第一天

Node.js:安装（path?）、使用

Node.js ：

由c++编写

运行js代码

数据库MySQL,mongodb

后台业务，逻辑，功能接口

官网--下载:基数版本--开发

偶数版本--稳定版

Node -v：查看

Path：配置，查看，修改，安装？？？？？？

1. 部署简单
2. 注重约定（约定：底层 fun（a,b,c）{})
3. 进程：（Window任务管理器）

PID查看

（ 程序（记事本-PID）--代码---内存

程序（txt）相似，内存不一样）

4.线程：

5.计算机核心： CPU（“3+4”） ---计算机任务

CPU一次 只能运行一个任务---- ：

一个进程（进程为线程的容器）

（至少一个线程为进程其服务）

1. js为单线程

7. Node.js 使用：先同步，再异步（定时器/大量使用）

同步：按顺序执行，

异步：不等待，直接执行下一步,\*\*\*\*\*

1）实现形式：

A.回调--读取文件--执行一个事件 function(){ alert(‘’) }

B.监听--当事件发生后，状态改变，该怎么做:

console.log(‘’) -- 先先执行 （同步）

process.on(‘exit’,function(){ （异步）

setTimeout(**function**(){

console.log('进程退出');

},1000)

（ --监听后--再执行 退出-完毕）

})--全局对象：储存 查看 当前进程信息

console.log(‘’) -- 先执行 （同步）

阻塞-（一定是同步的）：等待，

非阻塞：单线程 异步

事件驱动-（运行机制）：node

1. node.js 中放入文件:‘命令’

node demo1.js(js文件) --enter键\*\*\*\*\*\*\*\*\*

（console.log）：

\_dirname:当前文件路径目录

\_filename：当前文件

Demo3.js 打开当前文件

Cmd \*\*.js 打开当前文件

1）Ctrl+alt+delete / ctrl+shift+esc:

打开资源管理器（Window任务管理器）--进程 --内存(查看)

2）建立一个定时器

8.Node-运行机制:

1)function aa(){

console.log(1)

}

function bb(){

console.log(2)

}

exports.a=aa 暴露多个

exports.b=bb

2)var cc=require('./mdl.js')

//console.log(cc.a) // 打印 对象

cc.a() //函数 调用

读取文件：min.js

Var fs=require('a') ：引入fs模块，使用其方法

fs.readFile(‘a.txt’ , ’utf-8’, function(err，data){

1）If(err) 错误

2）console.log(data) 读取完后的结果

写入文件：

fs.writeFile(‘b.text’,data,function(err,data){

--复制（写入）到的文件名，内容，函数

If(err)

console.log(‘复制完毕’)

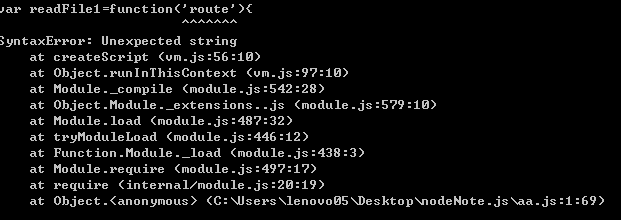
})

})

--创建的目标文件名，编码格式，回调函数

第二天

1.参数



-- fs模块-----”功能”（封装）--(min.js)

2.node.js搭建服务器:

1） <https://www.baidu.com:80/aaaa?id=123>：

https:双方约定的通讯格式

[www.baidu.com:请求服务器（DNS域名解析）](http://www.baidu.com:请求服务器（DNS域名解析）)，

网络指定 ‘服务器’

:80 ：端口号 （名称加端口号）

指定服务器里面的指定的服务

（默认的端口号可省略）

/aaaa 请求当前web服务上面的--资源，

服务端会按此部分内容，解决处理

？Id=123 查询参数

‘ 浏览器’ request 发出请求

‘ 服务器’ response 做出响应

1. 创建http

localhost/127.0.0.1 --本机地址

创建文件，读取html文件，打开创建的IP地址，访问到html文件

1. ‘包’：

Package.json: common.js ：使用require加载，

使用exports或 moduel..exports暴露

npm+ install 本地安装 模块(c盘)\*\*\*\*\*\*\*\*

install -g 全局安装\*\*\*\*\*\*

uninstall 卸载

install -force强行安装

-list 查看命令列表

init --得到package.json

第三天

1.express安装: --cc文件夹

1)npm install -g express-generator(全局安装 express 脚手架)

2)express bird(初始化 项目/文件夹名称) \*8\*\*8\*

3)cd bird (进入)

(app.js:入口文件)

cnpm install (安装依赖包)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--（未成功打开浏览器的情况）

1. npm start:( 启动)/[ node ./bin/www <---package.json ]--打开浏览器localhost:3000

【6) npm init

npm install mysql

var mysql=require(‘mysql’) 引入

1. 下载：

断言模块,

mongodb ： -----mongo.js：

npm init

npm install assert

(assert:代替if（err）{throw err} else{} )

cnpm install mongodb

1. router.js --CC文件夹

var path=require('path')

//内置模块，不用下载 设置静态资源路径，上传图片

app.use(express.static(path.join(\_\_dirname,'img)))】

1. cnpm install jquery

5.后台--接口：

cnpm install node-dev -g

(启动命令：node-dev app-文件夹名称)

-----》更改内容后无需再用命令，直接刷新

1. express:1)+2)+3);cnpm install 2.node app 3.cnpm install express-session --save

4.cnpm install mongodb 5.npm install formidable 6.cnpm install images 7..cnpm install connect-flash 8.node app.js

9.

2.cnpm安装：npm install -g cnpm -registry=https://registry.npm.taobao.org

1. ionic

vue

1. git:

安装：版本，配置

<https://git-scm.com/download/win>

1. $ git init (-所在文件夹)

--出现 .git文件夹（右键使其显示）

1. 网址上克隆-zip--到本地文件夹\*\*\*\*\*\*\*\*\*\*\*
2. git bash here--鼠标右键--所在文件夹)

8.（则能直接访问他人网页）:

1) $ git pull 文件地址（github）

(下载其他人的文件)

2)cd 进入项目（鼠标点击），

3）npm run dev（error--》4)）

4）cnpm install （项目依赖）

5)npm run dev (启动项目)

5.创建秘钥：$ ssh-keygen -t rsa -C “email地址”

公钥pub：

6. $ git log历史记录

$ git reflog 所有记录

$ git status 状态

$ git config -g

$ git add 提交

$ git diff 查看修改的内容

$ cat a.txt 查看内容

$ git checkout -- a.txt 撤销修改的内容

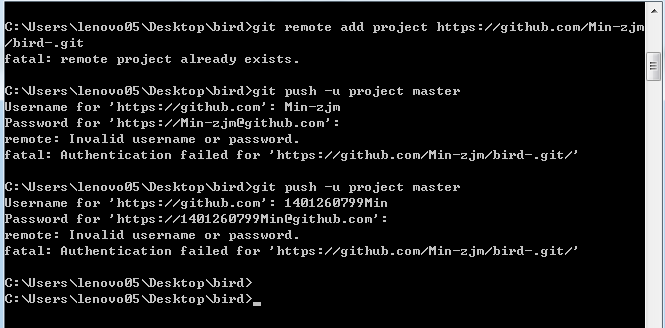
$ git commit

git remote rm origin

git remote add

REPL (read -eval -print-loop)

第四天



1. Git 上传文件
2. 同步.异步执行顺序：：four.js

写入文件的延生；

3.io：

所有涉及到到数据传输，

无阻塞io : 事件队列：先进先出

buffer编码格式--应用--four.js：

（http://nodejs.cn/api/buffer.html）

: 异步读取--同步事件：

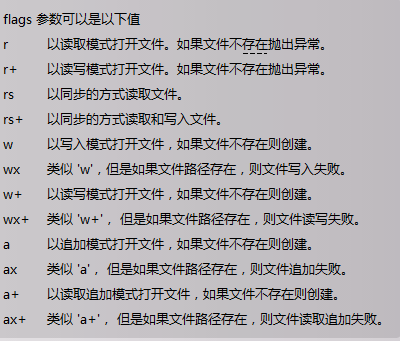
var fs=require(‘fs’)

//fs.readFile(‘a.txt”,’编码‘，function（err,data）{

//Console.log(‘data’)

//}）

flog 参数应用 --写入bas.txt（创建了一个txt，写入小明（写入实现了’非‘覆盖）--four.js



\*\*流的运用：--four.js（这边读出 写入到那边

(数据交换)，加速，减少内存）

1）写入

2) 读取

第五天

CC文件夹

**路由，router.js：**  **网页获取图片，上传，**

1.//创建一个express对象--创建一个服务器

var express=require('express');

var app=express();

--express.js(cc文件夹)

1. app.get / app.use(一个参数，两个参数)
2. var path=require('path') ############

//内置模块，不用下载 --'上传图片'\*\*\*\*\*\*\*\*\*\*\*A.

//console.log(path,join(\_\_dirname,'img'))

//express.static() //设置静态资源路径

app.use(express.static(path.join(\_\_dirname,'img')))//加载-路径(输入：[http://localhost:5000/hos.jpg)\*\*\*\*\*\*](http://localhost:5000/hos.jpg)******)

4. var express=require('express');

var app=express();

var path=require('path')

1. router：router.js

第六天

####################

2.Micsoft Access(微软数据库)

1.Oracle(甲骨文)

3.mongoDB-小，方便快捷,

MySQL-大中小公司

1. 数据库 管理系统 组成：

服务器，客服端--cs

服务器，浏览器--bs

1. 关系型 (表)-行/列对齐
2. 当前数据库：（4567）

A.安装：

1. mysql -h localhost -u

名(rooter)字 -p

2）输入密码

B.数据库操作：

（计算机程序：mysql（unicode那 款）

-->密码： admin）

1）show databases ; 查看有几款数据库

2）*create database test1;*\*创建数据库 (没有的情况下,创建test--数据库名称)

3）show databases;

4）use test1; 使用（切换）数据库\*\*\*\*\*

5）查看切换到的数据库：

select database();

6）drop database test1; 删除数据库

C.表格：

1. *create table t2( \*-创建表格 表头t2*

*-->id int not null auto\_increment primary key,*

( id int(10) )

(意为：表头输入数字,且不能为空,列自增)

-->*name char(20),*

(string：定长0-255:最小长度)

（varchar(20); 变长字符：最大长度） -->*sex char(2),*

-->*birth date);*

（日期：（birth） date 年.月.日）

1. show tables; （查看创建的表格）
2. describe t2; (查看表格结构 /内容)
3. drop table t2; (删除表格)

1. *insert into t2 value*

*(1,’小明’,‘男’,’1993-11-01’),*

*(2,’小明’,’女’,’1993-11-12’);*

\*增：插入(int:数字)

1. *select \* from t2;*

\*查：查询插入内容

1. delete from t2 where name=’小明’;

\*删除

1. update t2 set id=’3’where name=’小明’；

\*改

C.\*使用js 写mysql：

1. npm init

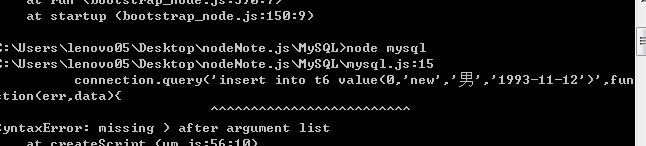
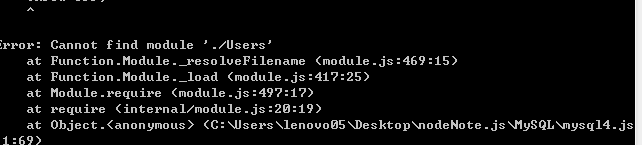
2.npm install mysql

3. var mysql=require(‘mysql’) 引入

-->mysql.js

1. node mysql.js
2. select \* from t6;

第七天

1. 封装：mysql2.js mysql3.js
2. 单引号（双引号）
3. 
4. 
5. serves.msc 查看 系统 服务
6. Mongo：启动/客服端（cmd-管理员身份运行）

mongo -help：测试安装否

启动：Net start MongoDB

关闭：Net stop MongoDB

1. .show dbs(databases) 查看
2. db.createCollection(‘t1’)

创建数据库

1. mysql:大小写不敏感

mongo:大小写敏感

第七天

1.mongo的使用:

1）启动/客服端：cmd（cmd-管理员身份运行《--开始-程序：）

输入mongo

1. mongo.exe（不建议）

命令：

报错：err

ok:0

ok:1

1. show dbs 所有 数据库 ；

db当前 数据库\*\*\*\*\*\*\*\*\*

1. use dbs1 创建数据库-半

（有意义/显示：有table）

/ 切 换

5)db.createCollection(‘table1’)

创建表格---*集合*(=table-表格)

查看表格:

Show tables/show collections

删除表格:db.table1.drop()

D.查询表格内容：db.table1.find()

【条件查询：

db.table1.find({id:”7”})

db.table1.find({id:”7”},{name:1})

db.table1.find({age:”7“,name:”1“})

：多条件查询 某个name:1】

1. 插入 内容：（免费给id），db.table1.insert({“name”:”min”,”sex”:”w”,”birth”:”1993-11-12”})

支持js代码：条件for循环：

A)for(var i=0;i<2;i++) db.table1.insert({’age’:10+i})

$gt: >

$lt : <

$gte : >=

$lte : <=

{‘age’:5}

B).修改：db.table1.update({‘age’:{$gt:5}},{$set:{‘name’:’666’}},false,true)

B.修改（更新） db.table1.update({name:’b’},{name:’bill’}) 更改name为bill

第一个参数：条件

修改器：

$set 修改\*条件为name:’b’:

db.table1.update({name:’b’},{$set:{name:’bill’}})

只修改第一条-第四个参数:

db.table1.update({name:’b’},{$set:{name:’bill’}}，false,false)

修改所有匹配条件的-第四个参数：

db.table1.update({name:’b’},{$set:{name:’bill’}}，false,true)

第三个参数：找不到，会进行添加：db.table1.update({name:’b’},{$set:{name:’bill’}}，true,true)--尽量不用

$unset 删除一条中 某个字段

db.table1.update({‘name’:’bill’},{$unset:{‘name’:’bill’}}，false,true)

$inc 对数据类型进行加减 2+3=5(数字格式）

db.table1.update({‘name’:2},{$inc:{‘name’:3}}，false,true)

C.清空 集合内容db.table1.remo1ve()

db.table1.remove({‘name’:’bill’}) :删除 某条内容

8）show dbs 查看

9)db.dropDatabase()

（use dbs1） 删除数据库

1. 下载：断言模块,mongodb ： -----mongo.js：

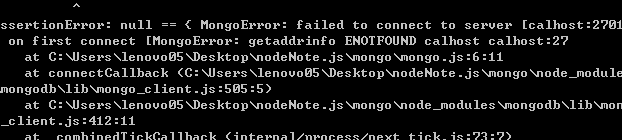
1）npm init

2）npm install assert

assert:代替if（err）{throw err} else{}

3）cnpm install mongodb:

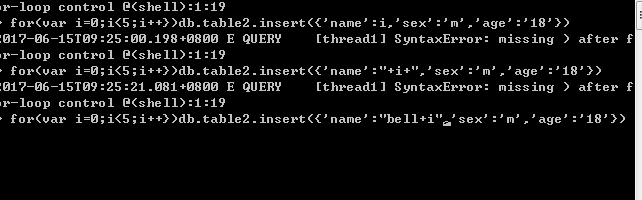
Node.js 链接mongo数据库操作方法

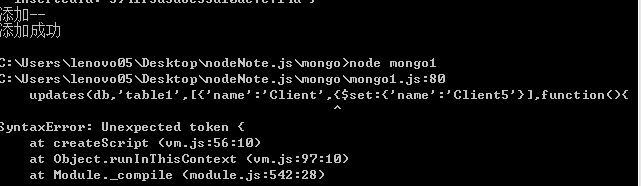


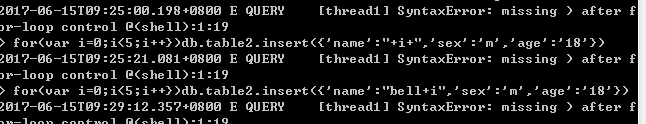
1. node.mongo
2. db.table1.find()
3. Js代码 : mongo.js

第八天

1.cataloglist(数据与服务器)

  
封装：





手动创建express--》expressRouter



后台：express，接口

第九天

1. 接口 是写出来的



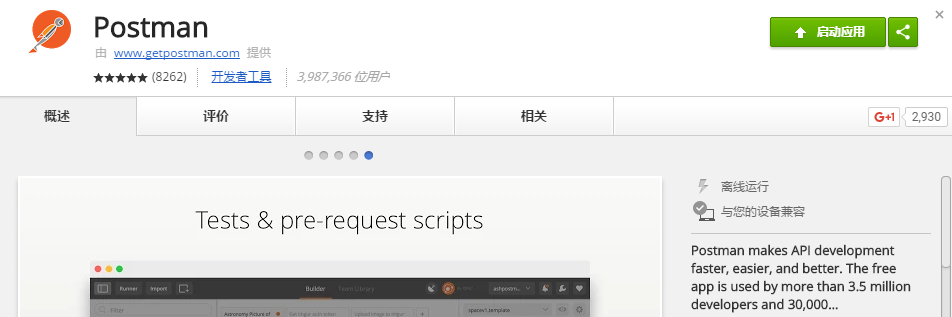
1. 验证数据：

翻墙软件--下载postman，测试post、get

1）

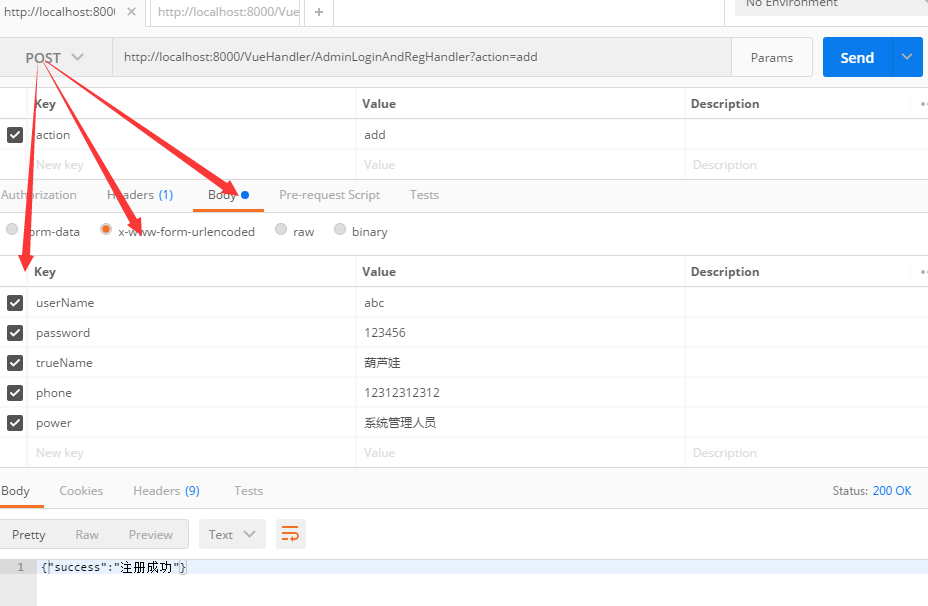


2）

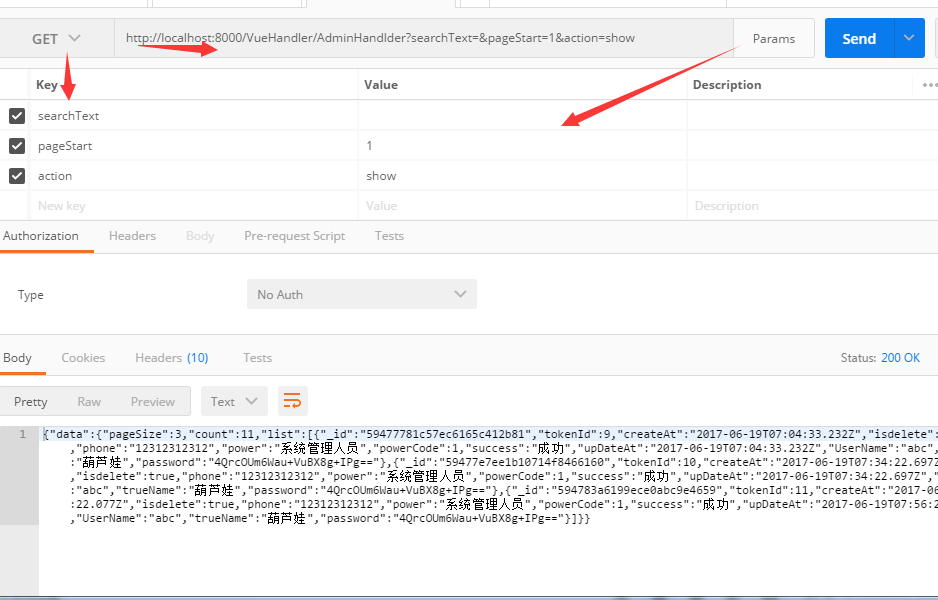


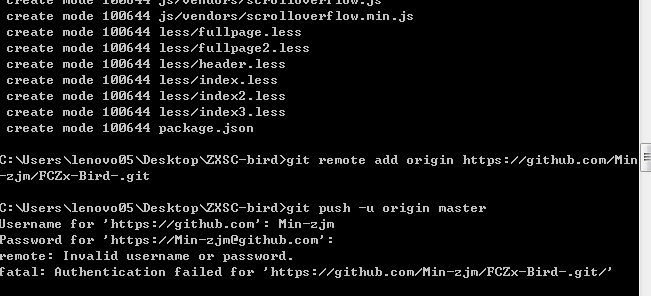
1. 输入ajax数据

post：



get:



3.git

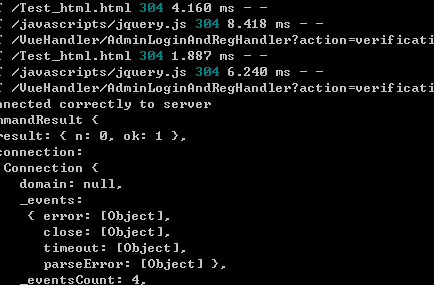
4. post: =>req.body

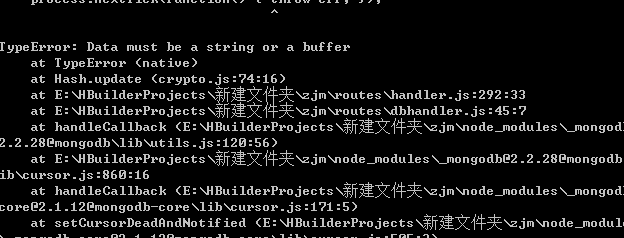
get: =>req.query

数组：res.send(‘{“ ”:” ”}) json格式

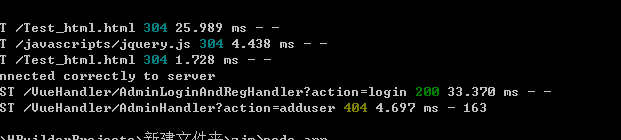
字符串：res.end(‘ ’)

5.









第十天

1. Vue：构建用户界面的****渐进式框架****（view-个人-中国人）
2. <https://cn.vuejs.org>
3. M-model：数据保存

V-view:界面

VM-业务逻辑

C：controller

1. 指令：V-
2. 和angular：都不兼容低版本IE
3. Angular:

var app=angular.module(‘app’,[]);

app.controller(“XXX”,function(){

$scope.msg=”welcome”

}

//html:

div ng-controller=’XXX’ {{m}}

Vue:

<div id=’box’>

{{msg}} //data

{{hello}}

</div>

var c=new Vue({

el:’#box’,

data{

msg:’welcome’ ,

hello:’你好’

}

})

1. 下载：

npm init

npm install vue

复制-粘贴-引入vue.js

1. 指令：作用在html标签上
2. v-model双向绑定”value:值”

v-bind绑定”属性:class,src ------”

v-for=‘(i,k,index) in options’循环（参数）

v-show/if 显示，隐藏

自定义-指令

第十一天

1. Git
2. 接口 引入的项目
3. app删除按钮
4. Car 项目

{}

好