

## 监听器

1. 四个范围对象: application, session, request
2. 开发步骤:
  - a. 编写监听器, 实现接口
  - b. 配置web.xml
3. 监听对象的创建和销毁
  - a. requestServletRequestListener  
session:HttpSessionListener  
application:ServletContextListener
  - b. 每个监听对象各自提供两个方法:监听开始, 监听结束的方法
  - c. ServletContext在servlet容器启动时 自动创建
4. 监听对象中属性的变更
  - a. requestServletRequestAttributeListener  
session:HttpSessionAttributeListener  
application:ServletContextAttributeListener

## session对象的特殊监听(不需要配web.xml)

### 1. session的钝化和活化

钝化: 内存->硬盘

活化: 硬盘->内存

### 2. session对象的四种状态

监听绑定和解绑: HttpSessionBindingListener

- a. session.setAttribute("a", xxx) 对象a绑定到session中
- b. session.removeAttribute("a") 将对象a从session中解绑

监听活化和钝化: HttpSessionActivationListener

- c. 钝化
- d. 活化

绑定和解绑(具体看代码):

```
System.out.println("绑定:Bean对象(将Bean对象绑定到session域), 绑定的对象"+this+", sessionId"+event.getSession().getId());
```

this: 代表绑定的对象

event: 可以获取存储a对象的session对象

## 钝化和活化

如何钝化和活化:配置

tomcat安装目录/conf/context.xml

<!--通过配置实现钝化和活化

    maxIdleSwap:最大空闲时间, 如果超过改时间, 将会被钝化

    FileStore:通过该类 具体实现 钝化操作

    directory:相对路径

-->

<Manager

    className="org/apache/catalina/session/PersistentManager" maxIdleSwap="5"

>

    <Store className="org/apache/catalina/session/FileStore"

    directory="lq"/>

</Manager>

钝化,活化的本质就是 序列化 和反序列化:序列化,反序列化需要实现接口

总结:钝化,活化 实际执行 是通过context.xml中进行配置而进行

活化:session中获取某一个对象时,如果对象不存在,则直接从之前钝化的文件中获取

HttpSessionActivationListener只是负责在session钝化和活化时 予以监听

需要实现Serializable