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

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

# 模块1：新加入用户后，给管理员发送通知。

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2 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

任务3

票据审核通知

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。