

Session 的使用

主讲教师：（大地）

合作网站：www.itying.com

一、 Session 简单介绍.....	1
二、 Session 的工作流程.....	2
三、 express-session 的使用:.....	2
四、 express-session 的常用参数:.....	3
五、 express-session 的常用方法:.....	4
六、 负载均衡配置 Session，把 Session 保存到数据库里面.....	4
七、 Cookie 和 Session 区别.....	5

一、Session 简单介绍

session 是另一种记录客户状态的机制，不同的是 **Cookie** 保存在客户端浏览器中，而 **session** 保存在服务器上。

Session 的用途：

session 运行在服务器端，当客户端第一次访问服务器时，可以将客户的登录信息保存。

当客户访问其他页面时，可以判断客户的登录状态，做出提示，相当于登录拦截。

session 可以和 **Redis** 或者**数据库**等结合做持久化操作，当服务器挂掉时也不会导致某些客户信息(购物车)丢失。

二、Session 的工作流程

当浏览器访问服务器并发送第一次请求时，服务器端会创建一个 session 对象，生成一个类似于 key,value 的键值对，然后将 key(cookie)返回到浏览器(客户端)，浏览器下次再访问时，携带 key(cookie)，找到对应的 session(value)。客户的信息都保存在 session 中

三、express-session 的使用:

1.安装 express-session

```
cnpm install express-session --save
```

2.引入 express-session

```
var session = require("express-session");
```

3.设置官方文档提供的中间件

```
app.use(session({  
  
  secret: 'keyboard cat',  
  
  resave: true,  
  
  saveUninitialized: true
```

```
)))
```

4.使用

设置值 `req.session.username = "张三";`

获取值 `req.session.username`

四、express-session 的常用参数:

```
app.use(session({
  secret: '12345',
  name: 'name',
  cookie: {maxAge: 60000},
  resave: false,
  saveUninitialized: true
}));
```

行加密的字符串.这个

参数	作用
secret	一个 String 类型的字符串，作为服务器端生成 session 的签名。
name	返回客户端的 key 的名称，默认为 connect.sid,也可以自己设置。
resave	强制保存 session 即使它并没有变化,。默认为 true。建议设置成 false。 don't save session if unmodified
saveUninitialized	强制将未初始化的 session 存储。当新建了一个 session 且未设定属性或值时，它就处于

	未初始化状态。在设定一个 cookie 前，这对于登陆验证，减轻服务端存储压力，权限控制是有帮助的。（默认：true）。建议手动添加。
cookie	设置返回到前端 key 的属性，默认值为{ path: '/', httpOnly: true, secure: false, maxAge: null }。
rolling	在每次请求时强行设置 cookie，这将重置 cookie 过期时间（默认：false）

五、express-session 的常用方法:

```
req.session.destroy(function(err) {    /*销毁 session*/
})
req.session.username='张三';    //设置 session
req.session.username    //获取 session
req.session.cookie.maxAge=0;    //重新设置 cookie 的过期时间
```

六、负载均衡配置 Session，把 Session 保存到数据库里面

1. 需要安装 express-session 和 connect-mongo 模块

2. 引入模块

```
var session = require("express-session");
const MongoStore = require('connect-mongo')(session);
```

3. 配置中间件

```
app.use(session({
```

```
secret: 'keyboard cat',  
  
resave: false,  
  
saveUninitialized: true,  
  
rolling: true,  
  
cookie: {  
  
    maxAge: 100000  
  
},  
  
store: new MongoStore( {  
  
    url: 'mongodb://127.0.0.1:27017/student',  
  
    touchAfter: 24 * 3600 // time period in seconds  
  
})  
  
}))
```

七、Cookie 和 Session 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。