

express中操作cookie和session教程

一、操作cookie:

1. 设置cookie(给客户端“种”cookie):
直接使用`res.cookie('','',{})`即可。
2. 获取cookie(要第三方中间件):
 - * 安装: `npm i cookie-parser`
 - * 引入: `const cookieParser = require('cookie-parser')`
 - * 使用: `app.use(cookieParser())`
3. 返回给客户端一个cookie:
 - * `res.cookie('username','peiqi',{maxAge:1000*60*60})`

备注: 1.cookie是以: key-value的形式存在的, 前两个参数分别为: key、value。
2.maxAge用于配置cookie有效期(单位毫秒)。
3.如果不传入maxAge配置对象, 则为会话cookie, 随着浏览器的关闭cookie会自动会消失。
4.如果传入maxAge, 且maxAge不为0, 则cookie为持久化cookie, 即使用户关闭浏览器, cookie也不会消失, 直到过了它的有效期。

4. 接收客户端传递过来的cookie:
 - * `req.cookies.xxx` : 获取cookie上xxx属性对应的值。
- 备注: cookie-parser中间件会自动把客户端发送过来的cookie解析到request对象上。

二、操作session (cookie配合session) :

- 1.下载安装: `npm i express-session --save` 用于在express中操作session
- 2.下载安装: `npm i connect-mongo --save` 用于将session写入数据库 (session持久化)
- 3.引入express-session模块:
`const session = require('express-session');`
- 4.引入connect-mongo模块:
`const MongoStore = require('connect-mongo')(session);`
- 5.编写全局配置对象:

```
app.use(session({
  name: 'userid', //设置cookie的key, 默认值是: connect.sid
  secret: 'atguigu', //参与加密的字符串 (又称签名)
  saveUninitialized: false, //是否在存储内容之前创建会话
  resave: true, //是否在每次请求时, 强制重新保存session, 即使他们没有变化
  store: new MongoStore({ //session持久化
    url: 'mongodb://localhost:27017/sessions_container',
    touchAfter: 24 * 3600 //修改频率 (例: //在24小时之内只更新一次)
  }),
  cookie: {
    httpOnly: true, // 开启后前端无法通过 JS 操作cookie
    maxAge: 1000*30 // 设置cookie的过期时间
  },
}));
```
- 6.向session中添加一个xxxx, 值为yyy: `req.session.xxxx = yyy`
- 7.获取session上的xxx属性: `const {xxx} = req.session`

整个过程是:

- 1.客户端第一次发起请求, 服务器开启一个session专门用于存储这次请求的一些信息。

2. 根据配置对象的信息，服务器决定是否进行：**session**持久化等其他操作。
2. 与此同时服务器创建了一个**cookie**，它的**key**我们可以自己指定，但是它的**value**一定是上一步**session**的唯一标识。
3. 服务器将我们指定好的内容添加进**session**对象，例如：`req.session.xxxx = yyy`。
4. 等请求再次过来时，客户端的请求中包含着之前“种”的**cookie**。
5. 服务器检查携带过来的**cookie**是否有效，决定是否去读取对应**session**中的信息。