

云开发公开课

客流 小程序



目录

云开发项目实战-SCRM 小程序	2
1.1 项目概述	2
1.2 任务目标	2
1.3 准备工作	3
1.4 实战架构	3
1.5 实战任务	3
1.5.1 绘制界面原型	3
1.5.2 创建小程序·云开发环境	8
1.5.3 初始化云开发能力	10
1.5.4 创建底部 Tab	11
1.5.5 创建每一个主体功能对应的目录和页面	14
1.5.6 细化并实现每个主要界面的功能	14
1.6 完成标准	19

云开发项目实战-SCRM 小程序

1.1 项目概述

随着微信小程序不断发展壮大,传统的 CRM 厂商也在不断向微信上迁移,毕竟微信的背后是巨大的用户和流量,还有极其方便的移动支付渠道。用微信小程序来做 CRM,和以前的做法存在一些比较明显的差异,主要包括这几个方面:

- 用户的使用习惯不同
- 技术上的实现方式不同
- 运营方面的做法不同

所以,业内给这一类的 CRM 应用起了个新名字,叫做 Social CRM,简称 SCRM。

1.2 任务目标

本主题将会为您讲解:

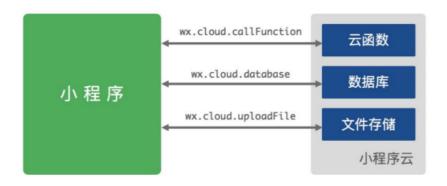
- SCRM 在商业层面如何理解,为什么目前会变得流行起来?
- 当前市面上 SCRM 的发展状况如何?
- SCRM 类应用有哪些通用的业务场景和组件?
- SCRM 在技术上具体怎么实现?

最后,带你上手实际写一个 SCRM 小程序,完整覆盖小程序+云开发的完整流程。

1.3 准备工作

在 PC 中下载小程序项目脚手架,下载地址: https://gitee.com/mumu-osc/scrm

1.4 实战架构



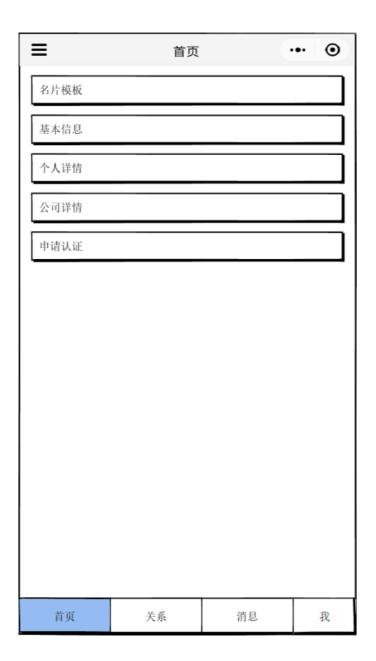
1.5 实战任务

*有能力的同学推荐尝试自行编写代码,不要偷看参考答案噢!

1.5.1 绘制界面原型

选择一款你熟悉的绘图工具,绘制出整个应用的界面原型。界面原型的主要作用是用来定义整个应用的功能结构,以及数据结构,防止应用跑偏。

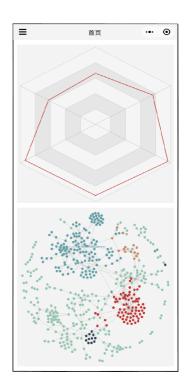




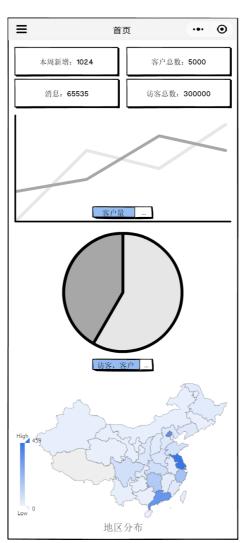












1.5.2 创建小程序•云开发环境

1、打开微信 IDE,创建一个新的小程序项目,并选择导入树洞 Demo 的目录,AppID 修改为你已经注册好的小程序 AppID。



2、打开项目后,点击顶部工具栏上的"云开发"



3、自定义填入环境名称和环境 ID, 点击"确定"创建云开发环境,即可进入云开发控制台。



4、小程序后续需要使用云开发提供的云数据库能力,数据库基于 MongoDB,需要先创建一个集合。进入云开发控制台,找到"数据库",点击左侧边栏的"+"按钮创建集合。



5、我们一共需要创建7个集合, 名称和功能如下:

名称	作用
actions	用来记录用户的行为,如: visit/msg/collection/request等。
card-details	用来记录名片详情,包括基本信息和图文详情等。
card-items	名片项目, 简略信息, 只包含: 真实姓名、昵称、用户名、手机号、职位、公司名。
relations	用来记录用户之间的关系。
statistics-analytics	用来存储每个用户的分析数据:如今日接收到多少条消息、发送了多少 条消息等。
statistics-first-page	用来存储每个用户个人首页的统计数据。
users	用来记录用户资料。

1.5.3 初始化云开发能力

打开 scrm 目录下的 app.js 文件,在 onLaunch 事件最后初始化云开发能力,env 参数为上一任务中创建的环境 ID,traceUser 参数值为 true,并保存。

```
config.json • app.js
 ▶ 🗀 msg-list
 ▶ ☐ relation-analytics
 ► ☐ relation-customer
 ► 🗀 relation-visitor
                                                   App({
                                                      onLaunch: function () {
 ▶ □ subject-action
                                                        let that=this;
if (!wx.cloud) {
▼ 🗁 pages
 ► 🗀 card-detail
                                                           console.error('请使用 2.2.3 或以上的基础库以使用云能力')
 ► 🗀 choose-card-template
 ▶ 🗀 data-main
                                                           wx.cloud.init({
                                                             // env 参数说明:
// env 参数决定接下来小程序发起的云开发调用(wx.cloud..xx)
// 此处请填入环境 ID,环境 ID 可打开云控制台查看
// 如不填则使用默认环境(第一个创建的环境)
 ▶ □ edit-basic-info
 ▶ ☐ edit-company-info
 ▶ □ edit-detail-images
 ▶ 🗀 edit-main
 ▶ 🗀 index
                                                              traceUser: true,//自动跟踪用户
 ▶ 🗀 logs
 ▶ 🗀 map
 ▶ 🗀 mine
 ▶ 🗀 msg-main
                                                        const db=wx.cloud.database();
 ▶ 🗀 poster-list
                                                        const _=db.command;
 ▶ [ ¬ relation-main
 ▶ 🗀 share-main
                                                        // 展示本地存储能力
var logs = wx.getStorageSync('logs') || [];
logs.unshift(Date.now());
 ▶ 🗀 verify
                                                         wx.setStorageSync('logs', logs);
  wxss app.wxss
 () sitemap.json
                                                        wx.login({
.gitignore
                                                           success: res => {
  console.log(res);
README.md
(o) project.config.json
```

1.5.4 创建底部 Tab

打开 app.js,增加这些内容,创建出底部的 Tab 页:

```
"tabBar": {
    "selectedColor": "#13227a",
    "borderStyle": "white",
    "backgroundColor": "#ffff",
   ▶ 🗀 msg-list
   ► ☐ relation-analytics
   ▶ ☐ relation-customer
   ▶ ☐ relation-visitor
   ▶ 🗀 subject-action
                                                                                "text": "首页",
"pagePath": "pages/index/index",
"iconPath": "assets/images/home-1.png",
"selectedIconPath": "assets/images/home-2.png"

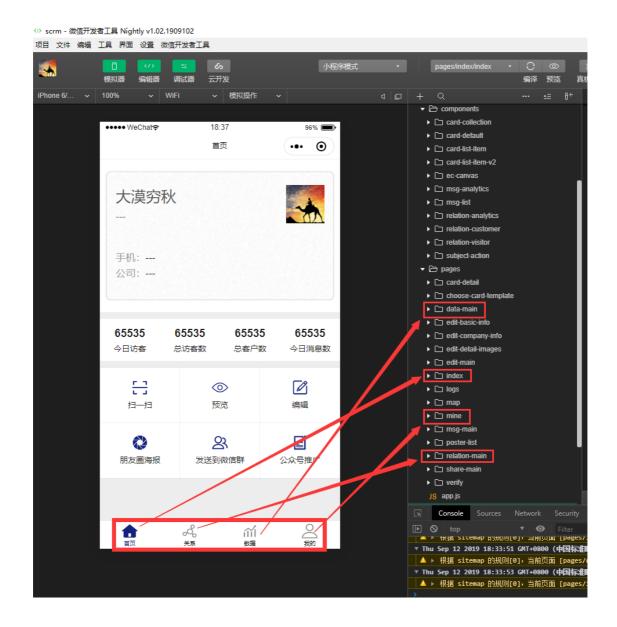
▼ D pages

   ▶ 🗀 card-detail
   ► 🗀 choose-card-template
   ▶ 🗀 data-main
   ▶ □ edit-basic-info
                                                                                "text": "关系",
"pagePath": "pages/relation-main/relation-main",
"iconPath": "assets/images/relation-1.png",
"selectedIconPath": "assets/images/relation-2.png"
   ▶ 🗀 edit-company-info
   ▶ 🗀 edit-detail-images
   ▶ 🗀 edit-main
                                                                           ▶ 🗀 index
   ▶ 🗀 logs
   ▶ [☐ map
   ▶ 🗀 mine
   ▶ 🗀 msg-main
   ▶ 🗀 poster-list
   ► ☐ relation-main
                                                                                "text": "我的",
"pagePath": "pages/mine/mine",
"iconPath": "assets/images/user-1.png",
"selectedIconPath": "assets/images/user-2.png"
   ▶ 🗀 verify
   JS app.js
       s app.wxss
   () sitemap.json
  .gitignore
                                                                      sitemapLocation : sitemap.json ,
"permission": {
  ☐ README.md
  (o) project.config.json
                                                       /miniprogram/app.json 1.8 KB % for-921-workshop
Record Console Sources Network Security AppData Audits Sensor Storage Trace Wxml
```

最终效果如下:



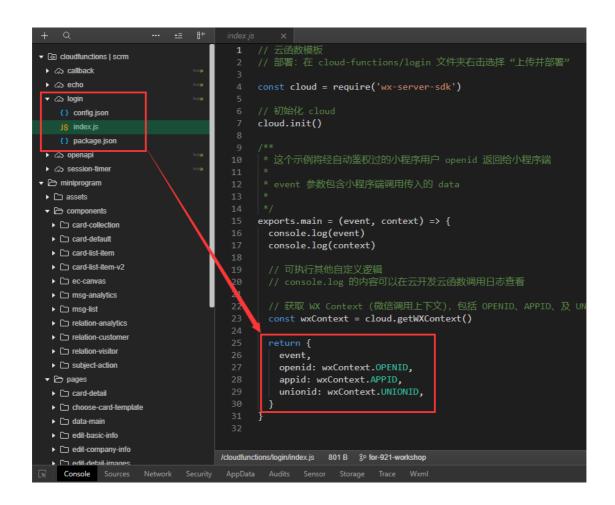
1.5.5 创建每一个主体功能对应的目录和页面



1.5.6 细化并实现每个主要界面的功能

调用 login 云函数

创建一个 login 云函数,用来获取 openid 等资料。



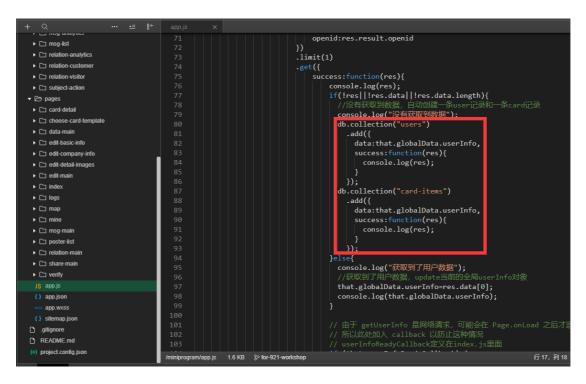
在 app.js 中调用 login 函数,获取用户的 openid,并实现我们自己的业务功能,核心业务流程描述如下:

```
► ☐ relation-analytics
                                                                                                                                                                                                                                                    如果能根据openid在自己的数据库中获取到用户信息,则update当前的全局userInfo对象
否则,自动创建一条user记录和一条card记录
     ▶ [ ¬ relation-custome
     ▶ □ relation-visitor
    ▶ ☐ subject-action
                                                                                                                                                                                                                                         wx.cloud.callFunction({

▼ D pages

                                                                                                                                                                                                                                                       name:'login',
complete:res=>{
    ▶ □ card-detail
                                                                                                                                                                                                                                                                       console.log(res);
if(!res||!res.result||!res.result.openid){
                                                                                                                                                                                                                                                                                 console.log("调用失败");
     ▶ ☐ edit-detail-images
    ▶ □ edit-main
                                                                                                                                                                                                                                                                           that.globalData.userInfo.openid=res.result.openid;
    ▶ 🗀 index
                                                                                                                                                                                                                                                                          db.collection("users")
                                                                                                                                                                                                                                                                                               openid:res.result.openid
    ▶ 🗀 map
                                                                                                                                                                                                                                                                                 })
.limit(1)
     ▶ 🗀 mine
                                                                                                                                                                                                                                                                                                t({
    success:function(res){
        console.log(res);
        if(!res||!res.data.length){
            rescappy by participation of the consoleration of 
     ▶ 🗀 share-main
                                                                                                                                                                                                                                                                                                                            console.log("没有获取到数据");
db.collection("users")
      () app.ison
                                                                                                                                                                                                                                                                                                                                   .add({
data:that.globalData.userInfo,
          app.wxss
     () sitemap.json
                                                                                                                                                                                                                                                                                                                                              success:function(res){
  console.log(res);
 gitignore .
   (o) project.config.json
                                                                                                                                              /miniprogram/app.js 1.6 KB % for-921-workshop
```

写入 users 和 card-items 数据



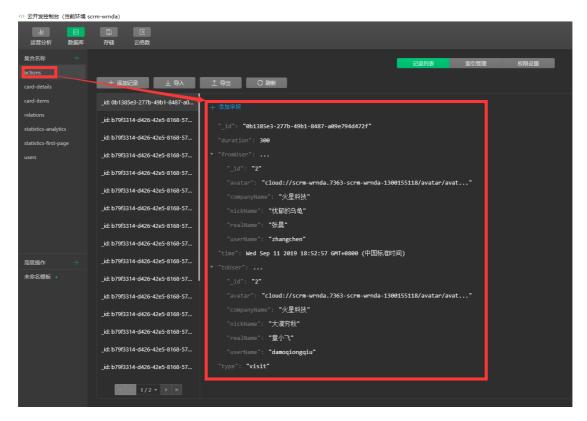
对于首次进入小程序的用户,在获取到微信授权之后,自动给用户创建 2 条记录: users 集合和 card-items 集合。

获取用户动态列表

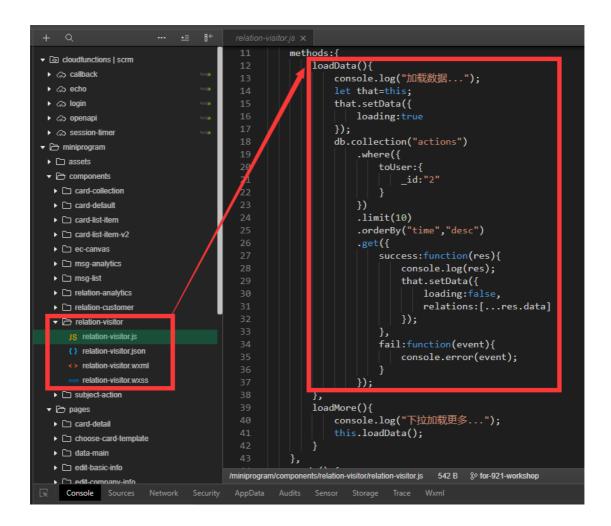
用户动态的界面效果如下:



我们在 actions 集合中记录了这些数据:



我们在前端来加载这些数据:



1.6 完成标准

- 1、完成界面原型所定义的全部功能。
- 2、上线运行。



关注腾讯云「云开发」 **上小程序,用云开发**