Node是一个基于Chrome V8引擎的Javascript运行环境

* 打开cmd，输入node，进入node工作环境

运行JS文件：进入文件同级目录，执行命令node example.js

* 模块分为3类：核心模块，第三方模块，自定义模块

## fileSystem

const fs = require('fs');

引入fs模块

fs.readFile('./ca.html', (err, data) => {

if(err) throw err;

console.log(data);

})

const data = new Uint8Array(Buffer.from('Node.js中文网'));

fs.writeFile('文件.txt', data, (err) => {

if (err) throw err;

console.log('文件已被保存');

});

异步地将数据写入到一个文件，如果文件已存在则覆盖该文件

如果 options 是一个字符串，则它指定字符编码：

fs.writeFile('文件.txt', 'Node.js中文网', 'utf8', callback);

fs.writeFile(file, data[, options], callback)

* file [<string>](http://nodejs.cn/s/9Tw2bK) | [<Buffer>](http://nodejs.cn/s/6x1hD3) | [<URL>](http://nodejs.cn/s/5dwq7G) | [<integer>](http://nodejs.cn/s/SXbo1v) 文件名或文件描述符。
* data [<string>](http://nodejs.cn/s/9Tw2bK) | [<Buffer>](http://nodejs.cn/s/6x1hD3) | [<TypedArray>](http://nodejs.cn/s/oh3CkV) | [<DataView>](http://nodejs.cn/s/yCdVkD)
* options [<Object>](http://nodejs.cn/s/jzn6Ao) | [<string>](http://nodejs.cn/s/9Tw2bK)
  + encoding [<string>](http://nodejs.cn/s/9Tw2bK) | [<null>](http://nodejs.cn/s/334hvC) **默认值:** 'utf8'。
  + mode [<integer>](http://nodejs.cn/s/SXbo1v) **默认值:** 0o666。
  + flag [<string>](http://nodejs.cn/s/9Tw2bK) 参阅[支持的文件系统标志](http://nodejs.cn/s/JjbY8n)。**默认值:** 'w'。
* callback [<Function>](http://nodejs.cn/s/ceTQa6)
  + err [<Error>](http://nodejs.cn/s/qZ873x)

如果 data 是一个 buffer，则 encoding 选项会被忽略。

## http

const http = require('http');

var server = http.createServer();

server.listen('8000');

http.createServer([options][, requestListener])返回新建的 http.Server 实例。

教程上listen函数参数可以有回调函数，

server.listen(8000, function(){

console.log('nihao')

})

官方手册上没说这种用法 ，不知道为啥

server.on('request', (request, response)=>{

console.log(request.url);

response.write('world');

response.write('11111');

response.end();

});

* server绑定request事件
* request请求对象 response响应对象，给客户端发送消息
* response.end();

使用结束响应，否则一直等待

也可以有参数response.end('zaijian')

* response.write()可以使用多次
* request.url 拿到请求路径/ /nihao/123 /nihao/123/

qq浏览器会自动请求127.0.0.1:8000/favicon.ico

* request.method 拿到请求方法GET POST

response.write('你好');

write响应中文的时候，浏览器显示乱码，浣犲ソ，应该是因为默认按GBK编码的html内容。但是postman能正确显示，检查Network里边的response也可以，应该是默认utf-8编码。

response.setHeader('Content-type', 'text/plain; charset=utf-8');

然后显示就正确了

* response.setHeader('Content-type', 'text/html; charset=utf-8');

text/plain按纯文本解析，也可以按html解析，换成text/html既可

* response的Headers正常情况下有这三个：
* Connection: keep-alive
* Content-Length: 871
* Date: Sun, 01 Dec 2019 06:31:43 GMT

setHeader以后会加上一条

* Content-type: text/html; charset=utf-8

如果响应的HTML文件里包含图片，那么浏览器解析HTML解析到图片时，会另外发送一个请求。所以响应时应该判断发起的请求是请求什么，分为html和其他资源两类

if(request.url=='' || request.url=='/'){ //如果url什么都没有

response.setHeader('Content-type', 'text/html; charset=utf-8');

fs.readFile('./ca.html', (err, data)=>{

if(err){

throw err;

}

response.end(data);

})

}else{ //如果url请求其他资源

fs.readFile('.'+request.url, (err, data)=>{

if(err){

throw err;

}

response.end(data); //响应其他资源时不需要设置header

})

}

响应图片时不需要设置header

fs.readdir(path[, options], callback) 读取目录的内容。回调有两个参数 (err, files)，其中 files 是目录中的文件名的数组（不包括 '.' 和 '..'）。

fs.stat(path[, options], callback)

回调有两个参数 (err, stats)，其中 stats 是一个 fs.Stats 对象。

bigint <boolean> 返回的 fs.Stats 对象中的数值是否应为 bigint 型。默认值: false。

callback <Function>

不建议在调用 fs.open()、 fs.readFile() 或 fs.writeFile() 之前使用 fs.stat() 检查文件是否存在。 而是应该直接打开、读取或写入文件，如果文件不可用则处理引发的错误。

要检查文件是否存在但随后并不对其进行操作，则建议使用 fs.access()。

例如，给定以下的文件夹结构：

- txtDir

-- file.txt

- app.js

以下程序将会检查给定路径的统计信息：

得到的输出将会类似于：

true

Stats {

dev: 16777220,

mode: 16877,

nlink: 3,

uid: 501,

gid: 20,

rdev: 0,

blksize: 4096,

ino: 14214262,

size: 96,

blocks: 0,

atimeMs: 1561174653071.963,

mtimeMs: 1561174614583.3518,

ctimeMs: 1561174626623.5366,

birthtimeMs: 1561174126937.2893,

atime: 2019-06-22T03:37:33.072Z,

mtime: 2019-06-22T03:36:54.583Z,

ctime: 2019-06-22T03:37:06.624Z,

birthtime: 2019-06-22T03:28:46.937Z

}

false

Stats {

dev: 16777220,

mode: 33188,

nlink: 1,

uid: 501,

gid: 20,

rdev: 0,

blksize: 4096,

ino: 14214074,

size: 8,

blocks: 8,

atimeMs: 1561174616618.8555,

mtimeMs: 1561174614584,

ctimeMs: 1561174614583.8145,

birthtimeMs: 1561174007710.7478,

atime: 2019-06-22T03:36:56.619Z,

mtime: 2019-06-22T03:36:54.584Z,

ctime: 2019-06-22T03:36:54.584Z,

birthtime: 2019-06-22T03:26:47.711Z

}

## 第三方模块

package.json

package-lock.json就是锁定安装时的包的版本号，并且需要上传到git，以保证其他人在npm install时大家的依赖能保证一致。

CommonJS

exports.a=1;

此JS文件可当做一个模块，导出变量a

const nihao=require('./ew.js');

console.log(nihao.a);

其他模块导入

* require的参数为绝对路径或相对路径，相对路径必须加../或./，可以不写后缀名
* 没有指名路径，就是加载了核心模块或者第三方模块。指明路径就是自定义模块
* 每个模块，即每个js文件里有一个module对象，代表模块自身。module对象有一个exports属性，exports.a=1这个exports就是module.exports的引用。也可以通过module.exports.a=1导出数据
* 模块加载顺序：遇到第三方模块，找node\_modules文件夹，从当前文件的同级目录开始找，找不到就到上一级目录，找不到就一直往上找，直到E盘、根目录…

## 数据库

官方文档中无任何关于数据库的

npm install mysql

第三方模块

var mysql=require('mysql')