

腾讯 Ntrip SDK 开发指南

目 录

1. 版本变更历史	3
2. SDK 概述.....	3
2.1 SDK 功能描述	3
2.2 接入流程与参数配置	3
2.2.1 接入流程	3
2.2.2 参数配置	4
3. SDK API 接口	5
3.1 TX_NTRIP_SETUSERAGREEPRIVACY	5
3.2 TX_NTRIP_START.....	5
3.3 TX_NTRIP_STOP	6
3.4 TX_NTRIP_SEND_NMEA	6
3.5 TX_NTRIP_START_LOG	6
4. ANDROID 平台 API 接口	7
4.1 TENCENTNTRIPSETTINGS	7
4.2 TENCENTNTRIPMANAGER	7
4.3 TENCENTRTCMListener	8
4.4 TENCENTNTRIPLOGListener	8
5. LINUX 平台 SDK 集成示例	9
5.1 SDK 目录结构	9
5.2 代码示例	9
6. ANDROID 平台 SDK 集成示例.....	11
6.1 构建工程.....	11
6.2 代码示例.....	11

1. 版本变更历史

版本	时间	内容
1.0	20230731	初始版本
1.1	20230804	增加 Android 版本
1.2	20230913	增加隐私合规接口

2. SDK 概述

2.1 SDK 功能描述

RTK 定位是目前常用的 GNSS 高精度定位方式之一，其实现不仅依赖终端定位算法，还需要基准站的原始观测数据用于差分定位。该部分的基准站数据来自于网络 RTK 服务，腾讯位置面向不同行业和终端设备，提供了不同方式接入腾讯网络 RTK 服务，帮助终端实现高精度差分定位。根据终端是否已集成 Ntrip 协议，可分为两类方式。

第一类 Ntrip 账号接入，该类方式需要定位终端支持标准的 Ntrip 协议，只需要获取腾讯网络 RTK 账号，即可进行终端 RTK 定位，无需额外的集成工作。

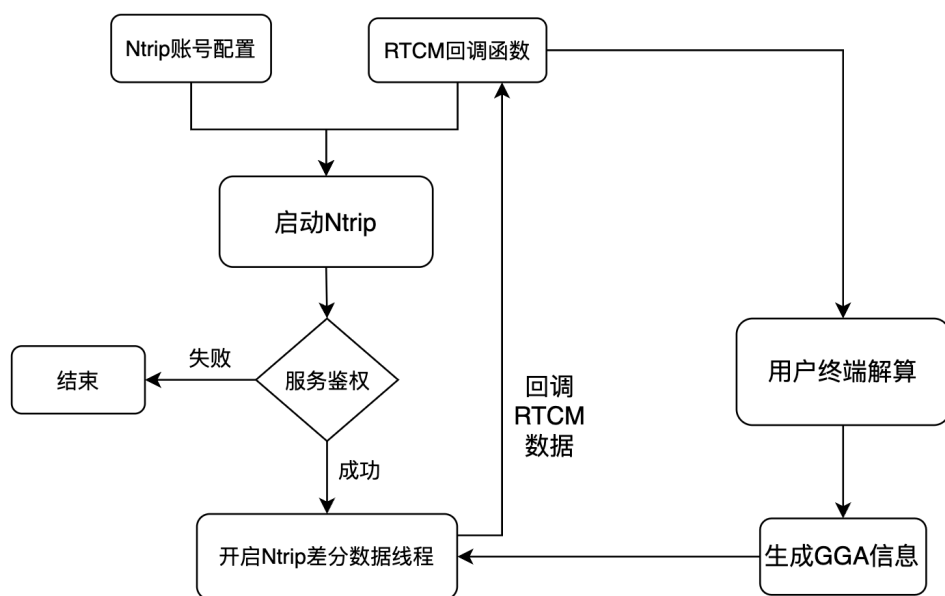
第二类 Ntrip SDK 接入，该类方式适合各类硬件开发商，通过适配腾讯位置提供的 Ntrip SDK，可获得网络 RTK 服务，集成方式扩展性和安全性都更好，但是需要根据硬件平台的不同，适配 SDK。

本文档主要说明第二类接入方式，目前支持 Android、Linux 和 Windows 环境，其他操作环境如嵌入式环境，可能需要额外的适配工作，请联系我们获取。

2.2 接入流程与参数配置

2.2.1 接入流程

用户通过移植腾讯 NTRIP SDK 库文件，调用封装好的 API 接口，传入正确 NTRIP 账号配置，上报自身概略 GGA 位置信息，即可接入腾讯网络 RTK 服务，获取到差分数据。主要流程如下图：



2.2.2 参数配置

该部分主要是对 Ntrip 账号进行配置，部分配置项如端口、挂载点需要根据使用需求进行配置，配置项的详细信息也可参考腾讯位置服务官网：<https://lbs.qq.com/rtk/>

参数名称	说明
hostname	网络 RTK 域名，默认为 cors.tencent.com
port	接入端口，默认使用 8002 和一般地图底图框架一致
mountPoint	接入挂载点，可根据终端支持的卫星系统进行配置
brdint	RTCM 播发间隔，默认使用 1s
username	网络 RTK 账号
password	网络 RTK 密码
device_id	设备 ID，需保持设备 ID 唯一
device_type	设备类型

端口配置说明：

端口号	端口8001	对应坐标框架ITRF2014参考历元2020.0
	端口8002	对应坐标框架WGS84参考历元2005.0
	端口8003	对应坐标框架CGCS2000参考历元2000.0
	端口8004	对应坐标框架ITRF2014参考历元2018.0
	端口8005	对应坐标框架WGS84参考历元2010.0

挂载点配置说明：

挂载点	RTCM32_GRC	RTCM32_GNSS	RTCM32_GNSS2	RTCM32_GRECJ	RTCM32_GRECJ2
卫星系统	GRC	GREC	GREC	GRECJ	GRECJ
配置	3星6频	4星11频	4星13频	5星14频	5星16频
频点	GPS L1/L2 BDS B1I/B2I GLONASS G1/G2	GPS L1/L2/L5 BDS B1I/B2I/B3I GLONASS G1/G2 Galileo E1/E5a/E5b	GPS L1/L2/L5 BDS B1I/B2I/B3I/B1C/B2a GLONASS G1/G2 Galileo E1/E5a/E5b	GPS L1/L2/L5 BDS B1I/B2I/B3I GLONASS G1/G2 Galileo E1/E5a/E5b QZSS L1/L2/L5	GPS L1/L2/L5 BDS B1I/B2I/B3I/B1C/B2a GLONASS G1/G2 Galileo E1/E5a/E5b QZSS L1/L2/L5

3. SDK API 接口

3.1 tx_ntrip_setUserAgreePrivacy

功能描述：设置是否同意用户隐私协议，正常使用 Ntrip，需要同意用户隐私协议

函数定义：void tx_ntrip_setUserAgreePrivacy(int isAgree)

函数参数：是否同意用户隐私协议，0 为不同意，1 为同意

返回值：无

3.2 tx_ntrip_start

功能描述：传入 Ntrip 配置参数和 RTCM 回调函数，启动腾讯 Ntrip

函数定义：int tx_ntrip_start(tx_ntrip_config* ntrip_settings, tx_ntrip_rtcm_callback rtcm_callback)

函数参数：Ntrip 配置参数结构体，RTCM 回调函数

返回值：1：OK，其他：FAIL

tx_ntrip_config 结构体定义

参数编号	数据格式	成员名称	参数描述 (Units)
1	char	hostname[128]	Ntrip 服务域名
2	int	port	Ntrip 接入端口
3	char	moutPoint	Ntrip 接入挂载点
4	int	brdint	Ntrip 播发间隔
5	char	username[32]	Ntrip 账号
6	char	password[32]	Ntrip 密码
7	char	device_id[128]	设备 ID
8	char	device_type[128]	设备类型

tx_ntrip_rtcm_callback 回调函数定义：typedef void (*tx_ntrip_rtcm_callback)(const char *buff, int len);

3.3 tx_ntrip_stop

功能描述：停止腾讯 Ntrip

函数定义：int tx_ntrip_stop()

函数参数：无

返回值：1：OK，其他：FAIL

3.4 tx_ntrip_send_nmea

功能描述：更新 NMEA 消息到腾讯 NTRIP

函数定义：int tx_ntrip_send_nmea(const char* nmea)

函数参数：nmea 字符串

返回值：1：OK，其他：FAIL

3.5 tx_ntrip_start_log

功能描述：开启 NTRIP 日志记录

函数定义：int tx_ntrip_start_log(TX_NTRIP_LOG_LEVEL level, tx_ntrip_log_callback log_callback)

函数参数：日志记录级别，日志回调函数

返回值：1：OK，其他：FAIL

TX_NTRIP_LOG_LEVEL 定义：

枚举值	字段说明
0: TX_NTRIP_LOG_DEBUG	日志级别为 debug
1: TX_NTRIP_LOG_INFO	日志级别为 info
2: TX_NTRIP_LOG_WARN	日志级别为 warning
3: TX_NTRIP_LOG_ERROR	日志级别为 error
4: TX_NTRIP_LOG_NONE	不输出日志

3.6 tx_ntrip_stop_log

功能描述：停止 NTRIP 日志记录

函数定义：int tx_ntrip_stop_log()

函数参数：无

返回值：1：OK，其他：FAIL

3.7 tx_ntrip_get_version

功能描述：获取 NTRIP SDK 版本号

函数定义：void tx_ntrip_get_version(char* version)

函数参数：接收 NtripSDK 版本号字符串

返回值：无

4. Android 平台 API 接口

Android 提供 TencentNtripSettings 类用于配置 NTRIP, 两个 Listener 接口用于处理 RTCM 数据和日志, TencentNtripManager 类用于管理 Ntrip 接入的所有操作。

4.1 TencentNtripSettings

类型	方法名	说明
TencentNtripSettings	setDeviceId(String deviceId)	设置设备 device ID
TencentNtripSettings	setDeviceType(String deviceType)	设置设备类型
TencentNtripSettings	setHostName(String hostname)	设置差分服务域名
TencentNtripSettings	setInterval(int interval)	设置播发间隔
TencentNtripSettings	setMountPoint(String mountPoint)	设置挂载点
TencentNtripSettings	setPassword(String password)	设置密码
TencentNtripSettings	setUserName(String username)	设置用户名
TencentNtripSettings	setPort(int port)	设置端口
String	getDeviceId()	获取设备 ID
String	getDeviceType()	获取设备类型
String	getHostName()	获取差分服务域名
int	getInterval()	获取播发间隔
String	getMountPoint()	获取挂载点
String	getPassword()	获取密码
String	getUserName()	获取用户名
int	getPort()	获取端口

4.2 TencentNtripManager

类型	方法名	说明
static TencentNtripManager	getInstance(Context context)	获取 TencentNtripManager 实例
void	setUserAgreePrivacy(boolean isAgree)	设置用户是否同意隐私协议，需要用户同意后才能正常使用相关功能
String	getVersion	获取版本号
void	addNtripLogListener(TencentNtripLogListener listener)	增加 Ntrip Log 监听
void	addRtcmlListener(TencentRtcmlListener listener)	增加 RTCM 数据监听
void	removeNtripLogListener(TencentNtripLogListener listener)	移除 Ntrip Log 监听
void	removeRtcmlListener(TencentRtcmlListener listener)	移除 RTCM 数据监听
int	startNtrip(TencentNtripSettings settings)	开启 NTRIP，相关配置可参考之前的描述
int	sendNmea(String nmea)	发送 GGA 信息
void	startLog(int level)	开启日志记录， level=0, 日志级别为 Debug; level=1, 日志级别为 Info; level=2, 日志级别为 Warn; level=3, 日志级别为 Error; level=4, 日志级别为 None
void	stopNtrip()	停止 NTRIP
void	stopLog()	停止日志记录

4.3 TencentRtcmlListener

类型	方法名	说明
void	onRtcmlCallback(byte[] rtcml)	Rtcml 数据回调

4.4 TencentNtripLogListener

类型	方法名	说明
void	onNtripLogListener(char[] log)	日志数据回调

5. Linux 平台 SDK 集成示例

5.1 SDK 目录结构



用户在自己工程中导入 txntrip_interface 头文件，默认提供 x86_64 架构的库文件，如果需要其他 Linux 的架构可联系我们获取。使用方法可以参考 5.2 中的使用示例，搭建自己的应用，值得注意的是需要同意用户隐私协议后，才能正常使用功能。

5.2 代码示例

```

1  #include <stdio.h>
2  #include <unistd.h>
3  #include "txntrip/txntrip_interface.h"
4
5  void init_config(tx_ntrip_config* p_ntrip_config);
6  void start_rtcn_test();
7  static void rtcn_callback(const char *buff, int len)
8  {
9      printf("CALLBACK [rtcn] recv, len:%d\n", len);
10 }
11
12 int main()
13 {
14     // 测试获取rover_rtcn
15     start_rtcn_test();
16
17     while (1) {
18         sleep(1);
19     }
20     return 0;
21 }
22
  
```

```

23 // ntrip rtcm 测试
24 void start_rtcm_test() {
25 //设置用户隐私权限
26 tx_ntrip_setUserAgreePrivacy(1);
27 // 启动日志模块
28 tx_ntrip_start_log(TX_NTRIP_LOG_DEBUG, log_callback);
29 tx_ntrip_config ntrip_config;
30 // 初始化ntrip配置
31 init_config(&ntrip_config);
32 tx_ntrip_start(&ntrip_config, rtcm_callback);
33 char str_buf[256] = {0};
34 // 传入gga
35 sprintf(str_buf, "$GNGGA,030529.00,3025.86189669,N,11417.18133192,E,1,29,0.61,29.
36 7846,M,-14.2709,M,,*52\r\n");

```

```

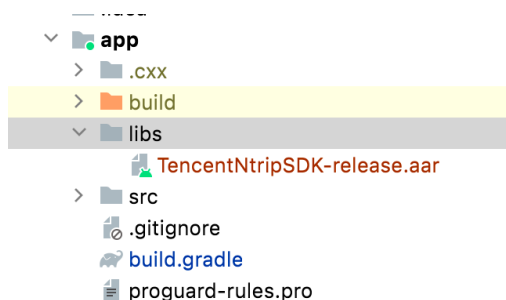
37
38 tx_ntrip_send_nmea(str_buf);
39 sleep(20);
40 tx_ntrip_stop();
41 tx_ntrip_stop_log();
42 }
43
44 // 初始化config参数, 以下均为必填项
45 void init_config(tx_ntrip_config* p_ntrip_config)
46 {
47 sprintf(p_ntrip_config->hostName, "cors.tencent.com"); // 请求域名
48 p_ntrip_config->port = 8002; // 请求端口
49 p_ntrip_config->brdint = 1; // rtcm播发间隔
50 sprintf(p_ntrip_config->mountPoint, "RTCM32_GRECJ2"); // 挂载点
51 sprintf(p_ntrip_config->userName, "*****"); // 用户名
52 sprintf(p_ntrip_config->password, "*****"); // 密码
53 sprintf(p_ntrip_config->device_id, "*****"); // 设备id
54 sprintf(p_ntrip_config->device_type, "*****"); // 设备类型
55 }

```

6. Android 平台 SDK 集成示例

6.1 构建工程

(1) 将 aar 包放入 libs 依赖中，配置 gradle



在 build.gradle 文件的 dependencies 中增加配置

```
1 implementation fileTree(includes: ['*.jar', '*.aar'], dir: 'libs')
```

(2) 代码混淆配置

如果需要混淆您的工程，请在 module 里找到 proguard-rules.pro 文件，添加如下混淆脚本：

```
1 -keepattributes *Annotation*
2 -keepclasseswithmembernames class * {
3     native <methods>;
4 }
5 -keep class c.t.**{*;}
6 # sdk api
7 -keep class com.tencent.map.ntrip.**{*;}
8 -keep class com.tencent.map.ntrip.impl.Ntrip_JNI {*;}
9 -dontwarn c.t.**
```

(3) 权限配置

差分服务请求需要开启网络请求权限

```
1 <!-- 访问网络，某些位置信息需要从网络服务器获取 -->
2 <uses-permission android:name="android.permission.INTERNET"/>
```

6.2 代码示例

(1) 设置用户隐私接口，获取 TencentNtripManager 实例

```
1 TencentNtripManager ntrip=TencentNtripManager.getInstance(getApplicationContext());
2 ntrip.setUserAgreePrivacy(true); //用户协议设置为同意后，才能正常使用
```

(2) 配置 Ntrip 参数

```
1 TencentNtripSettings settings = new TencentNtripSettings();
2 settings.setHostName("cors.tencent.com").setPort(8002).setMountPoint("****")
3 .setUserName("****").setPassword("****")
4 .setDeviceId("****").setDeviceType("****")
5 .setInterval(1);
```

其中缺省的部分需要用户根据账号和实际使用要求进行配置，参数说明可以参考开发文档中 2.2.2 的账号配置。其中对于当前账号，每台设备的 DeviceId 需要配置唯一，不能重复。DeviceType 根据实际设备类型进行配置，例如手机配置为 Phone 等。

(3) 注册监听

用户可通过实现 RTCM 和 NTRIP Log 的监听，来对接收的差分数据进行处理，并对产生的日志进行分析。

```
1 public class rtcmlistener implements TencentRtcmListener{
2     @Override
3     public void onRtcmCallback(byte[] rtcm){
4         // do your work
5     }
6 }
7 public class ntripLoglistener implements TencentNtripLogListener{
8     @Override
9     public void onNtripLogListener(char[] log){
10         // do your work
11     }
12 }
13 ntrip.addRtcmListener(new rtcmlistener());
14 ntrip.addNtripLogListener(new ntripLoglistener());
```

(4) 启动日志和 NTRIP

日志级别和 Ntrip 配置参考上面的描述。

```
1 ntrip.startLog(1);  
2 ntrip.startNtrip(settings);
```

(5) 发送 GGA

用户根据自身位置，调用接口发送 GGA 信息。该示例只发送了一次 GGA 信息，常规情况下 GGA 信息需要定频发送。

```
1 ntrip.sendNmea("$GNGGA,084800.00,2425.66614388,N,10925.04991572,E,4,28,0.3,100.8  
14,M,0.000,M,1.0,3350*67\r\n");
```

(6) 停止日志和 NTRIP

```
1 ntrip.stopLog();  
2 ntrip.stopNtrip();
```