

Filter过滤器，JavaWeb中的一个组件，与Servlet类似，可以捕获到请求和响应，根据项目的需要对请求和响应进行处理。它的位置是处于Servlet之前，客户端请求进入服务端之后，首先会被过滤器捕获，经过过滤器的处理，再传给相应的Servlet；同理从Servlet返回的响应首先会被Filter捕获，进行处理之后再传给客户端。

相当于介于客户端和Servlet之间的一个组件，对客户端和Servlet之间的数据交互进行一个过滤（双向）。

使用

- 1.创建一个自定义的Java类，实现javax.servlet.Filter接口。
- 2.在web.xml配置被过滤器捕获的请求/响应的映射。

生命周期与Servlet类似，init，doFilter，destroy。

doFilter方法中执行完过滤器的逻辑之后，需要进行filterChain.doFilter(request,response)方法调用，否则请求/响应无法向后传递。

过滤器的应用

- 统一设置请求的编码，解决中文乱码问题。
- 屏蔽敏感词。
- 限制用户对某些资源的访问权限。

思路：过滤器拦截download.jsp，判断发出该请求的客户端对应的用户登录状态，如果是登录状态，放行，否则作出相应的处理：挑战到登录页面，让游客登录。

url : <http://localhost:8080/download/download.jsp> --> uri

Url : <http://localhost:8080/logout.do>

获取<http://localhost:8080/uri>

监听器 Listener

是吗是监听器？

是一个对象，专门用来监听其他对象，监听其他对象所发的操作或状态的改变，然后作出相应的处理，当被监听的对象发生某种变化时，立即采取相应的行动。

监听器的分类

- 监听域对象的创建和销毁（request，session，application）
- 监听域对象中属性的新增，删除，修改事情
- 监听绑定到HttpSession中的某个对象的状态

监听域对象的创建和销毁

- 实现ServletContextListener, ServletRequestListener, HttpSessionListener
- 在web.xml中配置

```
package com.southwind.listener;

import javax.servlet.ServletRequestEvent;
import javax.servlet.ServletRequestListener;

public class MyListener implements ServletRequestListener {
    @Override
    public void requestInitialized(ServletRequestEvent sre) {
        System.out.println("创建了一个request对象");
    }

    @Override
    public void requestDestroyed(ServletRequestEvent sre) {
        System.out.println("销毁了一个request对象");
    }
}
```

```
<!-- 启用监听器 -->
<listener>
    <listener-class>com.southwind.listener.MyListener</listener-class>
</listener>
```

```
package com.southwind.listener;

import javax.servlet.http.HttpSessionEvent;
import javax.servlet.http.HttpSessionListener;

public class SessionListener implements HttpSessionListener {
    @Override
    public void sessionCreated(HttpSessionEvent se) {
        System.out.println("创建了一个session");
    }

    @Override
    public void sessionDestroyed(HttpSessionEvent se) {
        System.out.println("销毁了一个session");
    }
}
```

```
package com.southwind.listener;

import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;

public class ApplicationListener implements ServletContextListener {
    @Override
    public void contextDestroyed(ServletContextEvent sce) {
        System.out.println("销毁了一个application对象");
    }

    @Override
    public void contextInitialized(ServletContextEvent sce) {
        System.out.println("创建了一个application对象");
    }
}
```