[tripPlan ---imgUploader 1](#_Toc23082)

[tripPlan ----direcotive 1](#_Toc29389)

~~[解决bug1：路标问题。](#_Toc1854)~~ [1](#_Toc1854)

~~[已由他人解决bug2：wechat邀请员工](#_Toc1809)~~ [2](#_Toc1809)

[bug3 日期无法显示儿童节等节日。暂未解决 2](#_Toc1940)

~~[任務1：新加入用户后，给管理员发送通知。（已完）](#_Toc8138)~~ [5](#_Toc8138)

~~[微信添加员工](#_Toc31958)~~ [5](#_Toc31958)

~~[批量添加员工](#_Toc23506)~~ [5](#_Toc23506)

~~[手动添加员工](#_Toc26351)~~ [6](#_Toc26351)

~~[任務 申請單多行程支持](#_Toc28185)~~ [10](#_Toc28185)

~~[行程单前端展示多行程](#_Toc15630)~~ [10](#_Toc15630)

[任务 票据审核通知 （已完） 10](#_Toc18690)

[target实际指向是一个cached对象，包含5个函数，create，update，destory， 11](#_Toc2073)

~~[给鲸力发通知](#_Toc31151)~~ [11](#_Toc31151)

[模拟数据源 11](#_Toc20817)

[爬虫爬取数据 12](#_Toc23152)

[爬虫流程： 12](#_Toc3955)

[四大航空公司 12](#_Toc15865)

fsevents@^1.0.0

# tripPlan ---imgUploader

D:\node\nodeproject\jing\qmtrip\public\mobile\tripPlan\controller.js

ajax request

D:\node\nodeproject\jing\qmtrip\api\attachment\upload.ts

# tripPlan ----direcotive

D:\node\nodeproject\jing\qmtrip\www\libs\directives\trip-plan\index.ts

### ~~解决bug1：路标问题。~~

点击路标输入框，触发D:\node\nodeproject\jing\qmtrip\nglibs\directives\selector\index.ts的函数showSelectorDlg函数，生成对话框

暂未知如何出发以下的api。

on business-> landmark map--> html & controller

D:\node\nodeproject\jing\qmtrip\nglibs\services\modal-dlg\map\dialog.html

### ~~已由他人解决bug2：wechat邀请员工~~

通过接口和decorator注册了一系列的方法，如createdAt,deletedAt(D:\node\nodeproject\jing\qmtrip\common\model\common.ts)

D:\node\nodeproject\jing\qmtrip\www\ionic\company\staff-invited.content.html

invited-link中的createLink，试用createLink

D:\node\nodeproject\jing\qmtrip\common\model\common.ts

处理该模块的ts

D:\node\nodeproject\jing\qmtrip\www\ionic\company\staff-invited.ts

生成goInvitedLink,此时跳转到以下网页，使用decorator进行了

D:\node\nodeproject\jing\qmtrip\api\\_types\staff\invited-link.ts

# bug3 日期无法显示儿童节等节日。暂未解决

D:\node\nodeproject\jing\qmtrip\nglibs\services\modal-dlg\date\date-month.html

登陆模块

D:\node\nodeproject\jing\qmtrip\www\ionic\login\index.ts

问题

window.cordova

bs

定义了angular的入口，

D:\node\nodeproject\jing\qmtrip\nglibs\models\index.ts

slow api balabala

D:\node\nodeproject\jing\qmtrip\common\api\index.ts

数据库结构地址：

D:\node\nodeproject\jing\qmtrip\api\sql\init.sql

表结构：

D:\node\nodeproject\jing\qmtrip\common\model\index.ts

数据库

D:\node\nodeproject\jing\qmtrip\common\api\sql\_update.ts

程序入口：

---root目录下的server.js,此时初始化一系列服务，如apiserver，httpserver，等

initAPISql

initAPI

---initTcpServer(),tcp服务建立了self.\_.server (net.createServer)

---initShoeServer(),shoe服务建立了self.\_.websock

---同时初始化了service(client目录），包括本地，远程：D:\node\nodeproject\jing\qmtrip\common\api\remote.ts,远程调用使用了reconnect-core模块，提供callback函数

其中使用了shoe，zone，、

初始化initSql服务时，读取了api模块的文件，去除了client和sql模块，同时将这些文件地址补全后，初始该文件列表

---运行后，提示API监听端口4102， web端监听端口4002

---访问web，localhost:4002,进入

http://localhost:4002/#/login/?backurl=http:%2F%2Flocalhost:4002%2F

存在backurl

同时，D:\node\nodeproject\jing\qmtrip\common\api\index.ts的initShoeServer启动。

---

登陆入口：

当未登录状态时，访问localhost:4002/时，进入

http://localhost:4002/#/login/?backurl=http:%2F%2Flocalhost:4002%2F

===========================================================================

Models.xxx此时调用的就是该文件下的service的包。

D:\node\nodeproject\jing\qmtrip\nglibs\models\index.ts

ClientModels: D:\node\nodeproject\jingli\qmtrip\nglibs\models\index.ts

D:\node\nodeproject\jingli\qmtrip\common\model\remote.ts 定义了函数定义如，get

notice存在于

D:\node\nodeproject\jing\qmtrip\api\notice\index.ts

D:\node\nodeproject\jing\qmtrip\api\\_types\notice\notice.ts

### ~~任務1：新加入用户后，给管理员发送通知。（已完）~~

staff表中每个员工存在roleId

company存在companyId

D:\node\nodeproject\jingli\qmtrip\public\corp\StaffManage\controller.js

是staffManagement控制器

用户注册时，系统进行数据库操作位置(wechat share)

D:\node\nodeproject\jing\qmtrip\api\staff\index.ts

### ~~微信添加员工~~

D:\node\nodeproject\jing\qmtrip\www\ionic\login\invited-staff-two.html

D:\node\nodeproject\jing\qmtrip\www\ionic\login\invited-staff-two.ts

处理：

D:\node\nodeproject\jing\qmtrip\www\ionic\login\invited-staff-two.ts

第130添加处理notify请求

D:\node\nodeproject\jing\qmtrip\api\company\index.ts

存在给企业发送通知的范例

### ~~批量添加员工~~

D:\node\nodeproject\jing\qmtrip\www\corp-mgr\index\index.html

D:\node\nodeproject\jing\qmtrip\www\corp-mgr\index\index.ts

处理batchImportStaff的实现

D:\node\nodeproject\jing\qmtrip\api\staff\index.ts

处理notify：

在92行

### ~~手动添加员工~~

D:\node\nodeproject\jing\qmtrip\www\ionic\department\new-staff.html

D:\node\nodeproject\jing\qmtrip\www\ionic\department\new-staff.ts

在249行

总的处理办法：

D:\node\nodeproject\jing\qmtrip\api\notice\index.ts

此文件给定了创建appMessage

-----------------------------------------------------------------------------------------------------

代码如下啊：

//below sending notice to company admin

let company=await API.company.getCompanyById($scope.form.companyId);

let managers=await company.getManagers({withOwner:true});

//给各个企业管理员发送通知

await Promise.all(managers.map( (manager) => {

return API.notify.submitNotify({

accountId: manager.id,

key: "qm\_notify\_admins\_add\_staff",

values: {

staff:$scope.form.name

}

});

}));

-----------------------------------------------------------------------------------

var Company=require("api/\_type/company"); //added by jack

//below sending notice to company admin: added by jack

let company=await Company.getCompany({id: $scope.form.companyId});

let managers=await company.getManagers({withOwner:true});

await Promise.all(managers.map( (manager) => {

return API.notify.submitNotify({

accountId: manager.id,

key: "qm\_notify\_admins\_add\_staff",

values: {

staff:$scope.form.name

}

});

}));

----------------------------------------------------------------------------------------------------

backdetail

let managers=$scope.staff.company.getManagers({withOwners:false});

Promise.all(managers.map((manager)=>{

API.notice.submitNotify({

accountId:manager.id,

key:"qm\_notify\_admins\_add\_staff",

values:{

staff:$scope.staff.name

}

});

}));

----------------------------------------------------------------------------------------------------

let staff = $scope.staff;

let managers=await staff.company.getManagers({withOwners:false});

console.log("this is in backtoDetail: ", managers);

await Promise.all(managers.map((manager)=>{

console.log("in managers");

}));

--------------------------------------------------------------------------

async function sendNoticeToAdmins(){

//below sending notice to company admin: added by jack

API.require("notify");

await API.onload();

console.log("api load: ",API);

let managers=await $scope.staff.company.getManagers({withOwners:false});

console.log("new-staff.ts: retrieve staff: ",managers);

await Promise.all(managers.map( (manager) => {

return API.notify.submitNotify({

accountId: manager.id,

key: "qm\_notify\_admins\_add\_staff",

values: {

staff:$scope.form.name

}

});

}));

console.log("add new staff send notice done");

}

--------------------------------------------------------------------------

notice的标准定义

D:\node\nodeproject\jing\qmtrip\api\notice\index.ts

notice的创建和template的生成、submitNotify的声明

D:\node\nodeproject\jing\qmtrip\api\notify\index.ts

根据companyId获取company对象

全局查Company.get

API.company.getCompanyById(staff.companyId)

Models.company.get(staff.companyId)

问题：

D:\node\nodeproject\jing\qmtrip\api\\_decorator.ts

此文件

let company = await Models.company.get(tp["companyId"]);//此处为什么不能用tp.company

D:\node\nodeproject\jing\qmtrip\public\corp\StaffManage\controller.js

D:\node\nodeproject\jing\qmtrip\public\agency\companyList\controller.js

createStaff

==========================================================

clientexports decorator

處理方法定義：

D:\node\nodeproject\jing\qmtrip\api\staff\index.ts ：150行定義了sendNoticeToAdmains函數

微信分享處理位于

D:\node\nodeproject\jing\qmtrip\api\auth\index.ts 460用於處理wechat share的notice send

手動添加用戶時，在new-staff中調用位于api/staff/index中的sendNoticeToAdmins

D:\node\nodeproject\jing\qmtrip\www\ionic\department\new-staff.ts ：337和353 前段調用

D:\node\nodeproject\jing\qmtrip\api\\_types\staff\staff.ts 165 供前段調用

批量增加用戶

D:\node\nodeproject\jing\qmtrip\api\staff\index.ts ：749調用sendNoticeToAdmin

===============================================================================================

D:\node\nodeproject\jing\qmtrip\api\\_types\staff\staff.ts

getBatchAddStaffEmail()

console.log('testServerFunc');

skipLink@<%=appMessageUrl || '' %> 在appmessage.html如何處理

https://zhuanlan.zhihu.com/p/20297283?columnSlug=FrontendMagazine

<http://blog.wolksoftware.com/decorators-reflection-javascript-typescript>

### ~~任務 申請單多行程支持~~

左侧的弹出总框D:\node\nodeproject\jingli\qmtrip\www\ionic\index.html 其后台处理逻辑

D:\node\nodeproject\jingli\qmtrip\www\ionic\controller.ts

D:\node\nodeproject\jingli\qmtrip\www\ionic\index.html 管理企业，消息，退出页面。

处理流程：

Trip-type-select.ts(选择出行类型：住宿、交通)，确定后跳转到Create1.ts，处理destination的选择问题，完成destination的选择后跳转到booking-detail.ts,（选择时间、policy、等）完成后，next函数跳转booking-complete.ts，（显示出发、目的地等信息，跳转到预算计算https://t.jingli365.com/ionic/trip/budget.html

访问某个js文件

/#/login/invted-staff-three

处理逻辑：

D:\node\nodeproject\jingli\qmtrip\www\ionic\trip\budget.ts中通过传递budget的id号调用API.travelBudget.getBudgetInfo，获取用户出差提交的具体信息，包括departurePlace，destination2，destination3， hotel，subsidy等。

由budget.ts文件进入申请单，detail.html，传入approveId

=====================================================================================

### ~~行程单前端展示多行程~~

D:\node\nodeproject\jingli\qmtrip\www\ionic\trip\list.html 我的行程显示页面。

D:\node\nodeproject\jingli\qmtrip\www\ionic\trip-approval\list.html 出差请示显示具体信息

============================================================================================

### 任务 票据审核通知 （已完）

D:\node\nodeproject\jingli\qmtrip\www\ionic\invoice\invoice-detail.html

**await** staff.getNoticeToAdmins({  
 **companyId**:company.id,  
 **name**:staff.**name**});

Models

---nglibs创建的模块nglibs的服务：D:\node\nodeproject\jingli\qmtrip\nglibs\models\index.ts, 此时的models实际上是ModelRemoteOld远程请求cached的对象，并重新加载。

使用createService的方式

D:\node\nodeproject\jingli\qmtrip\common\model\remote.ts

D:\node\nodeproject\jingli\qmtrip\common\model\cached.ts

D:\node\nodeproject\jingli\qmtrip\common\model\abstract.ts

---存在与后台代码api/\_type/index.ts中，建立Models的exports变量。并且实现了ModelDelegate, 并且在api/\_service.ts中实现了initModels函数，为exports.Models， 进行了真正的实现

D:\node\nodeproject\jingli\qmtrip\api\\_types\index.ts

D:\node\nodeproject\jingli\qmtrip\common\model\delegate.ts //实现了this.target的具体指向。其实例化方式在D:\node\nodeproject\jingli\qmtrip\api\\_types\index.ts的函数initModels方式，在D:\node\nodeproject

\jingli\qmtrip\api\\_service.ts中实现该方法，至此，D:\node\nodeproject\jingli\qmtrip\api\\_types\index.ts中的export var Models变量得以实例化，并且，应该在内存中存在一分，可以查看容器

target实际指向是一个cached对象，包含5个函数，create，update，destory，

D:\node\nodeproject\jingli\qmtrip\common\model\sequelize.ts

D:\node\nodeproject\jingli\qmtrip\api\\_service.ts

后台使用时，实现缓存技术，当某个模块已在缓存中时，此时

D:\node\nodeproject\jingli\qmtrip\common\weakcache.ts

### ~~给鲸力发通知~~

仅有email和mobile，原型函数在D:\node\nodeproject\jingli\qmtrip\api\notify\index.ts的notifyDesignatedAccount()

D:\node\nodeproject\jingli\qmtrip\www\ionic\invoice\invoice-detail.ts

D:\node\nodeproject\jingli\qmtrip\nglibs\models\index.ts

D:\node\nodeproject\jingli\qmtrip\api\tripPlan\index.ts在函数saveTripDetailInvoice内掉调用函数notifyDesignatedAcount(函数与上同名)添加了给鲸力管理员发送通知。

# 模拟数据源

D:\node\nodeproject\jingli\qmtrip\api\travelBudget\index.ts 获取票据信息446 line和432行

# 爬虫爬取数据

代码hotel-api中的datasource分支。

D:\node\nodeproject\jingli\hotel-api\task\task-mgr.ts该文件中的handleTask()函数用于处理

## 爬虫流程：

该文件中的start()函数开始后，建立三张表，同时生成worker的多进程，用于处理事物，然后调用调用work.ts中的start函数。进而同过bind(this), 回调task.mgr.ts中的handleTask()，通过获取handlers的数组中的某个任务，同时将参数task传入，从而调用不同的handlers，如ctripHotelHandler()进行处理。该函数中调用了lib/下的文件进行数据爬取，如hotel数据，通过调用hotel.list()函数，

D:\node\nodeproject\jingli\hotel-api\task\worker.ts

# 四大航空公司

东航网页，生成参数js

<http://www.ceair.com/resource4/js/calendar/urlParse.js> 从url中解析参数

<http://www.ceair.com/resource4/js/calendar/events_ow.js> 生成url

Js调用控制

<http://www.ceair.com/resource4/js/calendar/events.js>

<http://www.ceair.com/resource4/js/calendar/calendar.js> 生成了Factory实例，引用了doSearch.js

<http://www.ceair.com/resource4/js/calendar/doSearch.js> 分解生成参数，提交post请求

Factory.js中存在shop(车间)，factory

获得http://www.ceair.com/ftpupload/resource2/js/public/data/city.js存在\_cityData字段

## 南航

\* 南航的北京城市码为PEK，不是BJS

\* flytype字段是动态改变的，如北京->青岛，该值为2

欧铁数据爬取

选择出行地和目的地时，调用的接口为：D:\node\nodeproject\jingli\common-api\api\place\index.ts

选择地点弹出框D:\node\nodeproject\jingli\qmtrip\nglibs\services\modal-dlg\city\dialog.html,其控制器

D:\node\nodeproject\jingli\qmtrip\nglibs\services\modal-dlg\city\index.ts, 继而回到初始页面

D:\node\nodeproject\jingli\qmtrip\www\ionic\trip\create.ts的queryAllPlaces，然后调用common-api的API.place.queryPlace进行postgresql数据库查询。该数据库为times\_common\_api的schema的place

生成预算

提交的请求参数：

{"budgets":[],"query":{"originPlace":"CT\_132","destinationPlace":"CT\_257","leaveDate":"2017-03-30T10:00:00.000Z","goBackDate":"2017-04-03T10:00:00.000Z","latestArrivalDateTime":"2017-03-30T10:00:00.000Z","earliestGoBackDateTime":"2017-04-03T10:00:00.000Z","isNeedTraffic":true,"isRoundTrip":true,"isNeedHotel":true,"businessDistrict":"23.183928,113.257353","hotelName":"广州A精品酒店","subsidy":{"hasFirstDaySubsidy":true,"hasLastDaySubsidy":true,"template":{"target":{"id":"38589340-091d-11e7-abbe-717d6188408e","subsidyMoney":"111.00","name":"222","travelPolicyId":"a25defe0-0884-11e7-8d40-11cd100850f0","isInternal":false,"createdAt":"2017-03-15T01:17:56.214Z","updatedAt":"2017-03-15T01:17:56.214Z","deletedAt":null,"created\_at":"2017-03-15T01:17:56.214Z","updated\_at":"2017-03-15T01:17:56.214Z","deleted\_at":null},"$model":{"target":{"$cache":{"id":"SubsidyTemplate","cache":{}}}},"$fields":{},"$resolved":{},"$parents":{}}}},"createAt":1489991356417}

D:\node\nodeproject\jingli\qmtrip\www\ionic\trip\create.ts的函数nextStep

D:\node\nodeproject\jingli\qmtrip\api\travelBudget\index.ts的函数getTravelPolicyBudget, 继而跳转到getTrafficBudget

,函数getTrafficBudget调用了函数getCityInfo