**Servlet容器的匹配过程**

2009-07-08 14:22 网络 中国IT实验室 [我要评论(0)](http://developer.51cto.com/art/200907/134840.htm#commment) 字号：[T](javascript:setfont(12);" \t "_self) | [T](javascript:setfont(16);)

[一键收藏，随时查看，分享好友！](javascript:favorBox('open');)

本文介绍Servlet容器的匹配过程，以及介绍匹配规则和顺序，包括精确路径匹配，最长路径匹配和扩展匹配等。

AD： [2013云计算架构师峰会超低价抢票中](http://wot.51cto.com/cloud2013/)

Servlet和filter是J2EE开发中常用的技术，使用方便，配置简单，老少皆宜。

估计大多数朋友都是直接配置用，也没有关心过具体的细节，今天遇到一个问题，上网查了Servlet的规范才发现，Servlet和filter中的url-pattern还是有一些文章在里面的，总结了一些东西，放出来供大家参考，以免遇到问题又要浪费时间。

一、Servlet容器对url的匹配过程：

当一个请求发送到Servlet容器的时候，容器先会将请求的url减去当前应用上下文的路径作为Servlet的映射url，比如我访问的是 http://localhost/test/aaa.html，我的应用上下文是test，容器会将http://localhost/test去掉， 剩下的/aaa.html部分拿来做Servlet的映射匹配。这个映射匹配过程是有顺序的，而且当有一个Servlet匹配成功以后，就不会去理会剩下 的Servlet了（filter不同，后文会提到）。其匹配规则和顺序如下：

1. 精确路径匹配。例子：比如ServletA 的url-pattern为 /test，ServletB的url-pattern为 /\* ，这个时候，如果我访问的url为http://localhost/test ，这个时候容器就会先进行精确路径匹配，发现/test正好被ServletA精确匹配，那么就去调用ServletA，也不会去理会其他的 Servlet了。

2. 最长路径匹配。例子：ServletA的url-pattern为/test/\*，而ServletB的url-pattern为/test/a/\*，此 时访问http://localhost/test/a时，容器会选择路径最长的Servlet来匹配，也就是这里的ServletB.

3. 扩展匹配，如果url最后一段包含扩展，容器将会根据扩展选择合适的Servlet.例子：ServletA的url-pattern：\*.action

4. 如果前面三条规则都没有找到一个Servlet，容器会根据url选择对应的请求资源。如果应用定义了一个default Servlet，则容器会将请求丢给default Servlet（什么是default Servlet？后面会讲）。

根据这个规则表，就能很清楚的知道Servlet的匹配过程，所以定义Servlet的时候也要考虑url-pattern的写法，以免出错。

对于filter，不会像Servlet那样只匹配一个Servlet，因为filter的集合是一个链，所以只会有处理的顺序不同，而不会出现只选择一个filter.Filter的处理顺序和filter-mapping在web.xml中定义的顺序相同。

二、url-pattern详解

在web.xml文件中，以下语法用于定义映射：

以“/‘开头和以”/\*“结尾的是用来做路径映射的。

以前缀“\*.”开头的是用来做扩展映射的。

“是用来定义default Servlet映射的。

剩下的都是用来定义详细映射的。比如： /aa/bb/cc.action

所以，为什么定义“/\*.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，导致Servlet容器无法判断。