

Node

- HTTP模块
- URL模块
- supervisor 修改代码后自动重启
- Node模块
- 完全符合CommonJS规范的包目录一般包含如下这些文件
 - package.json: 包描述文件
 - bin: 用于存放可执行二进制文件的目录
 - lib: 用于存放JavaScript代码的目录。
 - doc: 用于存放文档目录
 - 可以在npmjs.com找到需要的包
- NPM介绍
 - <https://www.npmjs.com>
 - 允许用户从NPM服务器下载别人编写的第三方包到本地使用。silly-datetime
 - 允许用户从NPM服务器下载并安装别人编写的命令行程序（工具）到本地使用。
 - 允许用户将自己编写的包或者命令行程序上传到NPM服务器供别人使用。
- **清空命令行 cls**
- NPM命令
 - npm -v 查看npm版本
 - npm install ModuleName 安装模块
 - npm uninstall moduleName 卸载模块
 - npm list 查看当前目录下已安装的node包
 - **npm info moduleName 查看模块版本**
 - npm install moduleName@1.8.0 安装指定版本模块
- **package.json**
 - 创建:
 - npm init 或 npm init --yes 强制生成
 - 安装模块
 - 创建好package.json以后，安装模块需要把模块写入package.json这个配置文件
 - npm install moduleName --save

- 写入到package.json的dependencies中
 - npm install moduleName --save-dev
 - 写入到package.json里面的devDependencies中
- npm install 通过package.json加载模块
- 内容：
 - dependency配置当前程序所依赖的其他包
 - devDependency配置当前程序所依赖的其他包，只会下载模块，而不会下载这些模块的测试和文档框架
 - “dependencies”:{
 - "ejs":"^2.3.4",
 - "express":"^4.13.3"
 - }
 - ^表示第一位版本号不变，后面两位取最新的
 - ~表示前两位不变，最后一个取最新
 - *表示全部取最新
- 安装淘宝镜像
 - <http://npm.taobao.org/>
 - npm install -g cnpm --registry=<https://registry.npm.taobao.org>
- FS模块 文件系统模块
 - fs.stat 检测是文件还是目录 **这个是异步，注意注意有坑的**
 - fs.mkdir 创建目录
 - fs.writeFile 创建写入文件
 - fs.appendFile 追加文件
 - fs.readFile读取文件 //获取数据为buffer
 - fs.readdir读取目录
 - fs.rename重命名
 - fs.rmdir删除目录
 - fs.unlink 删除文件
 - fs.createReadStream 文件流的方式读数据 **分块读取 广播**
 - const fs=require('fs');
 - var fileReadStream=fs.createReadStream('data.json')
 - let count=0;
 - var str=""
 - fileReadStream.on('data',(chunk)=>{ //接收到数据触发
 - console.log(` \${++count}接收到: \${chunk.length}`);

- str+=chunk;
- })
- fileReadStream.on('end',()=>{
 - console.log('----结束----');
 - console.log(count);
 - console.log(str);
- })
- fileReadStream.on('error',(error)=>{
 - console.log(error)
- })

- fs.createWriteStream 写入流

- var fs=require('fs');
- var data='我是从数据库获取的数据,我要保存起来嘞';
- var writeStream=fs.createWriteStream('data.json'); **//文件不存在会创新文件再写入**
- writeStream.write(data,'UTF8'); **//可以多次写入 必须在end之前**
- writeStream.end();
- writeStream.on('finish',function(){
 - console.log("写入完成");
- })
- writeStream.on('error',function(err){
 - console.log(err.stack);
- })
- console.log("程序执行完毕")

- 管道流

- 通过读取一个文件的内容并将内容写入到另外一个文件中。
- **var fs=require("fs");**
- **//创建一个可读流**
- **var readerStream=fs.createReadStream('input.txt');**
- **//创建一个可写流**
- **var writerStream=fs.createWriteStream('output.txt');**
- **//管道读写操作**
- **//读取input.txt文件内容，并将内容写入到output.txt文件中**
- readerStream.pipe(writerStream);
- console.log("程序执行完毕");

幕布 - 思维概要整理工具
