

# Ruby CGI Session

CGI::Session 可以为用户和CGI环境保存持久的会话状态，会话使用后需要关闭，这样可以保证数据写入到存储当中，当会话完成后，你需要删除该数据。

## 实例

```
#!/usr/bin/ruby
require 'cgi'
require 'cgi/session'
cgi = CGI.new("html4")
sess = CGI::Session.new( cgi, "session_key" => "a_test",
"prefix" => "rubysess.")
lastaccess = sess["lastaccess"].to_s
sess["lastaccess"] = Time.now
if cgi['bgcolor'][0] =~ /[a-z]/
sess["bgcolor"] = cgi['bgcolor']
end
cgi.out{
cgi.html {
cgi.body ("bgcolor" => sess["bgcolor"]){
"The background of this page" +
"changes based on the 'bgcolor'" +
"each user has in session." +
"Last access time: #{lastaccess}"
}
}
}
```

访问 `/cgi-bin/test.cgi?bgcolor=red` 将跳转到指定背景颜色的页面。

会话数据存在在服务器的临时文件目录中，`prefix` 参数指定了会话的前缀，将作为临时文件的前缀。这样你在服务器上可以轻松的认识不同的会话临时文件。

## CGI::Session 类

CGI::Session 保持了用户与 CGI 环境的持久状态。会话可以在内存中，也可以在硬盘上。

## 类方法

Ruby 类 Class CGI::Session 提供了简单的方法来创建 session:

```
CGI::Session::new( cgi[, option])
```

启用一个新的 CGI 会话并返回相应的 CGI::Session 对象。选项可以是可选的哈希，可以是以下值：

- **session\_key:** 键名保存会话 默认为 `_session_id`。
- **session\_id:** 唯一的会话 ID。自动生成

- **new\_session:** 如果为true，为当前会话创建一个新的Session id。如果为 false, 通过 session\_id 使用已存在的 session 标识。 如果省略该参数，如果可用则使用现有的会话，否则创建一个新的。
- **database\_manager:** 用于保存 sessions 的类，可以是 CGI::Session::FileStore or CGI::Session::MemoryStore。默认为 FileStore。
- **tmpdir:** 对于 FileStore, 为 session 的错存储目录。
- **prefix:** 对于 FileStore, 为 session 文件的前缀。

实例化方法

序号	方法描述
1	<b>[]</b> 返回给定 key 的值。查看实例。
2	<b>[]=</b> 设置给定 key 的值。 查看实例。
3	<b>delete</b> 调用底层数据库管理的删除方法。对于 FileStore, 删除包含 session 的物理文件。对于 MemoryStore, 从内存中移除 session 数据。
4	<b>update</b> 调用底层数据库管理的更新方法。对于 FileStore, 将 session 写入到磁盘中。对于 MemoryStore则无效果。

 [点我分享笔记](#)