



nodejs_express_s...l.rar
3.84KB

在express中使用session则必须使用"express-session模块"

如果把session保存在mysql数据库中，则还必须追加引用"express-mysql-session"模块和"mysql模块";

```
const session = require('express-session');  
var MySQLStore = require('express-mysql-session');
```

在express中使用session的关键代码示例

```
const express = require("express");  
const app = express();  
const session = require('express-session');  
var MySQLStore = require('express-mysql-session');  
//创建与mysql相关的参数，注意要先在数据库创建对应的表哟  
const options = {  
  host: 'localhost',  
  port: 3306,  
  user: 'root',  
  password: '123',  
  database: 'mytest', //数据库名,  
  schema: {  
    tableName: 'sessions', //表名  
    columnNames: { //字段名  
      session_id: 'sessionid',  
      expires: 'expires',  
      data: 'data'  
    }  
  }  
};  
app.use(session({
```

```

    key: 'session_cookie_name', //cookie中保存sessionid值的键名,可省, 默认为
connect.sid
    secret: 'session_cookie_secret',//sessionid值加密的盐
    store: new MySQLStore(options),
    resave: false,//可省, 默认值是true,开发中都用false,为true已经过时
    saveUninitialized: false,//可省, 默认是true,开发中使用false, true不适用,为true时,
    不管是否登陆, 都会生成一个session,保存在数据库中。
    //即设置保存sessionid的cookie,可省, 默认为{ path: '/', httpOnly: true, secure:
    false, maxAge: null }
    cookie: {
        path: '/', //可省, 默认为"/", cookie在什么路径下有效, 默认为'/'
        httpOnly: true, //可省, 默认为false,设置浏览器能否操作当前cookie,当为true
    时, 则浏览器中js不能操作cookie
        secure: false, //可省, 默认为false; 值为true是在https中使用的
        maxAge: null, //默认值为null, 值为null时, 则浏览器中cookie随浏览器的关闭
    而销毁, 设置为整型时, 而时间过了才销毁
    }
    });

//为session设置值
req.session.username="小明";
var 取出session中的值
var username = req.session.username;

```