# 项目各种依赖的版本

## springcloud

2021.0.1

## springcloud alibaba

2021.0.1.0

## nacos

2.0.4

## springboot

2.6.7

## mybatis

3.5.9

## mybatis-plus

3.5.1

# 项目git地址

https://codeup.aliyun.com/62b3ddc464c8a06be2d612d6/houduan/yydj.git

# 测试环境

## 管理后台

https://cs.ht.huizedaojia.cn/

## 接口地址

https://cs.app.huizedaojia.cn/

## H5

https://cs.static.huizedaojia.cn/

## 测试环境服务器信息



IP地址：39.107.52.167（公网地址）

用户名：root

密码：nB6sxrY77FoWt5VH

用户名：jsh\_294

密码：2PHxop1GatG8xxdu

内存大小：8GB

## 测试环境数据库信息

ip地址：39.107.52.167

用户名：yydj\_392

密码：W70jauX2CHxzlNbM

版本：8.0.30

## 测试环境redis信息

端口：16379

密码：LM9NPN8zJFizywBT

## 测试环境nacos

地址：http://39.107.52.167:8848/nacos/

用户名：nacos\_test

密码：123456

## nacos安装地址

# 生产环境

## 服务器信息

ip地址：49.232.16.185

用户名：root

密码：H4PSVKUiKcPHKYz2

用户名：jsh\_294

密码：2PHxop1GatG8xxdu

## 数据库版本

mysql8.0+

说明：生产环境的mysql没有部署在docker中。

# 我的手表imei号

356223502921220

imei位数：15位

我的用户id：

# 心电咨询问题

咨询惠泽天下心之声心电分析群这个群

# 手表咨询问题

咨询即爱技术支持群

# 体检机咨询问题

数据端口对接群

# 模块

## yydj-gateway（网关模块）

## yydj-auth（认证中心）

## yydj-autha（安卓后端服务）

### controller类

#### DevicesController类（设备管理接口）

##### hasVh（检查是否有体检机）

##### friHasVhSendNotice（给有体检机的人发送通知）

##### historyPen（手表电子围栏历史定位查询接口）

##### watchStatus（手表状态记录查询）

##### watchLocation（当前手表的位置查询接口）

##### setLocationMode（设置定位方式）

##### setLocationMode（查询定位方式）

##### pageBattery（展示Battery数据）

##### listData

##### initReMemberDataMaxAndMinVo

##### listEcg

##### pageEcg

##### listVh

##### showData（首页实时数据）

##### hasBind（检测是否绑定所有）

##### hasBindDevice（检测是否绑定设备）

##### runWays（运动轨迹查询）

##### penRun

##### pan

##### setTuiSongPan

##### getTuiSongPan

##### setPan

##### hasSos

##### sos

##### sos

##### info

##### bindDevice

##### unbindDevice

##### setWatch

##### setAlarmClock

##### deleteByid

##### updateAlarmClock

##### alarmClocks

##### setMeasurement

##### getMeasurement

##### heartRates

##### bloodPressure

##### bloodOxygen

##### temperature

##### remoteData

##### remoteData

##### remoteData

##### restReTime

##### remoteAddress

##### defaultInnfo

##### sendAmr

##### getAmrMember

##### getAmr

#### ReDataHealthReportController类（健康管理）

#### MemberLetterController类（私信管理接口）

##### list（私信列表）

##### infoList（私信会话列表）

##### send（发送私信）

##### accountImport（导入单个账号）

POST请求

##### accountImport（导入单个账号）

PUT请求

##### multiAccountImport（导入多个账号）

##### accountCheck（查询账号是否已经导入im）

##### adminSetMsgRead（设置单聊消息已读）

##### adminSetPortrait（设置资料）

#### MemberController类（客户端接口）

##### imsign（即时通讯用户签名）

##### tuiSongSet（查询用户推送设置）

##### tuiSongSet

##### hasPhone（检测是否绑定手机号）

## yydj-api（接口模块 5个）

### yydj-api-asystem

app系统接口模块

### yydj-api-client

客户端模块api

### yydj-api-content

视频内容模块Api

### yydj-api-device

设备模块api

### yydj-api-system

系统接口

## yydj-common（通用模块 7个）

### yydj-common-core（核心模块）

### yydj-common-datascope（权限范围）

### yydj-common-datasource（多数据源）

### yydj-common-log（日志记录）

### yydj-common-redis（缓存服务）

### yydj-common-security（安全模块）

### yydj-common-swagger（swagger系统模块）

## yydj-modules（业务模块 9个）

### yydj-modules-asystem（app系统模块）

### yydj-modules-client（客户端模块）

### yydj-modules-content（视频内容模块）

### yydj-modules-device（设备模块）

#### controller类

RemoteApiController类（接口回传地址）

### yydj-modules-file（文件服务）

### yydj-modules-gen（代码生成）

### yydj-modules-job（定时任务）

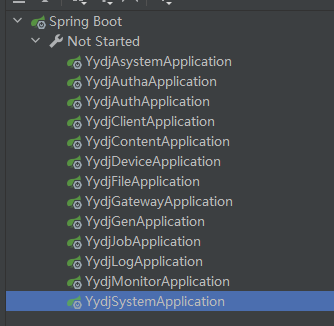
### yydj-modules-log（日志模块）

### yydj-modules-system（系统模块）

## yydj-visual（图形化管理模型）

### yydj-monitor（监控中心）

# 服务（13个）



有13个

# 功能说明

## 系统管理

### 用户管理

### 角色管理

### 菜单管理

### 部门管理

### 岗位管理

### 字典管理

### 参数设置

### 通知公告

### 日志管理

#### 操作日志

#### 登陆日志

### 版本管理

## 系统监控

### 在线用户

### 定时任务

### Sentinel控制台

### Nacos控制台

### Admin控制台

## 系统工具

### 表单构建

### 代码生成

### 系统接口

## 羊羊当家管理

### 养老金管理

### 手表数据

#### 数据列表

#### 运动列表

### 历史健康统计详情

### 作品管理

#### 作品列表

#### 作品审核

#### 评论列表

#### 评论审核

### 类别管理

### 用户管理

#### 用户管理

### 文章管理

#### 协议列表

#### 帮助反馈

### 报警管理

#### 报警记录

#### 报警处理

### 血压建模

登录接口

login

# 数据字典

## 用户性别

男

女

未知

## 菜单状态

显示

隐藏

## 系统开关

正常

停用

## 任务状态

正常

暂停

## 任务分组

默认

系统

## 系统是否

是

否

## 通知类型

通知

公告

## 通知状态

正常

关闭

## 操作类型

新增

修改

删除

授权

导出

导入

强退

生成代码

清空数据

查询

## 系统状态

成功

失败

## 养老金算法

每分等于多少步

## 短视频上传格式

mp4,flv

## app上传格式

apk

## 视频状态

## 平台类型格式

## 通知类型格式

## 评论状态

## 视频分类

## 视频类别

## 视频分类推荐

## 健康值参数设定值

## app初始化背景图片

## 健康管理异常视频

## 报警类型

## 是否有病史

## 是否解绑

# 枚举

## TuiSongType（推送类型）

### DI\_DIAN\_LIANG（低电量报警）

*FAN\_WEI*

*SOS*

*TUO\_SHOU*

*XIN\_LV*

*YUN\_DONG*

*GAO\_YA\_ZHI*

*DI\_YA\_ZHI*

*TI\_WENG*

## NoticeType（通知类型）

## ReWarningType（报警类型）

### 说明

这个枚举里面的类型都是厂商回传接口中定义的

# 项目中的接口

## auth/logout（后台系统退出登陆）

## auth/login（后台系统登陆）

### 请求参数

### 开发思路

1、

## 后台验证码

### 接口地址

/code

### 开发思路

1、RouterFunctionConfiguration类中的routerFunction方法

2、

## 发送短信验证码

### 接口地址

/sms

### 开发思路

1、ValidateSmsCodeHandler类中的

### 说明

使用了阿里云短信服务接口

## 首页实时数据

### 接口地址

showData/{memberId}

### 开发思路

1、

查询re\_data\_health表

### 涉及的表

## 手表状态记录查询

### 接口地址

devices/watchStatus

### 开发思路

## 测量心率

### 接口地址

/device/devices/heartRates

### 开发思路

1、（autha模块）DevicesController类中的heartRates方法

## 测量体温

### 接口地址

### 开发思路

## 健康管理查询

### 接口地址

autha/health\_report/page

### 开发思路

1、ReDataHealthReportController类中的list方法

### 涉及的表

re\_data\_health\_report

## 健康管理统计查询

## 手机app上面的手表微聊

### 接口地址

/devices/sendAmr

### 开发思路

1、 DevicesController类sendAmr方法（modules-device模块）

2、调用三方接口发送语音

2、保存数据到re\_audio表,type的值为1，表示手表自动发起

### 涉及的表

yy\_devices

re\_audio

## 文件上传请求

### 接口地址

upload

### 开发思路

1、SysFileController类中的upload方法。

### 说明

1、目前文件上传到腾讯oss

## system/menu/getRouters

### 功能说明

## auth/dashboard/memberNum

### 功能说明

获取首页总数及新增用户数（app用户端/设备端）

### 开发思路

1、DashBoardController类中的memberNum方法

## auth/dashboard/order

### 功能说明

平台今日成交量



### 开发思路

1、DashBoardController类中的order方法

## auth/dashboard/sexData

### 功能说明

查看性别数据



### 开发思路

1、DashBoardController类中的sexData方法

## auth/dashboard/reWarningListData

### 功能

查询用户预警信息展示

### 请求方法

GET

### 请求参数

无

### 返回参数

### 开发思路

1、DashBoardController类中的reWarningListData方法

执行下面的sql语句获取数据

SELECT \* FROM yy\_notice WHERE TYPE>=4 AND DATE\_FORMAT(create\_time,'%Y-%m-%d') = '2023-04-03';

2、

### 涉及的表

yy\_notice

### sql语句

SELECT \* FROM yy\_notice WHERE TYPE>=4 AND DATE\_FORMAT(create\_time,'%Y-%m-%d') = '2023-04-03';

## /auth/dashboard/memberListData

### 功能

查询用户信息展示

## auth/dashboard/warningNumData

### 功能

查询预警分类

### 开发思路

1、

## auth/member/memberList

### 功能

用户列表（后台菜单中）

## 绑定设备（app端）

### 接口地址

autha/devices/bindDevice

### 开发思路

1、DevicesController类中的bindDevice方法（autha模块）

2、根据串号查询设备表

SELECT \* FROM yy\_devices WHERE im = '356231024325944';

3、绑定设备

（1）根据串号查询设备记录数

SELECT count(\*) FROM yy\_devices WHERE im = '356231024325944';

（2）如果没有记录数，插入数据到yy\_devices表

（3）执行下面的sql

select count(\*) from yy\_member\_devices where yy\_devices\_id = and is\_flag = 0;

（4）如果查询的结果为空，插入数据到yy\_member\_devices表

（5）如果查询的结果不为空，执行下面的sql

select \* from yy\_member\_devices where yy\_member\_id = and yy\_devices\_id = and is\_flag = 0

更新yy\_member\_devices表

4、根据用户id查询用户信息，然后更新yy\_member表，会更新imei字段。

### 涉及的表

devices

yy\_member\_devices

yy\_member

### 涉及的sql

SELECT count(\*) FROM yy\_devices WHERE im = '356231024325944';

select count(\*) from yy\_member\_devices where yy\_devices\_id = and is\_flag = 0;

select \* from yy\_member\_devices where yy\_member\_id = and yy\_devices\_id = and is\_flag = 0

select \* from yy\_member where id=’’;

## 解绑设备（app端）

### 接口地址

/autha/devices/unbindDevice

### 开发思路

1、DevicesController类中的unbindDevice方法（autha模块）

2、根据用户id查询用户信息

### 涉及的表

yy\_member

yy\_devices

yy\_member\_devices

### 涉及的sql

## 头像上传接口

### 接口地址

autha/upload

## 添加或更新紧急联系人

### 接口地址

/autha/devices/sos

### 请求方法

POST

### 开发思路

1、DevicesController类中的sos方法（autha模块）

2、获取前端传过来的参数

### 涉及的模块

autha

client

### 涉及的表

yy\_member\_sos

### 涉及的sql

SELECT yy\_member\_devices.id, im AS imei, ptype, dev\_type

FROM yy\_member\_devices

LEFT JOIN yy\_devices yd ON yd.id = yy\_member\_devices.yy\_devices\_id

WHERE yy\_member\_devices.yy\_member\_id = 2090 AND yy\_member\_devices.is\_flag = 0

### 涉及的三方接口

设置SOS号码

http://api.jiai.pro:8080/jiai/sos

## 微信登录

### 接口地址

autha/oauth/callback/{source}

### 请求参数

{

    "code":"001111ll2v9T9b4Wejml29a3zy1111lE",

    "state":"wx\_login"

}

说明：这两个参数都是微信回调返回的，code每次都会变，不是固定的

### 返回参数

{

"code": 500,

"msg": "invalid code, rid: 647d81cc-17963ece-39651dc4",

"data": {}

}

### 开发思路

1、TokenController类中的login方法（autha模块）

## 手机号登录

### 接口地址

autha/oauth/callback/{source}

### 请求参数

## 微信登录绑定手机号

### 接口地址

autha/bindPhoneNumber

### 开发思路

1、TokenController类中的bindPhoneNumber方法（autha模块）

2、获取前端传过来的参数

3、判断手机号是否存在，查询yy\_member表

4、如果没有查询到用户信息，那么插入信息到yy\_member\_third\_auth表，插入信息到yy\_member表

5、创建token

### 涉及的表

yy\_member\_third\_auth

新增加了一个字段：phone

## 用户信息查询

### 接口地址

autha/client/info/{yyMemberId}

### 开发思路

## 文件上传接口

### 接口地址

autha/upload

### 开发思路

1、UpLoadFileController类中的upload方法（autha模块）

2、SysFileController类中的upload方法（module-file模块）

3、

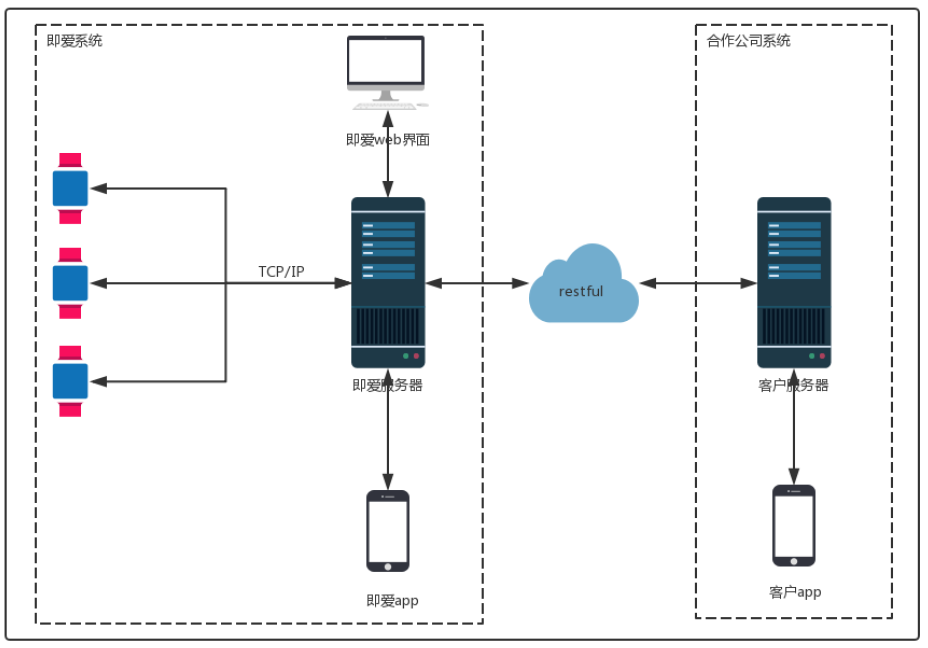
## /autha/devices/watchStatus

### 功能说明

## /autha/watchLocation/{memberId}

# 项目中对接的三方接口

## 架构示意图



IThirdDevApiService接口类

## 心率相关的接口

### 心率测量请求

Res putHeartRates(String imei);

## 心电相关的接口

### ECG心电

Res<EcgVo> ecg(EcgDto ecgDto);

## 血压相关的接口

### 心率血压血氧血糖温度数据

Res health(HealthDto health);

### Res putBloodPressure(String imei);

血压测量其请求

### Res patchBloodPressure(BloodPressureDto bloodPressureDto);

校准血压

## 血氧相关的接口

### 血氧测量请求

Res putBloodOxygen(String imei);

## 体温相关的接口

### 体温测量请求

Res putTemperature(String imei);

这是请求测量，不是取体温，返回成功，表示已经收到请求，这时不会有数据，数据通过第四章接口异步返回。

### 设置心率/血压/体温计测量周期

Res postHealthInterval(String imei, String type, String interval);

## 闹钟相关的接口

### Res postAlarm(AlarmDto alarm);

设置闹钟

### Res deleteAlarm(String cid, String imei);

删除闹钟

### Res putAlarm(AlarmDto alarmDto, String cid);

更新闹钟

## 亲情号相关的接口

### Res postFamilyNumber(String imei, String name, String mobileNo);

设置亲情号码

### Res deleteFamilyNumber(String imei, String id);

删除亲情号码

### Res putFamilyNumber(String imei, String name, String mobileNo, String id);

修改亲情号码

### Res put10FamilyNumber(String imei, List<WatchesSosBeanDto> watchesSosBeanDtoList);

一次更新10个亲情号码

## SOS相关的接口

### 设置SOS号码

Res putSos(String imei, List<WatchesSosBeanDto> watchesSosBeanDtoList);

## 预警相关的接口

### 预警数据

Res warning(WarningDto warningDto);

说明：这个没有三方接口

## 语音相关的接口

### 语音服务

Res putAudio(String imei, String type, String audio);

## 定位相关的接口

### 定位方式修改

Res postPositionMode(String imei, String mode);

### 定位请求

Res putLocation(String imei);

### 根据经纬度获取地址

JSONObject getLatAndLngByAddr(String longitude, String latitude);

### 自由设置定位时间

Res postPositionTime(String imei, String mode, String intervalTim, String gpsFlg);

### 定位数据

Res location(String imei, String type, String lat, String lon);

## 设备相关的接口

### 设备信息设置

Res putWatches(WatchesDto watchesDto)

### 查询设备信息

Res<WatchesVo> postWatches(String imei);

### 恢复出厂设置

Res postRestore(String imei);

### 远程关机

Res postClose(String imei);

### 设备电量/计步数据/温度计电量

Res beatHeart(BeatHeartDto beatHeart);

### 蓝牙地址

Res bluetooth(String imei, String blueTooth);

### 确认阅读

Res confirm(String imei, String oriSeq);

### 发送文字到设备展示

Res putInformation(String imei, List<JiaiInformationItemDto> jiaiInformationItemDtosl);

# 厂商三方接口（pdf文件）

## 设备信息设置

## 查询设备信息

## 心率测量请求

## 血压测量请求

## 校准血压

## 定位方式修改

## 恢复出厂设置

## 设置亲情号码

## 一次更新10个亲情号码

## 删除亲情号码

## 修改亲情号码

## 设置SOS号码

## 定位请求

## 语音服务

### 重要说明

音频类型格式只支持amr格式。

## 设置心率、血压、体温计测量周期

## 发送文字到设备展示

## 自由设置定位时间

## 设置闹钟

## 修改闹钟

## 删除闹钟

## 远程关机

## 体温测量请求

## 血氧测量请求

## 发送文字到设备展示（无需确认回复）

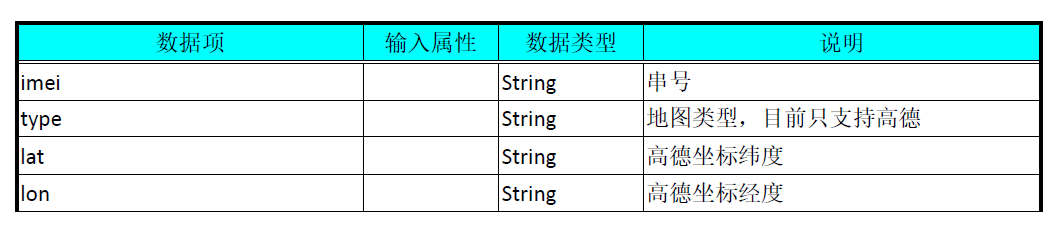
# 接口回传地址

## 地址

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

## 定位数据

http://127.0.0.1:8080/location



### 使用场景

### 接口地址

/device/noticeReback/location

### 请求方法

POST

### 请求参数

{

    "imei": "",

    "type": "",

    "lat": "",

    "lon": ""

}



### 返回参数

### 开发思路

1、RemoteApiController类中的location方法（device模块）

2、获取请求参数并转换为data对象

3、发送电子围栏预警功能

4、将数据插入re\_location表

5、根据经纬度获取地址

6、更新预警定位信息（re\_warning表）

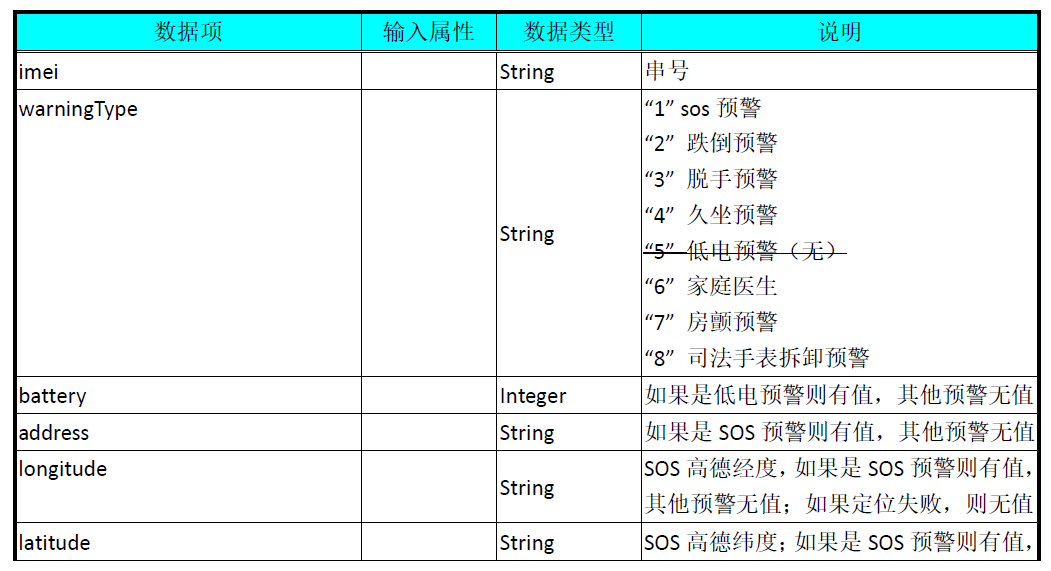
### 涉及的表

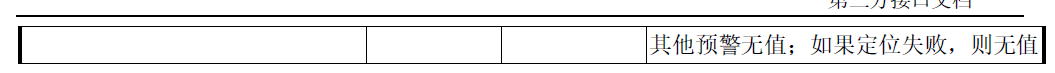
re\_location

re\_warning

## 预警数据

http://127.0.0.1:8080/warning





### 接口地址

device/noticeReback/warning

### 请求方法

POST

### 请求参数

{

    "imei": "",

    "warningType": "",

    "battery": 10,

    "address": "",

    "longitude": "",

    "latitude": ""

}

### 返回参数

### 开发思路

1、RemoteApiController类中的warning方法（device模块）

2、获取前端传过来的数据，封装为ReWarning对象

3、根据警告信息发送通知

（1）发送短信消息

（2）发送预警通知和推送通知

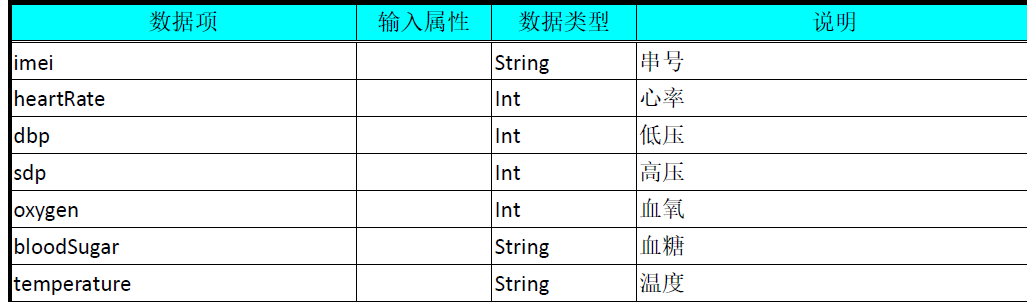
### 涉及的表

re\_warning

yy\_notice

## 心率血压血氧血糖温度数据（必须实现）

http://127.0.0.1:8080/health



### 接口地址

device/noticeReback/health

### 请求方法

post

### 请求参数

{

    "bloodSugar": "0.0",

    "dbp": 78,

    "heartRate": 76,

    "imei": "356223502921220",

    "oxygen": 97,

    "sdp": 124,

    "temperature": ""

}

### 返回参数

{

    "code": 200,

    "msg": null,

    "data": {}

}

### 开发思路

1、RemoteApiController类中的health方法（device模块）

2、将前端传的数据插入到re\_data表中

3、根据imei查询用户id

### 涉及的表

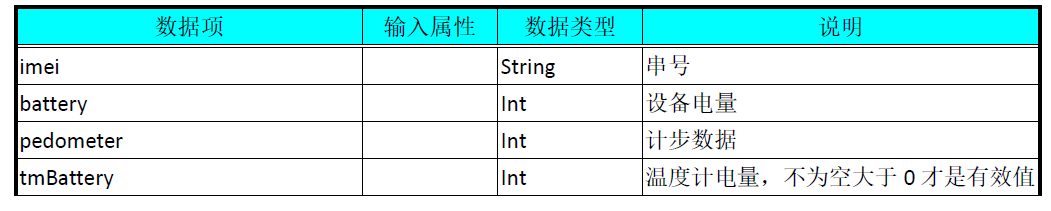
re\_data

re\_data\_health

re\_data\_health\_report

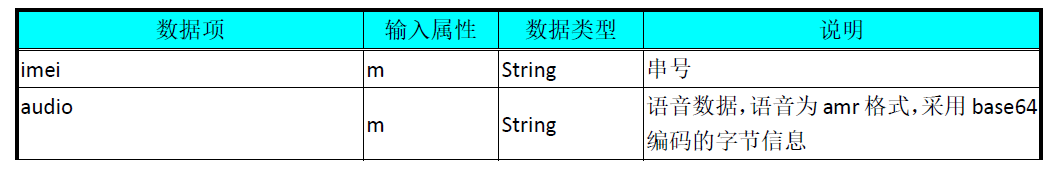
## 设备电量/计步数据/温度计电量（必须实现）

http://127.0.0.1:8080/beatHeart



## 语音服务（必须实现）

http://127.0.0.1:8080/audio



### 接口地址

device/noticeReback/audio

### 请求方法

POST

### 请求参数

{

    "imei":"356223502921220",

    "audio":"5L2g5aW9"

}

### 返回参数

{

    "code": 200,

    "msg": null,

    "data": {}

}

### 开发思路

1、RemoteApiController类中的audio方法（device模块）

2、保存信息到re\_audio表

3、发送通知和推送

### 涉及的表

re\_audio

### sql语句

SELECT \* FROM re\_audio WHERE imei = '' AND TYPE = '1' ORDER BY create\_time DESC LIMIT 1;

## ECG心电（选择实现）

http://127.0.0.1:8080/ecg

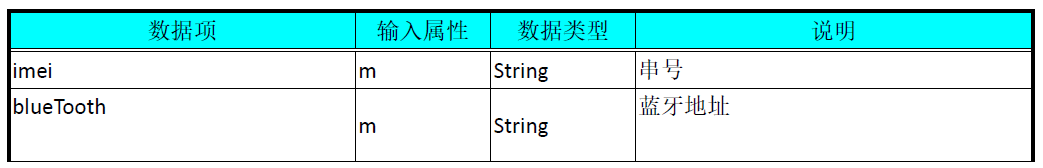
该接口抄送的为原始心电数据，需要自行进行分析，并按接口内容返回分析结果。



Response

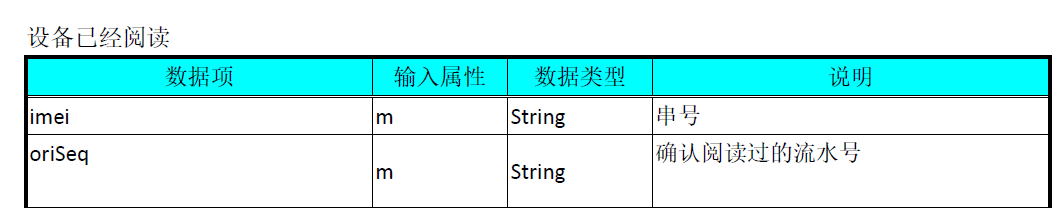
## 蓝牙地址（选择实现）

http://127.0.0.1:8080/bluetooth



## 确认阅读（选择实现）

http://127.0.0.1:8080/confirm



## RRI数据分析（选择实现）

# 表

## re\_audio（语音服务表）

## re\_battery（电量表）

## re\_data（数据表）

## re\_data\_health（健康表）

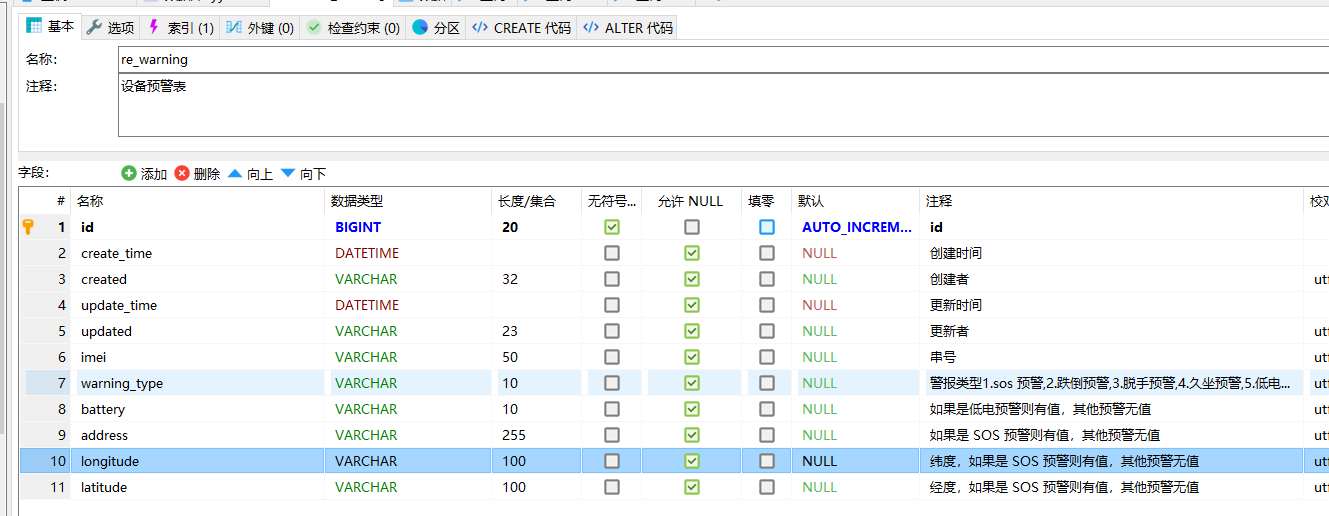
## re\_data\_health\_report（健康报告表）

## re\_ecg（心电数据表）

## re\_location（设备坐标定位表）

## re\_vh（体验机数据表）

## re\_warning（设备预警表）



## sys\_config（参数配置表）

## sys\_dept（部门表）

## sys\_dict\_data（字典数据表）

## sys\_dict\_type（字典类型表）

## sys\_job（定时任务调度表）

## sys\_job\_log（定时任务调度日志表）

## sys\_logininfor（系统访问记录）

## sys\_menu（菜单权限表）

## sys\_notice（通知公告表）

## sys\_oper\_log（操作日志记录表）

## sys\_post（岗位信息表）

## sys\_role（角色信息表）

## sys\_role\_dept（角色部门关联表）

## sys\_role\_menu（角色菜单关联表）

## sys\_user（用户信息表）

## sys\_user\_post（用户岗位关联表）

## sys\_user\_role（用户角色关联表）

## yy\_alarm\_clock

## yy\_alarm\_clock\_back（用户闹钟设置表）

## yy\_app\_version（版本控制表）

## yy\_comment（评论表）

## yy\_device\_sos\_clock\_error\_log

## yy\_error\_log（异常日志表）

## yy\_hot（热门关键词表）

## yy\_member（客户表）

## yy\_devices（设备表）

## yy\_member\_devices（客户设备关联表）

## yy\_member\_address\_book（会员通讯录）

## yy\_notice（通知通告表）

### type

## yy\_ecg\_order（心电套餐订单表）

# 工具类

## RedisService

# 通知和推送的区别

通知（Notification）和推送（Push）是两种不同的方式，用于向用户传递信息或通知。它们在应用程序和移动设备之间进行消息传递，但存在一些区别。

## 通知（Notification）

通知是由应用程序发送到用户设备上的一种消息，用于向用户传递信息、提醒或通知。

通知通常以弹出窗口、图标提示、声音或振动等形式出现在用户的设备上，以引起用户的注意。

通知可以包含文本、图像、链接等信息，用于向用户传达特定的消息或提醒。

通知可以是即时的，也可以是预定的，例如日程提醒、社交媒体更新或应用程序的状态更改。

通知通常是由应用程序自身触发和发送的，用户可以通过设备的通知中心或设置来管理和控制通知的显示和行为。

## 推送（Push）

推送是由后台服务器发送到设备的一种消息，用于向应用程序传递信息或通知。

推送消息是基于事件或条件触发的，例如新消息、新闻更新、系统提醒等。

推送消息可以在应用程序未打开的情况下传递给用户，以提供实时的信息和通知。

推送消息通常通过移动设备操作系统的推送服务（如苹果的APNs和安卓的FCM）进行传输和接收。

推送消息可以包含文本、图像、链接等信息，用于向应用程序传递特定的内容或指令。

应用程序需要在接收到推送消息后进行处理和展示，以便向用户显示相关的信息和通知。

## 总结

通知是由应用程序发送给用户设备上的消息，旨在向用户传递信息或提醒，用户可以在设备上管理和控制通知的显示和行为。推送是由后台服务器发送给应用程序的消息，用于向应用程序传递信息或通知，可以在应用程序未打开的情况下传递给用户。

# 问题

## 1、Read timed out executing GET http://yydj-device/device/re-data-health-report/healthRationByPeriod?period=1&memberId=2030",

**访问某个接口读取超时**

## 2、Unable to connect to Redis; nested exception is io.lettuce.core.RedisConnectionException: Unable to connect to localhost/<unresolved>:6379

# amr文件

移动设备的音频文件格式。