被加载？

news  
|--- 1.jpg  
localhost/news/1.jpg  
localhost/news/servlet/FirstServlet

注意：所有的静态资源都是由缺省servlet--defaultServlet加载而来。

3. 拓展：request对象和response对象

a. 接口为什么可以创建对象？

HttpServletRequest是一个接口，为什么request对象依然可以用来发送请求数据？HttpServletResponse是一个接口，为什么response对象依然可以用来发送响应数据？

原因：真正创建对象的并不是这两个接口而是这两个接口下的子实现

类。HttpServletRequestWrapper HttpServletResponseWrapper两个子实现类创建了request和response对象。



b. request对象和response对象何时创建？

servlet完成初始化操作之后，service方法身上的request和response对象也会被创建出来。

## 1. Servlet标签问题

```
<servlet>
  <servlet-name></servlet-name>
  <servlet-class>cn.tedu.FirstServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name></servlet-name>
  <url-pattern>/servlet/FirstServlet1</url-pattern>
</servlet-mapping>
```

- a. <servlet>注册servlet标签
- b. <servlet-mapping>servlet映射标签。
- c. 标签由两部分组成的原因：为了便于多个不同的映射路径映射到同一个servlet身上。这种方式只需要重写servlet-mapping部分配置并修改url-pattern即可。

## 2. servlet映射路径通配形式

如果有大量相似的路径都映射到同一个servlet身上，多次重写servlet-mapping配置代码量仍然很大，这是可以使用servlet映射中通配的形式来定义url-pattern.

- a. 定义方式：
  - i. \*.后缀。
  - ii. 以"/"开头以"/"结束

*.do案例：	以"/"开头以"/"结束：案例
<pre>&lt;servlet-mapping&gt;   &lt;servlet-name&gt;     FirstServlet   &lt;/servlet-name&gt;   &lt;url-pattern&gt;*.do&lt;/url-pattern&gt; &lt;/servlet-mapping&gt;</pre>	<pre>&lt;servlet-mapping&gt;   &lt;servlet-name&gt;     FirstServlet   &lt;/servlet-name&gt;   &lt;url-pattern&gt;/servlet/*&lt;/url-pattern&gt; &lt;/servlet-mapping&gt;</pre>

## 3. Servlet的一些细节(3)

- 对于如下的一些映射关系：
  - Servlet1 映射到 /abc/\*
  - Servlet2 映射到 /\*
  - Servlet3 映射到 /abc
  - Servlet4 映射到 \*.do （永远匹配级最低）
- 问题：
  - 当请求URL为 "/abc/a.html" ， "/abc/\*" 和 "/" 都匹配，哪个servlet响应
    - Servlet引擎将调用Servlet1。
  - 当请求URL为 "/abc" 时， "/abc/\*" 和 "/abc" 都匹配，哪个servlet响应
    - Servlet引擎将调用Servlet3。
  - 当请求URL为 "/a/a.do" 时， "/a/\*" 和 "\*.do" 都匹配，哪个servlet响应

- Servlet引擎将调用Servlet1。
- 当请求URL为 “/a.do” 时，“/\*” 和 “\*.do” 都匹配，哪个servlet响应
  - Servlet引擎将调用Servlet2。
- 当请求URL为 “/xxx/yyy/a.do” 时，“/\*” 和 “\*.do” 都匹配，哪个servlet响应
  - Servlet引擎将调用Servlet2。

url-pattern与通配的匹配程度越高越有限加载。\*.do匹配级别永远最低。

#### 4. Servlet的一些细节(4)

- a. Servlet是一个供其他Java程序（Servlet引擎）调用的Java类，它不能独立运行，它的运行完全由Servlet引擎来控制 and 调度。
- b. 针对客户端的多次Servlet请求，通常情况下，服务器只会创建一个Servlet实例对象，也就是说Servlet实例对象一旦创建，它就会驻留在内存中，为后续的其他请求服务，直至web容器退出，servlet实例对象才会销毁。
- c. 在Servlet的整个生命周期内，Servlet的init方法只被调用一次。而对一个Servlet的每次访问请求都导致Servlet引擎调用一次servlet的service方法。对于每次访问请求，Servlet引擎都会创建一个新的HttpServletRequest请求对象和一个新的HttpServletResponse响应对象，然后将这两个对象作为参数传递给它调用的Servlet的service()方法，service方法再根据请求方式分别调用doXXX方法。
- d. 在servlet被访问到时会创建一个servlet对象驻留在内存中，如果有多个servlet资源被访问，则每个servlet 在被初次访问的时候都会创建一个各自的servlet对象保存在tomcat内存当中。

#### 5. Load-on-startup标签

```
<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>>false</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

- a. 如果一个servlet标签中配置了load-on-startup标签，则在tomcat服务器启动的时候就会自动访问一次这个servlet，让这个servlet的对象驻留在内存中一份，为后边的请求提供访问。
- b. load-on-startup标签中间可以配置一个数字，这个数字可以采用0及其以上的数字，一般采用1及其以上的数字。如果这个数字越小，则优先级越高，如果数字越大则优先级越低。注意：尽管各个servlet之间存在优先级关系，但是仍然都是随着tomcat服务器的启动而加载。

#### 6. 缺省servlet

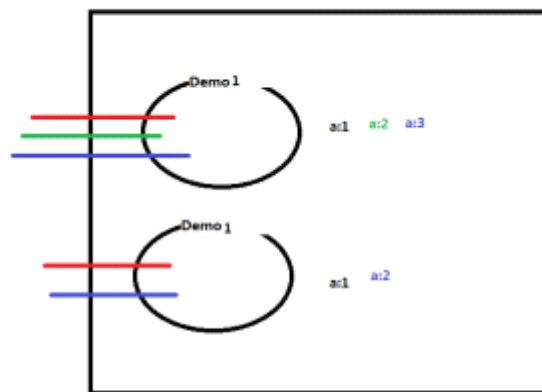
- a. 如果某个Servlet的映射路径仅仅为一个正斜杠 (/)，那么这个Servlet就成为当前Web应用程序的缺省Servlet。
- b. 凡是在web.xml文件中找不到匹配的<servlet-mapping>元素的URL，它们的访问请

求都将交给缺省Servlet处理，也就是说，缺省Servlet用于处理所有其他Servlet都不处理的访问请求。

- c. 在<tomcat的安装目录>\conf\web.xml文件中，注册了一个名称为org.apache.catalina.servlets.DefaultServlet的Servlet，并将这个Servlet设置为了缺省Servlet。
- d. 当访问Tomcat服务器中的某个静态HTML文件和图片时，实际上是在访问这个缺省Servlet。
- e. 如果访问的资源虚拟路径不存在，依然是交给缺省servlet处理。

## 7. servlet多线程并发安全的问题

- a. 在面对局部变量的时候，所有线程操作的都是各自独立的变量，这些变量之间是没有联系的，相互不产生影响。
- b. 面对全局变量的时候存在线程并发安全问题，这个问题可以通过加锁来解决，但是添加锁的时候要注意：锁对象一定要统一。锁住部分的代码一定尽可能的少，只锁住必要部分代码，如果锁住过多的代码则会导致程序的运行效率。
- c. 尝试利用sun公司提供的SingThreadModel接口解决多线程并发安全问题。发现一个servlet实现这个接口之后，面对多并发的時候会在tomcat内存中保留多个相同servlet的对象，这些对象会同时为所有的用户提供服务，所以导致了用户初次访问闲置的servlet提供服务的线程，按照接口思路应该为每一个新的用户都提供一个新的servlet对象，但是结果表明事实是调用内存中已有的一个servlet对象提供的服务。显然结果不正确，当前接口如果被实现会导致线程安全问题更加复杂，不要使用当前接口。



### d. 解决线程安全问题：

- i. 加锁：合理添加锁，锁住关键部分代码，尽可能少的降低代码的执行效率。
- ii. 尽可能多定义局部变量少定义全局变量，因为全局变量不可避免的会出现多线程并发安全问题。

