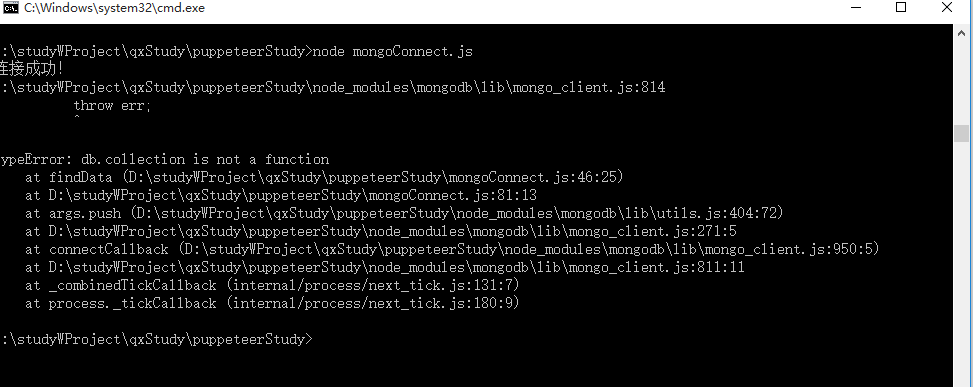
MongoDB Node.js驱动程序是被官方所支持的原生Node.js驱动程序，他是至今为止最好的实现， 并且得到了MongoDB官方的支持。MongoDB团队已经采用MongoDB Node.js驱动程序作为标准方法。

npm install mongodb@1.4.3 // MongoDB Node.js驱动程序

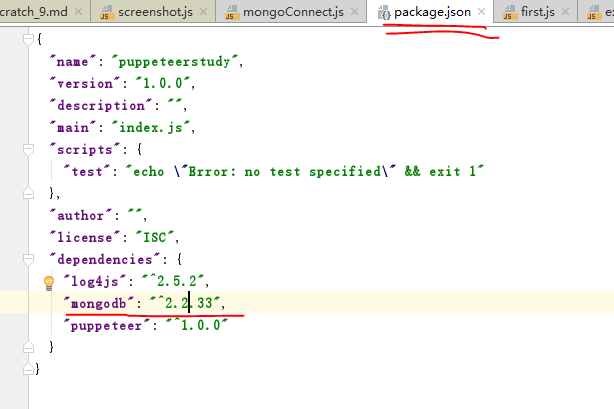
npm install mongoose@3.8.8 //mongoose模块

**var** MongoClient = *require*(**'mongodb'**).MongoClient;  
  
MongoClient.connect(  
 **'mongodb://10.200.200.157:27017/wd3metadata\_for\_157'**,  
 {  
 */\* db: { w: 1, native\_parser: false },  
 server: {  
 poolSize: 5,  
 socketOpations: { connectTimeoutMS: 500 },  
 auto\_reconnect: true  
 },  
 replSet: {},  
 mongos: {}\*/* }, **function**(err, db) {  
 **if** ( err ) {  
 ***console***.log(**'连接失败！'**);  
 } **else** {  
 ***console***.log(**'连接成功！'**);  
  
 *findData*(db, **function**(result) {  
 *//显示结果* ***console***.log(result);  
  
 *//关闭数据库* db.close();  
 });  
 }  
 }  
);  
  
*//定义函数表达式，用于操作数据库并返回结果***var** *findData* = **function**(db, callback) {  
 *//获得指定的集合* **var** collection = db.**collection**(**'UrlParameters'**);  
 *//要查询数据的条件* **var** where={**Md5**: **"6a07f73dbfb6dd8fef92c97ccaa3de36"**};  
 *//要显示的字段* **var** set={**Md5**:1,**Parameter**:1};  
 collection.find(where,set).toArray(**function**(err, result) {  
 *//如果存在错误* **if**(err)  
 {  
 ***console***.log(**'Error:'**+ err);  
 **return**;  
 }  
 *//调用传入的回调方法，将操作结果返回* callback(result);  
 });  
}

**问题记录**



原因： 版本问题。运行“npm install mongodb”命令而不指定版本，默认是安装最新版本3.0.2。3.0的api已经更新和以前的版本不一样



再运行“npm install”

**Mongoose**

实际开发中估计更多会选用类似mongoose的模块来操作来提升开发效率。

schema是mongoose里会用到的一种数据模式，可以理解为表结构的定义；每个schema会映射到mongodb中的一个collection，它不具备操作数据库的能力。

Schema 之所以能够定义documents, 是因为他可以限制你输入的字段及其类型.

Mongooose中，有三个比较重要的概念，分别是Schema、Model、Entity。它们的关系是：Schema生成Model，Model创造Document，Model和Document都可对数据库操作造成影响，但Model比Document更具操作性

　　Schema用于定义数据库的结构。类似创建表时的数据定义(不仅仅可以定义文档的结构和属性，还可以定义文档的实例方法、静态模型方法、复合索引等)，每个Schema会映射到mongodb中的一个collection，Schema不具备操作数据库的能力

　　Model是由Schema编译而成的构造器，具有抽象属性和行为，可以对数据库进行增删查改。Model的每一个实例（instance）就是一个文档document

　　Document是由Model创建的实体，它的操作也会影响数据库

mongoose支持的基本类型有:

String,Number,Date,Buffer,Boolean,Mixed,ObjectId,Array

其中, Mixed和ObjectId是mongoose中特有的,ObjectId实际上就是\*\*\_id\*\*的一个映射. 同样,mongoose里面有着和所有大众数据库一样的东西.

记得每次给添加字段时，要往这个table.json里面添加。由于nodejs这个服务器的改动，更改table.json往往需要重启游戏服务的。

{

"\_id":"t0001",

"isOpen":true ,

"timeInterval":86400000, //时间间隔：1天

"lastRunTime":"2014-07-10 10:21:12", //最新一次截图时间：此时此刻

"parameters":'{

"url":"https://item.taobao.com/item.htm?spm=a21bo.2017.201876.43.3f9a0030Oaxhh8&scm=1007.12493.92624.100200300000005&id=523387189238&pvid=6da310d8-fc0b-4870-905b-d3764fa6e4d2",

"device":"computer",

"viewport":{"width":1920,"height":600}

}'

}

{

"\_id":"t0014",

"isOpen":true ,

"timeInterval":86400000,

"lastRunTime":"2018-02-01 13:21:12",

"parameters":'{

"url":"https://item.taobao.com/item.htm?spm=a21bo.2017.201876.43.3f9a0030Oaxhh8&scm=1007.12493.92624.100200300000005&id=523387189238&pvid=6da310d8-fc0b-4870-905b-d3764fa6e4d2",

"device":"iPhone 6",

"viewport":{"width":375,"height":600}

}',

"screenshotImages":[

"180212180534\_item.taobao.com\_ver01.jpg",

"180212180634\_item.taobao.com\_ver02.jpg"

]

}

**事件'SIGINT'**

捕获当前进程接收到的信号（如按下了 ctrl + c）：

process.on(**'SIGINT'**, **function** () {  
 **mongoose**.connection.close(**function** () {  
 logger.info(**'Mongoose disconnected through app termination'**);  
 process.exit(0);  
 });  
});

**https://try-puppeteer.appspot.com/**

isAll?ExpressQueryTotal:null