**监听器（Listener）**

1 监听器：

专门监听其他对象身上发生的时间或状态属性的改变，并及时作出相应的处理。

**Servlet监听器：**Servlet规范中的一种特殊的类，用于监听web应用中的ServletContext、ServletSession和ServletRequset的域对象的创建和销毁，以及域对象属性的变化。

2 分类：

2-1 监听域对象的创建和销毁

对象：ServletContext、HttpSession、HttpServletRequest

》域对象的生命周期



》监听器接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **域对象** | **监听器接口** | **创建域对象** | **销毁域对象** | **方法参数** |
| ServletContext | ServletContextListener | contextInitialized() | contextDestroyed() | ServletContextEvent sce |
| HttpSession | HttpSessionListener | sessionCreated() | sessionDestroyed() | HttpServletEvent se |
| ServletRequest | ServletRequestListener | requestInitialized() | requestDestroyed() | ServletRequestEvent sre |

2-2 监听域对象中属性的变更

包括属性的添加（attributeAdded）、替换（attributeReplaced）、移除（attributeRemoved）

》监听器接口

|  |  |
| --- | --- |
| **域对象** | **监听器接口** |
| ServletContext | ServletContextAttributeListener |
| HttpSession | HttpSessionAttributeListener |
| ServletRequest | ServletRequestAttributeListener |

》监听器的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **添加属性** | **替换属性** | **移除属性** | **方法的参数** |
| attributeAdded | attributeReplaced | attributeRemoved | ServletContextAttributeEvent scae |
| HttpSessionEvent se |
| ServletRequestAttributeEvent srae |

属性事件的对象有2个方法，getName()和getValue()，用于获取发生改变的属性名称和值，其中，属性替换事件的getValue方法获取的是被更改的属性原来的值。

2-3 监听session的绑定事件

》session域中对象的状态

|  |  |
| --- | --- |
| **状态** | **对应的监听器** |
| 绑定到session中 | HttpSessionBindingListener |
| 从session域中移除绑定 |
| 随session对象持久化到一个设备中 | HttpSessionActivationListener |
| 随session对象从一个存储设备中恢复 |

》监听器接口

|  |  |  |  |
| --- | --- | --- | --- |
| **监听器** | **方法** | **说明** | **参数** |
| HttpSessionBindingListener | valueBound() | 对象通过setAttribute方法绑定到session中 | HttpSessionBindingEvent event |
| valueUnbound() | 对象通过removeAttribute方法从session中解除绑定 |
| HttpSessionActivationListener | sessionWillPassivate() | JavaBean对象被“钝化”，即被写入磁盘，序列化 | HttpSessionEvent se |
| sessionDidActivate() | JavaBean对象被“活化”，即从磁盘读取到内存中，反序列化 |

**HttpSessionBindingEvent对象的方法：**

getName()：获取JavaBean对象被绑定到session中所用的属性名；

getValue()：获取JavaBean对象本身

getSession()：获取绑定javaB对象的session对象

**HttpSessionBindingEvent对象的方法：**

getSession()：获取绑定javaB对象的session对象

* session对象在web应用被关闭后并不会马上销毁，它是有一定的存活时间的；此时session对象会被序列化（文件）保存到Tomcat服务器的“work\catalina\localhost\项目名称”目录下，同时，如果绑定到session对象中的对象也实现了序列化接口的话，该对象也可以保存到磁盘中。
* 要实现HttpSessionActivationListener接口的功能，处理实现它的两个方法以为，绑定的类还需要实现Serializable序列化接口。
* 这两个接口用于监听JavaBean对象在session中的状态，是与特定的JavaBean对象绑定在一起，即JavaBean的类实现相应的监听器接口，JavaBean对象就可以感知自身的状态了，因而不用web.xml中进行注册。

3 创建监听器

3-1 实现监听器接口

根据需要监听的对象，实现相应的监听器接口。

ServletContextEvent类封装的web应用的ServletContext对象，对用getServletContext方法即可获得该对象。

三种监听器接口的API和用法基本一致

3-2 注册监听器

在web.xml中注册监听器

<listener>

<listener-class>监听器类的完整类名</listener-class>

</listener>

\*\* HttpSessionBindingListener和HttpSessionActivationListener这两个监听器不用注册

3-3 监听器应用

* ServletContextListener：最常用的监听器，可在web应用加载时进行资源的初始化操作，如创建数据库连接池、创建Spring的IOC容器、读取web应用的初始化参数等。
* 域对象属性的监听器使用较少。