

# Express Cookie 的基本使用

主讲教师：（大地）

合作网站：[www.itying.com](http://www.itying.com) （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

一、Cookie 简介.....	1
二、Cookie 特点.....	1
三、Cookie 的使用.....	2
四、加密 Cookie.....	3

## 一、Cookie 简介

● cookie 是存储于访问者的计算机中的变量。可以让我们用同一个浏览器访问同一个域名的时候共享数据。

● HTTP 是无状态协议。简单地说，当你浏览了一个页面，然后转到同一个网站的另一个页面，服务器无法认识到这是同一个浏览器在访问同一个网站。每一次的访问，都是没有任何关系的。

● Cookie 是一个简单到爆的想法：当访问一个页面的时候，服务器在下行 HTTP 报文中，命令浏览器存储一个字符串；浏览器再访问同一个域的时候，将把这个字符串携带到上行 HTTP 请求中。第一次访问一个服务器，不可能携带 cookie。必须是服务器得到这次请求，在下行响应报头中，携带 cookie 信息，此后每一次浏览器往这个服务器发出的请求，都会携带这个 cookie。

## 二、Cookie 特点

- cookie 保存在浏览器本地
- 正常设置的 cookie 是不加密的，用户可以自由看到；
- 用户可以删除 cookie，或者禁用它
- cookie 可以被篡改
- cookie 可以用于攻击
- cookie 存储量很小。未来实际上要被 localStorage 替代，但是后者 IE9 兼容。

### 三、Cookie 的使用

Express 中要使用 Cookie 的话，我们需要使用 cookie-parser 模块来实现

```
1.安装    cnpm instal cookie-parser --save

2.引入    var cookieParser = require('cookie-parser');

3.设置中间件

    app.use(cookieParser());

4.设置 cookie

    res.cookie("name",'zhangsan',{maxAge: 900000, httpOnly: true});

5. 获取 cookie

    req.cookies.name
```

#### Cookie 属性说明

属性	描述
domain	域名 name=value: 键值对, 可以设置要保存的 Key/Value, 注意这里的 name 不能和其他属性项的名字一样
Expires	过期时间 (秒), 在设置的某个时间点后该 Cookie 就会失效, 如 expires=Wednesday, 09-Nov-99 23:12:40 GMT
maxAge	最大失效时间 (毫秒), 设置在多少后失效
secure	当 secure 值为 true 时, cookie 在 HTTP 中是无效, 在 HTTPS 中才有效
Path	表示 cookie 影响到的路路径, 如 path=/. 如果路径不能匹配时, 浏览器则不发送这个 Cookie
httpOnly	是微软对 COOKIE 做的扩展。如果在 COOKIE 中设置了“httpOnly”属性, 则通过程序 (JS 脚本、applet 等) 将无法读取到 COOKIE 信息, 防止 XSS 攻击产生
singed	表示是否签名 cookie, 设为 true 会对这个 cookie 签名, 这样就需要用 res.signedCookies 而不是 res.cookies 访问它。被篡改的签名 cookie 会被服务器拒绝, 并且 cookie 值会重置为它的原始值

### 设置 cookie 的几种方法

```
res.cookie('rememberme', '1', { maxAge: 900000, httpOnly: true })  
res.cookie('name', 'tobi', { domain: '.example.com', path: '/admin', secure: true });  
res.cookie('rememberme', '1', { expires: new Date(Date.now() + 900000), httpOnly: true });
```

### 获取 cookie

```
req.cookies.name
```

### 删除 cookie

```
res.cookie('rememberme', '', { expires: new Date(0)});  
res.cookie('username', 'zhangsan', {domain: '.ccc.com', maxAge: 0, httpOnly: true});
```

## 四、加密 Cookie

### 1. 配置中间件的时候需要传参

```
var cookieParser = require('cookie-parser');  
app.use(cookieParser('123456'));
```

### 2. 设置 cookie 的时候配置 signed 属性

```
res.cookie('userinfo', 'hahaha', {domain: '.ccc.com', maxAge: 900000, httpOnly: true, signed: true});
```

### 3. signedCookies 调用设置的 cookie

```
console.log(req.signedCookies);
```