dingding-web

该项目目前尚未迁移，位于appService源码组中。该项目主要做移动端H5页面，项目中大部分会嵌入到app中，比如一些活动、优惠券等等。其中一大部分已经不再使用，

但没有删除。

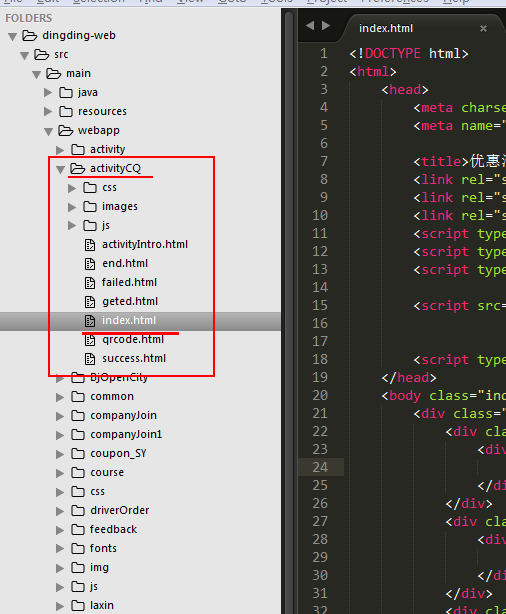
app嵌入页测试

线上地址：

测试地址：<http://test.pdhz-dddc.com/dingding-web/>

可以通过测试链接地址或线上地址+文件夹名字，则默认返回的是这个文件下的index.html 。

如：<http://test.pdhz-dddc.com/dingding-web/> aqctivityCQ/



各个项目模块说明

/dingding-web/src/main/webapp/....

css和js和page目录是一体的，里面涉及到的activity,billing,coupon,user模块没负责过，也不确定线上是否有在使用过。plugin目录存放的是所有需要用到的一些插件，

需要即引用即可。Common目录下的system.js封装了开发中用到的一些公用的方法，这里可能会经常使用到。

1 内嵌在客户端app内的H5页面(正常使用)

course: 使用教程页面 ----内嵌app

feedback: 意见反馈 ---内嵌app

service: 服务页面 ---内嵌app

tourism: 旅游路线 ---内嵌app

userGuides: 用户指南 ---内嵌app

xiecheng: 携程单详情页面 ---内嵌app

webDriverRegister/app: 专车车主认证 ---内嵌app

webXmlPages/404.html: 配置网络异常下的404页面 --内嵌app

rule: 各种协议，隐私说明，法律协议，常见问题等 ---内嵌app

specialCarMoneyRule: 专车计费规则 ---内嵌app

2 内嵌在客户端app内的H5页面(下架过期或未确定的)

laxin: 拉新活动 ---内嵌app(下架)

midAutumnActivity: 中秋活动 ---内嵌app(下架)

renteqsale: 以租代售 ---内嵌app(下架)

shareplan: 共享计划 ---内嵌app(下架)

shengdan: 圣诞元旦双旦活动 ---内嵌app(下架)

shop: 商城 ---内嵌app(下架)

store: 门店 ---内嵌app(下架)

regactive: 免费洗车和搭车现金券(下架)

activity: 老早之前的内嵌app里面的活动页面(不确定线上是否还在使用)

3 提供移动url的H5移动端页面 或 嵌入在微信公众号里面的H5页面

companyJoin: 扫码认证车主 --移动端页面(线上在使用)

superCar: 超级车微信端下单页面 ---移动url(超级车，不确定是否在使用)

companyJoin1: 最老版的扫码认证车主备份(已下架)

activityCQ：重庆优惠券活动页面--内嵌微信公众号(不确定线上是否还在使用)

BjOpenCity: 北京开城优惠券活动页面--内嵌微信公众号(不确定线上是否还在使用)

coupon\_SY: 沈阳优惠券活动--内嵌微信公众号(不确定线上是否还在使用)

driverOrder: 司机订单 ----内嵌在微信公众号里(不确定线上是否还在使用)

recruit: 司机招募移动端页面 ---url访问(不确定线上是否还在使用)

recruit\_pc：司机招募pc端页面 ---url访问(不确定线上是否还在使用)

H5页面开发总结和注意事项

1 通用方法或者格式

每个项目对应自己的目录，每个目录下存放对应的css,js,img目录。

2 ios和android加载web页面的形式

ios和android一般是加载web端提供的项目的路径，然后提供给客户端，客户端在对应的位置加载web的url。或者，客户端(app)请求后台服务器接口，

然后由后台服务器来控制跳转到哪个web页面，这时只需提供web页面的url给服务器即可。

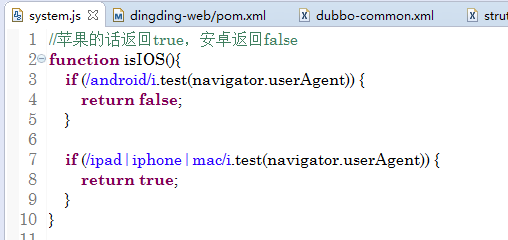
3 客户端和H5数据交互

第一种方式很简单，只适用于web端获取客户端传来的数据。通过在url后面拼接参数，然后才web端获取url后面的参数即可获得从客户端传给web端的部分值。

这种方式只适用于获取单个或少数值，多个即不适用。

第二种方式比较通用，既可以获取客户端传过来的值，又可以将值传给客户端，项目中一般采用这种方法。需要分ios和android来操作js。Common目录下封装了js

区分ios和android的方法，直接调用即可。/dingding-web/src/main/webapp/common/js/system.js

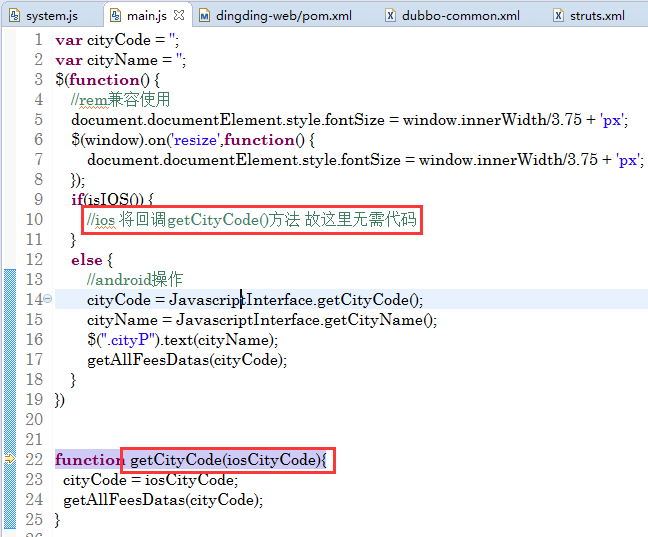


4 web端获取客户端传过来的值

**IOS**下，首先双方约定一个方法，这个方法的功能是需要获取ios端传过来的某些值，然后在js中全局定义这样一个方法，然后客户端会在加载url的时候调用js里面的

这个方法(真正意义上是ios在加载web url的时候回进行一个回调函数的调用，这个回调函数就是在js里面定义好的。)，然后我们就可以在js里定义的这个方法里面通过参数获取到客户端传过来的参数值。

eg: /dingding-web/src/main/webapp/specialCarMoneyRule/main.js



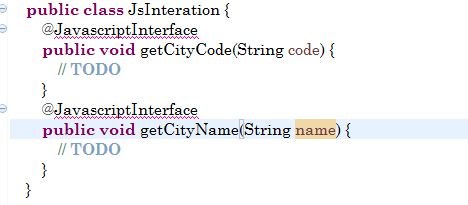
function getCityCode()就是IOS客户端加载web url的时候会回调的方法，只需要在js中定义这个回调函数，然后通过参数来获取IOS客户端传过来的值。

Android下,相对来说比较简单。只需要在客户端定义一个方法，然后这个方法返回web端需要的值，然后在web端就可以通过JavascriptInterface这个对象去调用客户端定义的方法，就可以取到值。

在上图中，cityCode = JavascriptInterface.**getCityCode()**; cityName = JavascriptInterface.**getCityName()**; 这两个方法是在Android客户端中定义的两个方法，每个方法对应返回web需要的值。

JavascriptInterface是Android下的一个WebView设置供JavaScript调用的交互接口。android.webkit.JavascriptInterface;

eg：java代码如下

 更详细的用法请参考：<http://www.imooc.com/article/1475>

5 web端传值给客户端

ios下,调用window.webkit.messageHandlers.iosMethod.postMessage(Object)即可，这里的iosMethod是ios客户端定义好的方法，然后通过调用这个方法的postMessage方法就能将值传给客户端，参数Object是一个json对象，客户端需要接收哪些字段值。

android下，通过调用JavascriptInterface.androidMethod(a,b,c)即可，这里的androidMethod是android客户端下定义好的方法用来接收web端传过去的值，需要哪些就传几个参数。

eg: //dingding-web/src/main/webapp/js/activity/sharecode.js 页面分享功能

