



用户记住密码的功能的实现

cookie与session的应用

HTTP协议的短板

- ❖ 众所周知，HTTP 是一个无状态协议，所以客户端每次发出请求时，下一次请求无法得知上一次请求所包含的状态数据，如何能把一个用户的状态数据关联起来呢？

cookie简介

Cookie是存储在客户端浏览器中的数据，我们通过Cookie来跟踪与存储用户数据。一般情况下，Cookie通过HTTP headers从服务端返回到客户端。多数web程序都支持Cookie的操作。

因为Cookie是存在于HTTP的标头之中，所以必须要在其他信息输出以前进行设置，类似于header函数的使用限制。

Cookie的处理步骤

- ❖ cookie 是 http 协议的一部分
- ❖ 1、服务器向客户端发送 cookie。
- ❖ 2、浏览器将 cookie 保存。
- ❖ 3、每次请求浏览器都会将 cookie 发向服务器。

PHP对Cookie的操作

PHP通过**setcookie** () 函数进行Cookie的设置，任何从浏览器发回的Cookie，PHP都会自动的将他存储在**\$_COOKIE**的全局变量之中，因此我们可以通过**\$_COOKIE['key']**的形式来读取某个Cookie值。

设置cookie

- ❖ 在PHP中，一般对COOKIE的设置操作是通过
- ❖ `setcookie (name,value,expire,path,domain)`
- ❖ 来完成的，此函数具有5个参数，分别是cookie名，cookie值，cookie过期时间，cookie有效路径和cookie的有效域。

Cookie的缺陷

- ❖ cookie 虽然很方便，但是使用 cookie 有一个很大的弊端，cookie 中的所有数据在客户端就可以被修改，数据非常容易被伪造，那么一些重要的数据就不能存放在 cookie 中了，而且如果 cookie 中数据字段太多会影响传输效率。

cookie相对不是太安全，容易被盗用导致cookie欺骗

单个cookie的值最大只能存储4k

每次请求都要进行网络传输，占用带宽

session简介

session是将用户的会话数据存储在服务端，没有大小限制，通过一个session_id进行用户识别，PHP默认情况下session id是通过cookie来保存的，因此从某种程度上来说，session依赖于cookie。但这不是绝对的，session id也可以通过参数来实现，只要能将session id传递到服务端进行识别的机制都可以使用session。

小结

- ❖ Cookie相当于一张通行证，由服务器进行发放，存储在用户端。
- ❖ Session更像是一个花名册，存储在服务器端，当用户登录时，运用用户所提供的信息来匹配到用户数据。