一，servlet初体验

在新建的webproject项目下编写一个servlet的话需要俩步：
1，编写一个java类实现servlet接口，使用关键字implements servlet
2，修改web.xml文件，给servlet提供一个可访问的URL地址
3，部署应用到Tomcat服务器
4，访问：http://localhost:8080/项目名称/xml中写的外部访问servlet的名称

新建项目的时候没有自带web.xml文件的解决办法：
https://blog.csdn.net/double_sweet1/article/details/80955435

复制servlet-class路径：点开对应的java文件，找到对应的构造函数，右键copy qui....Name，复制到servlet-class标签内部

验证servlet-class路径写得对不对：在servlet-class标签体内部按住Ctrl键，点击标签内容看能不能跳到对应的servlet文件

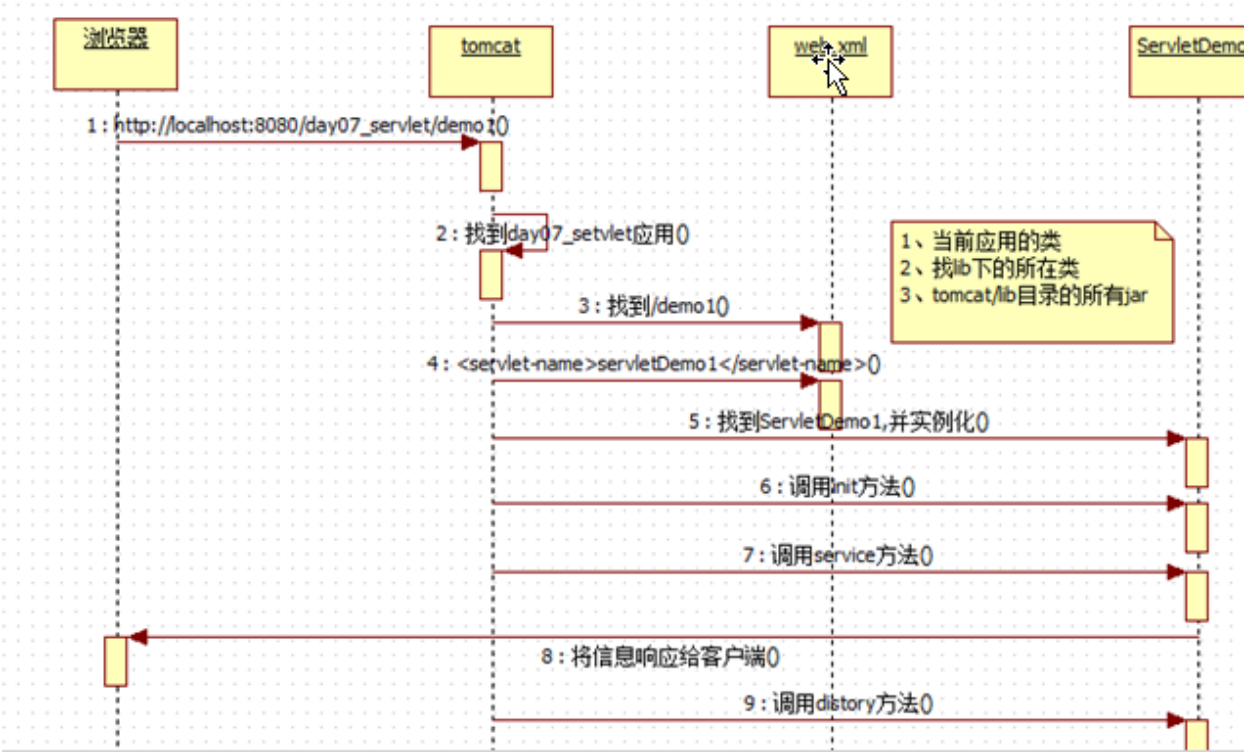
在web.xml中写入：

```
  <!-- 创建servlet实例 -->
  <servlet>
  <servlet-name>test1</servlet-name>-----实例化java类
  <servlet-class>
newS.test  -----java类的路径
</servlet-class>
  </servlet>

  <!-- 给servlet提供（映射）一个可供客户端访问的URI -->
  <servlet-mapping>
  <servlet-name>
  test1        -----项目名称，与servlet中servlet-name的值一致
  </servlet-name>
  <url-pattern>
  /demo      ------外部访问servlet的名称
  </url-pattern>
```

</servlet-mapping>

二，servlet的执行过程



当servlet类发生改变或者web.xml的文件发生改变的时候，就一定要重启服务器

小知识点：

如何让servlet在服务器启动时就创建：

<servlet>

<servlet-name>servletdemo</servlet-name>

<servlet-class>com.servlet.test</servlet-class>

<load-on-startup>2</load-on-startup>

</servlet>

三，servlet创建

创建servlet的三种方法

①实现javax.servlet.Servlet接口

②继承javax.servlet.GenericServlet类（适配器模式）

③继承javax.servlet.http.HttpServlet类（模板方法设计模式，模板设计模式实际上是多态）

service方法被doget和dopost分为俩部分，继承HttpServlet方法不能重写service方法

request.getRemoteAddr（）返回客户端访问服务器的IP地址

get和post提交请求出现乱码时的解决方式是不一样的

实际开发中使用第三种

servlet---->genericServlet---->HttpServlet---->继承HttpServlet

一个servlet可以配置多个映射路径，只要name一致就可以

servlet映射细节：通配符：*代表任意字符串

①url-pattern：*.do　　表示以*.字符串的请求都可以访问

②url-pattern：/*　　表示任意字符串都可以访问

③url-pattern：/action/*　表示以/action开头的请求都可以访问

匹配规则：

优先级：从高到低

绝对匹配---->/开头匹配---->扩展名方式匹配（后缀名）

如果url-pattern的值是/，表示执行默认映射，所有资源都是servlet

四，servlet的线程安全问题

①让其变为多实例，实现SingleThreadModel，

解决线程安全问题最佳的办法，不要写全局变量，而写局部变量

五，servlet获取配置信息

private　ServletConfig　config;

servletconfig的作用：一是可以获取servlet配置信息；二是获取servletcontext对象

①方式1

string s = config.getInitParameter（“配置文件标签名”）；获取配置文件中的信息

②方式2

string s1 = super.getServletConfig.getInitParameter（“配置文件标签名”）；获取配置文件中的信息

③方式3

string s2 = super.getInitParameter（“配置文件标签名”）；获取配置文件标签中的信息

六，servletcontext对象

servletcontext代表的是整个应用，一个应用只有一个servletcontext对象，单实例

作用：

①域对象：在一定范围内（当前应用），使多个servlet共享数据

setAttribute（string name，object value）向servletcontext对象的map中添加数据

getAttribute（string name）从servletcontext对象的map中取数据

removeAttribute（string name）根据name去移除数据

通过调用GenericServlet的getServletContext方法得到ServletConfig对象：

ServletContext application = this.getServletContext（）

②获取全局配置信息

③获取资源路径

a，string getRealPath(string path)根据资源名称得到资源的绝对路径

可以得到当前应用任何位置的任何资源

使用：

获取web-inf下的文件：/WEB-INF/文件名

获取src下面的文件：/WEB-INF/classes/文件名

获取src内包中的文件：/WEB-INF/classes/包名/文件名

string path = this.ServletContext（）.getRealPath（“/WEB-INF/文件路径”）；参数一定要以/开头

Properties pro = newProperties（）；

pro.load（new FileInputStream（path））；

System.out.println（pro.getProperties（“键名”））；//通过键名得到键值

实现servlet的请求转发：

```
//实现请求转发，目标是跳转到servletdemo1
ServletContext sc = this.getServletContext();
RequestDispatcher rd = sc.getRequestDispatcher("/servletdemo1");
rd.forward(request,response);将消息向下传递，传递完成后又回来了
```