27 让用户掏钱：开发实现付费会员支付页面

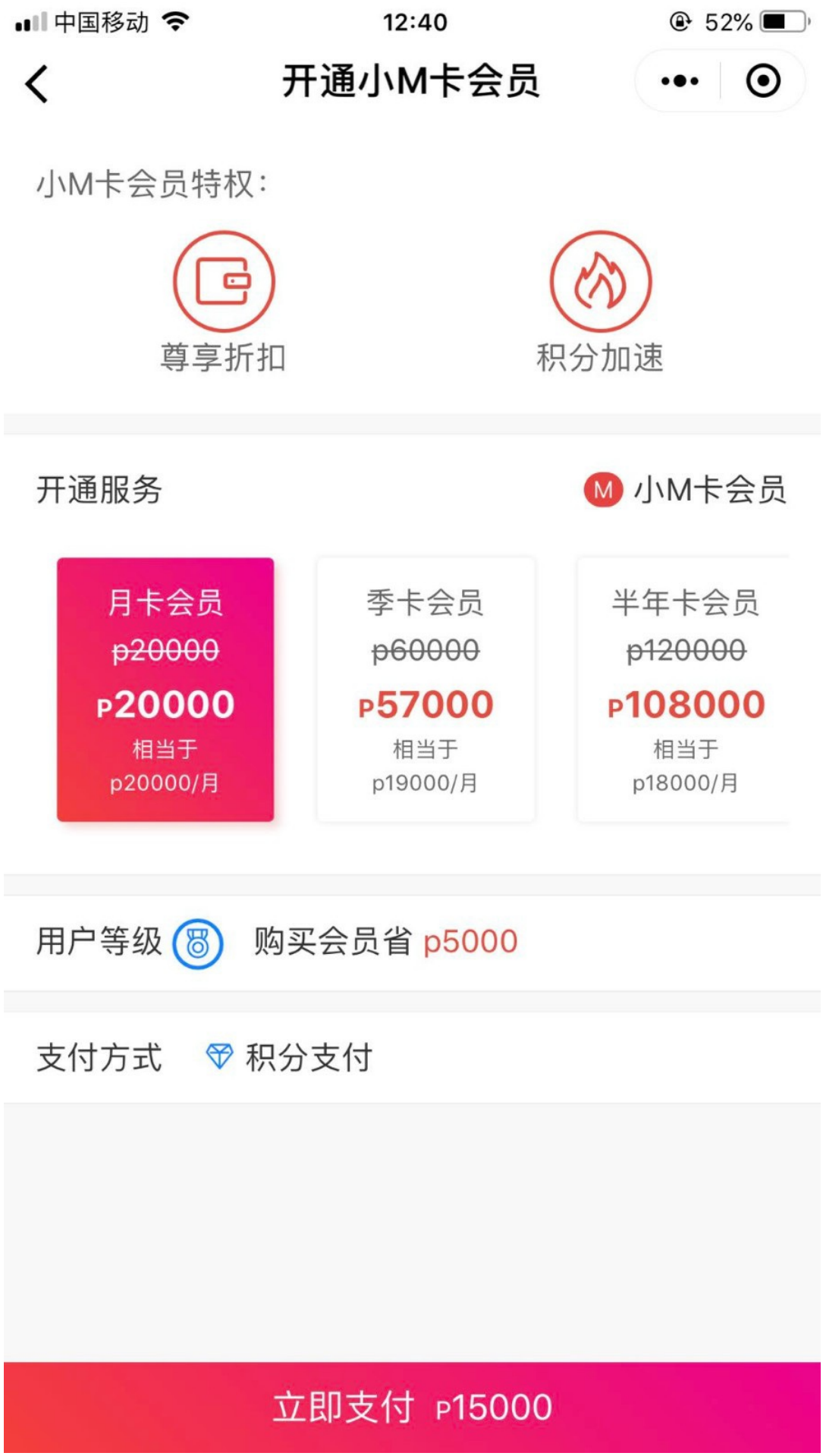
更新时间：2019-08-21 15:29:20

> 散步促进我的思想。我的身体必须不断运动，脑筋才会开动起来。
>
> —— 卢梭

上一节我们已经实现了支付接口，本节将实现支付页面的展示，并调用支付接口完成整个支付流程。

在第二节中，我们已经将付费会员套餐购买页面（如图 8 所示）进行了详细拆解，本节我们就按照“分类拆解法”的编程步骤一步一步完成页面的开发。

图 8 付费会员套餐购买页面



请先按照本章第二节数据库设计的内容，在云数据库中新建付费会员套餐表 `membership_plan` ，并将付费会员套餐的 **Json** 数据导入表中。

1. 数据服务

在付费会员套餐购买页面中，需要获取付费会员套餐表 `membership_plan` 的信息，因此我们首先要建立数据服务 `MembershipService` ，从付费会员套餐表中获取套餐信息。

`MembershipService` 类在 MembershipService.js 中实现，该类公开一个方法 `getMembershipPlans` 用于获取所有付费会员套餐信息：

```
/**
 * 从数据库获取付费会员套餐类型信息
 * @method getMembershipPlans
 * @for MembershipService
 * @param {function} successCallback(payMembershipInfo) 处理数据查询结果的回调函数
 * payMembershipInfo示例数据：
 *  [{
 *    id: 1,
 *    title: "月卡会员",
 *    listPrice: 20000,
 *    price: 20000,
 *    validity: 1
 *  },]
 */
getMembershipPlans(successCallback) {
  //执行数据库查询
  db.collection('membership_plan')
    .get()
    .then(res => {
      //回调函数处理数据查询结果
      typeof successCallback == "function" && successCallback(res.data)
    })
    .catch(err => {
      //访问数据库失败 的 系统异常处理
      //跳转出错页面
      wx.redirectTo({
        url: '../../errorpage/errorpage'
      })
      console.error(err)
    })
}
```

请注意方法注释的规范写法。

MembershipService.js 的完整代码请参考专栏源代码，具体代码位置见本节末尾图 9 。

2. 编程步骤

在本章第二节我们已经完成了图 8 的页面拆解工作，接下来根据拆解结果一步步完成页面的代码编写。

2.1 定义页面子部件及其排列顺序

付费会员套餐购买页面可以拆解为 5 个子部件：

- 子部件 1 ：小M卡会员特权介绍
- 子部件 2 ：付费会员套餐选择
- 子部件 3 ：优惠信息
- 子部件 4 ：支付方式
- 子部件 5 ：支付按钮

此外，在本章第四节我们完成的支付接口会返回支付结果，我们还需要两个子部件用于显示支付结果：

- 子部件 6 ：显示支付成功结果

- 子部件 7 ：显示支付失败结果

支付结果的显示也可以编写成单独页面，在支付接口返回结果后，跳转到新的页面显示支付结果。

在付费会员套餐购买页面实现支付结果显示，需要在显示支付结果子部件时隐藏其它子部件，即：

- 显示子部件 1 - 5 时隐藏子部件 6 与 子部件 7
- 显示子部件 6 时隐藏子部件 1 - 5 与 子部件 7
- 显示子部件 7 时隐藏子部件 1 - 5 与 子部件 6

因此，我们需要两个标志 `showPaySuccess` 和 `showPayFaild` 来控制显示哪些子部件。

从数据库读取套餐信息需要一定时间，还需要一个从数据库获取完数据的标志 `inited` ，当从数据库获取到数据后再显示页面。

```
data: {
  showPaySuccess: false, //是否显示支付成功结果
  showPayFaild: false, //是否显示支付失败结果
  inited: false, //是否已从数据库读取数据
},
```

在 WXML 页面模板中我们需要定义在什么条件下显示哪些子部件：

```
<!-- 当获取到用户数据后，并且不显示支付结果时，才显示子部件 1 - 5 -->
<view wx:if="{{inited && !showPaySuccess && !showPayFaild}}">
 <!-- 子部件 1 ：小M卡会员特权介绍 -->
 <view class="bg-white padding">
 </view>
 <!-- 子部件 2 ：付费会员套餐选择 -->
 <view class="weui-panel">
 </view>
 <!-- 子部件 3 ：优惠信息 -->
 <view class="weui-panel">
 </view>
 <!-- 子部件 4 ：支付方式 -->
 <view class="weui-panel">
 </view>
 <!-- 子部件 5 ：支付按钮 -->
 <view>
 </view>
</view>

<!-- 当支付接口返回支付成功时，显示支付成功结果 -->
<!-- 子部件 6 ：显示支付成功结果 -->
<view wx:if="{{showPaySuccess}}" class="weui-msg bg-white padding">
</view>

<!-- 当支付接口返回支付失败时，显示支付失败结果 -->
<!-- 子部件 7 ：显示支付失败结果 -->
<view wx:if="{{showPayFaild}}" class="weui-msg bg-white padding">
</view>
```

使用 WeUI 的 Panel 组件实现子部件之间由灰色色块区隔的效果，使用 WeUI 的 Msg 组件的 成功提示页与失败提示页可以快速的实现支付成功与支付失败结果的显示。

2.2 实现子部件 1 ：小M卡会员特权介绍

子部件 1 的显示元素包括：

标题

静态界面，无事件，无数据。

付费会员特权名称与图标

包含 4 个静态界面，2 个付费会员特权名称，2 个付费会员特权图标。

无事件，无数据。

付费会员套餐购买页面子部件 1 的实现代码与付费会员首页子部件 1 中显示元素 2 的实现代码基本一致，**具体实现代码请参考本章第三节内容或阅读专栏源代码，源代码位置见本节末尾图 9。**

2.3 实现子部件 2：付费会员套餐选择

子部件 2 的显示元素包括：

顶部标题栏

静态界面，无事件，无数据。

所有套餐的水平列表显示

交互界面，事件为用户屏幕滑动事件与用户点击屏幕事件，数据为所有付费会员套餐信息。

用户屏幕滑动事件的事件响应为：根据用户向左或向右滑动屏幕的方向，显示更短或更长会员有效期的付费会员套餐。

用户点击屏幕事件的事件响应为：将用户点击选中的付费会员套餐变为红底白字，其余付费会员套餐白底黑字，同时修改子部件 5 中用户需支付的价格。

所有付费会员套餐信息在付费会员套餐表中，该表数据需要从云数据库中获取。

在子部件 2 中需要显示所有付费会员套餐信息，因此需要调用数据服务 `MembershipService` 获取套餐信息，并保存到 data 中。

此外，还需要两个数据来存储用户选择的套餐ID `selectedId` 和 用户选择的套餐的价格 `price` ，并设置用户默认选中的套餐为第一个套餐。

以上内容在 JS 逻辑中实现：

```
//引用付费会员套餐的数据库访问类
import MembershipService from '../../../../../dataservice/MembershipService.js'
var membershipService = new MembershipService()

Page({
  /**
   * 页面的初始数据
   */
  data: {
    membershipPlans: [], //会员套餐列表
    selectedId: 0, //用户选中的套餐id
    price: 0, //用户选中套餐的价格
  },

  /**
   * 生命周期函数--监听页面加载
   */
  onLoad: function(options) {
    //获取小M卡会员套餐信息
    this.getMembershipPlans()
  },

  /**
   * 获取并设置小M卡会员套餐信息
   */
  getMembershipPlans: function() {
    var that = this
    //调用数据服务获取小M卡会员套餐信息
    membershipService.getMembershipPlans(
      //获取小M卡会员套餐信息回调函数
      function(membershipPlans) {
        that.setData({
          membershipPlans: membershipPlans,
          //设置用户默认选中的套餐为第一个套餐
          selectedId: membershipPlans[0].id,
          price: membershipPlans[0].price
        })
      })
  },
}
```

顶部标题栏的左右对齐可以使用 WeUI 的 Flex 组件实现，小 M 卡的 图标 M 可以使用 WeUI 的 Badge 组件实现。

在第五章第四节中我们介绍了小程序官方组件 scroll-view ， 该组件支持横向和纵向滚动，所有套餐的水平列表显示同样可以使用该组件实现。

以上内容在 WXML 页面模板中实现：

```
<!-- 子部件 2：付费会员套餐选择 -->
<view class="weui-panel">
  <view class="weui-panel__bd padding">
    <!-- 使用 WeUI 的 Flex 组件实现标题左右对齐 -->
    <view class="weui-flex align-center padding-bottom-sm ">
      <view class="weui-flex__item">
        开通服务
      </view>
      <!-- 使用 WeUI 的 Badge 组件实现 M 图标-->
      <view class="weui-badge weui-badge_space">M</view>
      <view class="weui-badge_after">
        小M卡会员
      </view>
    </view>
    <!-- 使用 scroll-view 实现横向滚动，横向滚动需要设置样式 white-space: nowrap; -->
    <scroll-view scroll-x="true" class="membership_plan_list_container">
      <!-- 列表渲染 -->
      <block wx:for="{{membershipPlans}}" wx:key="{{item.id}}">
        <!-- 绑定用户点击事件 -->
        <!-- 当前选中的套餐红底显示，未选中的套餐白底显示 -->
        <view class="membership_plan_tab_item {{ item.id == selectedId ? 'bg-gradual-red' : 'bg-white' }} radius shadow text-center text-df padding-s
m margin-sm" bindtap="onPlanItemClick" data-item="{{item}}">
          <view>{{item.title}}</view>
          <view class="line_through">p{{item.listPrice}}</view>
          <!-- 未选中的套餐的优惠价格用红字显示 -->
          <view class="text-xl text-bold {{ item.id == selectedId ? '' : 'text-red' }}">
            <text class="text-sm">P</text>{{item.price}}
          </view>
          <view class="text-xs">相当于</view>
          <view class="text-xs">p{{item.price/item.validity}}/月</view>
        </view>
      </block>
    </scroll-view>
  </view>
</view>
```

同时还需要在 JS 逻辑中实现用户点击会员套餐的事件：

```
/**
 * 用户点击会员套餐的事件
 */
onPlanItemClick: function(event) {
  var id = event.currentTarget.dataset.item.id;
  //如果用户点击的会员套餐与当前选中的会员套餐不一致，则切换当前选中的会员套餐
  if (this.data.selectedId != id) {
    this.setData({
      selectedId: id,
      price: this.data.membershipPlans[id - 1].price
    });
  }
},
```

2.4 实现子部件 3：优惠信息

子部件 3 的显示元素：

用户当前等级的购买付费会员套餐优惠特权

交互界面，无事件，数据为用户当前等级的特权信息。

用户当前等级的特权信息需要根据用户当前成长值，从用户等级与等级特权表中计算得出（详见第五章）。

子部件 3 需要调用数据服务获取用户当前等级的特权信息，具体实现的 JS 逻辑与第五章第四节 “2.2.3 实现显示元素 3：用户的等级图标” 中的 JS 逻辑基本一致，**具体实现代码请参考五章第四节内容或阅读专栏源代码，源代码位置见本节末尾图 9。**

在用户当前等级的特权信息中我们需要获取的内容是购买付费会员的折扣金额。

因此，我们需要在 data 中存储该数据：

```
data: {
  discount: 0, //用户会员等级对应的折扣金额
},
```

并在获取到用户的等级信息后，设置 discount 的值：

```
/**
 * 计算用户等级信息
 * "... 略" 的代码见第五章第四节 "2.2.3 实现显示元素 3 ：用户的等级图标"
 */
setLevelData: function(levels, myInfo) {
  // ... 略
  //设置数据用于显示
  // ... 略

  //设置用户当前等级特权享受的购买付费会员折扣金额
  if (myLevel.bonus.length == 3) {
    //只有用户等级大于 1 级时，才拥有购买付费会员折扣金额的特权
    that.setData({
      discount: myLevel.bonus[2].discount
    })
  }
  //在获取完数据后，显示界面
  // ... 略
},
```

在子部件 3 的 WXML 页面模板中，用 ColorUI 的头像元素与图标组件实现用户等级的镂空图标显示效果，用 ColorUI 的颜色样式设置用户等级图标的颜色。

此外，应该在用户的购买付费会员折扣金额大于 0 时才显示子部件 3 的内容。

```
<!-- 子部件 3 ：优惠信息 -->
<view wx:if="{{ discount > 0 }}" class="weui-panel">
  <view class="weui-panel__bd padding-lr padding-tb-sm">
    用户等级
    <view class="cu-avatar sm round bg-white line-{{ myLevel.icon }} solids margin-right-sm">
      <text class="icon-medal"></text>
    </view>
    购买会员省
    <text class="text-red">p{{ discount}}</text>
  </view>
</view>
```

2.5 实现子部件 4 ：支付方式

子部件 4 的显示元素：

积分支付方式

静态界面，无事件，无数据。

子部件 4 只需要在 WXML 页面模板中实现静态界面即可，实现代码类似子部件 3 ，**具体实现代码请阅读专栏源代码，源代码位置见本节末尾图 9。**

2.6 实现子部件 5 ：支付按钮

子部件 5 的显示元素：

支付按钮

交互界面，事件为用户点击屏幕事件，数据为用户购买付费会员套餐需要支付的积分。

用户点击屏幕事件的事件响应为：调用服务端的支付接口，并在页面中显示支付接口返回的支付结果。

用户购买付费会员套餐需要支付的积分 = 用户选中的付费会员套餐价格 - 用户当前等级的购买付费会员套餐优惠。

子部件 5 固定在页面底部，要实现该样式可以使用 form 与 button 组件配合自定义样式 `bottom_pay_button` 实现（自定义样式请阅读专栏源代码），颜色可以使用 ColorUI 的颜色样式设置：

```
<!-- 子部件 5 ： 支付按钮 -->
<form class="bottom_pay_button bg-gradual-red text-center text-df" bindsubmit="onPayButtonClick">
  <button form-type="submit">
    立即支付
    <text class="text-sm margin-left-xs">P</text>{{price - discount}}
  </button>
</form>
```

按钮的点击事件为调用第四节的云函数，并根据云函数的返回结果设置 `showPaySuccess` 和 `showPayFaild` ，来在页面中显示支付结果。

另外，**在支付按钮的点击事件中需要注意三个与用户体验有关的点：**

- **如果用户的当前可用积分不够购买套餐，直接在小程序客户端提示用户无法购买，不需要调用支付接口让用户等待；**
- **支付操作，需要弹出确认支付窗口进行二次确认，可以用小程序官方组件 showModal 实现；**
- **在调用云函数执行支付逻辑时需要提示用户正在支付中，云函数返回支付结果后取消提示，可以用小程序官方组件 showLoading 实现。**

按钮点击事件的具体 JS 逻辑如下：

```
/**
 * 购买会员套餐按钮点击事件
 */
onPayButtonClick: function(event) {
  if (this.data.myInfo.point < this.data.price - this.data.discount) {
    //如果用户的当前可用积分不够购买套餐，提示用户无法购买，不需要调用支付接口
    wx.showModal({
      content: '很抱歉，你的积分不足，无法购买',
      showCancel: false,
    })
  } else {
    var that = this
    //支付操作，需要弹出确认支付窗口进行二次确认
    wx.showModal({
      title: '积分支付',
      content: '你即将支付p' + (this.data.price - this.data.discount).toString(),
      confirmText: "确认",
      cancelText: "取消",
      success: function(res) {
        //用户确认支付
        if (res.confirm) {
          //在调用云函数执行支付逻辑时，需要在页面中提示用户正在支付中
          wx.showLoading({
            title: '支付中',
          })
          //调用支付接口进行支付
          that.pay(event)
        }
      }
```

```
    });
  }
 },

 /**
  * 调用支付接口进行支付
  */
 pay: function(event) {
  var that = this
  //调用云函数执行支付逻辑
  wx.cloud.callFunction({
   name: 'payMembership',
   data: {
    membershipPlanId: that.data.selectedId,
   },
   success: function(res) {
    //根据云函数返回值，决定显示支付成功、支付失败或其他提示
    if (res.result.data == true) {
     that.setData({
      showPaySuccess: true
     })
    } else if (res.result.errMsg == "很抱歉，你的积分不足，无法购买") {
     wx.showModal({
      content: '很抱歉，你的积分不足，无法购买',
      showCancel: false,
     })
    } else {
     that.setData({
      showPayFaild: true
     })
    }
   },
   fail: function(err) {
    //调用云函数失败，显示支付失败提示界面
    that.setData({
     showPayFaild: true
    })
   },
   complete: function() {
    //云函数返回支付结果后取消正在支付中的提示
    wx.hideLoading();
   }
  })
 }
```

2.7 实现子部件 6 与 子部件 7 ：显示支付结果

子部件 6 与 子部件 7 可以基于 WeUI 的 Msg 组件的成功提示页与失败提示页做一些修改实现。

这里给出子部件 6 的 WXML 页面模板代码：

```
<!-- 当支付接口返回支付成功时，显示支付成功结果 -->
<!-- 子部件 6 ： 显示支付成功结果 -->
<view wx:if="{{ showPaySuccess }}" class="weui-msg bg-white padding">
 <view class="weui-msg__icon-area">
  <icon type="success" size="93"></icon>
 </view>
 <view class="weui-msg__text-area">
  <view class="weui-msg__title">支付成功</view>
  <view class="weui-msg__desc">你已开通小M卡会员特权
  </view>
 </view>
 <view class="weui-msg__opr-area">
  <view class="weui-btn-area">
   <navigator open-type='navigateBack' url="">
    <button class="weui-btn" type="primary">返回</button>
   </navigator>
  </view>
 </view>
</view>
```

子部件 7 请各位同学基于 WeUI 的 Msg 组件的失败提示页自行实现，也可以参考专栏源代码。

由于篇幅所限，包含完整样式的 **WXML** 页面模板代码请查阅专栏源代码 **membershipplan.wxml** 与 **membershipplan.wxss**，完整的 **JS** 逻辑代码见 **membershipplan.js**，具体代码位置见本节末尾图 **9** 。

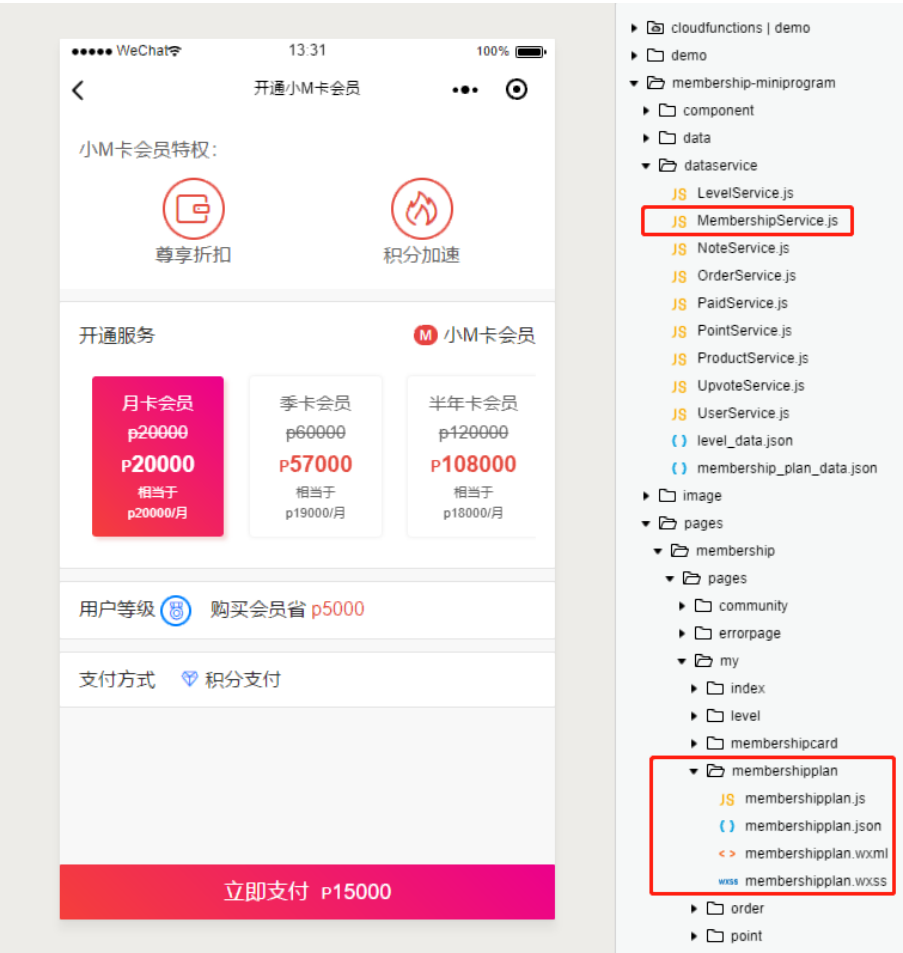
3. 专栏源代码

本专栏源代码已上传到 GitHub，请访问以下地址获取：

https://github.com/liujiec/Membership-ECommerce-Miniprogram。

本节源代码内容在图 9 红框标出的位置。

图 9 本节源代码位置



下节预告

从下一节开始，我们将开始搭建商城模块，实现商品展示、搜索、购物车、购买等功能。

实践环节

实践是通往大神之路的唯一捷径。

本节实操内容：

- 编写代码完成图 **8** 所示的页面，如碰到问题，请阅读专栏源代码学习如何实现。

}

26 准备好收银机：开发实现付费会员支付接口

28 商城业务设计