Servlet生命周期：

* init（等初始化完成才会接收请求） 、
* service（doGet、doPost等）、
* destroy(等处理完所有请求后才执行销毁操作，可以通过shutdown.bat执行)

ServletConfig：对应某个serlvet的配置信息，用于初始化serlvet初始化数据

ServeltContext：表示可访问整个web应用的资源。

读取属性，读取初始化参数，访问程序文件等

以”/”作为开头的路径的问题。

1. 本应用的根 this.getServletContext().getResourceAsStream("/a.txt");
2. Web服务器的应用集合的根 form action=”/ctx/sss.jsp”

redirect重定向 全新请求，有不同的请求数据

forward请求转发：同一请求，有相同的请求数据

ServletContextListener：监听应用初始化、应用销毁

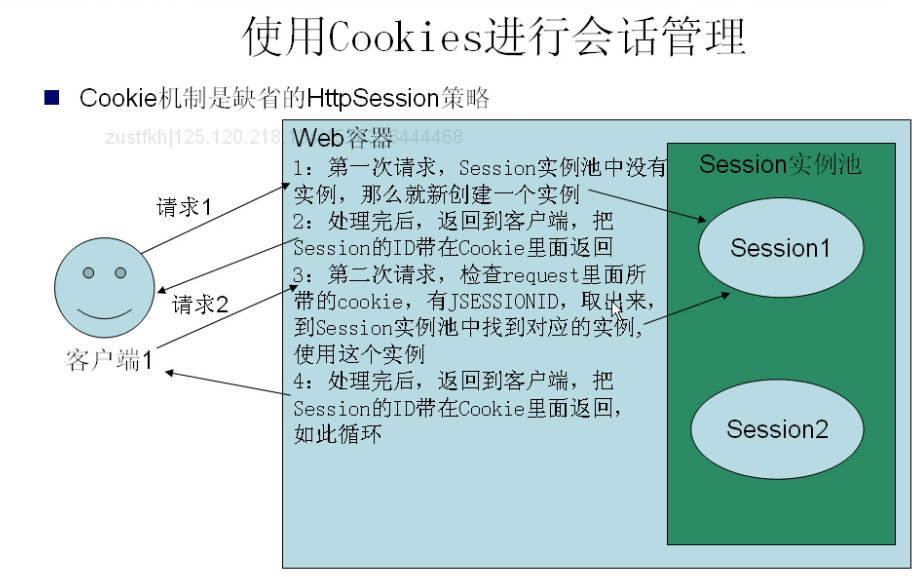
Session：每个客户端根据jsessionID对应一个session内存块，存储于服务器内存中。

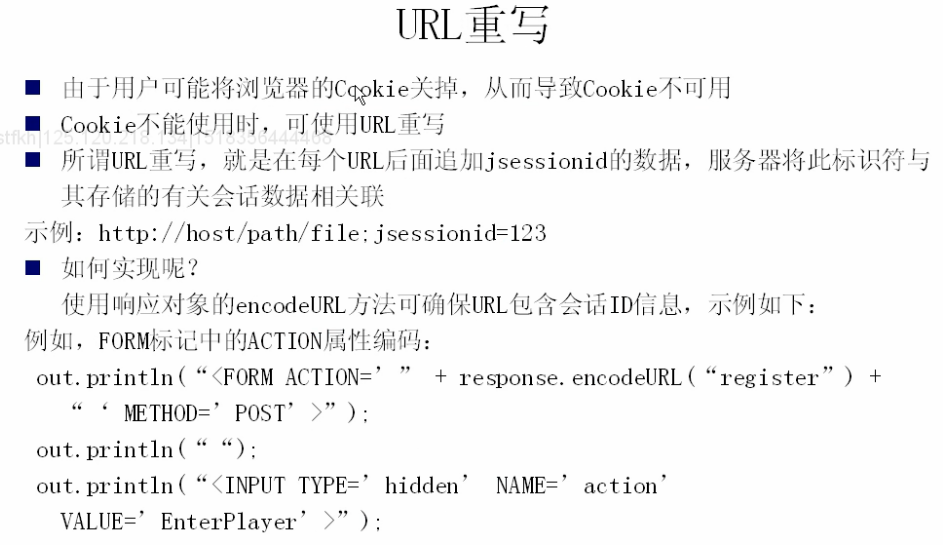
ServletContext服务端共享，不区分客户端

Session服务端共享块，区分客户端

Cookies：该域名下（路径）的所有Cookies在每次一请求中均发送给web服务器。

Cookie机制是缺省的HttpSession策略。默认会将sessionID保存到cookies中来区分浏览器





如果客户端禁止cookies，可以使用如下替代：



servlet并发问题 –> synchronized

web应用中六中属性范围

|  |  |  |
| --- | --- | --- |
| **范围** | **线程安全** | **说明** |
| 局部变量 | 是 | 每个线程各自保存 |
| 实例变量 | 否 | 多线程共享 |
| 类变量 | 否 |  |
| 请求范围 | 是 | ApI：ServletRequest接口  （request） |
| 会话范围 | 否（理论上存在） | 用户可能有多个浏览器活动，访问同一个web资源。  HttpSession（session） |
| 应用范围 | 否 | Web应用中所有servlet共用资源  ServletContext（application） |

Web安全

* 认证（权限匹配）、
* 授权（权限分配）、
* 维护数据完整性（https）、
* 访问跟踪（每次访问记录过程、产生日志）、
* 处理web攻击