PHP 实践之路 - 面向对象篇课程

主讲：孙胜利

新浪微博：@私房库

微博主页：http://weibo.com/sifangku

第三章 数据库中保存会话

一、cookie 回顾

二、session 回顾

三、在数据库里保存 session 信息

cookie 回顾

技术目的：

存储少量数据在客户端，每个网站都可存储一些属于自己网站的 cookie 数据，

这些 cookie 数据在每次访问对应网站的时候会自动的传递给服务器端，

PHP 会自动的读取发送过来的 cookie 数据到$_COOKIE 中，

服务器端返回数据给客户端时可以指定客户端存储数据到 cookie 中

session 回顾

1、session 技术的目的：

对访问网站的所有客户进行自动登记，自动登记的目的是为了自动区分各自的身份。

2、具体的技术细节是什么？

当有客户初次访问我们的网站（这边的初次，指的是我们的网站开启 session 技术之后的初次）：

1）PHP 为每一个初次访问我们网站的用户分配一个唯一的身份号码（sessionid）

2）PHP 在服务器上为该客户建立相关的资料库（资料库里可以保存各种各样的相关数据）

假如其中有个客户 A，在我们网站上注册过会员，且这个客户 A 想现在登录的话我们可以这样子做：

1>让这个用户输入用户名和密码，如果正确无误，我们就可以断定这个人是我们网站的会员

2>这时我们就可以在资料库里保存这个客户的用户名、会员的 ID 号等等

3）服务器端在返回数据（这些数据里也包括我们服务器为这个客户分配的身份号码）给客户端时，都会要求客户端将 身份号码 保存起来（默认是保存在 cookie 中）

当客户再次请求我们的网站时

1）会自动带上 cookie 中的数据（身份号码）

2）php 根据这个身份号码，自动的取得保存在服务器上的相对应的资料库里的资料

3）我们也可以根据资料库里面的具体内容（比如是否有用户名）来判断这个客户是否是我们网站的注册会员，来确定他的身份

4）我们依然可以在资料库里存各种各样的内容

5）其他的客户也是同样的道理

注：

1）存放在客户端的 cookie 中的卡号（sessionid）是会过期的

session.cookie_lifetime

session.name

指定会话名以用做 cookie 的名字。只能由字母数字组成，默认为 PHPSESSID。

在服务器端可以通过 session_name()来获取该名字

提示：把 sessionid 放在在 url 里，以 GET 方式传递给服务器端或者用隐藏表单的方式传给服务器端也是可以的，这种情况适用于客户端禁用 cookie 机制或者服务器端也禁用使用 cookie 机制来保存身份号码的情况。

建议对于禁用浏览器 cookie 功能的客户，可以直接给出让其开启 cookie 的提示。

2）服务器端保存的每一个访问我们网站的客户的资料库即 session 信息也是会过期的

session.gc_maxlifetime

过期的 session，PHP 会自动的回收（被回收的时机取决于概率的大小）

session.gc_probability　　默认为 1

session.gc_divisor　　默认为 100

此概率用 gc_probability/gc_divisor 计算得来。例如 1/100 意味着在每个请求中有 1% 的概率启动 gc 进程(garbage collection 垃圾回收)。

在数据库里保存 session 信息

1、建立数据表

#	名字	类型	整理	属性
1	id	char(32)	utf8_general_ci	
2	data	varchar(2550)	utf8_general_ci	
3	expire	int(11)		UNSIGNED

2、使用 session_set_save_handler()设置用户自定义会话存储函数（接管 session 的默认存取机制）

自 PHP 5.4 之后可以使用以下形式的参数

session_set_save_handler(实现了 SessionHandlerInterface 接口的对象)

实现 SessionHandlerInterface 接口必须实现下列方法：

1）abstract public bool open (string $save_path , string $name)

启动会话时执行（session_start()）

2）abstract public string read (string $session_id)

读取会话数据到$_SESSION 时执行（session_start()）

注意：

1>该函数可以返回空字符串

2>该函数也可以返回特定格式的字符串，这个字符串会被 PHP 自动反操作成数组形式放入 $_SESSION 中，具体是怎么操作的，我们就不用管了，这是 PHP 内部做的事情！

PHP 配置文件：session.serialize_handler 定义用来序列化 / 解序列化的处理器名字。

当前支持 ：

php，默认

php_serialize ，注：自 PHP 5.5.4 起可以使用

wddx，　注：如果 PHP 编译时加入了 WDDX 支持，则只能用 WDDX

3>其他形式的数据都是无效的！不能被 PHP 内部操作成功且放入$_SESSION 中

3）abstract public bool gc (int $maxlifetime)

垃圾收集器回调周期，以清除旧的会话数据被 PHP 内部调用。

4）abstract public bool write (string $session_id , string $session_data)

存储会话数据时执行(一般是在 PHP 执行即将结束时)

5）abstract public bool destroy (string $session_id)

销毁会话数据时执行，比如执行 session_destroy()时

6）abstract public bool close (void)

在 write 函数执行完成时执行，它像类里面的析构函数