Node.js

nodejs简介

nodejs不是一门语言，不是库，不是框架，是一个JavaScript运行时环境，即nodejs可以解析和执行JavaScript代码，以前只有浏览器能解析执行js代码，因为nodejs现在js可以完全脱离浏览器运行。

nodejs中的JavaScript

没有BOM、DOM

EcmaScript

在node中为JavaScript提供了一些服务器级别的操作API

文件读写

网络服务的构建

网络通信

nodejs特性

event-driver事件驱动

non-blocking I/O model非阻塞IO模型（异步）

lightweight and efficient 轻量和高效

npm包生态系统

基于nodejs开发，是世界上最大的开源库生态系统

绝大多数JavaScript相关包都存放在了npm上，这样做的目的是为了让开发人员更方便的去下载使用

如安装jquery：npm install jquery

node执行js文件

1. 创建编写js脚本文件
2. 打开终端，定位到脚本文件所属目录
3. 输入node 文件名 执行对应文件

注意：文件名不能用node.js，最好也不用中文

node操作文件

引入文件系统模块fs，在fs核心模块中提供了所有文件操作相关的API

读文件

//使用require引入fs模块

var fs = require(‘fs’);

//读取文件：第一个参数是文件路径，第二个参数是回调函数，回调函数中有两个参数：data和error

//如果读取成功data是数据对象，error是null；如果读取失败data是undefined，error是错误对象

fs.readFile(‘url’, function(data, error) {

node读取文件内容显示在终端都是二进制数据，要想转换成字符可用data.toString()方法

})

写文件

//写文件三个参数：1.文件路径2.文件内容3.回调函数

//回调函数只接受一个参数error

fs.writeFile(‘url’, 文件内容, function(error) {})

node搭建web服务

搭建基础服务模型

//1.需要引入核心模块http

var http = require(‘http’)

//2.用http.createServer()方创建web服务器，该方法返回一个server实例

var server = http.createServer()

//3.注册request请求事件，当客户端请求发过来就会自动触发服务器的request请求事件，然后执行第二个参数回调函数

server.on(‘request’, function() {

console.log(‘收到请求’);

})

//4.启动服务：绑定端口号

server.listen(8000, function() {

console.log(‘服务启动成功，localhost:8000进行访问’)

})

//5.在node终端执行该主体文件即可启动服务

node web服务文件

服务端发送响应

request请求事件处理函数需要接收两个参数：

request请求对象

response响应对象

server.on(‘requset’, function(request, response) {})

request.url：返回访问者所访问的路径，如访问http://127.0.0.1:8000/a，则返回/a

response.write(data)：该方法用于服务端返回数据，可以多次调用，但是最后要用response.end()方法结束响应，否则服务端不发送

response.end(data)：结束write方法发送数据的方法，参数可以填要发送的数据，直接发送不用提前写write方法

node后端返回json数据

JSON.stringify(非字符串数据) //将非字符串转换为字符串数据