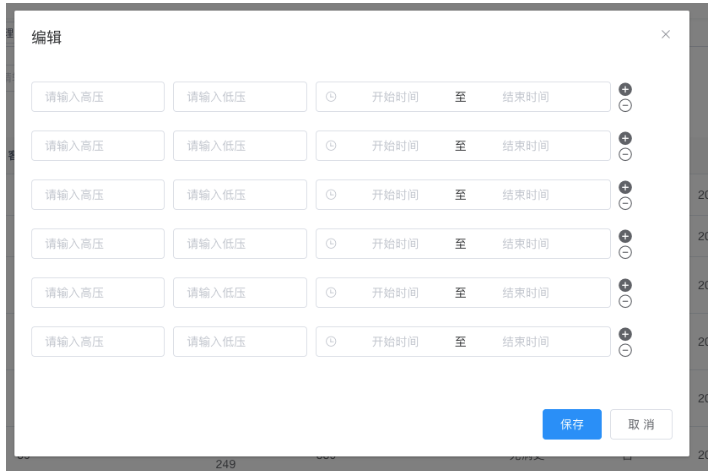
# 需求地址

http://192.168.9.20/ld\_gongyinglian/1/%e7%be%8a%e7%be%8a%e5%bd%93%e5%ae%b6v2.0.6/#g=1&p=%E6%88%91%E7%9A%84-%E7%B3%BB%E7%BB%9F%E8%AE%BE%E7%BD%AE-%E8%AE%BE%E5%A4%87%E8%AE%BE%E7%BD%AE-%E6%B5%8B%E9%87%8F%E5%91%A8%E6%9C%9F

# 血压建模

菜单：羊羊当家管理-血压建模





说明：

1、点击编辑，弹出弹窗，支持添加，删除血压值。

2、最多设置24条。

3、用户解绑设备，数据清除。

4、血压数据传到厂商，厂商血压校准后，将血压高低值数据返回到手

表，再传到APP。

## 需要开发的接口

### 查询进入血压建模列表

#### 接口地址

memberDevice/memberDeviceList

#### 请求方法

**POST**

#### 请求参数

{

    "pageNum": 1,

    "pageSize": 10,

    "imei": "",

    "nikeName": ""

}

#### 返回参数

#### 开发思路

1、MemberDeviceController类中的memberDeviceList方法（auth模块）

2、MemberDevicesController类中的getList方法（device模块）

#### 涉及的表

yy\_member\_devices

yy\_member

#### sql语句

SELECT  
DISTINCT  
ymd.yy\_member\_id,  
ym.nike\_name,  
ym.imei,  
ymd.yy\_devices\_id,  
ymd.create\_time,  
ymd.is\_illness,  
ymd.is\_flag  
from yy\_member\_devices ymd  
left join yy\_member ym on ym.id = ymd.yy\_member\_id

### 点击编辑查询血压建模

#### 接口地址

memberDevice/queryBloodPressureModel/{imei}

#### 请求方法

GET

#### 请求参数

imei

#### 返回参数

#### 开发思路

需要启动的服务：

MemberDeviceController类（auth模块）queryBloodPressureModel方法

#### 涉及的表

#### sql语句

执行的sql语句：

select start\_time as startTime,end\_time as endTime,sdp,dbp from re\_data a where a.imei = #{imei}

### 添加血压值

#### 涉及的模块

yydj-auth

yydj-device

#### 接口地址

memberDevice/addMemberDeviceList

#### 开发思路

1、MemberDeviceController类中的addMemberDeviceList（auth模块）

IReDataService接口

需要调用校准血压这个三方接口

patchBloodPressure

cpid、key、imei要匹配

cpid：1889

key：a62943d910e2f79f0021c7842982720b

imei：必须是真实的手表的imei

#### 涉及的表

yy\_blood\_model

#### sql语句

insert into re\_data(create\_time, created,update\_time,updated,dbp,sdp,start\_time,end\_time) values

();

### 定时任务（10s执行一次）

#### 开发思路

1、BloodPressureScheduledTask类中的

#### cron表达式

0/10 \* \* \* \* ?

#### 涉及的表

yy\_blood\_model

#### 涉及的sql语句

SELECT a.id,a.member\_id,a.imei,a.dbp,a.sdp FROM yy\_blood\_model a where a.is\_execute = '0' ORDER BY a.create\_time DESC limit 1

UPDATE yy\_blood\_model a SET a.is\_execute ='1' WHERE a.id = #{id}

### 定时任务（每五分钟执行一次）

#### 开发思路

1、BloodPressureSecondScheduledTask类

2、查询血压建模表中的所有数据

3、调用三方接口：校准血压建模

#### cron表达式

0 0/5 \* \* \* ?

#### 涉及的表

yy\_blood\_model

#### 涉及的sql语句

SELECT a.member\_id,a.imei,a.dbp,a.sdp FROM yy\_blood\_model a ORDER BY a.create\_time DESC

#### 涉及的三方接口

校准血压

### 备份血压建模

#### 接口地址

auth/memberDevice/backupBloodPressureModel

#### 开发思路

1、MemberDeviceController类中的backupBloodPressureModel方法（auth模块）

2、MemberDevicesController类中的backupBloodPressureModel方法。

#### 涉及的模块

auth

device

#### 涉及的表

yy\_blood\_model\_backup

#### sql语句

普通的insert into

### 查询备份列表

#### 接口地址

auth/memberDevice/queryBackupBloodPressureList

#### 开发思路

1、MemberDeviceController类中的queryBackupBloodPressureList方法（auth模块）

2、

#### 涉及的表

yy\_blood\_model\_backup

yy\_member

#### sql语句

**SELECT** ybmb.member\_id,ym.nike\_name,ybmb.backup\_time **FROM** yy\_blood\_model\_backup ybmb **LEFT** **JOIN** yy\_member ym **ON** ybmb.member\_id = ym.id

**WHERE** ym.nike\_name = 'CC'

**ORDER** **BY** ybmb.backup\_time **DESC**

### 查询备份列表-详情

#### 接口地址

auth/memberDevice/queryBackupBloodPressureDetail

#### 开发思路

1、MemberDeviceController类中的queryBackupBloodPressureDetail方法。

2、

#### 涉及的表

yy\_blood\_model\_backup

#### sql语句

**SELECT** dbp, sdp, start\_time **as** startTime, end\_time **as** endTime

**FROM** yy\_blood\_model\_backup **WHERE** member\_id =

# 登录-本机号码一键登陆

## 接口地址

/oauth/callback/onClicklogin

## 开发思路

1、TokenController类中的onClicklogin方法（autha模块）

2、

## 涉及的表

没有涉及到表

# 微信登陆，绑定手机号

## 接口地址

/autha/bindPhoneNumber

## 开发思路

1、TokenController类中的bindPhoneNumber方法

## 涉及的表

yy\_member

yy\_member\_third\_auth

## sql语句

# 历史健康趋势图

## 需求

健康维度：

健康（health）：90-100

亚健康（subHealth）：80-89

欠佳（poor）：60-79

未知（unknown）：<60 （0-59）

周占比：比如健康一周测了8次，所有项总测次数50次，那健康占比就是8除以50。

月占比：（倒数第二周占比-月平均值）+（倒数第一周占比-月平均值）+月平均值

年占比：（倒数第二个月占比-年平均值）+（倒数第一个月占比-年平均值）+年平均值

倒数第二个月占比：根据当前时间算

没有周平均值

月平均值=周占比加起来/周数

年平均值=月占比加起来/月数

## 项目检测指标

心率：heart

低压：dbp

高压：sdp

体温：temperature

血氧：oxygen

餐前血糖：blood\_sugar\_pre

餐后血糖：blood\_sugar

总胆固醇：cho

甘油三酯：tri

尿酸：uri

//心率范围：60-100  
static final String *HEART\_RATE\_RANGE* = "60-100";  
//低压范围  
static final String *LOWER\_DBP\_RANGE* = "58-85";  
//高压范围  
static final String *UPPER\_SDP\_RANGE* = "90-130";  
//体温范围  
static final String *TEMPERATURE\_RANGE* = "36.0-37.1";  
//血氧范围  
static final String *OXYGEN\_RANGE* = "94-99";  
//餐前血糖  
static final String *BLOOD\_SUGAR\_PRE* = "3.9-6.1";  
//餐后血糖  
static final String *BLOOD\_SUGAR* = "3.9-7.8";  
//总胆固醇范围：  
static final String *CHO\_RANGE* = "2.8-5.18";  
//甘油三脂  
static final String *TRI\_RANGE* = "0.56-1.72";  
//尿酸  
static final String *URI\_RANGE* = "89-416";

## 接口

### 查询本周（和上周）健康占比

#### 说明

1：周 2：月 3：年

#### 开发思路

1、ReDataHealthReportController类中的healthRationByWeekPeriod

方法（autha模块）

#### 涉及的模块

autha

model-client

models-device

#### 涉及的表

re\_data\_health\_report

#### sql语句

### 查询本月（和上月）健康占比

#### 开发思路

1、ReDataHealthReportController类中的healthRationByMonthPeriod方法

2、

#### 涉及的表

re\_data\_health\_report

### 查询本年（上年）健康占比

#### 开发思路

#### 涉及的表

re\_data\_health\_report

### 本周（月，年）项目检测异常

#### 接口地址

autha/health\_report/queryWeekDetectionException/{period}/{timeRange}/{targetMemberId}

#### 开发思路

1、ReDataHealthReportController类中的queryDetectionException方法（autha模块）

2、

#### 涉及的表

re\_data\_health\_report

#### 涉及的sql

SELECT create\_time as createTime,

heart\_rate as heartRate,

dbp,

sdp,

oxygen,

blood\_sugar\_pre as bloodSugarPre,

blood\_sugar as bloodSugar,

temperature,

cho,

tri,

uri

FROM re\_data\_health\_report

WHERE yy\_member\_id = #{memberId}

AND create\_time

BETWEEN #{startDateTime} AND #{endDateTime}

order by create\_time DESC

### 查看详情

这是一个地址，没有涉及到接口

测试环境详情页地址：

https://cs.static.huizedaojia.cn/yq/pages/index/history?keyword=heart

生产环境详情页地址：

### 项目检测异常更多页

#### 接口地址

autha/health\_report/queryWeekDetectionException/{period}/{timeRange}/{targetMemberId}

#### 开发思路

1、ReDataHealthReportController类中的queryWeekDetectionExceptionMore（autha模块）

2、

#### 涉及的表

re\_data\_health\_report

#### 涉及的sql语句

# 手表微聊

## 需求

1、微聊的信息在消息通知列表展示。

2、用户列表按最后一次消息时间倒序排序。（前端去开发）

3、微聊发布出去，提示”消息无法发送，请检查设备是否正常“。

## 接口

### 获取手表的IM userID

#### 接口地址

autha/letter/getWatchIMUserId

#### 开发思路

1、MemberLetterController类中的getWatchIMUserId方法（autha模块）

#### 涉及的腾讯IM的接口

查询账号是否已经导入im

导入单个账号

拉取资料

设置资料

#### 涉及的模块

autha

client

#### 手表头像地址

https://yydj-1313075478.cos.ap-beijing.myqcloud.com/2023/05/18/%E6%89%8B%E8%A1%A8%E5%A4%B4%E5%83%8F\_20230518142812A001.png

#### 涉及的表

#### sql语句

### 发单聊消息之后回调（手机app给手表发送语音）

#### 基本说明

这是腾讯云IM进行的回调

#### 回调地址

device/noticeReback/callbackAfterSendMsg

#### 开发思路

1、RemoteApiController类中的callbackAfterSendMsg方法（device模块）

2、获取到腾讯回传过来的参数

3、将语音文件url进行base64编码，获取base64字符串

4、调用厂商的语音接口将base64字符串发送给手表。

#### 说明

需要在腾讯云控制台地址配置回调地址

测试环境地址

https://cs.app.huizedaojia.cn/device/noticeReback/callbackAfterSendMsg

生产环境地址

https://app.huizetianxia.xn--fiqs8s/device/noticeReback/callbackAfterSendMsg

## 手表给手机发语音

1、手表发送语音是走的厂商的三方接口 （http://api.jiai.pro:8080/jiai/audio）

2、回传到咱们的服务器（https://app.huizetianxia.xn--fiqs8s/device/noticeReback/audio）

3、在app的通知中展示消息

### 上传语音文件路径

# 心率短信

## 需求

心率报警提醒的时候短信通知只保留一条

## 开发思路

1、ReDataHealthBuilder

## 说明

这涉及到代码的修改，没有涉及到数据库。

# 大屏app

## 需求

1、返回app用户的定位和数量。（这个没有做）

2、返回心电的订单（购买时间、金额）

## 今日心电订单展示

### 接口地址

auth/dashboard/ecgOrderList

### 开发思路

1、DashBoardController类中的ecgOrderList方法。

### 涉及的模块

auth

api-client

modules-client

### 涉及的表

yy\_ecg\_order

### 涉及的sql语句

--查询今日的订单

SELECT create\_time, actual\_amount  
FROM yy\_ecg\_order  
where create\_time BETWEEN #{startDate} AND #{endDate}  
and order\_status=1  
ORDER BY create\_time DESC

# 大屏设备端

## 需求

1、首页现在都是假数据，体检机和手表数据获取，每10秒获取一次。

2、此页面非常卡，需要优化（前端优化）

3、总数及新增用户数（完成）

（1）设备使用人数：取设备总数，每卖一个设备，会录一个设备串号

（2）今日新增用户：当日0点后录入设备串号的的数量相加

（3）本月新增用户：本月1号开始统计，到本月最后一天结束

（4）绑定设备人数：使用APP绑定设备的人数

4、平台今日成交量，写死（写为多少）就按照目前页面中的，不用修改

5、用户性别对比，写死（写为多少）就按照目前页面中的，不用修改

6、实时在线分布图：取实时在线设备的地位，台数增加到6位

7、今日预警分类

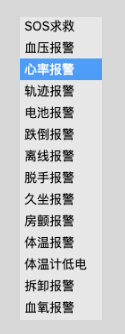
（1）体检机：心率，血压与手表重叠，血糖（餐前，餐后），甘油三酯，胆固醇，尿酸，其他分类是手表的

（2）报预警了就统计

（3）sos，点了就统计

8、报警信息

报警类型：（14个）



9、预警信息展示

（1）状态：设备返回的

（2）只展示报警中的数据

## 接口

### auth/dashboard/warningNumData

#### 说明

报警类型数据

#### 请求方法

GET

#### 请求参数

没有

#### 返回参数

#### 开发思路

1、DashBoardController类中的warningNumData方法（auth模块）

#### 涉及的表

没有涉及的表

### auth/dashboard/queryAlarmInfo

#### 功能说明

查询报警信息，一打开页面就是已读消息。



#### 开发思路

1、DashBoardController类中的queryAlarmInfo方法（auth模块）

2、

#### 涉及的表

re\_warning

yy\_member

#### 涉及的sql

//查询报警消息

SELECT rw.imei, ym.phone, ym.nike\_name as nikeName, rw.warning\_type as warningType, rw.address, rw.create\_time as createTime,rw.status

FROM re\_warning rw,

yy\_member ym

WHERE rw.imei = ym.imei ORDER BY rw.create\_time DESC

//更新阅读状态

UPDATE re\_warning SET read\_status ='1' WHERE read\_status = '2' and create\_time <=’’

### auth/dashboard/queryAlarmButtonFlashing

#### 基本说明

查询是否有报警信息

#### 开发思路

1、DashBoardController类中的queryAlarmButtonFlashing方法

2、

#### 涉及的表

re\_warning

yy\_member

#### sql语句

SELECT COUNT(\*)  
FROM re\_warning rw,  
 yy\_member ym  
WHERE rw.imei = ym.imei  
 and rw.read\_STATUS = '2'  
ORDER BY rw.create\_time DESC

### auth/dashboard/memberNum

#### 基本说明

查询设备端用户的数目

#### 请求方法

GET

#### 请求参数

type:device

#### 开发思路

1、DashBoardController类中的memberNum方法（auth模块）

2、

#### 涉及的表

yy\_devices

### auth/dashboard/order

#### 基本说明

今日呼叫数量，不需要修改，就按目前的就可以

#### 请求方法

GET

#### 请求参数

没有

#### 开发思路

1、

### auth/dashboard/queryOnlineDevicePosition

#### 功能说明

查询实时在线设备（手表）的定位

#### 开发思路

1、DashBoardController类中的queryOnlineDevicePosition方法（auth模块）

#### 涉及的模块

auth

api-device

device

#### 涉及的表

yy\_devices

re\_location

说明：re\_location表中的imei字段添加了普通索引，yy\_devices表中的im字段添加了普通索引。

#### 涉及的sql

SELECT DISTINCT imei FROM re\_location where imei IN (SELECT DISTINCT im FROM yy\_devices ORDER BY create\_time DESC)

SELECT lat, lon  
FROM re\_location  
WHERE imei = #{imei}  
ORDER BY create\_time DESC LIMIT 1

### auth/dashboard/reWarningListData

#### 功能说明

查询用户预警信息展示

#### 请求方法

GET

#### 开发思路

1、DashBoardController类中的reWarningListData方法

#### 涉及的表

yy\_notice

#### 涉及的sql语句

SELECT \* FROM yy\_notice WHERE TYPE = 4 AND DATE\_FORMAT(create\_time,'%Y-%m-%d') = '2023-05-23';

# SOS紧急联系人定时任务

## 基本说明

跑定时任务去更新已绑定手表的紧急联系人

## 开发思路

1、SosTask

## 涉及的表

yy\_member\_devices

yy\_devices

yy\_member\_sos

## sql语句