

监听器

监听器用于监听web应用中某些对象、信息的创建、销毁、增加，修改，删除等动作的发生，然后作出相应的响应处理。当范围对象的状态发生变化时，服务器自动调用监听器对象中的方法。常用于统计在线人数和在线用户，系统加载时进行信息初始化，统计网站的访问量等等。

分类：

按监听的对象划分，可以分为：

- ServletContext对象监听器
- HttpSession对象监听器
- ServletRequest对象监听器

按监听的事件划分

- 对象自身的创建和销毁的监听器
- 对象中属性的创建和消除的监听器
- session中的某个对象的状态变化的监听器

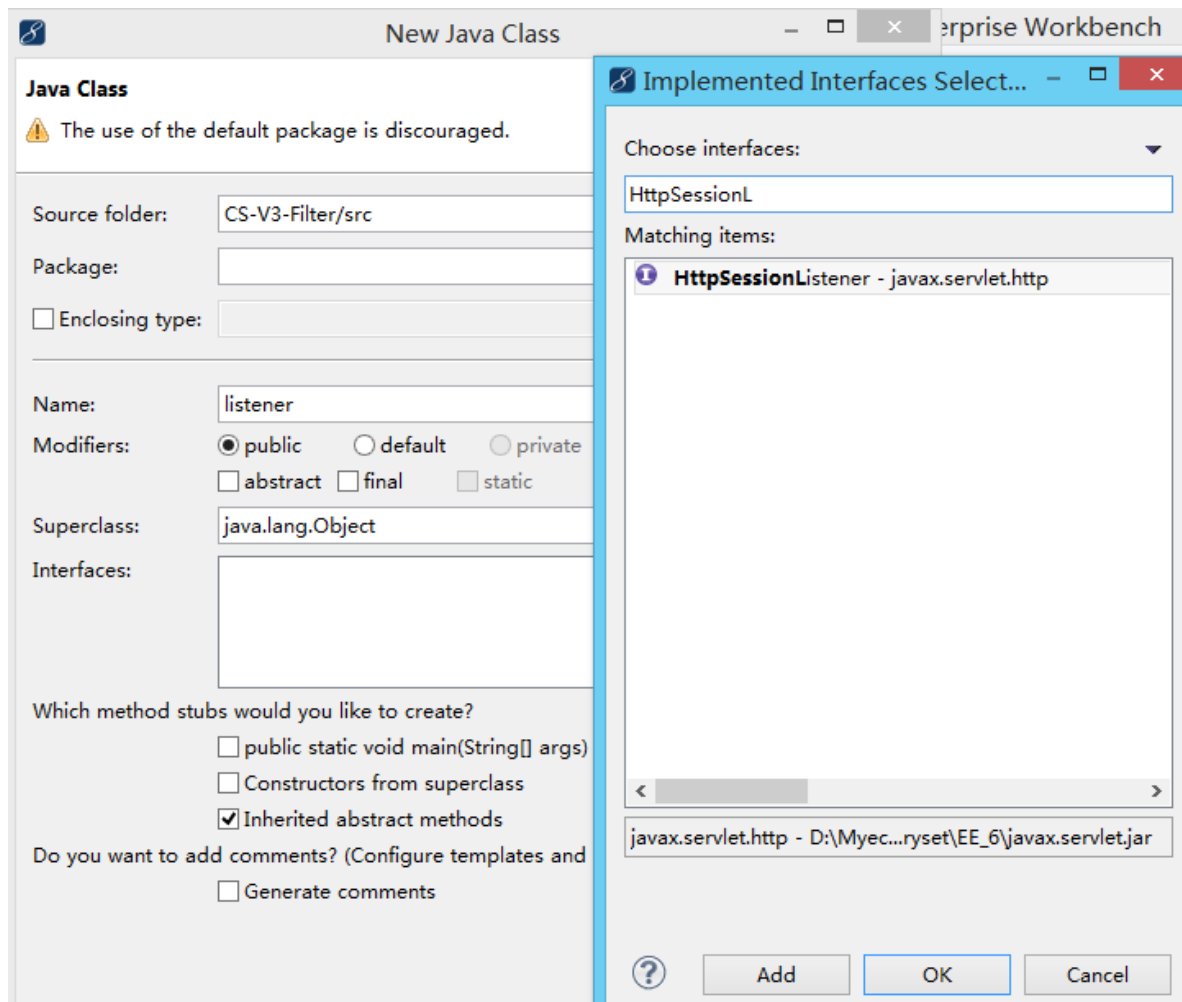
HttpSessionListener示例：

示例：用监听器统计网站在线人数

原理：每当有一个访问连接到服务器时，服务器就会创建一个session来管理会话。那么我们就可以通过统计session的数量来获得当前在线人数。

所以这里用到的是HttpSessionListener。

1：创建监听器类，实现HttpSessionListener接口。



2: 重写监听器类中的方法

```
1 public class onLineCount implements HttpSessionListener {
2
3     public int count=0; //记录session的数量
4     public void sessionCreated(HttpSessionEvent arg0) { //监听session的创建
5         count++;
6         arg0.getSession().getServletContext().setAttribute("Count", count);
7     }
8
9
10    @Override
11    public void sessionDestroyed(HttpSessionEvent arg0) { //监听session的撤销
12        count--;
13        arg0.getSession().getServletContext().setAttribute("Count", count);
14    }
15
16 }
```

3: 在web.xml中配置监听器。注意：监听器>过滤器>servlet，配置的时候要注意先后顺序

```
1 <listener>
2     <listener-class>com.ygj.control.onLineCount</listener-class>
3 </listener>
```

在Servlet3.0中，监听器的配置可以直接在代码中通过注释来完成，无需在web.xml中再配置。

```

1  @WebListener    //在此注明以下类是监听器
2  public class onLineCount implements HttpSessionListener {
3
4      public int count=0;
5      public void sessionCreated(HttpSessionEvent arg0) {
6          count++;
7          arg0.getSession().getServletContext().setAttribute("Count", count);
8      }
9
10
11     @Override
12     public void sessionDestroyed(HttpSessionEvent arg0) {
13         count--;
14         arg0.getSession().getServletContext().setAttribute("Count", count);
15     }

```

4: 在显示在线人数处通过session.getAttribute("Count")即可获取在线人数值。

其他常用监听器:

除了上面监听session建立与销毁的listener外, 还有以下几个常用的监听器。

1: 监听session属性的增加、移除以及属性值改变的HttpSessionAttributeListener

```

@Override
public void attributeAdded(HttpSessionBindingEvent arg0) {
    // TODO Auto-generated method stub
}

@Override
public void attributeRemoved(HttpSessionBindingEvent arg0) {
    // TODO Auto-generated method stub
}

@Override
public void attributeReplaced(HttpSessionBindingEvent arg0) {
    // TODO Auto-generated method stub
}

```

2: 监听web上下文的初始化（服务器已准备好接收请求）与销毁的ServletContextListener

```

@Override
public void contextDestroyed(ServletContextEvent arg0) {
    // TODO Auto-generated method stub
}

@Override
public void contextInitialized(ServletContextEvent arg0) {
    // TODO Auto-generated method stub
}

```

3: 监听web上下文属性的增加、删除、属性值变化的ServletContextAttributeListener

```

@Override
public void attributeAdded(ServletContextAttributeEvent arg0) {
    // TODO Auto-generated method stub

}

@Override
public void attributeRemoved(ServletContextAttributeEvent arg0) {
    // TODO Auto-generated method stub

}

@Override
public void attributeReplaced(ServletContextAttributeEvent arg0) {
    // TODO Auto-generated method stub

}

```

4: 监听request的创建与销毁的ServletRequestListener

```

@Override
public void requestDestroyed(ServletRequestEvent arg0) {
    // TODO Auto-generated method stub

}

@Override
public void requestInitialized(ServletRequestEvent arg0) {
    // TODO Auto-generated method stub

}

```

5: 监听request的属性的增加、删除、属性值变化的ServletRequestAttributeListener

	ServletContext域	HttpSession域	ServletRequest域
域对象的创建与销毁	ServletContextListener	HttpSessionListener	ServletRequestListener
域对象内的属性的变化	ServletContextAttributeListener	HttpSessionAttributeListener	ServletRequestAttributeListener

监听器原理

底层是使用观察者模式。