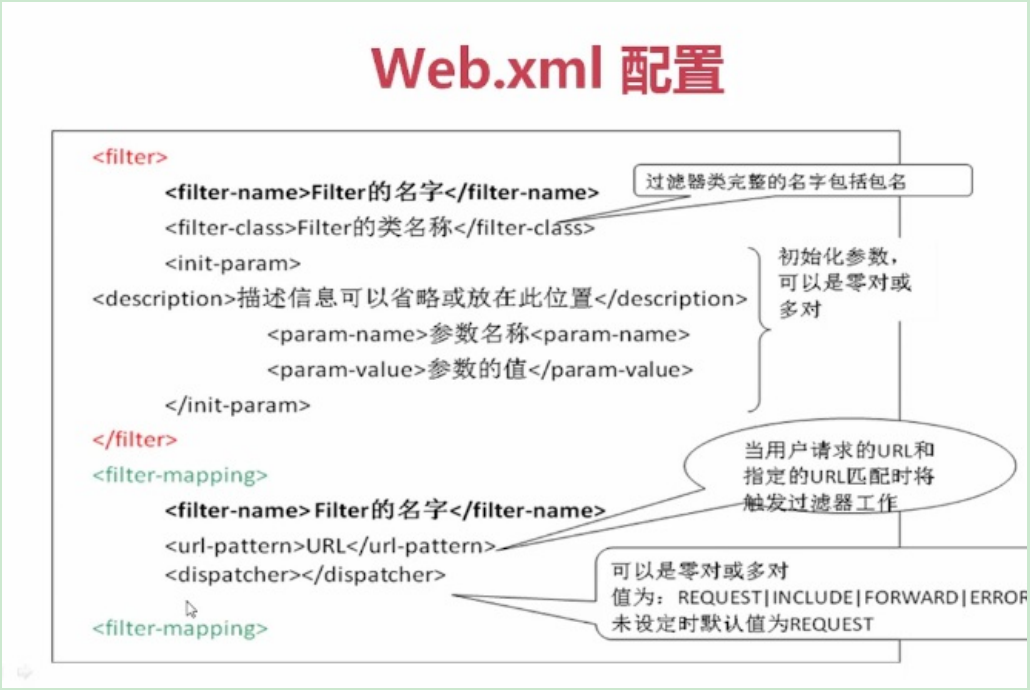
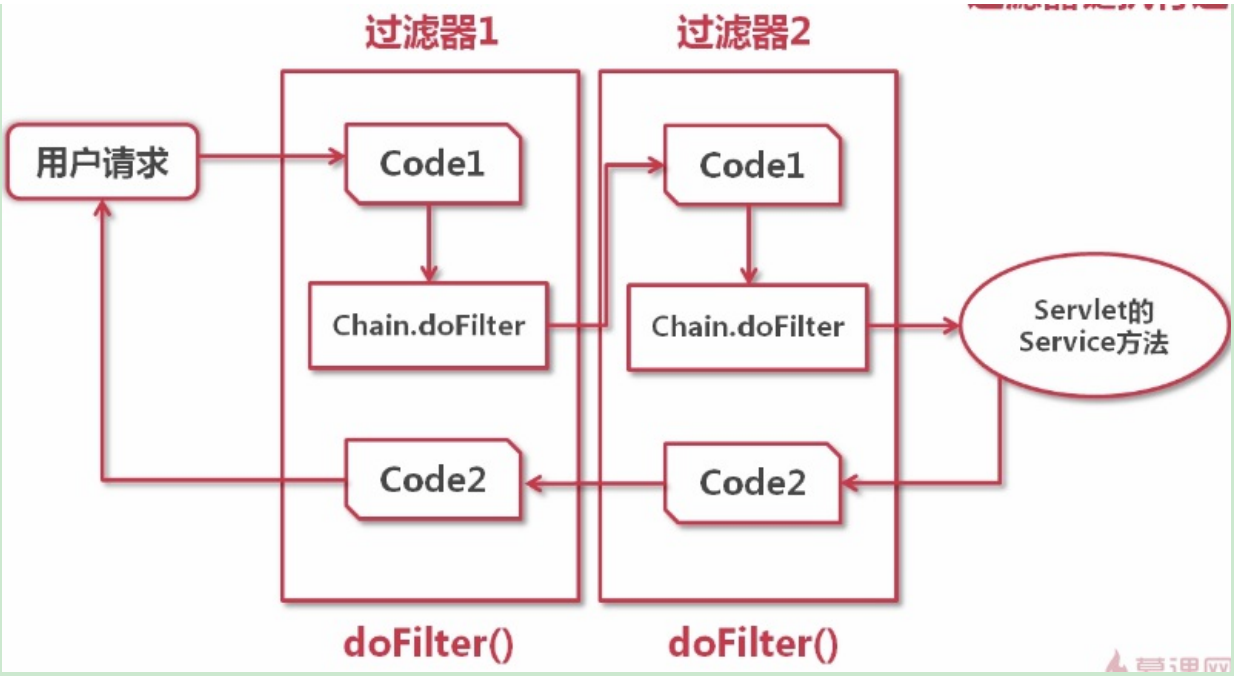
过滤器与监听器 ..报表

过滤器是一个服务端的概念，它课截取客户端的请求与响应信息，并对这些信息进行过滤
工作原理
生命周期
实例化
初始化
过滤
销毁
过滤器的配置



创建Filter类 继承Filter接口
然后在Web.xml中配置

过滤器链
过滤器链的一个执行过程，和单个过滤器相类似，只是执行Chain.doFilter方法后会继续的执行下一个过滤器放行开关之前的代码，直到所有的过滤器方法都执行完毕，到达Servlet的Service方法，然后在倒序的执行过滤器放行开关之后的代码。如下图所示：
为了形成一个过滤器链
1：创建第二个过滤器
2：在web.xml文件中配置第二个过滤器，并且使他们的过滤URL是一致的
3：部署项目
4：测试项目
5：验证我们刚刚所学的理论知识——过滤器链的执行顺序



过滤器的分类

servelet3.0加入了异步处理分类（功能）

@WebFilter用于将一个类声明为过滤器，该注解将会在部署时被容器处理，容器将根据具体的属性配置将相应的类部署为过滤器。

@WebFilter 的常用属性

属性名	类型	描述
filterName	String	指定过滤器的 name 属性，等价于 <filter-name>
value	String[]	该属性等价于 urlPatterns 属性。但是两者不应该同时使用。
urlPatterns	String[]	指定一组过滤器的 URL 匹配模式。等价于 <url-pattern> 标签。
servletNames	String[]	指定过滤器将应用于哪些 Servlet。取值是 @WebServlet 中的 name 属性的取值，或者是 web.xml 中 <servlet-name> 的取值。
dispatcherTypes	DispatcherType	指定过滤器的转发模式。具体取值包括： ASYNC、ERROR、FORWARD、INCLUDE、REQUEST。
initParams	WebInitParam[]	指定一组过滤器初始化参数，等价于 <init-param> 标签。
asyncSupported	boolean	声明过滤器是否支持异步操作模式，等价于 <async-supported> 标签。
description	String	该过滤器的描述信息，等价于 <description> 标签。

监听器

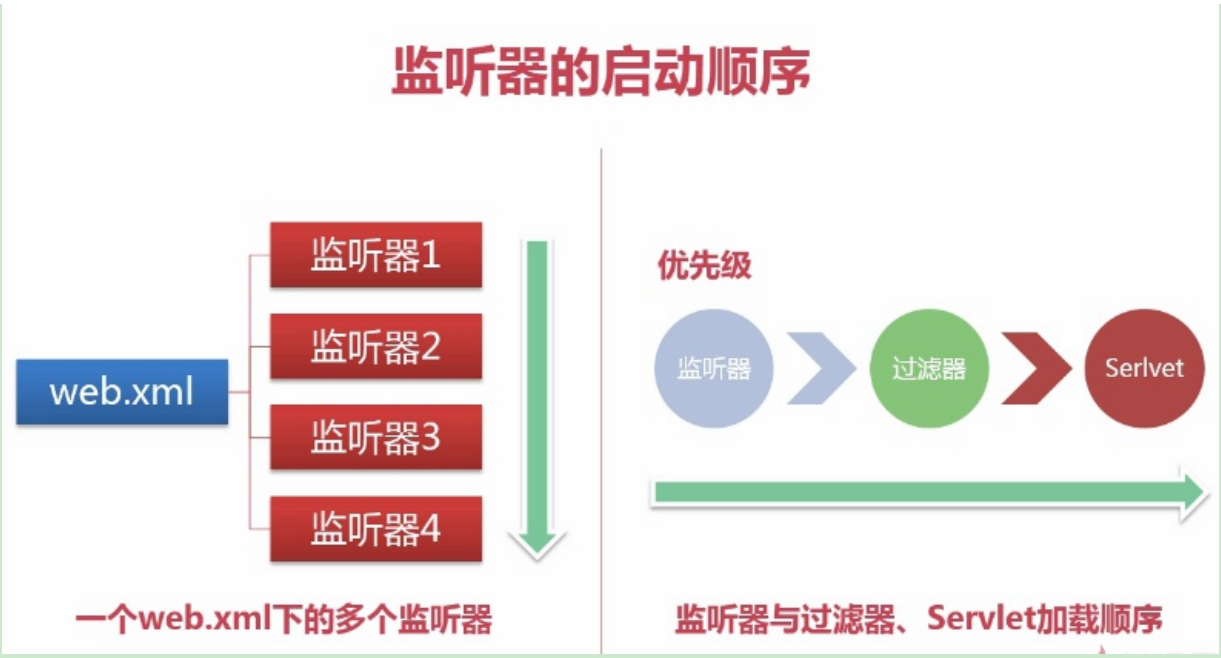
Web监听器的用途：

1，统计在线人数和在线用户。

2，系统启动时加载初始化信息。

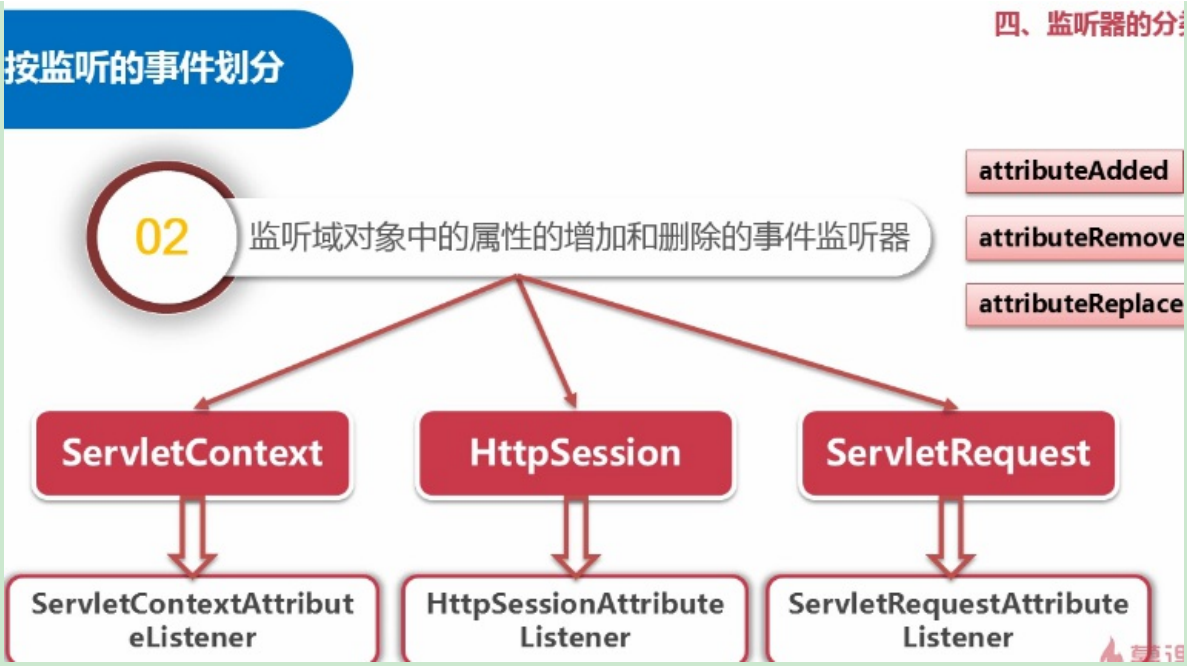
3，统计网站访问量。

4，跟Spring结合。



监听器监听的对象
1 用于监听应用程序环境对象(ServletContext)的事件监听器
2 用于监听用户会话对象（HttpSession）的事件监听器
3 用于监听请求消息对象 （ServeltRequest）的事件监听器
2.按事件划分
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

ServletContext ServletContextListener这个接口 用途主要是 定时器 全局属性对象
HttpSession 实现了 HttpSessionListener这个接口 主要用途是统计在线人数 记录访问日志
ServletRequest 实现了ServletRequestListener这个接口 主要用途是读取参数，记录访问历史
（request监听的是用户的每一个访问请求）
三个方法：
request的
attributeAdded attributeRemoved attributeReplaced (增加，删除，替换)
注意三个方法的在不同作用域的执行时间
HttpSessionListener监听的创建和销毁
1.什么时候创建？
启动服务器第一次访问浏览器，会创建一个session
2.什么时候销毁？
a.关闭服务器
b.关闭浏览器过一段时间session过期
c.不关闭浏览器，session超时
可以在web.xml配置session超时的时间
<session-config>
<session-timeout>0</session-timeout>
</session-config>
三种监听器增加删除更新



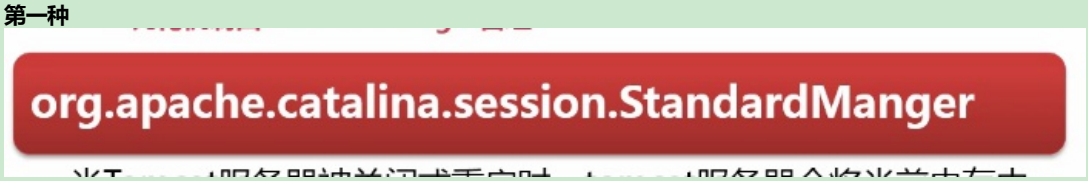
绑定到Httpsession域中的对象状态的事件监听器

HttpSession中的对象状态：
（1）绑定：通过setAttribute保存到session对象当中；
（2）解除绑定：removeAttribute；
（3）钝化：将session对象持久化到存储设备上；
（4）活化：将session对象从存储设备上进行恢复。

钝化机制

Session钝化机制SessionManager管理

第一种



- 当Tomcat服务器被关闭或重启时，tomcat服务器会将当前内存中的Session对象钝化到服务器文件系统中；
- 另一种情况是Web应用程序被重新加载时，内存中的Session对象也会被钝化到服务器的文件系统中。
- 钝化后的文件被保存:Tomcat安装路径/work/Catalina/hostname/applicationname/SESSION.ser

第二种

org.apache.catalina.session.Persistentmanager

- 首先在钝化的基础上进行了扩张。第一种情况如上面1，第二种如上2，第三种情况，可以配置主流内存的Session对象数目，将不长使用的Session对象保存到文件系统或数据库，当用时再重新加载。
- 默认情况下，Tomcat提供两个钝化驱动类，
- org.apache.Catalina.FileStore和org.apache.Catalina.JDBCStore。

Servlet规范
HttpSession BindingListener
绑定:valueBournd方法
解除绑定:valueUnbound方法

HttpSessionActivationListener
钝化:sessionWillPassivate方法
活化:sessionDidActivate方 法

这两个监听器不需要在web.xml注册

使用servlet3.0前提条件
1 使用servlet3.0新标准jar包
JDK必须在1.6以上版本
编译器的编译级别wei6.0
在web.xml文件中，使用3.0规范
使用支持servlet3.0特性的web容器，比如tomcat7
使用servlet监听器的用法

@WebListener

该注解用于将类声明为监听器，被 @WebListener 标注的类必须实现以下至少一个接口：
ServletContextListener
ServletContextAttributeListener
ServletRequestListener
ServletRequestAttributeListener
HttpSessionListener
HttpSessionAttributeListener

无法去定义监听器的顺序

报表

报表的定义
以格式化的形式输出数据
对数据进行分组 汇总 计算等操作
通过报表 图表 或嵌入图片图像等形式来丰富数据的显示

报表生成的关键要素

1 后台数据抽取

2 数据项逻辑运算

3 前台表格展现

报表在项目中的地位

1 面向管理层与决策层

2 充分展现系统数据价值

统计信息的特征

1 数量性

2 综合性