

# Ruby CGI Cookie

HTTP协议是无状态协议。但对于一个商业网站，它需要保持不同的页面间的会话信息。

如用户在网站注册过程中需要跳转页面，但又要保证之前填写的信息不丢失。

这种情况下 Cookie 很好的帮我们解决了问题。

## Cookie 是如何工作的？

几乎所有的网站设计者在进行网站设计时都使用了Cookie，因为他们都想给浏览网站的用户提供一个更友好的、人文化的浏览环境，同时也能更加准确地收集访问者的信息。

### 写入和读取

Cookies集合是附属Response对象及Request对象的数据集合，使用时需要在前面加上Response或Request。

用于给客户机发送Cookies的语法通常为：

当给不存在的Cookies集合设置时，就会在客户机创建，如果该Cookies已存在，则会被代替。由于Cookies是作为HTTP传输的头信息的一部分发给客户机的，所以向客户机发送Cookies的代码一般放在发送给浏览器的HTML文件的标记之前。

如果用户要读取Cookies，则必须使用Request对象的Cookies集合，其使用方法是：需要注意的是，只有在服务器未被下载任何数据给浏览器前，浏览器才能与Server进行Cookies集合的数据交换，一旦浏览器开始接收Server所下载的数据，Cookies的数据交换则停止，为了避免错误，要在程序和前面加上response.Buffer=True。

### 集合的属性

- **1.Expires属性**：此属性用来给Cookies设置一个期限，在期限内只要打开网页就可以调用被保存的Cookies，如果过了此期限Cookies就自动被删除。如：设定Cookies的有效期到2004年4月1日，到时将自动删除。如果一个Cookies没有设定有效期，则其生命周期从打开浏览器开始，到关闭浏览器结束，每次运行后生命周期将结束，下次运行将重新开始。
- **2.Domain属性**：这个属性定义了Cookies传送数据的唯一性。若只将某Cookies传送给\_blank">搜狐主页时，则可使用如下代码：
- **3.Path属性**：定义了Cookies只发给指定的路径请求，如果Path属性没有被设置，则使用应用程序的缺省路径。
- **4.Secure属性**：指定Cookies能否被用户读取。
- **5、Name=Value**：Cookies是以键值对的形式进行设置和检索的。

## Ruby 中处理Cookies

你可以创建一个名为 cookie 的对象并存储文本信息，将该信息发送至浏览器，调用 CGI.out 设置cookie的头部：

### 实例

```
#!/usr/bin/ruby
require "cgi"
cgi = CGI.new("html4")
cookie = CGI::Cookie.new('name' => 'mycookie',
'value' => 'Zara Ali',
'expires' => Time.now + 3600)
```

```
cgi.out('cookie' => cookie) do
  cgi.head + cgi.body { "Cookie stored" }
end
```

接下来我们回到这个页面，并查找cookie值，如下所示：

实例

```
#!/usr/bin/ruby
require "cgi"
cgi = CGI.new("html4")
cookie = cgi.cookies['mycookie']
cgi.out('cookie' => cookie) do
  cgi.head + cgi.body { cookie[0] }
end
```

CGI::Cookie对象实例化时包含以下参数：

参数	描述
name	规定 cookie 的名称。
value	规定 cookie 的值。
expire	规定 cookie 的有效期。
path	规定 cookie 的服务器路径。
domain	规定 cookie 的域名。
secure	规定是否通过安全的 HTTPS 连接来传输 cookie。