后端交接文档——

* 后端综述

使用Node + Express 开发服务端，前后端的交互使用http方式进行交互，所有接口均遵循restful api规范。数据库使用MongoDB。

* 名词解释

- Node：一个基于 Chrome V8 引擎的 JavaScript 运行环境。使用了一个事件驱动、非阻塞式 I/O 的模型（非阻塞模式的IO处理给Node带来在相对低系统资源耗用下的高性能与出众的负载能力，非常适合用作依赖其它IO资源的中间层服务。），使其轻量又高效（有说法说Node可以认为是数据密集型分布式部署环境下的实时应用系统的完美解决方案）。其包管理器名为npm，基本是全球目前最大的开源库生态系统。Node的缺点在于可靠性低，

单进程，单线程，只支持单核CPU，不能充分的利用多核CPU服务器。一旦这个进程崩掉，那么整个web服务就崩掉了。但这些缺点可以利用程序健壮性加以解决，比如我们使用pm2来帮助我们启动多个进程、对Node进程进行监听和崩溃后的自动重启等；

- Chrome V8 引擎：Google 开源的、高性能的 JavaScript 引擎。由 C++ 编写，并用在 Google 开源浏览器 Chrome 中。实现了 ECMA-262 第 5 版描述的 ECMAScript。V8 编译和执行 JavaScript 源代码，处理对象内存分配，对不再使用的对象进行回收。V8 的垃圾回收是其性能的关键；

- Express：是一种保持最低程度规模的灵活 Node.js Web 应用程序框架，为 Web 和移动应用程序提供一组强大的功能。可以为各种 HTTP 实用程序方法和中间件，快速方便地创建强大的 API；

- Restful api：一种软件架构风格、设计风格，而不是标准，只是提供了一组设计原则和约束条件。主要用于客户端和服务器交互类的软件。基于这个风格设计的软件可以更简洁，更有层次，更易于实现缓存等机制。使用可以参考：<http://www.ruanyifeng.com/blog/2014/05/restful_api>

- MongoDB：一个介于关系数据库和非关系数据库之间的产品，本身数据非关系型数据库，但是它却是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似json的bson格式，因此可以存储比较复杂的数据类型。

* 后端目录结构：

- app // 所有后端文件、资源的总目录

- config // 配置文件

- controllers // 业务逻辑处理

- middlewares // 中间件

- models // 数据库model（或者说比较好点的理解方式是各种表的Schema）

- resources // 图片之类的资源

- routes // api

- schedules // 定时任务

- test // 测试文件（娃娃机项目由于开发太急基本没写）

- utils // 自定义的各种常用方法

- validators // 验证器中间件

- app.js // 启动文件

- server.js // 服务启动时，各种中间件、路由api的注册文件

* 启动文件app.js

启动时候，首先调用server.js，初始化express服务，之后链接数据库，成功链接的情况下，依次读取并注册所有models，之后读取所有定时任务schedules，之后可定义一些需要做的其他事情（比如娃娃机项目之后是读取数据库，将房间信息保存到内存中方便之后调用），最后完成端口监听，完成启动。

* server.js

该文件是注册中间件和路由的文件，比较关键的几个中间件是：

1. 一个专门给支付宝支付使用的刷新请求头部的中间件；
2. bodyParser.json()：转换请求体，之后可以使用req.body拿到用户请求传输来的json对象；
3. requestInfo.getUserInfo：处理token用的，可以判断操作是哪个管理员，之后可以用req.user获取到这个管理员的信息；
4. requestInfo.getVisitorInfo：也是处理token用的，只是判断的是哪个用户，之后可以用req.visitor获取到这个用户的信息；

* 路由routes

通过index.js读取该目录下的所有文件，加载所有路由，一个模块一个文件。基本定位问题都是可以通过这个目录下找到对应的接口，再定位到出问题的方法。

* models

数据库的表结构，后端数据库采用mongodb，所以在node中，使用mongoose链接数据库，并定义表结构，进行数据操作。

主要数据库的增删改查操作可以参考：<http://blog.csdn.net/u012810020/article/details/54582051>

其中，比较特殊的字段或表有以下几个：

user表中的password：这是一个虚拟字段，数据库中本身并不存在；

user表中的hashedPassword：用户密码加密后的字符串；

user表中的salt：用户密码加密过程中，添加的随机字符串；

fs.chunks和fs.files表：这两个表是由gridfs-stream依赖创建的，该依赖用户处理用户上传或需要下载的资源；

jobs表：该表是由agenda依赖创建的，该依赖用户处理定时触发的一些方法；

* 验证器validators

该文件夹下的文件主要用于验证用户传入的参数是否正确，也数据中间件一类；

* utils：

该文件夹下的文件主要是一些比较通用的方法，或者在多个地方都会使用到的方法，比较典型的是以下几个：

1. pay：支付宝、微信、IOS支付的文件夹；
2. push：推送功能（待完成）；
3. appConfig.js：读取配置文件的类，所有配置文件中的值，可以通过该类返回的对象获取；
4. appConst.js：各种常量统一放置的位置；
5. assertion.js：用于处理验证器中的一些自定义的方法；
6. authentication.js：控制权限时统一使用的方法；
7. easyRouter.js：较为底层的，控制http请求和返回数据的方法，每个route都使用到了；
8. encryption.js：token加密和解密的方法；
9. excelExport.js：excel导出的方法；
10. logger.js：自定义的日志输出方法；
11. mongooseHelper.js：帮助处理数据库提交或者返回的字段的方法；
12. serverUtils.js：其他通用方法，比如随机生成指定格式的字符串、日期转换等；

* 配置文件configs

- config.js：主要的配置文件，不管是开发环境还是上线后的生产环境，该配置文件下的内容都有效；

- development.js：仅开发环境有效的文件，启动app.js前需要定于环境变量NODE\_ENV=development，NODE\_DEBUG=true；

- production.js：仅生产环境有效的文件，启动app.js前需要定于环境变量NODE\_ENV=production，NODE\_DEBUG=false；

- excelSystem.xml：导出excel时候的需要的文件；

* 开发新模块大致流程：

--> 开发时间充裕，应该预先编写test文件

--> 设计数据格式，定义model

--> 定义接口数量，初步定义route，主要可以完成各个接口的可调用权限

--> 对于post，put这样的创建/修改接口，定义数据验证器validators

--> 定义controllers，完成各种数据的处理

--> 完善route，将validators和controllers串联起来

--> 调用test

* 账号、密码：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 账户 | 密码 | IP | port |  |
| 阿里云 | tiansheDolls | NextGen2204 |  |  |  |
| 娃娃机实例，远程连接 |  | 139379 |  |  |  |
| SSL登录 | root | NextGen615 | 39.104.66.47 | 22598 |  |
| 数据库-总管理员 | tianshe | NextGen615 |  |  |  |
| 抓东东数据库管理员 | dolls | tsDollsDbPsd |  |  |  |
| 后台页面登录账号 | admin | tiansheDoll615 |  |  |  |
| tianshemedia阿里云SSL | root | NextGen615 | 39.104.53.167 | 22598 | 实例名：tianshemedia |
| tianshemedia阿里云远程连接 |  | 290908 |  |  |  |
| H5娃娃机新阿里云SSL | root | NextGen615 | 39.104.61.134 | 22598 | 实例名：h5dolls |
| H5娃娃机新阿里云远程连接 |  | 186880 |  |  |  |

抓东东公司内测试机：192.168.1.195；SSL登录账号密码都是tianshe

* 关于部署

1. 代码使用git上传到源码机，主要命令：git commit,git push
2. 初次创建服务器，需要在服务器相应目录下，使用git remote命令，添加远程仓库地址（例：git remote add origin [git@210.22.157.42:KoanSun/dolls\_grab.git）](mailto:git@210.22.157.42:KoanSun/dolls_grab.git）)
3. 更新代码，主要命令：git pull
4. 如果前端有更新，则允许gulp release重新压缩前端代码
5. 使用pm2 restart all重启所有服务（初次启动，使用pm2 start ecosystem.json。需要停止服务，使用pm2 delete all）

* 关于部署服务器

1. 确保系统64位（因为mongodb数据库在32位系统上存储上限2G）
2. 安装mongoDb数据库（$ apt-get install mongodb）
3. 配置mongodb

$ cd /usr/bin

$ ./mongo

$ use tencent

$ db.addUser('tianshe', 'NextGen615')

允许远程配置

编辑文件：/etc/mongod.conf

修改后的内容如下：

bind\_ip = 0.0.0.0

port = 27017

auth=true

重启：$ /etc/init.d/mongodb restart

1. 安装openssh-server（用于ssh链接）

$ apt-get install openssh-server

使用

$ service ssh start

$ /etc/init.d/ssh start

打开服务

（如果需要修改port，则通过、etc/ssh下的两个配置文件修改）

1. 安装node
   1. 安装依赖

$ sudo apt-get install g++

$ sudo apt-get install libssl-dev

$ sudo apt-get install python

$ sudo apt-get install make

* 1. 安装node

$ wget http://nodejs.org/dist/v6.9.4/node-v6.9.4.tar.gz

$ tar zxvf node-v6.9.4.tar.gz

$ cd node-v6.9.4/

$ ./configure

$ sudo make install

* 1. 测试

$ node -v

1. 安装pm2（npm install pm2 -g）
2. 安装gulp（npm install gulp -g）
3. 启动项目（gulp release，pm2 start ecosystem.json）

数据库查询：

1. 登录阿里云->控制台->左侧工具栏选“云服务器ECS”->点“远程连接”要查询的服务器
2. 登录ssl用户名密码
3. 输入mongo进入数据库
4. 然后use dolls，进入抓东东的数据库
5. 使用db.auth('账号','密码')，完成权限校验
6. 输入查询语句