

## 课程介绍:

通过前面知识点的学习，我们对于请求的处理已经可以说比较灵活了，但是还不够。我们再介绍两个重要的对象 `ServletContext` 对象和 `ServletConfig` 对象

## ServletContext 对象:

问题:

`Request` 解决了一次请求内的数据共享问题，`session` 解决了用户不同请求的数据共享问题，那么不同的用户的数据共享该怎么办呢？

解决:

使用 `ServletContext` 对象

作用:

解决了不同用户的数据共享问题

原理:

`ServletContext` 对象由服务器进行创建，一个项目只有一个对象。不管在项目的任意位置进行获取到的都是同一个对象，那么不同用户发起的请求获取到的也就是同一个对象了，该对象由用户共同拥有。

特点:

服务器进行创建

用户共享

一个项目只有一个

生命周期:

服务器启动到服务器关闭

作用域:

项目内

使用:

获取 `ServletContext` 对象

使用作用域进行共享数据流转

获取 `web.xml` 中的全局配置

获取 `webroot` 下项目资源流对象

获取 `webroot` 下资源绝对路径

案例: 网页浏览器次数统计, 详见源码

## ServletConfig 对象:

问题:

使用 `ServletContext` 对象可以获取 `web.xml` 中的全局配置文件,

在 `web.xml` 中

每个 `Servlet` 也可以进行单独的配置, 那么该怎么获取配置信息呢?

解决:

使用 `ServletConfig` 对象

作用：

ServletConfig 对象是 Servlet 的专属配置对象，每个 Servlet 都单独拥有一个 ServletConfig 对象，用来获取 web.xml 中的配置信息。

使用：

获取 ServletConfig 对象

获取 web.xml 中 servlet 的配置信息