

后盾网 人人做后盾

www.houdunwang.com

PHP会话控制

后盾网 2011-2016 v5.0

无状态

- 无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

无状态性客户端与服务器端连接通信运行速度快，服务器应答也快。因为无状态，协议对事务处理是没有记忆性的是独立的。所以为了满足后面事务需要前面的事务有关信息，所以事务有关的信息要在协议外发送。

HTTP特性

Cookie

- 最早是网景公司的前雇员Lou Montulli在1993年3月的发明。Cookie是当你浏览某网站时，服务器存储在你机器上的一个小文本文件，再次来到该网站时，网站通过读取Cookie，得知你的相关信息，就可以做出相应的动作。

Cookie传递

- Cookie是利用了网页代码中的HTTP头信息进行传递的，浏览器的每一次网页请求，都可以伴随Cookie传递。
服务器将Cookie添加到网页的HTTP头信息中，伴随网页数据传回到你的浏览器，浏览器会根据你电脑中的Cookie设置选择是否保存这些数据。
如果浏览器不允许Cookie保存，则关掉浏览器后，这些数据就消失。

什么是COOKIE?

生命周期

- Cookie有一个Expires（有效期）属性，这个属性决定了Cookie的保存时间，服务器可以通过设定Expires字段的数值，来改变Cookie的保存时间。

如果不设置该属性，或设为0，那么Cookie只在浏览网页期间有效，关闭浏览器，这些Cookie自动消失。

COOKIE生命周期

浏览器处理Cookie方式

- 数量
 - 各个浏览器中会不同，同一浏览器版本也不同
 - IE8允许每个域保存50个cookie
 - Firefox允许每个域保存50个cookie
- 大小：
 - 一般来说允许4Kb，不同浏览器，不同版本略有不同
- 当Cookie的数量达到上限时的删除方式
 - 1：将最近使用最少的Cookie删除
 - 2：随机删除

COOKIE数量与大小

setcookie (name ,value, expire,path)

- name Cookie 的名字
value Cookie 的值
expire Cookie 过期的时间
path Cookie 在服务器端的有效路径

示例

- setcookie("webname","后盾网");

生成Cookie

操作对象

- 通过序列化实现

serialize

- `string serialize (mixed $value)`
产生一个可存储的值的表示

unserialize

- `mixed unserialize (string $str [, string $callback])`
从已存储的表示中创建 PHP 的值

COOKIE储存数组或对象

- session从用户访问页面开始，到断开与网站连接为止，形成一个会话的生命周期。在会话期间，分配客户唯一的一个SessionID，用来标识当前用户，与其他用户进行区分。
- session会话时，sessionid会分别保存在客户端和服务端两个位置，对于客户端使用临时的Cookie保存（Cookie名称为PHPSESSID）或者通过URL字符串传递，服务器端也以文本文件形式保存在指定的Session目录中。
- session通过ID接受每一个访问请求，从而识别当前用户、跟踪和保持用户具体资料，以及Session变量（在Session活动期间，可在Session中存储数字或文字资料），比如session_name等等，这些变量信息保存在服务器端。

SESSION是什么？

session_start()

- 开始一个会话或者返回已经存在的会话，判断客户端有无session_id，如果没有，在服务器端写入SESSION文件（或者通过数据库等完成）发送写session_id的cookie头信息。如果有客户端发来的session_id找到session数据

session使用方法

- 同get post等使用方法相同，\$_SESSION['var']即可使用

开启SESSION

session_id()

- 获取设置session_

session_name()

- 获取设置session_name

更改session配置

`unset($_SESSION['web'])`

- 删除session变量

`$_SESSION=array()`

- 删除所有session变量，不删除session文件

`session_unset()`

- 释放当前在内存中已经创建的所有\$_SESSION变量，但不删除session文件以及不释放对应的session id

`session_destroy()`

- 删除当前用户对应的session文件以及释放session id，内存中的\$_SESSION变量内容依然保留

释放用户的session所有资源

- `session_unset();session_destroy();`

清除SESSION

`session_save_path()`

- 会话数据的路径。如果指定的路径,将数据保存到路径中。

更改session保存位置

`session.gc_probability = 1`

- session清除基数

`session.gc_divisor = 100`

- 定义在每次初始化会话时，启动垃圾回收程序的概率。这个收集概率计算公式如下： $\text{session.gc_probability} / \text{session.gc_divisor}$ ；对会话页面访问越频繁，概率就应当越小。建议值为1/1000~5000。

`session.gc_maxlifetime = 1440`

- 超过此参数所指的秒数后，保存的数据将被视为'垃圾'并由垃圾回收程序清理。

SESSION垃圾回收

通过session完成用户登录模块

作业
