Dnode：

介绍：<http://www.jianshu.com/p/0e17f7419cc8> ; https://github.com/substack/dnode

Json形式，无需要IDL(Interface Definition Language).

Shoe提供stream interface，支持带有callback的websocket。

强类型RPC

COM/CORBA

https://webapplog.com/node-docker/

Browserify

用法：

Browserify client.js -o static/bundle.js

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

static async sendNoticeToAdmins(params:{companyId:string,name:string,noticeTemplate:string}):Promise<any>{

let company = await Models.company.get(params.companyId);

console.log("addbyhand: companyId: ",params.companyId);

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

managers.map((manager)=>{

console.log(manager.id);

console.log(manager.name);

});

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

return API.notify.submitNotify({

accountId: manager.id,

key: params.noticeTemplate,

values: {

staff:params.name

}

});

}));

}

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

**async** getNoticeToAdmins(params:{**companyId**:**string**,**name**:**string**,**noticeTemplate**:**string**}):Promise<**any**>{  
  
 **return** API.staff.*sendNoticeToAdmins*({**companyId**:params.**companyId**,**name**:params.**name**,**noticeTemplate**:params.**noticeTemplate**});  
}

# D:\node\nodeproject\jingli\qmtrip\api\staff\index.ts

static async sendNoticeToDesignatedAccount(params:{accountId?:string,name?:string,noticeTemplate?:string,email?:string,mobile?:string}):Promise<any>{

// let company = await Models.company.get(params.Id);

// console.log("addbyhand: companyId: ",params.accountId);

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

return API.notify.submitNotify({

accountId: manager.id,

key: params.noticeTemplate,

values: {

staff:params.name

}

});

}

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

@RemoteCall()

async getNoticeDesignatedAccount(params:{accountId?:string,name?:string,noticeTemplate?:string,email?:string,mobile?:string}):Promise<any>{

return API.staff.sendNoticeDesignatedAccount({accountId:params.accountId,name:params.name,noticeTemplate:params.noticeTemplate,email:params.email,mobile:params.mobile});

}

**async function** *sendNoticeToJinli*(){  
 *//added by jack* **var** current=Staff.*getCurrent*();  
 **let** managers=[{**mobile**:**'13810529805'**,**email**:**'notice@jingli365.com'**}];  
 **await *Promise***.all(managers.map((manager)=>{  
 current.getNoticeDesignatedAccount({  
 **accountId**:**""**,  
 **name**: current.name,  
 **noticeTemplate**: **"qm\_notify\_jingli\_upload\_invoice"** });  
 }));  
  
 **let** notice=**await** current.getNoticeToAdmin({  
  
 });  
}

Create1.ts

*/\*\*  
 \* Created by dev on 2017/3/8.  
 \*/***import** moment = **require**(**'moment'**);  
**import** { Staff } **from 'api/\_types/staff/staff'**;  
**import** {EApproveChannel} **from "api/\_types/approve/types"**;  
  
**var** msgbox = ***require***(**'msgbox'**);  
  
  
**var** defaultTrip = {  
 **beginDate**: moment().add(3, **'days'**).startOf(**'day'**).hour(18).toDate(),  
 **endDate**: moment().add(4, **'days'**).startOf(**'day'**).hour(9).toDate(),  
 **place**: undefined,  
 **placeName**: **''**,  
 **reason**: **''**,  
  
 **traffic**: **true**,  
 **fromPlace**: undefined,  
 **round**: **true**,  
  
 **hotel**: **false**,  
 **hotelPlace**: undefined  
};  
**type** TripDefine = **typeof** defaultTrip;  
**function** *TripDefineFromJson*(obj: **any**): TripDefine{  
 **if**(obj == undefined)  
 **return** defaultTrip;  
 obj.**beginDate** = **new *Date***(obj.**beginDate**);  
 obj.**endDate** = **new *Date***(obj.**endDate**);  
 **return** obj as TripDefine;  
}  
  
  
**export async function** *CreateController*($scope, $storage, $loading, ngModalDlg, $ionicPopup, Models, City){  
 ***require***(**'./trip.scss'**);  
 ***API***.**require**(**'tripPlan'**);  
 **await *API***.*onload*();  
 $scope.**currentStaff** = **await** Staff.*getCurrent*();  
 **let** currentCompany = $scope.currentStaff.company;  
 *//$scope.staffName=$scope.currentStaff.name;  
 //$scope.staffAvatar=$scope.curentStaff.avatar;  
 //let currentCompany = $scope.currentStaff.company;* **var** ret = **await** $scope.currentStaff.testServerFunc();  
 ***console***.log(**'$scope.currentStaff.testServerFunc() return'**, ret);  
  
  
  
  
 $scope.$watch(**'trip.place'**,**function**(n,o){  
 **if**(n != undefined){  
 $scope.**trip**.**hotel** = **true**;  
 }  
 *//added by jack, if new value is generated by <ng-select-city> for destination place, then directed to a new detail page* ***window***.**location**.**href** = **"#/trip/booking-details.html"**;*// destination data should be sent by url* });  
  
 **let** trip;  
 **try** {  
 trip= *TripDefineFromJson*($storage.local.get(**'trip'**));  
 } **catch**(err) {  
 trip = {};  
 }  
  
 **if**(!trip.**regenerate**) {  
 trip = defaultTrip;  
 **await** $storage.local.set(**'trip'**, trip);  
 }**else** {  
 **var** today = moment();  
 **if** (!trip.**beginDate** || (**new *Date***(trip.**beginDate**) < **new *Date***())) {  
 trip.**beginDate** = today.startOf(**'day'**).hour(18).toDate();  
 }  
  
 trip.**regenerate** = **false**;  
 }  
  
 $storage.local.set(**'trip'**, trip);  
 $scope.**trip** = trip;  
 $scope.$watch(**'trip'**, **function**(){  
 $storage.local.set(**'trip'**, $scope.**trip**);  
 }, **true**);  
  
 $scope.$watch(**'trip.place.name'**, **function**($newVal, $oldVal) {  
 **if** ($newVal != $oldVal) {  
 $scope.**trip**.**hotelPlaceObj** = undefined;  
 $scope.**trip**.**hotelPlace** = **''**;  
 $scope.**trip**.**hotelName** = **''**;  
 }  
 });  
  
 **async function** *queryAllPlaces*(keyword: **string**, isAbroad: **boolean**){  
 **let** key= **'hot\_cities\_170228\_domestic'**;  
 **if** (isAbroad) {  
 key = **'hot\_cities\_170228\_abroad'**;  
 }  
 **if** (!keyword) {  
 **let** hotCities = $storage.local.get(key)  
 **if** (hotCities && hotCities[0] && hotCities[0].**id**) {  
 **return** hotCities;  
 }  
 }  
 **var** places = **await *API***.**place**.queryPlace({**keyword**: keyword, **isAbroad**: isAbroad});  
 **if** (!keyword) {  
 $storage.local.set(key, places);  
 }  
 **return** places;  
 }  
 **async function** *queryAbroadPlaces*(){  
 **let** key = **'abroad\_cities\_170228'  
 let** abroad = $storage.local.get(key);  
 **if**(!abroad){  
 abroad = **await *API***.**place**.queryCitiesGroupByLetter({**isAbroad**:**true**});  
 $storage.local.set(key,abroad);  
 }  
 **return** abroad;  
 }  
 **async function** *queryDomesticPlaces*(){  
 **let** key = **'domestic\_cities\_170228'**;  
 **let** domistic = $storage.local.get(key);  
 **if**(!domistic){  
 domistic = **await *API***.**place**.queryCitiesGroupByLetter({**isAbroad**:**false**});  
 $storage.local.set(key,domistic);  
 }  
 ***console***.log(domistic)  
 **return** domistic;  
 }  
 $scope.**placeSelector** = {  
 queryAll: *queryAllPlaces*,  
 queryAbroad: *queryAbroadPlaces*,  
 queryDomestic: *queryDomesticPlaces*,  
 display: (item)=> {  
 **if** (item.isAbroad && item.**code**) {  
 **return `**${item.**name**}**(**${item.**code**}**)`**;  
 }  
 **return** item.**name** }  
 };  
  
 $scope.**fromPlaceSelector** = {  
 queryAll: *queryAllPlaces*,  
 queryAbroad: *queryAbroadPlaces*,  
 queryDomestic: *queryDomesticPlaces*,  
 display: (item)=> {  
 **if** (item.isAbroad && item.**code**) {  
 **return `**${item.**name**}**(**${item.**code**}**)`**;  
 }  
 **return** item.**name** }  
 };  
 $scope.**projectSelector** = {  
 query: **async function**(keyword){  
 **var** staff = **await** Staff.*getCurrent*();  
 **var** options = {**where**:{**companyId**: staff.company.id}};  
 **if**(keyword){  
 options.**where**[**"name"**] = {**$like**: **'%'**+keyword+**'%'**};  
 }  
 **var** projects = **await** Models.**project**.find(options);  
 **return** projects;  
 },  
 display: (item)=>item.**name**,  
 create: **async function**(name){  
 **return** {  
 **name**: name,  
 **id**: undefined  
 }  
 },  
 done: **function**(val) {  
 $scope.**trip**.reason = val.**name** ? val.**name**: val;  
 }  
 };  
  
 $scope.**hotelPlaceSelector** = {  
 done: **function**(val) {  
 **if** (!val.point || !val.point.lat || !val.point.lng) {  
 $scope.showErrorMsg(**"获取住宿位置失败"**);  
 **return**;  
 }  
 $scope.**trip**.**hotelPlace** = val.point.lat + **","** + val.point.lng  
 $scope.**trip**.**hotelName** = val.**title**;  
 }  
 };  
  
 $scope.nextStep = **async function**() {  
 **let** trip = $scope.**trip**;  
  
 **let** number = 0;  
 **if**(trip.traffic){  
 number = number + 1;  
 }  
 **if**(trip.round){  
 number = number + 1;  
 }  
 **if**(trip.hotel){  
 number = number + 1;  
 }  
 **try**{  
 **await** currentCompany.beforeGoTrip({**number**: number});  
 }**catch**(e){  
 $ionicPopup.**alert**({  
 **title**: **'行程余额不足'**,  
 **template**: **'行程余额不足无法获取预算，请通知管理员进行充值或使用特别审批'** })  
 **return false**;  
 }  
 **if** ($scope.currentTpSts && $scope.currentTpSts.**length** && (!$scope.**subsidy** || !$scope.**subsidy**.**template**)) {  
 $scope.showErrorMsg(**'请选择补助信息'**);  
 **return false**;  
 }  
 **let** beginMSecond = ***Date***.now();  
 ***API***.**require**(**"travelBudget"**);  
 **await *API***.*onload*();  
  
 **if**(!trip.place || !trip.place.**id**) {  
 $scope.showErrorMsg(**'请填写出差目的地！'**);  
 **return false**;  
 }  
  
 **if**(!trip.reasonName) {  
 $scope.showErrorMsg(**'请填写出差事由！'**);  
 **return false**;  
 }  
  
 **if**(!trip.traffic && ! trip.hotel) {  
 $scope.showErrorMsg(**'请选择交通或者住宿！'**);  
 **return false**;  
 }  
  
 **if**(trip.traffic && (!trip.fromPlace || !trip.fromPlace.**id**)) {  
 $scope.showErrorMsg(**'请选择出发地！'**);  
 **return false**;  
 }  
  
 **if**(moment(trip.endDate).toDate().getTime() - moment(trip.beginDate).toDate().getTime() <= 0) {  
 $scope.showErrorMsg(**'到达时间不可晚于离开时间！'**);  
 **return false**;  
 }  
  
 **let** params = {  
 **originPlace**: trip.fromPlace? trip.fromPlace.**id** : **''**,  
 **destinationPlace**: trip.place ? trip.place.**id** : **''**,  
 **leaveDate**: moment(trip.beginDate).toDate(),  
 **goBackDate**: moment(trip.endDate).toDate(),  
 **latestArrivalDateTime**: moment(trip.beginDate).toDate(),  
 **earliestGoBackDateTime**: moment(trip.endDate).toDate(),  
 **isNeedTraffic**: trip.traffic,  
 **isRoundTrip**: trip.round,  
 **isNeedHotel**: trip.hotel,  
 **businessDistrict**: trip.hotelPlace,  
 **hotelName**: trip.hotelName,  
 **subsidy**: $scope.**subsidy** };  
  
 **if**(params.**originPlace** == params.**destinationPlace**){  
 msgbox.log(**"出差地点和出发地不能相同"**);  
 **return false**;  
 }  
  
 **let** front = [**'正在验证出行参数'**, **'正在匹配差旅政策'**, **'正在搜索全网数据'**, **'动态预算即将完成'**];  
 $loading.reset();  
 $loading.**start**({  
 **template**: **'预算计算中...'** });  
 **let** idx = 0;  
 **let** isShowDone = **false**;  
 **let** budget;  
 **let** timer = *setInterval*(**async function**() {  
 **let** template = front[idx++]+**'...'**;  
 **if** (idx >= front.**length**) {  
 *clearInterval*(timer);  
 isShowDone = **true**;  
 **if** (budget) {  
 **return** *cb*();  
 }  
 }  
 $loading.reset();  
 $loading.**start**({  
 **template**: template,  
 });  
 }, 1000);  
  
 **let** calTimer;  
 **try** {  
 calTimer = *setTimeout*( () => {  
 *alert*(**'系统错误，请稍后重试'**);  
 $loading.end();  
 }, 60 \* 1000);  
  
 budget = **await *API***.travelBudget.*getTravelPolicyBudget*(params);  
 **if** (isShowDone) {  
 *cb*();  
 }  
 } **catch**(err) {  
 *clearTimeout*(calTimer);  
 *clearInterval*(timer);  
 $loading.end();  
 *alert*(err.**msg** || err);  
 }  
 **let** endMSecond = ***Date***.now();  
 *spendMS*(beginMSecond, endMSecond);  
 **function** *spendMS*(begin, end) {  
 ***console***.info(**'开始时间:'**, begin, **'结束时间:'**, end, **'耗时:'**, end - begin);  
 }  
 **function** *cb*() {  
 *clearTimeout*(calTimer);  
 $loading.end();  
 ***window***.**location**.**href** = **"#/trip/budget?id="**+budget;  
 }  
 }  
  
 $scope.specialApprove = **async function**() {  
 ***API***.**require**(**"travelBudget"**);  
 **await *API***.*onload*();  
  
 **let** trip = $scope.**trip**;  
  
 **if**(!trip.place || !trip.place.**id**) {  
 $scope.showErrorMsg(**'请填写出差目的地！'**);  
 **return false**;  
 }  
  
 **if**(!trip.reasonName) {  
 $scope.showErrorMsg(**'请填写出差事由！'**);  
 **return false**;  
 }  
  
 *// if(!trip.traffic && ! trip.hotel) {  
 // $scope.showErrorMsg('请选择交通或者住宿！');  
 // return false;  
 // }* **if**(trip.traffic && (!trip.fromPlace || !trip.fromPlace.**id**)) {  
 $scope.showErrorMsg(**'请选择出发地！'**);  
 **return false**;  
 }  
  
 **let** params = {  
 **originPlace**: trip.fromPlace? trip.fromPlace.**id** : **''**,  
 **destinationPlace**: trip.place ? trip.place.**id** : **''**,  
 **leaveDate**: moment(trip.beginDate).toDate(),  
 **goBackDate**: moment(trip.endDate).toDate(),  
 **latestArrivalDateTime**: moment(trip.beginDate).toDate(),  
 **earliestGoBackDateTime**: moment(trip.endDate).toDate(),  
 **isNeedTraffic**: trip.traffic,  
 **isRoundTrip**: trip.round,  
 **isNeedHotel**: trip.hotel,  
 **businessDistrict**: trip.hotelPlace,  
 **hotelName**: trip.hotelName,  
 };  
  
 **if**(params.**originPlace** == params.**destinationPlace**){  
 msgbox.log(**"出差地点和出发地不能相同"**);  
 **return false**;  
 }  
  
 **try** {  
 $loading.end();  
 ***window***.**location**.**href** = **"#/trip/special-approve?params="**+***JSON***.stringify(params);  
 } **catch**(err) {  
 $loading.end();  
 *alert*(err.**msg** || err);  
 }  
 }