Listener,Filter和servlet都是配置在web.xml文件中。

Listener、Filter、Servlet定义：

Listener:

首先定义一个Listener，实现以下接口：

**HttpSessionListener**(用来监控session的创建，销毁等)

**ServletRequestListener**(用于监控servlet上下文request)

**ServletRequestAttributeListener**(用于监控request中的attribute的操作)

**[java]** [view plain](http://blog.csdn.net/javawebxy/article/details/50514046) [copy](http://blog.csdn.net/javawebxy/article/details/50514046)

1. **package** com.xy.web.listener;
2. **import** javax.servlet.ServletRequestAttributeEvent;
3. **import** javax.servlet.ServletRequestAttributeListener;
4. **import** javax.servlet.ServletRequestEvent;
5. **import** javax.servlet.ServletRequestListener;
6. **import** javax.servlet.http.HttpSessionEvent;
7. **import** javax.servlet.http.HttpSessionListener;
8. **import** org.slf4j.Logger;
9. **import** org.slf4j.LoggerFactory;
11. **public** **class** TestListener **implements** HttpSessionListener,ServletRequestListener,ServletRequestAttributeListener {
12. **private** Logger logger = LoggerFactory.getLogger(TestListener.**class**);
14. //sessionListener start!
15. **public** **void** sessionCreated(HttpSessionEvent arg0) {
16. logger.info(".......TestListener sessionCreated().......");
17. }
19. **public** **void** sessionDestroyed(HttpSessionEvent arg0) {
20. logger.info(".......TestListener sessionDestroyed().......");
21. }
22. //sessionListener end!
24. //requestListener start!
25. **public** **void** requestInitialized(ServletRequestEvent arg0) {
26. logger.info("......TestListener requestInitialized()......");
27. }
29. **public** **void** requestDestroyed(ServletRequestEvent arg0) {
30. logger.info("......TestListener requestDestroyed()......");
31. }
32. //requestListener end!
34. //attributeListener start!
35. **public** **void** attributeAdded(ServletRequestAttributeEvent srae) {
36. logger.info("......TestListener attributeAdded()......");
37. }
39. **public** **void** attributeRemoved(ServletRequestAttributeEvent srae) {
40. logger.info("......TestListener attributeRemoved()......");
41. }
43. **public** **void** attributeReplaced(ServletRequestAttributeEvent srae) {
44. logger.info("......TestListener attributeReplaced()......");
45. }
46. //attributeListener end!
47. }

Filter：

**[java]** [view plain](http://blog.csdn.net/javawebxy/article/details/50514046) [copy](http://blog.csdn.net/javawebxy/article/details/50514046)

1. **package** com.xy.web.filter;
3. **import** java.io.IOException;
5. **import** javax.servlet.Filter;
6. **import** javax.servlet.FilterChain;
7. **import** javax.servlet.FilterConfig;
8. **import** javax.servlet.ServletException;
9. **import** javax.servlet.ServletRequest;
10. **import** javax.servlet.ServletResponse;
12. **import** org.slf4j.Logger;
13. **import** org.slf4j.LoggerFactory;
15. **public** **class** TestFilter **implements** Filter {
16. **private** Logger logger = LoggerFactory.getLogger(TestFilter.**class**);
18. **public** **void** destroy() {
19. logger.info("..............execute TestFilter destory()..............");
20. }
22. **public** **void** doFilter(ServletRequest arg0, ServletResponse arg1,FilterChain arg2) **throws** IOException, ServletException {
23. logger.info("..............execute TestFilter doFilter()..............");
24. arg2.doFilter(arg0, arg1);
25. }
27. **public** **void** init(FilterConfig arg0) **throws** ServletException {
28. logger.info("..............execute TestFilter  init()..............");
29. }
30. }

Servlet：

**[java]** [view plain](http://blog.csdn.net/javawebxy/article/details/50514046) [copy](http://blog.csdn.net/javawebxy/article/details/50514046)

1. **package** com.xy.web.servlet;
3. **import** java.io.IOException;
5. **import** javax.servlet.ServletException;
6. **import** javax.servlet.http.HttpServlet;
7. **import** javax.servlet.http.HttpServletRequest;
8. **import** javax.servlet.http.HttpServletResponse;
10. **import** org.slf4j.Logger;
11. **import** org.slf4j.LoggerFactory;
13. **public** **class** TestServlet **extends** HttpServlet {
15. **private** Logger logger = LoggerFactory.getLogger(TestServlet.**class**);
17. /\*\*
18. \*
19. \*/
20. **private** **static** **final** **long** serialVersionUID = -4263672728918819141L;
22. @Override
23. **public** **void** init() **throws** ServletException {
24. logger.info("...TestServlet init() init..........");
25. **super**.init();
26. }
28. @Override
29. **public** **void** destroy() {
30. logger.info("...TestServlet init() destory..........");
31. **super**.destroy();
32. }
33. @Override
34. **protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp)
35. **throws** ServletException, IOException {
36. **this**.doPost(req, resp);
37. }
39. @Override
40. **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)
41. **throws** ServletException, IOException {
42. logger.info("...TestServlet doPost() start..........");
43. //操作attribute
44. request.setAttribute("a", "a");
45. request.setAttribute("a", "b");
46. request.getAttribute("a");
47. request.removeAttribute("a");
48. //操作session
49. request.getSession().setAttribute("a", "a");
50. request.getSession().getAttribute("a");
51. request.getSession().invalidate();
52. logger.info("...TestServlet doPost() end..........");
53. }
54. }

配置XML：

**[html]** [view plain](http://blog.csdn.net/javawebxy/article/details/50514046) [copy](http://blog.csdn.net/javawebxy/article/details/50514046)

1. <!-- 测试filter -->
2. **<filter>**
3. **<filter-name>**TestFilter**</filter-name>**
4. **<filter-class>**com.xy.web.filter.TestFilter**</filter-class>**
5. **</filter>**
6. **<filter-mapping>**
7. **<filter-name>**TestFilter**</filter-name>**
8. **<url-pattern>**\*.do**</url-pattern>**
9. **</filter-mapping>**
10. <!-- 测试servlet -->
11. **<servlet>**
12. **<servlet-name>**TestServlet**</servlet-name>**
13. **<servlet-class>**com.xy.web.servlet.TestServlet**</servlet-class>**
14. **</servlet>**
15. **<servlet-mapping>**
16. **<servlet-name>**TestServlet**</servlet-name>**
17. **<url-pattern>**/\***</url-pattern>**
18. **</servlet-mapping>**
19. <!-- 测试listener -->
20. **<listener>**
21. **<listener-class>**com.xy.web.listener.TestListener**</listener-class>**
22. **</listener>**

配置好以后启动项目。

加载顺序：

日志：

查看启动日志如下：

信息: Initializing Coyote HTTP/1.1 on http-80

............

2016-1-14 0:47:04 org.apache.catalina.core.ApplicationContext log  
*信息: ContextListener: contextInitialized()  
2016-1-14 0:47:04 org.apache.catalina.core.ApplicationContext log  
信息: SessionListener: contextInitialized()*

...........

**2016-01-14 00:47:23,369  INFO TestFilter:28 - ..............execute TestFilter  init().............**.  
2016-1-14 0:47:23 org.apache.catalina.core.ApplicationContext log  
**信息: Initializing Spring FrameworkServlet 'spring'**

...........

2016-1-14 0:47:25 org.apache.catalina.startup.Catalina start  
信息: Server startup in 21565 ms

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

结论：

**从启动日志来看，**

**启动的顺序为listener->Filter->servlet.**

**简单记为：理(Listener)发(Filter)师(servlet).**

**执行的顺序不会因为三个标签在配置文件中的先后顺序而改变。**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

生命周期：

日志：

访问项目路径：http://localhost/MyWebProject/common/test.do，访问action两次，打断点后查看日志情况：

第一次访问：

2016-01-14 00:03:03,991  INFO TestListener:26 - ......TestListener requestInitialized()......  
2016-01-14 00:03:04,001  INFO TestServlet:24 - ...TestServlet init() init..........  
2016-01-14 00:03:04,011  INFO TestFilter:23 - ..............execute TestFilter doFilter()..............  
2016-01-14 00:03:15,275  INFO TestServlet:42 - ...TestServlet doPost() start..........  
2016-01-14 00:03:16,255  INFO TestListener:36 - ......TestListener attributeAdded()......  
2016-01-14 00:03:16,853  INFO TestListener:44 - ......TestListener attributeReplaced()......  
2016-01-14 00:03:18,561  INFO TestListener:40 - ......TestListener attributeRemoved()......  
2016-01-14 00:03:20,065  INFO TestListener:16 - .......TestListener sessionCreated().......  
2016-01-14 00:03:22,908  INFO TestListener:20 - .......TestListener sessionDestroyed().......  
2016-01-14 00:03:25,624  INFO TestServlet:52 - ...TestServlet doPost() end..........  
2016-01-14 00:03:27,746  INFO TestListener:30 - ......TestListener requestDestroyed()......

第二次访问：

2016-01-14 00:04:08,908  INFO TestListener:26 - ......TestListener requestInitialized()......  
2016-01-14 00:04:08,909  INFO TestFilter:23 - ..............execute TestFilter doFilter()..............  
2016-01-14 00:04:14,385  INFO TestServlet:42 - ...TestServlet doPost() start..........  
2016-01-14 00:04:14,778  INFO TestListener:36 - ......TestListener attributeAdded()......  
2016-01-14 00:04:14,974  INFO TestListener:44 - ......TestListener attributeReplaced()......  
2016-01-14 00:04:15,342  INFO TestListener:40 - ......TestListener attributeRemoved()......  
2016-01-14 00:04:15,904  INFO TestListener:16 - .......TestListener sessionCreated().......  
2016-01-14 00:04:17,354  INFO TestListener:20 - .......TestListener sessionDestroyed().......  
2016-01-14 00:04:17,815  INFO TestServlet:52 - ...TestServlet doPost() end..........  
2016-01-14 00:04:19,044  INFO TestListener:30 - ......TestListener requestDestroyed()......

关闭项目，打印日志如下：

2016-1-14 0:40:15 org.apache.coyote.http11.Http11Protocol pause  
信息: Pausing Coyote HTTP/1.1 on http-80  
2016-1-14 0:40:16 org.apache.catalina.core.StandardService stop  
信息: Stopping service Catalina  
**2016-1-14 0:40:16 org.apache.catalina.core.ApplicationContext log  
信息: SessionListener: contextDestroyed()  
2016-1-14 0:40:16 org.apache.catalina.core.ApplicationContext log  
信息: ContextListener: contextDestroyed()**  
2016-01-14 00:40:16,561  INFO TestServlet:30 - ...TestServlet init() destory..........  
.......  
2016-01-14 00:40:22,091  INFO TestFilter:19 - ..............execute TestFilter destroy()..............

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

结论：

从启动，结束和运行时候的日志看：

**Listener生命周期：一直从程序启动到程序停止运行。**

**ServletRequestListener：每次访问一个Request资源前，都会执行requestInitialized()方法，方法访问完毕，都会执行requestDestroyed()方法。**

**HttpSessionListener：每次调用request.getSession()，都会执行sessionCreated()方法，执行session.invalidate()方法，都会执行sessionDestroyed()方法。**

ServletRequestAttributeListener：每次调用request.setAttribute()都会执行attributeAdded()方法，如果set的key在request里面存在，就会执行attributeReplacerd()方法，调用request.removeAttribute()方法，都会执行attributeRemoved()方法。

**Filter生命周期：程序启动调用Filter的init()方法**(永远只调用一次,具体看启动日志)**，程序停止调用Filter的**destroy()方法(永远只调用一次，具体看关闭日志)，doFilter()方法每次的访问请求如果符合拦截条件都会调用(程序第一次运行，会在servlet调用init()方法以后调用，不管第几次，都在调用doGet(),doPost()方法之前)。

Servlet生命周期：程序第一次访问，会调用servlet的init()方法初始化(只执行一次，具体看日志)，每次程序执行都会根据请求调用doGet()或者doPost()方法，程序停止调用destory()方法(具体看结束日志)。