Struts2的拦截器月过滤器

拦截器的工作原理：  
　　当接收到一个httprequest ,  
a) 当外部的httpservletrequest到来时   
b) 初始到了servlet容器 传递给一个标准的过滤器链   
c) FilterDispatecher会去查找相应的ActionMapper，如果找到了相应的ActionMapper它将会将控制权限交给ActionProxy   
d) ActionProxy将会通过ConfigurationManager来查找配置struts.xml   
i. 下一步将会 通过ActionInvocation来负责命令模式的实现（包括调用一些拦截Interceptor框架在调用action之前）   
ii. Interceptor做一些拦截或者初始的工作   
e) 一旦action返回，会查找相应的Result   
f) Result类型可以是 jsp或者freeMark 等   
g) 这些组件和ActionMapper一起返回给请求的url（注意拦截器的执行顺序）   
h) 响应的返回是通过我们在web.xml中配置的过滤器   
i) 如果ActionContextCleanUp是当前使用的，则FilterDispatecher将不会清理sreadlocal ActionContext;如果ActionContextCleanUp不使用，则将会去清理sreadlocals。   
　　拦截器实现原理：  
  
　　1、拦截器是基于java反射机制的，而过滤器是基于函数回调的。  
2、过滤器依赖于servlet容器，而拦截器不依赖于servlet容器。  
3、拦截器只能对Action请求起作用，而过滤器则可以对几乎所有请求起作用。  
4、拦截器可以访问Action上下文、值栈里的对象，而过滤器不能。  
5、在Action的生命周期中，拦截器可以多次调用，而过滤器只能在容器初始化时被调用一次。  
  
　　过滤器的作用：  
　　(1)执行Actions  
过滤器通过ActionMapper对象，来判断是否应该被映射到Action.如果mapper对象指示他应该被映射，过滤链将会被终止，然后Action被调用。这一点非常重要，如果同时使用SiteMesh filter，则SiteMesh filter应该放到该过滤器前，否则Action的输出将不会被装饰。  
(2)清除ActionContext  
过滤器为了确保内存溢出，会自动的清除ActionContext。这可能会存在一些问题，在和其它的框架集成时，例如SiteMesh。ActionContextCleanUp提供了怎么处理这些问题的一些信息。  
(3)维护静态内容  
过滤器也会维护在Struts2中使用的一些公共的静态的内容，例如JavaScript文件，CSS文件等。搜索/struts/\*范围内的请求，然后将/struts/后面的值映射到一些struts的公共包中，也可以在你的类路径中搜索。默认情况下会去查找以下包：org.apache.struts2.static.template。这样你只用请求/struts/xhtml/styles.css，XHTML UI主题默认的样式表将会被返回。同样，AJAX UI组件需要的JavaScript文件，也可以在org.apache.struts2.static包中被找到。如果你想加入其它被搜索的包，在web.xml中设置filter时，通过给"actionPackages"初始参数一个逗号隔开的包列表值来设定。

区别：

Filter基于回调函数，我们需要实现的filter接口中doFilter方法就是回调函数，而interceptor则基于java本身的反射机制,这是两者最本质的区别。  
Filter是依赖于servlet容器的，即只能在servlet容器中执行，很显然没有servlet容器就无法来回调doFilter方法。而interceptor与servlet容器无关。  
Filter的过滤范围比Interceptor大,Filter除了过滤请求外通过通配符可以保护页面，图片，文件等等，而Interceptor只能过滤请求。  
Filter的过滤例外一般是在加载的时候在init方法声明,而Interceptor可以通过在xml声明是guest请求还是user请求来辨别是否过滤

1：Filter 对在web.xml中该filter配置的路径进行单向拦截， request匹配Filter的路径时做一次拦截！  这个底层实现没想过， 嘿嘿！  
2：interceptor的拦截处理其实就是代理机制！   
Struts2官方文档那个图画的相当明了了。既然是代理，那么是双向拦截，拦截器在action的方法之前执行一次， 之后执行一次！  
根据那个图，你可以再对应的去看下StrutsPrepareAndExecuteFilter的源代码，前面一段代码产生Action的环境，判断mapping之类的， 可以跟进ExecuteOperations execute这个成员变量，下面如果前面一系列判断都满足的话， 会调用execute.executeAction， 这个里面的实现就是产生action的代理， 然后通过配置信息， 去获得这个action身上配置了哪些拦截器，然后执行拦截器的代码，拦截器中用ActionInvocation去通知下一个拦截器执行， 等到所有的拦截器都拦完了以后，执行action， 随后拦截器又会反向执行一遍（官方确实是这个意思， LZ可以自己写两个拦截器测试下看是不是这么会是）  
所以对于Interceptor是不对request进行处理的， 它只是对action起作用。Struts的那个核心过滤器拦截了action的请求之后产生action的代理后进行处理的！