

cookie、session

cookie、session

[什么是 Cookie?](#)

[在浏览器端操作cookie](#)

[新建cookie](#)

[获取cookie](#)

[在express操作cookie](#)

[写入cookie](#)

[读取cookie](#)

[cookie签名](#)

[删除cookie](#)

[session](#)

[session是什么](#)

[读写session](#)

[Session和Cookie的主要区别](#)

[案例](#)

什么是 Cookie?

Cookie 是一些数据, 存储于客户端电脑上的文本文件中。当 web 服务器向浏览器发送 web 页面时, 在连接关闭后, 服务端不会记录用户的信息。Cookie 的作用就是用于解决 "如何记录客户端的用户信息": 当用户访问 web 页面时, 他的名字可以记录在 cookie 中。在用户下一次访问该页面时, 可以在 cookie 中读取用户访问记录。cookie 以名/值对形式存储, 如下所示:

```
username=byesame
```

当浏览器从服务器上请求 web 页面时, 属于该页面的 cookie 会被添加到该请求中。服务端通过这种方式来获取用户的信息。

在浏览器端操作cookie

新建cookie

```
document.cookie="username=byesame";
```

您还可以为 cookie 添加一个过期时间 (以 UTC 或 GMT 时间)。默认情况下, cookie 在浏览器关闭时删除:

```
document.cookie="username=byesame; expires=Thu, 18 Dec 2013 12:00:00 GMT";
```

您可以使用 path 参数告诉浏览器 cookie 的路径。默认情况下, cookie 属于当前页面。

```
document.cookie="username=byesame; expires=Thu, 18 Dec 2013 12:00:00 GMT; path="/;
```

获取cookie

在 JavaScript 中, 可以使用以下代码来读取 cookie:

```
var x = document.cookie;
```

在express操作cookie

写入cookie

`res.cookie(name, value [, options])`

```
const express=require('express');
var server=express();
//set cookie
server.use('/a.html', function (req, res){
  // path cookie目录 maxAge 过期时间 毫秒
  res.cookie('user', 'zhang', {
    httpOnly: false,
    path: '/',
    maxAge: 30*24*3600*1000
  });
  res.send('ok');
});
server.listen(8080);
```

读取cookie

读取cookie需要用到中间件

```
const express=require('express');
const cookieParser=require('cookie-parser');
var server=express();
//get cookie
server.use(cookieParser());
server.use('/', function (req, res){
  console.log(req.cookies);
  res.send('ok');
});
server.listen(8080);
```

cookie签名

为了防止用户在客户端篡改cookie的值 需要用到cookie签名

```
const express = require('express');
const cookieParser = require('cookie-parser');
var server = express();
//cookie
//在解析cookie的时候需要将密钥传入
server.use(cookieParser('wesdfw4r34tf'));
// 签名为了防止用户自己篡改 cookie
server.use('/', function (req, res) {
  // 签名密钥
  req.secret = 'wesdfw4r34tf';
  // 签名
  res.cookie('user', 'yan', {signed: true});
  console.log('签名cookie: ', req.signedCookies);
  console.log('无签名cookie: ', req.cookies);
  res.send('ok');
});
server.listen(8080);
```

删除cookie

`res.clearCookie(name [, options])`

```
res.cookie('name', 'tobi', { path: '/admin' });
res.clearCookie('name', { path: '/admin' });
```

session

session是什么

在WEB开发中，服务器可以为每个用户浏览器创建一个会话对象（session对象），注意：一个浏览器独占一个session对象(默认情况下)。因此，在需要保存用户数据时，服务器程序可以把用户数据写到用户浏览器独占的session中，当用户使用浏览器访问其它程序时，其它程序可以从用户的session中取出该用户的数据，为用户服务。

读写session

在express中操作session需要中间件

```
const cookieSession = require('cookie-session');
//配置中间件
server.use(cookieSession({
  name: 'sess',//客户端cookie名字
  keys: ['aaa', 'bbb', 'ccc'],//cookie签名
  maxAge: 2 * 3600 * 1000,// session过期时间 毫秒
}));

server.use(function(req,res){
  //通过操作req.session来读写cookie
})
```

Session和Cookie的主要区别

1. cookie数据存放在客户的浏览器上，session数据放在服务器上。
2. cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗
考虑到安全应当使用session。
3. session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能考虑到减轻服务器性能方面，应当使用cookie。
4. 单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。
5. 建议：
将登陆信息等重要信息存放为session
其他信息如果需要保留，可以放在cookie中

案例

通过cookie或session来保存用户登录状态