

千寻SDK Android版开发指南

V1.1.0

法律声明

版权所有© 2019，千寻位置网络有限公司。保留一切法律权利。本文档包含的所有内容除特别声明之外，版权均属于千寻位置网络有限公司所有，受《中华人民共和国著作权法》及相关法律法规和中国加入的所有知识产权方面的国际条约的保护。未经本公司书面许可，任何单位和个人不得以任何方式(电子或机械,包括影印)或理由对该文档或其包含的任何产品、服务、信息、材料的任何部分进行使用、复制、修改、抄录、传播或其它产品捆绑使用、销售，否则将视为侵权，本公司必依法追究其法律责任。本文档并不代表供应商或其代理的承诺，千寻位置网络有限公司可在不作任何声明的情况下对本文档内容进行修改。本文档中提到的其它公司及其产品的商标所有权属于该商标的所有者。

千寻位置网络有限公司

联系邮箱: service@qxwz.com

官方网站: www.qxwz.com

一、开发准备

1. 概述

本文档用于说明千寻SDK Android版的接入方式，以帮助用户在项目中进行集成和使用

2. 使用流程

2.1 SDK使用流程大体如下:

1、先到[千寻官网](#)申请DSK(设备服务号)和DSS(设备服务秘钥)，并在控制中心绑定Device Id和Device Type等信息。2、配置好工程，调用对应的API接口完成操作。3、运行应用程序。

2.2 申请DSK(设备服务号)、DSS(设备服务秘钥)

DSK、DSS由千寻位置网提供，获取办法请见 [在线帮助文档](#)。将DSK、DSS与设备类型(Device Type: 建议填写机型)及设备Id(Device Id: 机器唯一标识，如IMEI或UUID)。

二、配置工程

1. Android Studio 配置工程

1.1 新建一个Android工程

1.2 导入aar包

将aar包复制到工程的libs目录下，并在build.gradle中添加以下代码，如图所示：

```
repositories {  
    flatDir {  
        dirs 'libs' //this way we can find the .aar file in libs folder  
    }  
}  
  
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
    implementation(name: 'libqwxz_release_xxxxxx', ext: 'aar')  
}
```

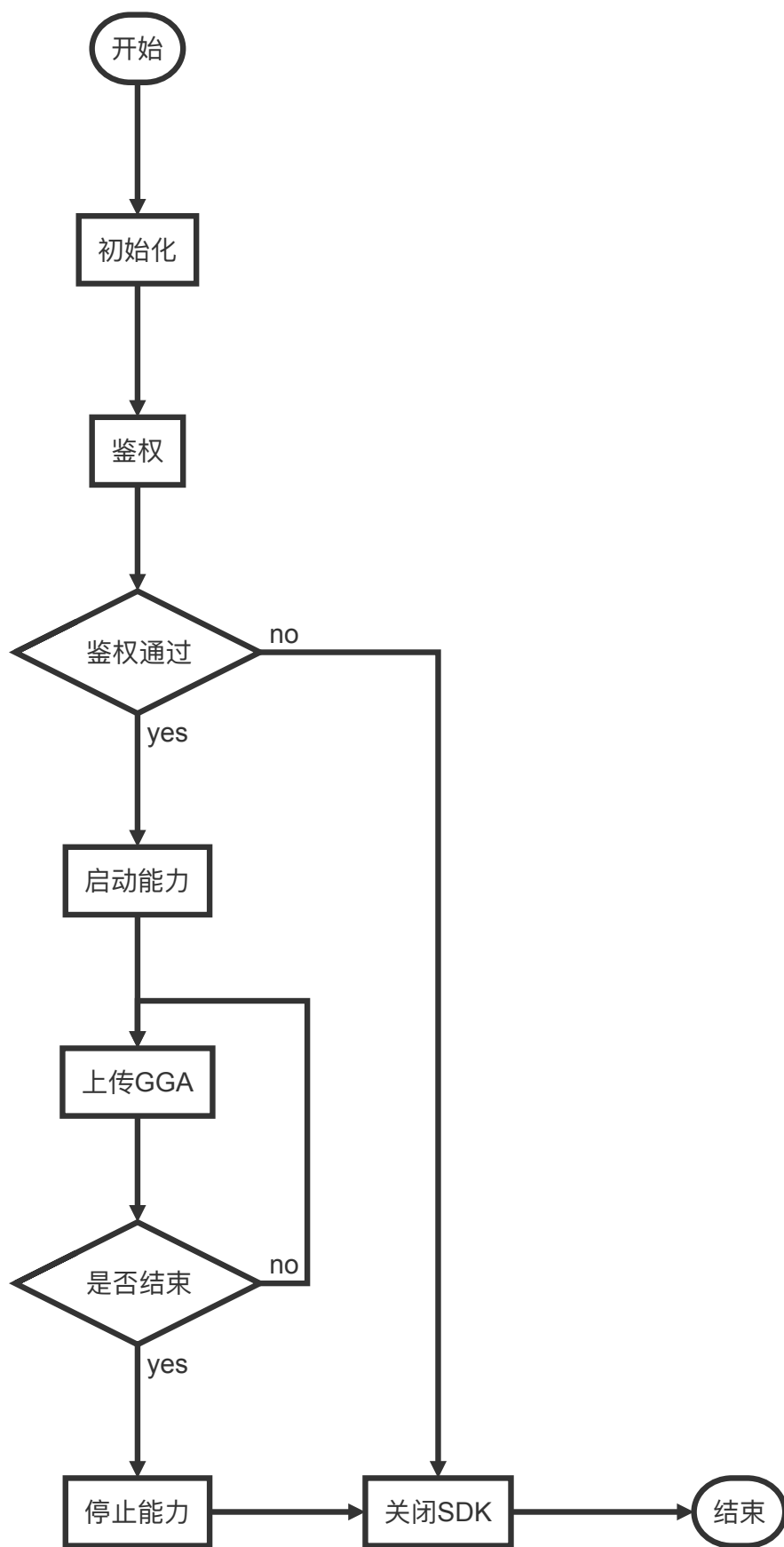
1.3 权限配置

在AndroidManifest.xml中配置权限：

```
<!--允许程序打开网络套接字-->  
<uses-permission android:name="android.permission.INTERNET" />
```

三、快速开始

1. 调用流程



2. 初始化

初始化方法负责接收用户传入的账号信息、回调接口实现、服务器信息、OSS配置等信息并初始化各子模块，其中账号信息和回调接口实例是必填参数，服务器信息和OSS配置是可选参数，代码示例如下：

```

protected class RtcmsDKCallbackImpl implements IRtcmsDKCallback {

    @Override
    public void onData(int dataType, byte[] data) {
        consolePrint("onData, dataType:" + dataType + ", len:" + data.length);
    }

    @Override
    public void onStatus(int code) {
        consolePrint("onStatus, code:" + code);
    }

    @Override
    public void onAuth(int code, List<CapInfo> caps) {
        consolePrint("onAuth, code:" + code);
        if (code == QXWZ_SDK_STAT_AUTH_SUCC) {
            consolePrint("onAuth, auth successfully.");
            for (CapInfo capInfo : caps) {
                consolePrint("onAuth, capInfo:" + capInfo.toString());
            }
        }
    }

    @Override
    public void onStart(int code, int capId) {
        consolePrint("onStart, code:" + code + ",capId:" + capId);
        if (code == Constants.QXWZ_SDK_STAT_CAP_START_SUCC) {
            consolePrint("onStart, start successfully.");
            isStart = true;
        } else {
            consolePrint("onStart, failed to start.");
            isStart = false;
        }
    }
}

SDKConfig sdkConfig = SDKConfig.builder()
    .setAccountInfo(
        AccountInfo.builder()
            .setKeyType(QXWZ_SDK_KEY_TYPE_DSK)
            .setKey(DSK)
            .setSecret(DSS)
            .setDeviceId(DEVICE_ID)
            .setDeviceType(DEVICE_TYPE)
            .build()
    )
    .setRtcmsDKCallback(new RtcmsDKCallbackImpl())
    .build();

RtcmsSDKManager.getInstance().init(sdkConfig);

```

3. 鉴权

SDK需完成鉴权后才能提供服务，鉴权方法是异步方法，鉴权结束后通过调用初始化时传入的回调接口实例中的onAuth方法通知用户鉴权结果，代码示例如下：

```
RtcmSDKManager.getInstance().auth();
```

4. 启动能力

鉴权成功后可启动能力获取相应服务，能力启动接口是异步方法，能力启动结束后通过调用初始化时传入的回调接口实例中的onStart方法通知用户启动结果，代码示例如下：

```
RtcmSDKManager.getInstance().start(QXWZ_SDK_CAP_ID_NOSR);
```

5. 上传GGA

能力启动成功后可通过每隔一段时间(间隔时间不短于1秒)上传GGA获取差分数据，差分数据通过调用初始化时传入的回调接口实例中的onData方法传递给用户，代码示例如下：

```
new Thread() {  
    public void run() {  
        while (isStart) {  
            RtcmSDKManager.getInstance().sendGga(GGA);  
            SystemClock.sleep(1000);  
        }  
    }  
}.start();
```

6. 停止能力

能力启动成功后可通过停止能力终止相关服务，示例代码如下：

```
RtcmSDKManager.getInstance().stop(QXWZ_SDK_CAP_ID_NOSR);
```

7. 关闭SDK

初始化后的任何阶段都可以关闭SDK，示例代码如下：

```
RtcmSDKManager.getInstance().cleanup();
```

四、接口详细说明

SDK所有API均通过RtcmSDKManager这个类提供，下面的所以接口均可以通过RtcmSDKManager.getInstance()获取RtcmSDKManager的单例后进行访问。

1. 初始化接口

用于SDK初始化，除特殊说明，其他接口均依赖该接口。

1.1 接口定义

```
int init(SDKConfig config);
```

1.2 参数说明

参数名	必填	参数类型	说明
config	是	SDKConfig	包含四个字段 accountInfo: 账号信息(必填) callback: 回调函数(必填) serverInfo: 服务器信息(选填) ossConfig: OSS配置信息

1.2.1 参数类型说明

- 类: SDKConfig 该类支持用构建者模式生成实例，例如：

```
SDKConfig.builder()  
    .setAccountInfo(accountInfo)//设置账号  
    .setRtcmSDKCallback(new RtcmSDKCallbackImpl())//设置回调  
    .setServerInfo(serverInfo)//设置服务器信息(可选)  
    .setOssConfig(ossConfig)//设置oss配置信息(可选)  
    .build();
```

- 类: AccountInfo 该类用来设置账号信息，支持用构建者模式生成实例，例如：

```
AccountInfo.builder()  
    .setKeyType(QXWZ_SDK_KEY_TYPE_DSK)//设置账号类型  
    .setKey(DSK)//设置账号  
    .setSecret(DSS)//设置密钥  
    .setDeviceId(DEVICE_ID)//设置设备ID  
    .setDeviceType(DEVICE_TYPE)//设置设备类型  
    .build();
```

- 接口: IRtcmSDKCallback 该接口用来声明用户需要实现的回调方法，声明如下：

```
public interface IRtcmSDKCallback {  
    void onData(int dataType, byte[] data, int len);//收到差分数据时会回调该方法  
    void onStatus(int code);//SDK状态变更时会回调该方法  
    void onAuth(int code, List<CapInfo> caps);//鉴权结束后会回调该方法  
    void onStart(int code, int capId);//能力启动结束时会回调该方法  
}
```

- 类：ServerInfo 该类用来设置千寻服务器信息，包括OPENAPI服务器的域名和端口号以及OSS服务器的域名和端口号，部分用户需要传入该类实例实现访问指定域名和端口号对应的服务器的目的，支持用构建者模式生成实例，例如：

```
ServerInfo serverInfo = ServerInfo.builder()
    .setOpenApiHost(OPENAPI_HOST) //设置OPENAPI服务器域名
    .setOpenApiPort(OPENAPI_PORT) //设置OPENAPI服务器端口号
    .setOssHost(OSS_HOST) //设置OSS服务器域名
    .setOssPort(OSS_PORT) //设置OSS服务器端口号
    .build();
```

- 类：OssConfig 该类用来设置OSS配置信息，包括OSS心跳间隔(默认值15s)和OSS重连间隔(默认值1s)，支持用构建者模式生成实例，例如：

```
OssConfig ossConfig = OssConfig.builder()
    .setHeartbeatInterval(30) //设置心跳间隔，单位秒
    .setRetryInterval(20) //设置重连间隔，单位秒
    .build();
```

1.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_INVALID_CONFIG	QXWZ_SDK_STAT_OK：初始化成功 QXWZ_SDK_ERR_INVALID_CONFIG：config非法,建议检查参数是否正确

2. 鉴权接口

用于设备鉴权，该接口是异步接口，接口调用成功不代表鉴权通过，鉴权结果将通过回调接口实例的onAuth方法通知开发者，鉴权回调函数说明见回调接口说明第1小节。

鉴权失败后SDK不会自动重试，请开发者自行重试。

2.1 接口定义

```
int auth();
```

2.2 参数说明

无

2.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_AUTHING	QXWZ_SDK_STAT_OK：接口调用成功 QXWZ_SDK_ERR_AUTHING：鉴权进行中，SDK会忽略本次鉴权请求,建议在收到鉴权回调后再执行后续操作

3. 能力启动接口

用于单个能力启动，该接口是异步接口，接口调用成功不代表启动成功，能力启动结果将通过回调接口实例的onStart方法通知开发者，能力启动回调函数说明见回调接口说明第1小节。

能力启动失败后SDK不会自动重试，请开发者自行重试。

3.1 接口定义

```
int start(int capId);
```

3.2 参数说明

参数名	必填	参数类型	说明
capId	是	int	需要启动的能力对应的ID

3.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_NOT_AUTHED QXWZ_SDK_ERR_ALREADY_STARTED QXWZ_SDK_ERR_CAP_NOT_IN_SERVICE QXWZ_SDK_ERR_CAP_INACTIVE QXWZ_SDK_ERR_CAP_EXPIRED QXWZ_SDK_ERR_CAP_PAUSE QXWZ_SDK_ERR_CAP_DISABLED QXWZ_SDK_ERR_CAP_NOT_FOUND	QXWZ_SDK_STAT_OK：接口调用成功 QXWZ_SDK_ERR_NOT_AUTHED：鉴权尚未通过,建议先调用鉴权接口且鉴权通过的情况下再调用该接口 QXWZ_SDK_ERR_ALREADY_STARTED：能力已经启动,建议不要重复启动能力 QXWZ_SDK_ERR_CAP_NOT_IN_SERVICE：能力不在服务中,该能力无法使用请联系业务 QXWZ_SDK_ERR_CAP_INACTIVE：能力未激活,该能力无法使用请联系业务 QXWZ_SDK_ERR_CAP_EXPIRED：能力已过期,该能力无法使用请联系业务 QXWZ_SDK_ERR_CAP_PAUSE：能力已暂停,该能力无法使用请联系业务 QXWZ_SDK_ERR_CAP_DISABLED：能力已禁用,该能力无法使用请联系业务 QXWZ_SDK_ERR_CAP_NOT_FOUND：能力未找到,该能力无法使用请联系业务

4. 能力停止接口

用于单个能力停止，该接口是同步接口，接口调用成功表示能力已成功停止。

4.1 接口定义

```
int stop(int capId);
```

4.2 参数说明

参数名	必填	参数类型	说明
capId	是	int	需要停止的能力对应的ID

4.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_NOT_AUTHED QXWZ_SDK_ERR_NOT_STARTED QXWZ_SDK_ERR_CAP_NOT_FOUND	QXWZ_SDK_STAT_OK：接口调用成功 QXWZ_SDK_ERR_NOT_AUTHED：鉴权尚未完成,建议先调用鉴权和能力启动接口且鉴权通过并且能力启动成功的情况下再调用该接口 QXWZ_SDK_ERR_NOT_STARTED：能力尚未启动,建议先调用能力启动接口且能力启动成功的情况下再调用该接口 QXWZ_SDK_ERR_CAP_NOT_FOUND：能力未找到,该能力无法使用请联系业务

5. SDK关闭接口

用于结束SDK生命周期，释放SDK占用的内存资源。该接口是同步接口，接口调用成功表示SDK已成功关闭。

5.1 接口定义

```
int cleanup();
```

5.2 参数说明

无

5.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_FAIL	QXWZ_SDK_STAT_OK：接口调用成功 QXWZ_SDK_ERR_FAIL：接口调用失败,请稍后重试

6. 版本获取接口

用于获取SDK版本号，该接口不依赖初始化接口。

6.1 接口定义

```
String getSDKVersion();
```

6.2 参数说明

无

6.3 返回值说明

返回值类型	说明
String	当前版本号

7. 能力信息获取接口

用于获取当前账号(设备)对应的能力信息，该接口需在鉴权成功后调用。

7.1 接口定义

```
List<CapInfo> getCapsInfo();
```

7.2 参数说明

无

7.3 返回值说明

- CapInfo 保存能力信息，字段定义如下：

```
private int capId; //能力ID
private byte state; //能力状态
private byte actMethod; //激活方式
private long expireTime; //能力过期时间
```

8. 上传GGA接口

用于上传GGA数据以获取差分数据，该接口是异步接口，差分数据将通过回调接口实例的onData方法传递给开发者，数据回调函数说明见回调接口说明第1小节。

上传GGA需在能力成功开启之后，能力开启之前调用该接口GGA数据会被丢弃，上传GGA期间发生断网等网络异常时，SDK会进行自动重连，直到网络恢复，开发者无需关注。

8.1 接口定义

```
int sendGga(String gga);
```

8.2 参数说明

参数名	必填	参数类型	说明
gga	是	String	GGA数据

8.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_INVALID_PARAM	QXWZ_SDK_STAT_OK: 接口调用成功 QXWZ_SDK_ERR_INVALID_PARAM: gga为空或len为0

9. 终端激活接口

开发者可在终端发起能力激活操作。

9.1 接口定义

```
int activate(int[] caps, IRtcmSDKActivateCallback callback);
```

9.2 参数说明

参数名	必填	参数类型	说明
caps	是	int[]	欲激活的能力数组, 以能力ID表示
callback	是	IRtcmSDKActivateCallback	激活回调接口, 详细参见回调接口说明第2小节

9.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_INVALID_PARAM QXWZ_SDK_ERR_ACTIVATING QXWZ_SDK_ERR_NOT_INITED	QXWZ_SDK_STAT_OK: 接口调用成功 QXWZ_SDK_ERR_INVALID_PARAM: 非法的输入参数, 请检查输入参数 QXWZ_SDK_ERR_ACTIVATING: 正在执行激活操作 QXWZ_SDK_ERR_NOT_INITED: SDK未初始化

10. 获取当前坐标系接口

获取当前坐标系的信息。

10.1 接口定义

```
int getCoordSys(IRtcmSDKGetCoordSysCallback callback);
```

10.2 参数说明

参数名	必填	参数类型	说明
callback	是	IRtcmSDKGetCoordSysCallback	坐标系查询回调接口, 详细参见回调接口说明第3小节

10.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_INVALID_PARAM QXWZ_SDK_ERR_GETTING_COORD_SYS QXWZ_SDK_ERR_NOT_INITED	QXWZ_SDK_STAT_OK: 接口调用成功 QXWZ_SDK_ERR_INVALID_PARAM: 非法的输入参数, 请检查输入参数 QXWZ_SDK_ERR_GETTING_COORD_SYS: 正在执行查询操作 QXWZ_SDK_ERR_NOT_INITED: SDK未初始化

11. 设置当前坐标系接口

设置当前坐标系的信息。

11.1 接口定义

```
int setCoordSys(int coordSysIndex, IRtcmSDKSetCoordCallback callback);
```

11.2 参数说明

参数名	必填	参数类型	说明
coordSysIndex	是	int	坐标系索引值, 1:ITRF2008 2:WGS84 3:CGCS2000
callback	是	IRtcmSDKSetCoordCallback	坐标系查询回调接口, 详细参见回调接口说明第4小节

11.3 返回值说明

返回值类型	取值范围	说明
int	QXWZ_SDK_STAT_OK QXWZ_SDK_ERR_INVALID_PARAM QXWZ_SDK_ERR_SETTING_COORD_SYS QXWZ_SDK_ERR_NOT_INITED	QXWZ_SDK_STAT_OK: 接口调用成功 QXWZ_SDK_ERR_INVALID_PARAM: 非法的输入参数, 请检查输入参数 QXWZ_SDK_ERR_SETTING_COORD_SYS: 正在执行设置操作 QXWZ_SDK_ERR_NOT_INITED: SDK未初始化

四、回调接口说明

1. 通用回调接口

1.1接口定义

SDK初始化时需传入回调接口实例, 用于接收鉴权结果、能力启动结果、差分数据以及SDK状态变更, 回调接口定义如下:

```
public interface IRtcmSDKCallback {  
    void onData(int dataType, byte[] data, int len);  
    void onStatus(int code);  
    void onAuth(int code, List<CapInfo> caps);  
    void onStart(int code, int capId);  
}
```

1.2. 回调方法说明

1.2.1 鉴权回调方法

该方法在鉴权结束后被调用，用于通知用户鉴权结果，方法定义如下：

1.2.1.1 方法定义

```
void onAuth(int code, List<CapInfo> caps);
```

1.2.1.2 参数说明

参数名	参数类型	说明
code	int	鉴权结果状态码，QXWZ_SDK_STAT_AUTH_SUCC表示鉴权成功，其他表示鉴权失败
caps	List	能力信息，仅在鉴权成功时该值为非空，CapInfo的说明见7.3

1.2.1.3 状态码说明表

状态码	值	说明
QXWZ_SDK_STAT_AUTH_SUCC	201	鉴权成功
QXWZ_SDK_ERR_NETWORK_UNAVAILABLE	-101	网络连接不可用,建议重试
QXWZ_SDK_ERR_CAP_GET_CONF_FAIL	-204	获取能力信息失败,请联系业务
QXWZ_SDK_ERR_CAP_MANUAL_ACT_REQUIRED	-207	能力需要手动激活,请联系业务
QXWZ_SDK_ERR_CAP_ACT_ON_TERM_REQUIRED	-208	能力需要在设备端激活,请联系业务
QXWZ_SDK_ERR_CAP_SYSTEM_ERROR	-211	能力系统错误,请联系业务
QXWZ_SDK_ERR_CAP_NOT_INCLUDE	-212	当前账号未包含该能力,请联系业务
QXWZ_SDK_ERR_AUTH_FAIL	-301	鉴权失败,建议重试,多次鉴权失败请请联系业务
QXWZ_SDK_ERR_NO_AVAIL_ACC	-302	无可账号,请联系业务
QXWZ_SDK_ERR_MANUAL_BIND_REQUIRED	-303	需手动绑定账号,请联系业务
QXWZ_SDK_ERR_ACC_BEING_PROCESSED	-304	账号正在绑定中,无需处理
QXWZ_SDK_ERR_ACC_NOT_BIND	-307	账号尚未绑定,请联系业务
QXWZ_SDK_ERR_ACC_EXPIRED	-308	账号已过期,请联系业务
QXWZ_SDK_ERR_ACC_NOT_ENOUGH	-309	账号数不够,请联系业务
QXWZ_SDK_ERR_ACC_UNSUPPORT_OP	-310	当前账号不允许此操作,请联系业务
QXWZ_SDK_ERR_INVALID_KEY	-311	账号或秘钥错误,请联系业务
QXWZ_SDK_ERR_CALL_API_FAIL	-401	调用OPENAPI失败,请联系业务
QXWZ_SDK_ERR_INVALID_API_RESP	-402	OPENAPI响应报文错误,请联系业务

1.2.2 能力启动回调方法

该方法在能力启动结束后被调用，用于通知用户能力启动结果，方法定义如下：

1.2.2.1 方法定义

```
void onStart(int code, int capId);
```

1.2.2.2 参数说明

参数名	参数类型	说明
code	int	能力启动结果状态码，QXWZ_SDK_STAT_CAP_START_SUCC表示启动成功，其他表示启动失败
capId	int	能力ID

1.2.2.3 状态码说明表

状态码	值	说明
QXWZ_SDK_STAT_CAP_START_SUCC	101	能力启动成功
QXWZ_SDK_ERR_CAP_START_FAIL	-203	能力启动失败,建议重试

1.2.3 数据回调方法

1.2.3.1 方法定义

该方法在获取到差分数据时被调用，用于将差分数据传给用户，方法定义如下：

```
void onData(int dataType, byte[] data, int len);
```

1.2.3.2 参数说明

参数名	参数类型	说明
dataType	int	数据类型
data	byte[]	数据内容

- type值与data数据类型的对应关系表

type值	data数据类型	说明
QXWZ_SDK_DATA_TYPE_RAW_NOSR	const qxwz_uint8_t*	len的值为data指向的二进制数据的字节数

1.2.4 状态回调方法

该方法在SDK状态变更后被调用，用于通知用户SDK当前状态，方法定义如下：

1.2.4.1 方法定义

```
void onStatus(int code);
```

1.2.4.2 参数说明

参数名	参数类型	说明
code	int	SDK当前状态值，code为非负时表示正常状态，code为负时表示异常状态

1.2.4.3 状态码说明表

状态码	值	说明
QXWZ_SDK_STAT_OK	0	状态正常
QXWZ_SDK_ERR_NETWORK_UNAVAILABLE	-101	网络连接不可用
QXWZ_SDK_ERR_GGA_OUT_OF_SERVICE_AREA	-201	GGA超出服务范围
QXWZ_SDK_ERR_INVALID_GGA	-202	GGA格式错误
QXWZ_SDK_ERR_ACC_DUPLICATED	-313	账号重复登录
QXWZ_SDK_ERR_INTERNAL_ERROR	-505	SDK内部错误,请联系业务
QXWZ_SDK_ERR_SERV_FAULT	-506	服务端错误, 请联系业务
QXWZ_SDK_ERR_NOSR_SERVICE_STOP	-507	NOSR服务停止, 请重新启动NOSR能力
QXWZ_SDK_ERR_UNKNOWN	-999	未知异常

2. 终端激活回调接口

2.1 接口定义

用于将终端激活结果通知开发者，回调接口定义如下：

```
public interface IRtcmSDKActivateCallback {  
    void onResult(int code, List<ActivateResult> Activates);  
}
```

2.2. 终端激活回调方法说明

该方法在终端激活结束后被调用，用于通知用户终端激活结果，方法定义如下：

2.2.1 方法定义

```
void onResult(int code, List<ActivateResult> Activates);
```

2.2.2 参数说明

参数名	参数类型	说明
code	int	QXWZ_SDK_STAT_OK表示成功，具体能力的激活成功与否参见参数 Activates ;其他状态码表示具体失败的原因，此时参数 Activates 不可使用，具体状态码请看2.2.2.2说明
Activates	List	能力信息，仅在能力激活成功时该值为非空，ActivateResult的说明见 2.2.2.1

2.2.2.1 ActivateResult 说明

```
public final class ActivateResult {
    private int capId;    //能力ID
    private int result;    //该能力激活结果,QXWZ_SDK_STAT_CAP_ACT_SUCC表示成功,其他表示具体失败原因
}
```

2.2.2.2 状态码说明表

状态码	值	说明
QXWZ_SDK_STAT_OK	0	状态正常
QXWZ_SDK_ERR_NETWORK_UNAVAILABLE	-101	网络连接不可用
QXWZ_SDK_ERR_CAP_GET_CONF_FAIL	-204	获取能力信息失败,请联系业务
QXWZ_SDK_ERR_CAP_MANUAL_ACT_REQUIRED	-207	能力需要手动激活,请联系业务
QXWZ_SDK_ERR_CAP_SYSTEM_ERROR	-211	能力系统错误,请联系业务
QXWZ_SDK_ERR_CAP_NOT_INCLUDE	-212	当前账号未包含该能力,请联系业务
QXWZ_SDK_ERR_ACC_NOT_BIND	-307	账号尚未绑定,请联系业务
QXWZ_SDK_ERR_ACC_EXPIRED	-308	账号已过期,请联系业务
QXWZ_SDK_ERR_ACC_UNSUPPORT_OP	-310	当前账号不允许此操作,请联系业务
QXWZ_SDK_ERR_INVALID_KEY	-311	账号或密钥错误,请联系业务
QXWZ_SDK_ERR_CALL_API_FAIL	-401	调用OPENAPI失败,请联系业务
QXWZ_SDK_ERR_INVALID_API_RESP	-402	OPENAPI响应报文错误,请联系业务

3. 获取坐标系回调接口

3.1 接口定义

用于将获取坐标系结果通知开发者，回调接口定义如下：

```
public interface IRtcmsdkGetCoordSysCallback {
    void onResult(int statusCode, CoordSysInfo CoordInfo);
}
```

3.2. 获取坐标系回调方法说明

该方法在获取坐标系结束后被调用，用于通知获取坐标系结果，方法定义如下：

3.2.1 方法定义

```
void onResult(int statusCode, CoordSysInfo CoordInfo);
```

3.2.2 参数说明

参数名	参数类型	说明
statusCode	int	坐标系查询执行结果，QXWZ_SDK_STAT_OK表示成功，其他表示具体失败原因，详见3.2.2.2
CoordInfo	CoordSysInfo	坐标系信息， 字段含义见3.2.2.1的说明

3.2.2.1 CoordSysInfo 说明

```
public final class Coord {
    private int index;           //坐标系索引,设置坐标系时需传入该值
    private int port;           //坐标系端口
    private String name;        //坐标系名
}

public final class CoordSysInfo {
    private List<Coord> coords;  //包含所有坐标系信息
    private int currCoordSysIndex; //当前坐标系索引
}
```

3.2.2.2 状态码说明表

状态码	值	说明
QXWZ_SDK_STAT_OK	0	状态正常
QXWZ_SDK_ERR_NETWORK_UNAVAILABLE	-101	网络连接不可用
QXWZ_SDK_ERR_ACC_NOT_BIND	-307	账号尚未绑定,请联系业务
QXWZ_SDK_ERR_ACC_EXPIRED	-308	账号已过期,请联系业务
QXWZ_SDK_ERR_ACC_UNSUPPORT_OP	-310	当前账号不允许此操作,请联系业务
QXWZ_SDK_ERR_INVAL_KEY	-311	账号或密钥错误,请联系业务
QXWZ_SDK_ERR_CALL_API_FAIL	-401	调用OPENAPI失败,请联系业务
QXWZ_SDK_ERR_INVAL_API_RESP	-402	OPENAPI响应报文错误,请联系业务

4. 设置坐标系回调接口

4.1 接口定义

用于将设置坐标系结果通知开发者，回调接口定义如下：

```
public interface IRtcmSDKSetCoordCallback {
    void onResult(int statusCode);
}
```

4.2. 设置坐标系回调方法说明

该方法在设置坐标系结束后被调用，用于通知设置坐标系结果，方法定义如下：

4.2.1 方法定义

```
void onResult(int statusCode);
```

4.2.2 参数说明

参数名	参数类型	说明
statusCode	int	坐标系设置执行结果，QXWZ_SDK_STAT_OK表示成功，其他表示具体失败原因，详见4.2.2.1

4.2.2.1 状态码说明表

状态码	值	说明
QXWZ_SDK_STAT_OK	0	状态正常
QXWZ_SDK_ERR_NETWORK_UNAVAILABLE	-101	网络连接不可用
QXWZ_SDK_ERR_ACC_NOT_BIND	-307	账号尚未绑定,请联系业务
QXWZ_SDK_ERR_ACC_EXPIRED	-308	账号已过期,请联系业务
QXWZ_SDK_ERR_ACC_UNSUPPORT_OP	-310	当前账号不允许此操作,请联系业务
QXWZ_SDK_ERR_INVALID_KEY	-311	账号或密钥错误,请联系业务
QXWZ_SDK_ERR_CALL_API_FAIL	-401	调用OPENAPI失败,请联系业务
QXWZ_SDK_ERR_INVALID_API_RESP	-402	OPENAPI响应报文错误,请联系业务

五、常见问题

1. 鉴权失败怎么办？

首先请关注auth方法的返回值，只有返回值为QXWZ_SDK_STAT_OK(0)时才会真正执行鉴权逻辑，若返回值不为0请根据状态码排查原因。当返回值为0时无论鉴权成功与否都会通过鉴权回调方法onAuth通知开发者，请关注code的值并决定是否进行重试，例如若code的值为QXWZ_SDK_ERR_NETWORK_UNAVAILABLE(-101)时表示鉴权是由于网络问题导致失败的，此时需要重试；若code的值为账号相关时请联系千寻业务人员。

2. 能力启动失败怎么办？

首先请关注start方法的返回值，只有返回值为QXWZ_SDK_STAT_OK(0)时才会真正执行能力启动逻辑，若返回值不为0请根据状态码排查原因。当返回值为0时无论能力启动成功与否都会通过能力启动回调方法onStart通知开发者，请关注code的值并决定是否进行重试，例如若code的值为QXWZ_SDK_ERR_CAP_START_FAIL(-203)时请判断当前网络是否断开，若断开请重试，若未断开说明SDK出现未知错误，请联系千寻业务人员。

3. 如何重置SDK?

调用cleanup接口后再调用init接口即可

4. 什么时候开始调用sendGga?

在NOSR能力启动成功之后，能力启动结果在能力启动回调方法中通知开发者，code的值为QXWZ_SDK_STAT_CAP_START_SUCC(201)时表示启动成功，其他值表示启动失败，开发者可通过一个全局标识位记录能力是否启动成功。

5. 多久调用一次sendGga?

建议调用间隔不小于1s。

6. 循环调用sendGga期间断网了怎么办?

SDK会自动尝试重连直到网络重新连接或SDK退出，无需做任何处理。