20 开发实现同步微信运动步数获得积分

更新时间：2019-08-13 17:02:10

> 人生的价值，并不是用时间，而是用深度去衡量的。
>
> ——列夫·托尔斯泰

在第五章第三节，我们已经开发实现了同步微信运动步数获得成长值的功能，同步微信运动步数获得积分与之类似。

本节将简单介绍微信运动步数获得积分的实现，重点是与各位同学一起实现完整的微信运动同步页面，如图 2 所示。

图 2 同步微信运动数据页面


中国移动
12:14
52%
同步微信运动数据
微信运动步数同步
请点击同步按钮开始同步微信运动数据
同步
同步成功
共同步积分
+117504
详情
积分
+85
运动日期:
2019/06/18
运动步数:
85
历史同步已增加积分:
0
本次同步增加积分:
85
积分
+2331
运动日期:
2019/06/17
运动步数:
2331
历史同步已增加积分:
0

1. 从微信运动获得积分

从微信运动获得积分的同步算法与获得成长值的同步算法基本一致，详细算法请回顾第五章第三节，将算法中的成长值改为积分即可。

为了实现图 2 所示页面，从微信运动获得积分需要增加一个算法步骤，记录获得积分的详细信息，并作为云函数返回值返回给页面。

记录获得积分的规则为：

- 如果用户某一天的微信运动步数从未同步为积分，则记录当天获得积分为微信运动步数；
- 如果用户某一天的微信运动步数已经同步为积分，需要判断微信运动步数是否更新。如果当天已同步的积分少于新获取的微信运动步数，则记录当天获得积分为新获取的微信运动步数 - 已同步的积分。

从微信运动获得积分的实现也与获得成长值类似，在云函数增加一个函数 `syncPoint` 来实现同步算法，云函数中需要增加一个数据 `weRunToPointData` 来记录本次同步获得积分的详细信息：

```
/**
 * 同步微信运动数据并更新到积分数据库
 * @param {array} weRunData 从小程序API获取到的微信运动数据
 * @return {array} 积分变动数据 {time,step,changePoints}
 */
async function syncPoint(weRunData) {
  const wxContext = cloud.getWXContext()
  //返回值，同步获得积分的详细信息
  var weRunToPointData = []
  //定义同步获得积分的详细信息内容
  for (var i in weRunData) {
    weRunToPointData.push({
      time: weRunData[i].timestamp, //微信运动时间
      step: weRunData[i].step, //微信运动步数
      changePoints: 0 //本次同步获得了多少积分
    })
  }

  //----------------算法步骤 2 对每一天的微信运动数据依次执行下列步骤----------begin
  for (var i in weRunData) {
    //略，参见第五章第三节源代码或专栏源代码
    if (queryResult.data.length <= 0) {
      //略，参见第五章第三节源代码或专栏源代码

      //如果用户某一天的微信运动步数从未同步为积分，则记录当天获得积分为微信运动步数
      weRunToPointData[i].changePoints = data.step
    } else {
      if (queryResult.data[0].changeGrowthValue < data.step) {
        //略，参见第五章第三节源代码或专栏源代码

        //如果用户某一天的微信运动步数已经同步为积分，并且当天已同步的积分少于新获取的微信运动步数，
        //则记录当天获得积分为新获取的微信运动步数 - 已同步的积分
        weRunToPointData[i].changePoints = data.step - queryResult.data[0].changePoints
      }
    }
  }
  //----------------算法步骤 2 对每一天的微信运动数据依次执行下列步骤----------end

  //算法步骤 3 更新用户当前可用积分
  //略，参见第五章第三节源代码或专栏源代码

  //算法步骤 4 调用风控规则校验
  //略，参见第五章第三节源代码或专栏源代码

  //函数返回同步获得积分的详细信息
  return weRunToPointData
}
```

然后，在云函数入口函数中调用同步方法并返回获得积分结果给页面显示：

```
exports.main = async (event, context) => {
  var weRunData = event.weRunData
  //调用从微信运动获得成长值的同步方法
  await syncGrowthValue(weRunData.data.stepInfoList)
  //调用从微信运动获得积分的同步方法，并返回获得积分结果给页面显示
  return await syncPoint(weRunData.data.stepInfoList)
}
```

完整的微信运动同步云函数 `syncWeRunToPoint` 请阅读本专栏源代码，源代码位置在图 3 中已用红框标出。

2. 同步微信运动数据页面

同步微信运动数据页面的实现按照“分类拆解法”的步骤一步一步完成。

2.1 拆解步骤

拆解步骤请回顾第二章第三节对“分类拆解法”的详细讲解内容。

同步微信运动数据页面包括 3 个子部件，从上到下分别是：

子部件 1 ：同步按钮区域

子部件 2 ：同步结果概要显示区域

子部件 3 ：同步结果详情显示区域

各子部件的详细拆解过程请各位同学自己动手实践，按照拆解步骤拆解图 2 的同步微信运动数据页面，在拆解完成后再继续阅读本节后续内容。这是本节内容的实践环节。

2.2 编程步骤

2.2.1 定义页面子部件及其排列顺序

首先在 WXML 页面模板中定义 3 个子部件的容器，并把它们按顺序排列：

```
<!-- 页面第一部分 同步按钮区域 -->
<view class="weui-panel">
</view>

<!-- 页面第二部分 同步结果概要显示区域 -->
<view class="weui-panel">
</view>

<!-- 页面第三部分 同步结果详情显示区域 -->
<view>
  <block>
     <!-- 同步结果列表渲染 -->
  </block>
</view>
```

子部件 1 和 子部件 2 使用 WeUI 的 Panel 实现，子部件 3 使用 `block` 进行列表渲染，显示列表数据。

2.2.2 实现子部件 1 ：同步按钮区域

拆解子部件 1 的结果如下：

显示元素 1 ：标题 微信运动步数同步

静态界面，无事件，无数据

显示元素 2 ：描述 请点击同步按钮开始同步微信运动数据

静态界面，无事件，无数据

显示元素 3 ：同步按钮

单条内容交互界面，事件为用户点击按钮事件，无数据

用户点击按钮事件的事件响应为：获取用户的微信运动步数，然后请求云函数 `syncWeRunToPoint` 进行同步，然后将云函数返回的同步结果设置到数据 `weRunToPointData` 中供子部件 3 ：同步结果详情显示区域显示，还需要计算本次同步用户获得的总积分数据 `addPointNum`，供子部件 2 ：同步结果概要显示区域显示。

考虑到云函数的执行时间可能较长，在请求云函数时在页面中显示同步中的提示话语会有更好的用户体验。

2.2.2.1 定义显示元素的排列顺序

显示元素 1 与显示元素 2 的样式可以用 WeUI 的 Panel 组件中的 `weui-media-box` 实现：

```
<!-- 页面第一部分 同步按钮区域 -->
<view class="weui-panel">
 <view class="weui-panel__bd">
  <view class="weui-media-box weui-media-box_text">
   <!-- 显示元素 1 标题-->
   <view class="weui-media-box__title weui-media-box__title_in-text">
    微信运动步数同步
   </view>
   <!-- 显示元素 2 描述-->
   <view class="weui-media-box__desc">
    请点击同步按钮开始同步微信运动数据
   </view>
  </view>
  <!-- 显示元素 3 -->
  <view>
  </view>
 </view>
</view>
```

2.2.2.2 实现显示元素 3 ：同步按钮

同步按钮点击后需要两个数据供子部件 2 和子部件 3 显示内容。

另外，访问云函数可能发生异常导致同步失败，因此还需要一个数据 `isFailed` 来标志云函数调用成功或失败，根据该标志在子部件中显示同步成功或同步失败的结果。

```
data: {
 weRunToPointData: [], //同步微信运动步数增加的积分详情
 addPointNum: 0, //同步微信运动步数增加的总积分
 isFailed: false //是否同步失败
},
```

接下来在 WXML 页面模板中定义同步按钮，并绑定按钮点击事件函数 `onSyncButtonClick` ，显示元素 3 的页面样式可使用 WeUI 的 `weui-cell` 实现：

```
<view class="weui-cell weui-cell_hide_line">
  <view class="weui-cell__bd">
    <button type="primary" bindtap='onSyncButtonClick'>同步</button>
  </view>
</view>
```

最后实现按钮点击事件函数 `onSyncButtonClick` ，需要实现的程序逻辑包括：

- 第一步：在页面中显示同步中的提示话语；

- 第二步：获取用户的微信运动步数；
- 第三步：请求云函数 `syncWeRunToPoint` 进行同步；
- 第四步：计算本次同步用户获得的总积分数据 `addPointNum`；
- 第五步：将云函数返回的同步结果和总积分设置到 `data` 中；
- 第六步：访问云函数发生异常，设置云函数调用失败标志 `isFailed` ；
- 第七步：在调用云函数成功或失败后，隐藏同步中的提示话语。

在第四章第三节已经详细讲解了第二步和第三步的代码实现，同步按钮事件只需要调用即可：

```
/**
 * 同步按钮点击事件
 */
onSyncButtonClick: function() {
  //第一步：在页面中显示同步中的提示话语
  wx.showLoading({
    title: '同步中',
  })
  //第二步、第三步直接调用 第四章第三节的函数 用户授权读取微信运动数据
  this.authorizeWeRun()
},
```

然后我们需要改写第四章第三节 `getStepInfoList` 函数的 `success` 方法实现第四步和第五步，同时在 `fail` 方法中实现第六步，在 `complete` 方法中实现第七步：

```
//引用时间格式化工具
const util = require('../../../../../utils/util.js')

/**
 * 调用云函数，获取解密后的微信运动步数
 * 云调用直接获取开放数据，详见https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/signature.html
 * 接口如果涉及敏感数据（如wx.getWeRunData），接口的明文内容将不包含这些敏感数据，
 * 而是在返回的接口中包含对应敏感数据的 cloudID 字段，数据可以通过云函数获取。
 */
getStepInfoList: function(weRunEncryptedData) {
  var that = this
  //云调用直接获取开放数据
  wx.cloud.callFunction({
    name: '云函数名称', //请修改为你建立的云函数名称
    data: {
      //通过 wx.cloud.CloudID 构造 CloudID
      //调用云函数时，这些字段的值会被替换为 cloudID 对应的微信运动数据
      weRunData: wx.cloud.CloudID(weRunEncryptedData.cloudID)
    },
    success: function(res) {
      var weRunToPointData = res.result
      //第四步：计算本次同步用户获得的总积分数据
      var addPointNum = 0
      for (var i in weRunToPointData) {
        addPointNum += weRunToPointData[i].changePoints
        //格式化时间
        weRunToPointData[i].time = util.formatDate(new Date(weRunToPointData[i].time * 1000))
      }
      //第五步：设置云函数返回的同步结果和总积分供页面显示
      that.setData({
        addPointNum: addPointNum,
        weRunToPointData: weRunToPointData.reverse()
      })
    },
    fail: function(err) {
      //第六步：访问云函数发生异常，设置云函数调用失败标志
      that.setData({
        isFailed: true
      })
    },
    complete: function() {
      wx.hideLoading(); //第七步：在调用云函数成功或失败后，隐藏同步中的提示话语
    }
  })
},
```

2.2.3 实现子部件 2：同步结果概要显示区域

拆解子部件 2 结果如下：

显示元素 1：同步成功提示文字

单条内容交互界面，无事件，无数据

在成功调用云函数获得返回值时显示

显示元素 2：本次同步用户获得的总积分

单条内容交互界面，无事件，数据为本次同步用户获得的总积分数据 `addPointNum`

在成功调用云函数获得返回值时显示

显示元素 3：同步失败提示文字

单条内容交互界面，无事件，无数据

在调用云函数出现异常时显示

子部件 2 的显示元素是否显示，由云函数调用失败标志 `isFailed` 决定，通过 WXML 的条件渲染语法 `wx:if` 实现，同步成功提示使用 WeUI 的 Preview 组件实现样式，同步失败提示可以使用 WeUI 的 Loadmore 组件的样式，也可以直接写在 WXML 的 `text` 标签中：

```
<!-- 页面第二部分 同步结果概要显示区域 -->
<!-- 同步成功提示 -->
<view class="weui-panel" wx:if="{{!isFailed}}">
 <view class="weui-panel__hd">
  同步成功
 </view>
 <view class="weui-form-preview__hd">
  <view class="weui-form-preview__item">
   <view class="weui-form-preview__label">共同步积分</view>
   <view class="weui-form-preview__value_in-hd line-red">+{{addPointNum}}</view>
  </view>
 </view>
</view>
<!-- 同步失败提示 -->
<view wx:if="{{isFailed}}">
 <view class="weui-loadmore weui-loadmore_line weui-loadmore_dot">
  <view class="weui-loadmore__tips weui-loadmore__tips_in-line"></view>
  <view>微信运动数据同步失败，请稍后重试</view>
 </view>
</view>
```

2.2.4 实现子部件 3：同步结果详情显示区域

拆解子部件 3 的结果如下：

显示元素 1：标题 详情

静态界面，无事件，无数据

显示元素 2：同步积分详情列表

多条内容交互界面，无事件，数据为云函数同步结果 `weRunToPointData`

在同步结果没有数据时（用户授权读取自己的微信运动数据，但该用户并未在微信中开启微信运动功能，调用 微信运动 API 获取到的微信运动记录数为 0 ）显示元素 2 中不会有详情记录显示，可以考虑该情况下隐藏子部件 3 的显示。

同步积分详情列表数据 `weRunToPointData` 的显示使用 WXML 的列表渲染语法 `wx:for` ，样式使用 WeUI 的 Preview 组件。

在列表渲染时，如果某条记录没有增加积分（changePoints = 0），可以考虑隐藏这条记录积分字段的显示，只显示微信运动数据的内容。

```
<!-- 页面第三部分 同步详情列表 -->
<!-- 同步结果没有数据时隐藏子部件 3 的显示 -->
<view wx:if="{{ weRunToPointData.length > 0 }}">
  <view class="weui-cells__title">详情</view>
  <!-- 列表渲染 列表显示同步详情记录 -->
  <block wx:for="{{weRunToPointData}}" wx:key="item.time">
    <view class="weui-form-preview">
   <!-- 如果某条记录没有增加积分，隐藏积分字段的显示 -->
      <view class="weui-form-preview__hd" wx:if="{{item.changePoints > 0}}">
        <view class="weui-form-preview__item">
          <view class="weui-form-preview__label">积分</view>
          <view class="weui-form-preview__value_in-hd line-red">+{{item.changePoints}}</view>
        </view>
      </view>
      <!-- 其他字段的显示代码请查阅专栏源代码 -->
    </view>
  </block>
</view>
```

由于篇幅所限，完整的 **WXML**页面模板代码请查阅专栏源代码 **walkingstepsync.wxml** 与 **walkingstepsync.wxss**，完整的 **JS** 逻辑代码见 **walkingstepsync.js**，具体代码位置见本节末尾图 **3** 。

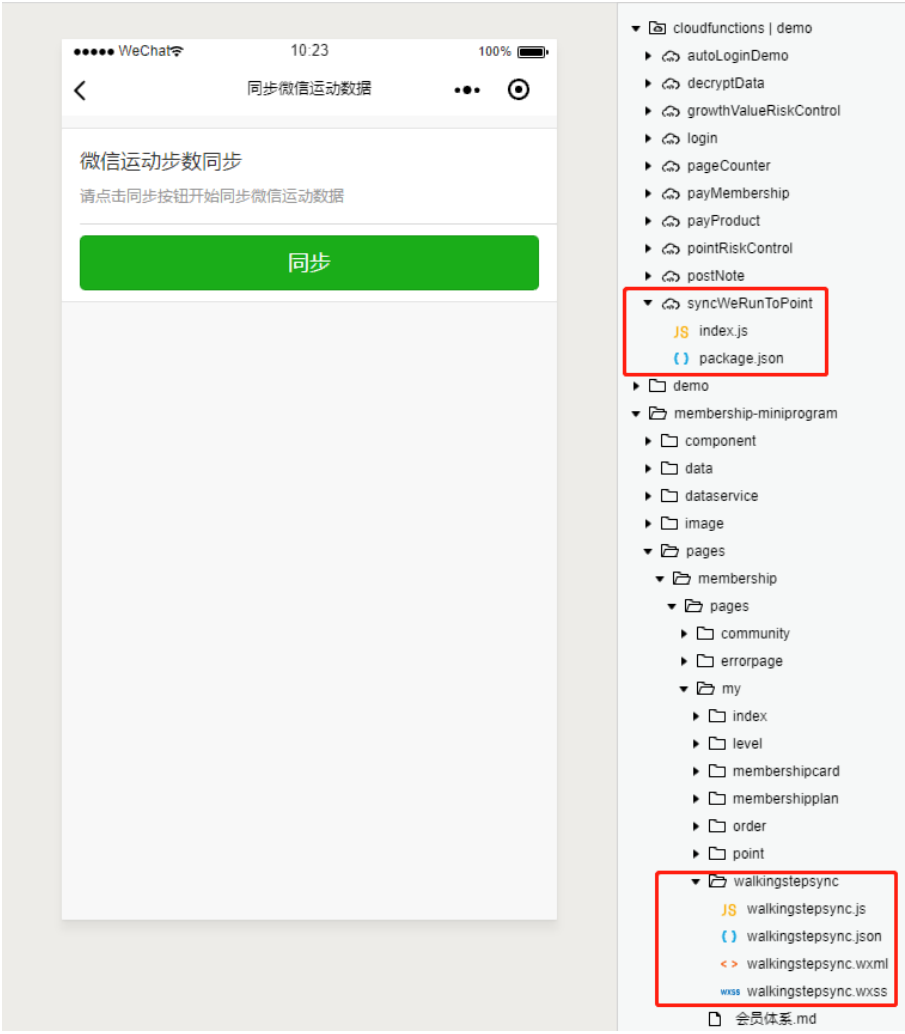
3. 专栏源代码

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 3 红框标出的位置。

图 3 本节源代码位置



下节预告

下一节，我们将实现积分体系页面，用户可在该页面中查看自己的积分变动记录。

实践环节

实践是通往大神之路的唯一捷径。

本节实操内容：

- 编写代码完成同步微信运动数据获得积分和成长值的完整云函数，如碰到问题，请阅读本专栏源代码学习如何实现。
- 请结合第二章第三节对“分类拆解法”的详细讲解内容，拆解图 **2** 的同步微信运动数据页面，然后将自己的拆解结果与本节 **2.2** 列出的拆解结果进行对照总结。
- 编写代码完成图 **2** 所示的页面，如碰到问题，请阅读本专栏源代码学习如何实现。

}

19 功能与数据库设计

21 开发实现积分体系页面