JavaWeb学习笔记

# Servlet对比Filter

|  |  |  |
| --- | --- | --- |
|  | Servlet | Filter |
| 概念 | 运行服务器端的java应用程序，具有独立于平台和协议的特性，并且可以动态的生成web页面，它工作在客户端请求与服务器响应的中间层。 | 可复用的代码片段，可以用来转换HTTP请求、响应和头信息。不能产生一个请求或者响应，它只是修改对某一资源的请求，或者修改某一响应。 |
| 生命周期 | 1.启动服务器时加载Servlet的实例；  2、web服务器启动时或接收到请求时，或者两者间的某时刻启动。init（）  3、从第一次到以后的多次访问，都是只调用doGet()或doPost()方法；  4、停止服务器时调用destroy()方法，销毁实例。 | 1、启动服务器时加载过滤器的实例，并调用init()方法来初始化实例；  2、每一次请求时都只调用方法doFilter()进行处理；  3、停止服务器时调用destroy()方法，销毁实例。 |
| 方法加载顺序 | 1).配置了<load-on-startup>，则Tomcat初始化时按值从小到大加载所有servlet的init方法。  2).没有配置<load-on-startup>，容器初始化时不加载。在请求匹配的url时进行加载，并且只加载最后一个servlet的init方法。其他的servlet不加载。 | 1).filter的init方法在容器初始化时加载。第一次加载顺序随机，以后再加载顺序以第一次顺序为准。  2).filter的doFilter方法在请求url时执行，如果有多个filter匹配，则按照<filter-mapping>顺序执行(前提是调用FilterChain的doFilter方法) |
| 总结 | filter流程是线性的，url传来检查之后，可保持原流程继续执行，被下一个filter,servlet接收等；而servlet 处理之后，不会继续向下传递。  filter功能可用来保持流程继续按照原来的方式进行下去，或者主导流程，而servlet的功能主要用来主导流程；  filter和servlet同时存在，且容器初始化都加载，则先加载filter再加载servlet的init方法。  如果请求的url既匹配filter又匹配servlet，并且servlet的init方法没有在容器初始化加载，则先加载匹配的servlet的最后一个servlet的init方法，再按顺序执行filter方法，最后再执行匹配的最后一个servlet方法  filter在Servlet销毁之后才销毁 | |