今天学习内容

1:总结node.js

1.1:node.js 运行于服务器端开发平台

1.2:node.js与前端js区别 数据类型

node.js第三方模块

前端js:DOM/BOM

1.3:模块:文件或者目录称为模块

文件模发起人node.js自动化

(function(exports,module,require,\_\_dirname,\_\_filename){

//自己代码

});

exports:向外导出成员

module:当前模块对象

require:fn 引入其它模块并创建指定模块对象

\_\_dirname:当前文件模块目录绝对路径

\_\_filename:

1.4:exports 和 module.exports

exports = modules.exports 底层操作

1.5:模块分类

1:自定义模块

#文件 m.js exports. app.js var m = require(“./m”);

#目录 node\_modules/m6/package.json

app.js require(“m6”);

2:原生模块

Buffer

url 解析url地址: url.parse(str,true)

http 创建http服务器/客户端

var server = http.createServer();

server.listen(3000);

server.on(“request”,(req,res)=>{})

fs

同步:阻塞

var data = fs.readFileSync(fileName);

fs.writeFileSync(fileName,data);

fs.appendFileSync(fileName,data);

异步:非阻塞

fs.readFile(fileName,(err,data)=>{});

fs.writeFile(fileName,data,(err)=>{});

fs.appendFile(fileName,data,(err)=>{})

querystring 解析查询字符串

3:第三方模块

mysql

1:创建连接池

var pool = mysql.createPool({connectionLimit:25});

2:pool.getConnection((err,conn)=>{});

3:conn.query(sql,[id],(err,result)=>{

if(err)throw err;

});

4:select result.length

express:

var app = express();

var server = http.createServer(app);

server.listen(3000);

app.get(请求地址”/list”,(req,res)=>{});

req.query.uid

req.body.uid 需要第三方插件

res.send(html);

res.sendFile(绝对路径+发送文件);

res.json({…});

2:学子商城->实现更多的功能[跨域]

模块一:后台管理员登陆

1. 客户端资源

所有静态资源放在 apache/htdocs/nodeadmin

请求地址 http://127.0.0.1/nodeadmin/index.html

1. 服务器端程序node.js

请求地址 http://127.0.0.1:3000/adminlogin

程序工作流程

1:启动node.js 服务器

2:启动apache 服务器(mysql启动)

3:客户请求 apache nodeadmin/index.html

4:发送登录请求 http://127.0.0.1:3000/adminlogin

注意事项:

如果客户端**跨域请求**服务器资源为了保存session工作正确

需要在ajax请求添加属性

xhrFields:{withCredentials:true}

#-----------------------------------------------------------------------

项目:{跨域}

两台web服务器 {nodejs 服务器业务功能}

{apaceh 服务器接收请求响应html静态资源}

一台数据库服务器 {数据 mysql服务器}

扩展思路与知识储备

(1)项目瓶颈:数据库

原理:数据库中数据保存在磁盘上的某一组文件

解决一:提供SQL查询效率

SELECT uid,uname,age FROM xz\_user;

数据库对象:索引 提高查询效率[select];

降低更新效率[update/delete/insert]

什么项目用:索引 查询多更新少

索引为什么快:工作原理

美团快递:送餐{死心眼} 数据库默认工作方式

美团快递:送餐{机灵}

ALTER TABLE xz\_user ADD INDEX (uname)

SELECT uid FROM xz\_user WHERE uname = “tom”;

400w-🡪没有加索引之前 查询一条记录 2s

400w->加索引之后 查询一条记录 0.00002s

解决二:主从数据库服务器{一个服务器负责写数据<主>}

{多个从服务器负责读数据<从>}

解决三:内存级数据库服务器{NOSQL} 极高数据读写 Redis

数据库读写速度太高->数据库压力太大{出问题} cpu 100%

社区:php/mysql 在线人数 t\_online 2 服务器down

$phpcount = 1; ->redis ++ -- 效率很好

tts:学生登录 8:45~9:30 ->mysql🡪Redis

微博/微信 微博[点赞] 120 Redis

(2)提高项目{PV Page View} 高性能服务器/页面静态化

*Nginx/*

(3)安全角度

SQL 注入 node.js ?

XSS 攻击 脚本 oa[报销/申请办公用品/审批] [确认]

document.all.readonly = true

过滤用户所有输入值/所有危险字符串判断

DDOS 攻击 向某个网站发送大量垃圾 防火墙

解决:网店->天猫->阿里云 6G/s

(4)数据分析数据挖掘(大数据/人工智能)

钱🡪一组新技术集合(hadoop/hbase/mapredure/hive…)

15T🡪分析用户行为->记录-(存储hadoop)

分析mapredure –》

….

人工智能:网易课堂:吴恩达