

15 开发实现同步微信运动步数获得成长值

更新时间：2019-07-24 13:30:00



“

如果说我比别人看得要远一点，那是因为我站在巨人的肩上。

——牛顿

”

在第四章第三节，我们已经完成了获取用户的微信运动步数的功能。本节将在此基础上，与各位同学一起开发实现同步微信运动步数获得成长值功能。

实现该功能的第一步是整理出从微信运动步数获得用户成长值的同步算法。

1. 从微信运动获得成长值的同步算法

微信运动 API 返回的结果是用户最近 30 天的运动数据。因此，根据用户最近 30 天的微信运动数据更新用户成长值时，有以下特殊场景需要考虑：

当我在早晨获取微信运动步数时，我当天的运动步数是 300 步；中午吃完午饭，我在外面散步了一会，到下午我再获取微信运动步数，我当天的运动步数变成了 2200 步。

第一次我同步微信运动步数时，我的成长值应该增加 300，第二次同步时，我的成长值应该增加 1900 而不是 2200，因为在早晨同步时我已经获得了 300 成长值，中午同步时只增加了 1900 步。

有两种处理方式处理这一情况，一种是将原记录中的获取成长值增加 1900，由 300 变为 2200；另一种是新增一条记录，记录用户获得成长值 1900，在这里我们采用前一种方式实现同步获取用户成长值。

由上述分析，我们可以得到同步微信运动步数到用户成长值有以下基本算法规则：

- 如果用户某一天的微信运动步数从未同步为成长值，则将当天的微信运动步数全部变为成长值；
- 如果用户某一天的微信运动步数已经同步为成长值，需要判断微信运动步数是否更新。如果当天已同步的成长值少于新获取的微信运动步数，需要将成长值更新为最新的微信运动步数；

- 1 微信运动步数 = 1 用户成长值（在第一节业务规则中已定义）。

此外，根据前一节的数据库设计，用户成长值需要记录到上一节我们设计的成长值获取记录表 `user_growth_value` 中。同时，还需要在用户表 `user` 中更新用户当前总成长值。

最后，根据前一节的风控规则设计，我们需要调用风控规则校验用户某一天的微信运动步数是否异常，并锁定异常用户。

完整的同步算法步骤如下：

1. 从微信运动 API 获得用户最近 30 天的运动数据 `weRunData`；
2. 对每一天的微信运动数据依次执行下列步骤：
 - 查询数据库 `user_growth_value` 中当天的微信运动数据是否已记录为成长值，使用运动数据中的 `timestamp` 字段作为每一天的日期标志；
 - 如果数据库不存在记录，则向数据库插入一条新的成长值获取记录，新增成长值 `changeGrowthValue` 即运动数据中的步数 `step`；
 - 如果数据库存在记录，则判断运动数据中的步数 `step` 是否大于已记录的成长值 `changeGrowthValue`。如果步数更大，需要将该记录的成长值 `changeGrowthValue` 更新为新的步数 `step`。
3. 使用 `user_growth_value` 的最新数据，累加每条记录中的成长值，得到最新的用户当前总成长值，并更新到数据库 `user` 中该用户记录的 `growthValue` 字段中；
4. 调用风控规则校验数据库 `user_growth_value` 中该用户的成长值记录是否有异常。

2. 从微信运动获得成长值的代码实现

如还未在云数据库中新建成长值获取记录表 `user_growth_value` 的同学，请先在云数据中新建该表。

从微信运动获得成长值需要在云函数中实现，在第三章第三节我们已经介绍过云函数可以有效防止恶意篡改数据的黑客攻击行为。

如果将实现代码写在小程序客户端的 JS 逻辑中，有技术能力的人完全可以跳过调用微信运动 API 获取微信运动步数的代码段，直接构造假的微信运动步数数据，然后调用同步算法步骤的第 2 步，将假数据直接提交到数据库中。

在云函数中访问云开发数据库的语法与在客户端的 JS 逻辑中不完全一致，在微信官方的“小程序开发文档”有详细的讲解，云函数访问云开发数据库的教程位置如下：

- 入口网址：[云函数访问云开发数据库](#)，如果微信官方文档改版导致链接失效，请按图索骥。
- 入口位置：在“小程序开发文档”的“云开发”栏目，如图 2 红框标出部分。

云函数与 JS 逻辑访问数据的主要区别是，云函数只能使用 `async`、`await` 的 `promise` 形式调用数据库。对 `async`、`await` 和 `promise` 不理解的同学可以参考这篇文章：[async/await 和 promise](#)，如果看不懂也没有关系，先看官方文档中的示例代码，以及本专栏的源代码，记住 `async`、`await` 的使用语法即可，待有更多编程实践经验后再回过头来研究。

图 2 云函数访问云开发数据库文档位置



实现获取成长值的第一步是从微信运动 API 获得用户最近 30 天的运动数据 `weRunData`，实现方式在第四章“第三节 获取微信运动步数”中已进行了详细的讲解，具体实现方式请回顾该内容。

然后，我们需要在第四章第三节建立的云函数的 `index.js` 中增加一个函数 `syncGrowthValue` 来实现算法步骤的第 2 步、第 3 步和第 4 步：

```
/**
 * 同步微信运动数据并更新到成长值数据库
 * @param {array} weRunData 从微信运动 API 获得的用户最近 30 天的运动数据
 */
async function syncGrowthValue(weRunData) {
  const wxContext = cloud.getWXContext()

  //-----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) {
    var data = weRunData[i]
    //查询数据库是否已存在该条微信运动记录
    queryResult = await db.collection('user_growth_value')
      .where({
        timestamp: data.timestamp,
        //云函数是在服务端操作，对所有用户的数据都有操作权限
        //在云函数中查询用户数据，需要添加openid的查询条件
        _openid: wxContext.OPENID
      })
      .get()
    if (queryResult.data.length <= 0) {
      //如果不存在记录，则向数据库插入微信运动记录
      await db.collection('user_growth_value')
        .add({
          data: {
            _openid: wxContext.OPENID, //云函数添加数据不会自动插入openid，需要手动定义
            date: db.serverDate(),
            changeGrowthValue: data.step,
            operation: "微信运动",
            timestamp: data.timestamp,
            orderId: '',
            noteId: ''
          }
        })
    } else {
      if (queryResult.data[0].changeGrowthValue < data.step) {
        //如果存在记录，但数据库步数少于小程序API返回步数，则向数据库更新微信运动记录
        await db.collection('user_growth_value').doc(queryResult.data[0]._id)
          .update({
            data: {
              date: db.serverDate(),
              changeGrowthValue: data.step
            }
          })
      }
    }
  }
  //-----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 更新用户当前总成长值-----begin
```

```

//第一步 获取所有成长值记录
// 先取出集合记录总数
const countResult = await db.collection('user_growth_value').count()
const total = countResult.total
// 计算需分几次取
const batchTimes = Math.ceil(total / MAX_LIMIT)
// 承载所有读操作的 promise 的数组
const tasks = []
for (let i = 0; i < batchTimes; i++) {
  const promise = db.collection('user_growth_value').where({
    _openid: wxContext.OPENID
  }).skip(i * MAX_LIMIT).limit(MAX_LIMIT).get()
  tasks.push(promise)
}
// 等待所有
var allGrowthValueRecords = (await Promise.all(tasks)).reduce((acc, cur) => ({
  data: acc.data.concat(cur.data),
  errMsg: acc.errMsg,
}))

//第二步 计算总成长值，并更新到user表
var totalGrowthValueNum = 0
allGrowthValueRecords.data.forEach(function(item) {
  totalGrowthValueNum += item.changeGrowthValue;
});
await db.collection('user')
  .where({
    _openid: wxContext.OPENID
  })
  .update({
    data: {
      growthValue: totalGrowthValueNum
    }
  })
}

//-----算法步骤 3 更新用户当前总成长值-----end

//算法步骤 4 调用风控规则校验
await cloud.callFunction({
  name: 'growthValueRiskControl',
  data: {
    openid: wxContext.OPENID
  }
})
}

```

风控规则云函数 `growthValueRiskControl` 的具体实现，将在本章第五节中详细讲解。

实现上述代码需要注意以下几点：

- 在小程序客户端的 JS 逻辑中进行查询，默认只能查询到当前用户有访问权限的数据，因此所有的数据操作都不需要考虑 `_openid` 字段。而云函数是在服务端操作，对所有用户的数据都有操作权限，因此在云函数中查询用户数据，需要添加 `_openid` 的查询条件，在向数据库中插入记录时，也需要设置 `_openid` 字段的值。
- **云开发数据库对每次查询返回的记录数量有限制，小程序客户端最多 20 条，云函数端最多 100 条。**在同步算法步骤 3 中需要计算用户所有成长值记录中的成长值之和，记录数可能超过 **100** 条，需要分多次读取（具体实现方式见代码中“第一步 获取所有成长值记录”的代码与注释）

此外，我们还需要在第四章第三节建立的云函数入口函数中调用同步方法：

```

exports.main = async (event, context) => {
  var weRunData = event.weRunData
  await syncGrowthValue(weRunData.data.stepInfoList) //调用从微信运动获得成长值的同步方法
  return weRunData.data.stepInfoList
}

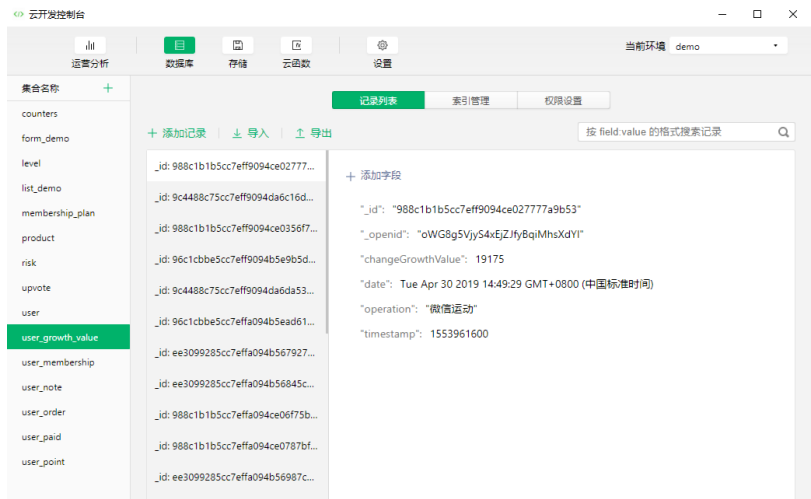
```

接下来我们还需要编写一个页面，在页面中提供一个同步按钮，让用户同步他的微信运动步数，并将同步结果显示出来供用户查看。

在你自己的小程序作品中，你也可以让系统自动定时同步每个小程序用户的微信运动步数，同时提供一个手工同步按钮，以便系统自动同步失败时，用户可以手动同步。

我们可以直接使用四章第三节建立好的页面，点击同步按钮，即可在数据库中查看到成长值获取记录表 `user_growth_value` 已经根据我们自己的微信运动步数，生成了成长值记录，如图 3 所示。

图 3 成长值获取记录表数据库记录



同时，我们还将看到在用户表 `user` 中看到自己的用户当前总成长值 `growthValue` 的数值已经更新。（请先阅读并实现本章第五节中“用户首次打开小程序自动注册”的功能，否则在 `user` 表中没有自己的用户记录，看不到总成长值信息）

过一段时间（例如2小时，在此期间有带着手机移动）后，我们再次点击同步按钮，并检查 `user` 表 `growthValue` 中增加的数值是否正好等于本次同步时今天的微信运动步数 - 上次同步时今天的微信运动步数，即可确定代码逻辑是否与算法思路一致。

完整的微信运动同步页面实现将在第六章第三节详细介绍。

下节预告

下一节，我们将实现用户成长体系页面，该页面根据从本节获取到的用户成长值，向用户显示等级、等级特权，以及用户体系介绍等内容。

实践环节

实践是通往大神之路的唯一捷径。

本节实操内容：

- 请结合本节与第四章第三节的内容，修改你在第四章第三节实践环节编写的源代码，实现微信运动获得成长值，并记录到成长值获取记录表 `user_growth_value` 中（如图 3 所示，在云数据库中能看到获得成长值的记录）。

