MONGODB：

* mongodb 是目前最好的面向文档的免费开源NoSqL数据库，也就是非关系型数据库。非关系型数据库还包括redis.
* Mongodb查询：find、aggregate.findone,findOneAndUpdate等
* Mongodb优势：面向文件的，高性能的，高可用性的，易扩展的，丰富的查询语句

REDIS：

* redis有哪些好处：速度快，因为数据存在内存中。类似于HashMap,HaspMap的优势就是查找和扩展的时间复杂度是0；支持丰富的数据类型：string、set、sort set、hash;支持事务，操作都是原子性，所谓原子性就是对数据的更改要么全部操作要么全部不操作；

丰富的特性：可用于缓存，消息，按key设置过期时间，过去后将会自动删除

* redis内存数据库集大小上升到一定时候，就会施行数据淘汰策略。

Mysql:

* Mysql数据库是开源的、关系型数据库。
* 常用命令：sc query mysql、sc delete mysql、net start mysql、show detabases(显示当前服务器上存在什么数据库)、create database mysqldata(创建一个数据库mysqldata)、use

Mysqldata(选择所创建的数据库)、show tables（查看表结构）、create table mytable(‘name’ varchar(20)、sex char(1))（创建一个数据库表）、insert into mytable values(‘hyq’,’M’)(插入记录)、drop table mytable(删除表)、update ...set ...where...(更新)、

Delete(删除记录)、

* 特别权限：all允许做任何事（和root一样）、usage:只允许登录-其他什么也不允许做

Sequelize:findAll、findOne、update、destory

NodeJS:

Nodejs采用的是单线程的处理机制(所有的I/O请求都采用非阻塞的工作方式及该代码在执行的时候，在他之后的代码无需等待可继续执行)。Nodejs是一个javaScript虚拟的运行环境。

NPM的好处：通过NPM可以安装和管理项目的依赖，并且能够指明依赖想的具体版本号。

NodeJS常用的模块：moment.js、angluarJS、Crypto、async、fs等

Async:

Series:顺序执行

Waterfall:顺序执行，并且传递值；中途出粗后面的函数不执行，直接将结果传到最后返回

Parallel(tesks,[callback]):多个函数并行执行，不会等待其他函数；如果中途出错，则立即将err和值传到最终的回调函数，其他未执行完毕的函数将不会再执行，但返回结果中会占一位

parallelLimit(tasks,limit,[callback])：限制并发执行的个数

Whilst:一直执行主函数，直到条件不满足，或者发生异常

Each:顺序执行

EachSeries:顺序执行，但如果其中一个异常，则其他不执行

**Promise:**

1.promise有三种状态：未完成、完成（fulfilled）、失败（rejected）

2.peomise的状态可以由未完成转换成完成或者未完成转换成失败

3.promise的状态转换只发生一次

闭包：本质上讲闭包就是将函数内部和函数外部链接起来的一座桥梁

js函数内部可以直接读取全局变量，在函数外部自然无法读取函数内的局部变量；

闭包的作用：读取函数内部变量；将这些变量的值始终保持在内存中

闭包的注意事项：防止内存泄漏；闭包会在父函数外部，改变父函数内部变量的值

**Nodejs:apply、call、bind的作用：**

Apply和call是为了改变某个函数运行时的上下文二存在的（就是为了改变函数内部this的指向）；

区别：apply的第二个参数是一个参数数组，call的第二个以及后面的参数都是数组里面的元素，就是说要全部列举出来。

Bind()--也是改变函数体内的this指向；

Bind会创建一个新的函数，称为绑定函数，当调用这个函数的时候，绑定函数会以创建他是传入的bind()方法的第一个参数作为this,传入bind()方法的第二个以及以后得参数加上绑定函数运行时本身的参数按照顺序作为原函数的参数来调用原函数；

Bind与apply、call最大的群别说bind不会立即调用及bind方法返回的仍然是一个函数，因此还需要（）或事件才出发执行或调用，其他两个会立即调用

基本数据类型：number、boolean、string、null以及undefined

复杂数据类型：array、function、object等

Nodejs中module.exports用来指定模块的；exports就是对module.exports的引用

ProtoType是用来给给累的实力定义函数、或者实现继承、也可以重写和调用弗雷函数；

A.prototype=new MyClass();

使用v8实现继承如下：

MyClass.prototype.\_proto\_=EventEmitter.prototype;//myClass继承EventEmitter

EventEmitter----nodejs事件机制

是nodejs基础的api之一，可以随处进行事件的监听和分发。如http请求的req.on(‘data’,function(){});req.on(‘end’,function(){})

传统文本终端下的studio|stdout|stderr对象都是流；

Nodejs重启的工具：forever(线上应用)\supervisor(多用于调试)

Git解决代码冲突问题：git stash暂时存储；git pull 拉取；git stash pop stash 还原暂存的内容；rebase和merge的区别：merge会合并另一个分支的内容；rebase也合并另一个分支的内容，但是会把本分支的commits顶到最顶端

**NODEJS如何捕获异步处理器抛出的异常：**

=>:在es6中（）=>....前代表传入参数，后代表结果

<!DOCTYPE html> 声明符合最新HTML5规范的文档

AngularJS: angularJS可以克隆和重复HTML元素；AnguarJS可以隐藏和显示HTML元素；AnguarJS可以在HTML元素"背后"添加代码;AngularJS支持输入验证

AnguarJS ng-controller:指令用于为你的应用添加控制器；在控制器中，可以使用scope对象来访问。

ng-repeat:指令用于循环输出指定次数的HTML元素，集合必须是数组或对象

ng-change:指令用于告诉AngularJS在HTML元素值改变时需要执行的操作。

ng-chang指令需要搭配ng-model指令使用

ng-change指令不会覆盖原生的onchange事件，如果除非该事件，ng-change表达式与原生的onchange事件都会执行

ng-change事件在值每次改变时触发，不需要等待执行过程或者失去焦点

ng-change事件只针对输入框值得该表，而不是通过javaScript来修改。

ng-model:指令绑定了HTML表单元素到scope变量中；如果scope中不存在变量，将会创建它。

监控事件：监控页面渲染

app.directive('onFinishRenderFilters', function ($timeout) {  
 return {  
 restrict: 'A',  
 link: function(scope, element, attr) {  
 if (scope.$last === true) {  
 $timeout(function() {  
 scope.$emit('ngRepeatFinished');  
 });  
 }  
 }  
 };  
});

//页面渲染完毕后进行页面图标样式的调整

$scope.$on('ngRepeatFinished', function (ngRepeatFinishedEvent) {  
 if($('.slider-element').data('slider') != null){  
 adjustSize($('.slider-element').data('slider').getValue());  
 }  
 //adjustPage();  
 $("div[ng-if=false]").css({"display": "none"});  
});

Datatables:是一款jquery表格插件。分页，及时搜索和排序

bootstrap框架：Bootstrap是最受欢迎的HTML、CSS和JS框架，用于开发相应式布局、移动设备有限的WEB项目。

async：異步：默認true;默認異步

scoket:

var server = http.createServer(function (request, response) {  
 response.writeHead(200, {"Content-Type": "text/html"});  
 response.write("WebSocket Start~~~~~~~~~~~~");  
 response.end("");  
}).listen(port);  
  
var socket = io.listen(server);

前臺：**var *socket*** = ***io***.connect(*$*(**"#socketURL"**).val());  
***socket***.on(**'detail'**, **function** (data) {

HTTP:

**var** options = {  
 **hostname**: ***meidiIp***,  
 **port**: ***meidiPort***,  
 **path**: ***meidiPath*** + url,  
 **method**: **'POST'**,  
 **headers**: {  
 **login**: login,  
 **password**: password,  
 **'Content-Type'**: **"application/json;UTF-8"**,  
 **"Content-Length"**: content.length  
 }  
};  
**console**.log(**"options+="** + **JSON**.stringify(options));  
  
var req = http.request(options, function (res) {  
 var dataStr = "";  
 res.setEncoding('utf8');  
 res.on('data', function (json) {  
 dataStr += json;  
 //console.log("return data:" + JSON.stringify(dataStr));  
 });  
 res.on('end', function () {  
 callback(dataStr);  
 });  
});  
req.write(content);  
req.on('error', function (e) {  
 console.log('problem with request: ' + e.message);  
});  
req.end();

**作为一个第三方API如何保证安全性即阻止恶意访问：**

* 在服务端设置白名单，同时判断对方带过来的token是否过去(这里的token自己随意组合，如：域名+过期时间，要加密)
* 服务端提供一个appName和AppKey,为了防止中间拦截修改，可以添加时间戳。比如appSecret可以是：业务参数+appName+aapKey,在进行MD5加密等

常用的加密方式：MD5(不可逆)、

Crypto来实现加密：

//哈希加密算法，不可逆 md5,sha1

var pass=crypto.createHash(“md5”); var psd=pass.update(password).digest(“base64”);

//密钥加密算法，不可逆 利用哈希算法，以一个密钥和一个消息为输入，生成一个 消息摘要为输出

Var secrectKey=” secrectKey”;

Var hmac=crypto.createHmac(“md5”, secrectKey)；

hmac.update(“password”).digest(“base64”);

**跨域问题：**

**r**es.header(“Access-Control-Allow-Origin");//设置权限

Linux:

1.查看文件路径：ls、ls -l,显示其他文件路径下的文件：ls/....;显示隐藏文件：ls -a

2.建立目录：mkdir;删除目录：rmdir aa;cd.. 打开上一层；cd 打开目录，.是当前目录；

3.复制文件到文件one中：cp install.log ./one;cp -r复制文件目录

4.创建文件：touch 11.txt

5.删除文件：rm aa.txt;rm -f aa.txt表示删除时不需要提醒用户

6.通配符：类似模糊匹配

7.显示当前所在目录：pwd

8.新增一个用户：在root权限下新增：useradd aaa;passwd \*\*\*

cd ~ 返回用户宿主目录

　　cd . 用户的当前目录

　　cd .. 返回上一级目录

vi文本编辑器

ps -A、ps -e显示所有进程

**HTTPS：**

客户端 ssl协议 服务器

1. 向服务器发送https请求，要求与服务器建立SSL连接
2. Web服务器收到请求后悔将网站的证书信息（证书中包含公钥）传递一份给客户端
3. 客户端的浏览器与web服务器开始协商ssl连接的安全等级，
4. 客户端随机（对称）秘钥，并使用公钥对秘钥进行加密，传递给web服务端
5. 服务端通过自己的私钥解密出回话秘钥
6. Web服务器利用回话秘钥加密与客户端之间的通信

Domain可以捕获异步异常

数组的遍历：foreach,过滤数组：fileter;要改变数组中的每个元素的值可以用map

Object.keys方法会返回一个由一个给定**对象**的自身可枚举属性组成的**数组**，数组中属性名的排列顺序和使用

Let a=[‘a’,’b’,’c’] Object.keys(a) //[‘0’.’1’,’2’]

Let obj={foo:’bar’,baz:42} Object.keys(obj) //[‘foo’,’baz’]

Let obj={0:’a’,1:’b’,2:’c’} Object.keys(obj) //[‘0’.’1’,’2’]

Object.assign 深度复制，不会改变值