

1. 监听器：专门用于对其他对象身上发生的**事件或状态改变**进行**监听和相应处理**的对象，当被监视的对象发生情况时，立即采取相应的行动。

2. Servlet 监听器：Servlet 规范中定义的一种特殊类，它用于监听 web 应用程序中的 **ServletContext**、**HttpSession** 和 **ServletRequest** 等域对象的**创建与销毁**事件，以及监听这些域对象中的**属性发生修改**的事件。

3. 按监听的事件类型 Servlet 监听器可分为如下三种类型：

- A. 监听域对象自身的**创建和销毁**的事件监听器
- B. 监听域对象中的属性的**增加、替换和删除**的事件监听器
- C. 监听**绑定到 HttpSession 域中的某个对象的状态**的事件监听器

4. Servlet 规范为每种事件监听器都定义了相应的接口，开发人员编写的事件监听器程序只需实现这些接口，web 服务器根据用户编写的事件监听器所实现的接口把它注册到相应的被监听对象上。

3.1 版本的监听器有：

5. 被监听对象中的属性的增加，删除和替换的事件，同一个事件在这三个接口中对应的**方法名称完全相同，只是接受的参数类型不同**。

6. 对象的绑定与解绑：实现了 HttpSessionBindingListener 接口的 JavaBean 对象可以感知自己被绑定到 Session 中和从 Session 中删除的事件。

不需要 web.xml 文件中进行注册，也不需要注解配置！若配置了可能导致启动过程中出现错误！

7. 监听器的配置，两种方式：

A. 注解的方式：使用的注解是**@WebListener**，**加在类名上方**——仅有一个 value 属性！

B. XML 配置，例：

```
<listener>
<listener-class>com.ddb.javaweb.listener.ServletContextListenerDemo</listener-
class>
</listener>
```

8. 相关顺序：

启动顺序：listener---filter---servlet

销毁顺序：servlet---filter---listener

9. 让一个会话失效，可以有如下两种参考方式：

1. 设置 x m l 。

```
<session-config>  
<session-timeout>30</session-timeout><!-- 30 分钟超时 -->  
</session-config>
```

2. 调用使无效的方法：

```
request.getSession().invalidate();
```

10. 典型应用：---统计当前在线人数

实现思路：

- A. 在 a p p l i c a t i o n 中存一个 s e t 集合，用来存储登录的用户名；
- B. 在有新的会话接入时，从 a p p l i c a t i o n 中取出 s e t 集合，增加新的用户，再将 s e t 集合设回 a p p l i c a t i o n 中；——可能需要处理中文用户名乱码的问题。
- C. 会话失效时，从 a p p l i c a t i o n 中取出 s e t ，从 s e t 中删除当前的用户名，同时将 s e t 设回 a p p l i c a t i o n 中。
- D. 返回 s e t 的 s i z e 就是在线的用户数。