

所有想要处理网络请求的类，都需要回答这三个问题：

- 你初始化时要做什么
- 你销毁时要做什么
- 你接受到请求时要做什么

servlet容器tomcat才是与客户端直接打交道，他监听了端口，请求过来后，根据url等信息，确定要将请求交给哪个servlet去处理，然后调用那个servlet的service方法，service方法返回一个response对象，tomcat再把这个response返回给客户端。

Servlet 生命周期

Servlet是单例的，整个服务器就只创建一个同类型Servlet。

- 启动服务器时加载Servlet的实例。
- Servlet 通过调用 **init ()** 方法进行初始化。
- Servlet 调用 **service()** 方法来处理客户端的请求。
- Servlet 通过调用 **destroy()** 方法终止（结束）。
- 最后，Servlet 是由 JVM 的垃圾回收器进行垃圾回收的。

可以在第一次接收请求时被创建，也可以在在服务器启动时就被创建，这需要在web.xml的< servlet>中添加一条配置信息 < load-on-startup>5< /load-on-startup>，当值为0或者大于0时，表示容器在应用启动时就加载这个servlet，当是一个负数时或者没有指定时，则指示容器在该servlet被请求时才加载。

init () 方法

init 方法被设计成只调用一次。它在第一次创建 Servlet 时被调用，在后续每次用户请求时不再调用。Servlet 创建于用户第一次调用对应于该 Servlet 的 URL 时，但是您也可以指定 Servlet 在服务器第一次启动时被加载。

service() 方法

Servlet 容器（即 Web 服务器）调用 service() 方法来处理来自客户端（浏览器）的请求，并把格式化的响应写回给客户端。service() 方法检查 HTTP 请求类型（GET、POST、PUT、DELETE 等），并在适当的时候调用 doGet、doPost、doPut, doDelete 等方法。

doGet() 方法

GET 请求来自于一个 URL 的正常请求，或者来自于一个未指定 METHOD 的 HTML 表单

doPost() 方法

POST 请求来自于一个特别指定了 METHOD 为 POST 的 HTML 表单

destroy() 方法

destroy() 方法只会被调用一次，在 Servlet 生命周期结束时被调用。destroy() 方法可以让您的 Servlet 关闭数据库连接、停止后台线程、把 Cookie 列表或点击计数器写入到磁盘，并执行其他类似的清理活动。在调用 destroy() 方法之后，servlet 对象被标记为垃圾回收。

web.xml的配置

url-pattern大致分为以下几种方式：

1，完全匹配

如：<url-pattern>/servlet/MyServlet.do</url-pattern>

2，目录匹配

如：<url-pattern>/servlet/*</url-pattern>

3, 扩展名匹配

如：<url-pattern>*.htm</url-pattern>

在web.xml文件中，以下语法用于定义映射：

1. 以” /’ 开头和以” /*” 结尾的是用来做路径映射的。
2. 以前缀” *. ” 开头的是用来做扩展映射的。
3. “/” 是用来定义default servlet映射的。
4. 剩下的都是用来定义详细映射的。比如： /aa/bb/cc.action

容器查找规则：

1, 容器会首先查找完全匹配，如果找不到，再查找目录匹配，如果也找不到，就查找扩展名匹配。

2, 如果一个请求匹配多个“目录匹配”，容器会选择最长的匹配。

例如：servletA的url-pattern为/test/*, 而servletB的url-pattern为/test/b/*, 此时访问http://localhost/test/b时，容器会选择路径最长的servlet来匹配，也就是这里的servletB。

注意：” /*.action” 这样一个看起来很正常的匹配会错。因为这个匹配即属于路径映射，也属于扩展映射，会导致容器无法判断。

