销售端

2018.3.5

业务

基本情况

- 销售端主要是H5开发的套壳项目(基于jsbridge)
- 超市与订单两个维度的统计数据展示
 - 装机0单超市 14天未下单超市 本月历史首单超市 本月历史二单超市 日 新增超市数
 - 。 日有效订单数量 日累积交易金额 月累计交易金额
- 显示超市相关信息<已注册与未注册>
 - 。 超市名称 联系人 详细地址 <联系方式>
- 对已注册超市进行拜访打点, 地点微调
- 新增未注册超市,并进行拜访记录
- 对不同角色的用户进行了功能限制
 - 。 出于不同角色的实际需要

"历史"

初期销售端是基于钉钉的H5项目,现为独立发版的H5套壳项目。

主要解决的问题:

- 1. 定位问题(ios10升级导致的定位问题,详情参考'销售端定位失效case分析')
- 2. 拍照禁止相册问题

版本情况

以下为2017年主要大版本以及主要业务展示,具体版本情况见"销售端版本详情"

环境&&账号

1. 测试环境

地址: http://qa.market-sales-h5.wmdev2.lsh123.com/

账号&密码:

wangxueqi01@lsh123.com 654321Mis

发布命令:

fisp release -wmr sales -d qa sh deploy.sh home qa

2. 线上环境

地址: http://xs.lshyg.cn/

账号&密码:

wangxueqi01@lsh123.com Wxq921210

线上包下载地址:

http://xs.lshyg.cn/home/download/app

技术

需要关注的组件

- 1. address-dialog.js map.js 主要是涉及了地图定位的使用,包括 但不限于地址校准,附近超市搜索,超市打点与拜访,显示高 德地图中超市等,主要涉及的页面有
 - viewmap \ marketinfo \ addrecord \ nearby
- 2. upload-image.js image-info.js 前者是涉及了图片上传,后者主要是针对iOS系统不会自行旋转竖向拍摄的照片而引入的**提取照片信息**组件,v1.3和v1.3.1中对图片上传策略调整较多,包括开放超市详情页上传图片入口和查看大图功能

重点理解的功能

- 1. 路由回退机制 由于销售端中部分页面可能出现页面层级较深需要回退的情况(例如:我的超市-超市列表-超市详情-拜访记录-修改拜访记录或未注册超市-区域筛选/销售筛选列表-超市列表-超市详情-新增拜访记录等),而销售端使用backbone实现的是单页面应用,因此不能使用传统意义上的页面回退;同时一个页面可能有多个入口(例如:新增未注册超市页面可能是从超市列表页进入的,也可能是从区域筛选列表页进入的),入口的判断也是个问题。这里需要着重看一下页面间参数是如何保存与使用的,如何判断页面回退逻辑的(之前有单独发过这一块的分析和总结)。
- 2. queryString 页面间的参数传递和使用与优供商城和之前的司机端不同,视图间的参数传递使用了queryString的方式,可以按需索取,处理机制都是在 router 中进行的,在第一次串讲过程中也对这方面做了简单的解释,需要体会一下这样做的好处。
- 3. 用户登录与用户资料更新的实现形式,将销售端账号与钉钉绑定的策略与优供商城不同,销售端的用户登录也是在 单页面应用 的体系中实现的,所以用户信息的更新与登录状态的调整都需要由前端来调整,这点区别需要理解和学习一下。

React Native部分

- 所在版本 提交售后版本,在超市订单详情处申请售后/编辑售后,进入选择商品页, 选择后编辑售后。
- 入□

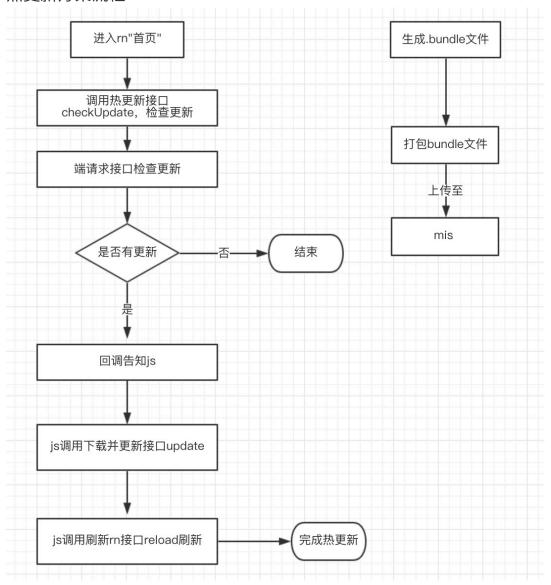
```
// sales -> widget -> mymarket -> order-view.js

JavascriptBridge.initRNBridge({
    token: that.getCookie('token'),
    receipt_order_id: receipt_order_id,
    order_id: order_id,
    host_url:host_url
```

• 项目地址

https://github.com/lianshang/sales-RN

- 热更新
 - 。 开发分支
 - sales-RN dev
 - spm-supply-fe feature-sales-hotupdate
 - 。 进展
 - 已开发完成,并测试完成,缺少mis后台管理优化。
 - 。 热更新方案流程



。 接口约定

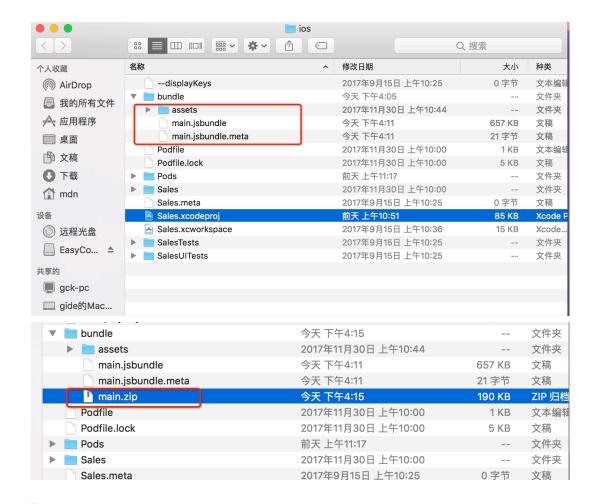
```
//1. 检查更新并下载接口
      接口名: checkUpdate
      参数:
      callback (修改)
           返回数据
       {
        ret: XX, //错误码返回 (有版本更新时为0; 无版本更新时为-1;
)
       content: {}, //下载文件时,后端传过来的数据
       msg: '', //错误数据, ret==0时, 即有版本更新的时候 为空
;否则不为空;
       }
      //2.更新
      接口名: update
      参数:
      content
      callback
      //3.重启rn
      接口名: reload
      参数:
      data
      callback
```

• 打bundle包流程

- 。 1.iOS 上传包
 - 1) 在根目录下执行

react-native bundle --entry-file index.ios.js --bundleoutput ./ios/bundle/main.jsbundle --platform ios --assetsdest ./ios/bundle --dev false

2) 找到bundle, 打包bundle下的所有文件 至 main.zip(必须为main.zip)



ios问题记录:

直接右键压缩 导致找不到main.zip格式

两个方案:

- 1. 你还按照这种方式打包,找到LSHomeWebVC第139行,改成 NSString *fileExtensionName = @"/main.jsbundle";
- 2. 本地代码不用动,先本地建个main文件夹,然后压缩这个main 文件夹,传到mis上去

■ 2.android 上传包

1) 在根目录下执行

react-native bundle --platform android --dev false -entry-file index.android.js --bundle-output
android/app/src/main/assets/index.android.bundle -assets-dest android/app/src/main/assets

2) 找到bundle, 打包bundle下的所有文件至XXX.zip

