

千寻云踪Android SDK

开发指南

(verison V1.1)

千寻位置网络有限公司

2016 年 09 月 上海

法律声明

版权所有© 2016 , 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、传播或与其它产品捆绑使用、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申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

目录

第一章 概述	5
1.1 产品简介	5
1.2 功能特点	5
1.2.1 实体管理	5
1.2.2 位置管理	5
1.2.3 配置管理	6
第二章 开发指南	7
2.1 准备工作	7
2.1.1 入驻千寻	7
2.1.2 创建应用	7
2.1.3 下载 SDK	7
2.1.4 配置环境	7
2.1.5 添加权限	8
2.1.6 结束配置	8
2.1.7 设置 appkey 和 appSecret	8
2.1.8 查看类参考	8
2.2 新手上路	8
2.2.1 上传位置	9
2.2.2 业务管理	12
2.2.3 C 类应用	13
2.2.4 更多功能	14

第三章 更新日志	15
3.1 千寻云踪 Android SDK V1.0 发布上线	15
3.1.1 发布时间	15
3.1.2 版本描述	15
3.1.3 详细描述	15
3.2 千寻云踪 Web API V1.1 发布上线	15
3.2.1 发布时间	15
3.2.2 版本描述	15
3.2.3 详情描述	15

第一章 概述

1.1 产品简介

千寻云踪 Android SDK 是千寻位置网面向 Android4.3.0 及以上版本的设备提供的一套位置开发调用接口。基于高并发、分布式、流式计算等技术，提供海量终端位置数据的上传、存储、实时追踪、历史轨迹查询、轨迹抽稀等各种功能；打造满足不同行业及应用需求的云服务，使得位置数据的接入及后续处理变得『易如反掌』。

该 SDK 适用于管理各种移动的人、车、物等实体，例如商用车、乘用车、手机、可穿戴设备等。基于该 SDK 进行二次开发，既可以上传 Android 设备的位置，也可以通过简单的实体管理、位置管理、配置管理，快速搭建功能强大的业务系统。

要使用该 SDK，需要申请 appKey 和 appSecret，请在千寻位置网(qxwz.com)注册账号，并按照帮助文档完成注册、认证、应用创建、appKey 和 appSecret 获取工作，具体流程可见开发指南的准备工作章节。

1.2 功能特点

1.2.1 实体管理

- (1) 实体操作：支持实体的创建、删除、修改、查询
- (2) 自定义扩展：支持自定义扩展字段，满足个性化的业务诉求

1.2.2 位置管理

- (1) 自定义扩展：轨迹点支持自定义扩展字段，满足个性化的业务需要
- (2) 位置上传：支持上传单个轨迹点、批量上传轨迹点；支持按用户设置的采集频率和上传频率自动完成数据的采集和上传；支持用户自主完成位置数据及属性信息的获取。

(3) 实时位置 : 支持查看指定实体列表的实时位置

(4) 历史轨迹 : 支持查询指定实体连续 10 天的历史轨迹

1.2.3 配置管理

(1) 抽稀配置 : 可以选择是否抽稀、设置抽稀延迟 , 抽稀程度。轨迹抽稀可以在保持轨迹基本形态前提下 , 降低存储成本、提高查询效率。

(2) 存储配置 : 自定义数据存储周期。

(3) 采集频率和上传频率配置 : 支持用户自主设置位置信息及其属性信息 , 系统自动按设置的频率进行数据采集和上传。

第二章 开发指南

2.1 准备工作

2.1.1 入驻千寻

开发者需要入驻千寻位置网，才能够使用千寻云踪 Android SDK。请在千寻官网(qxwz.com)注册用户，注册成功并通过个人/企业用户认证后，便可成为千寻位置网的开发者。

2.1.2 创建应用

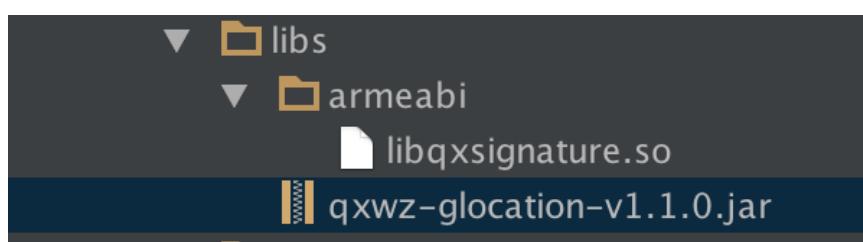
在千寻位置网官网的管理控制台成功创建应用后，会自动生成该应用的 appKey&appSecret；appKey 和 appSecret 是应用的凭证信息，也是调用 API/SDK 的唯一凭证，请务必妥善保管。

2.1.3 下载 SDK

请千寻位置网官网下载千寻云踪 Android SDK，根据下文配置环境，便可开启您的体验之旅。

2.1.4 配置环境

将下载的千寻云踪 Android SDK 包 (qxwz-glocation-v1.1.0.jar)，拷贝到工程的 libs 根目录下；将下载的千寻签名 so 拷贝到工程的 libs 下得 armeabi 目录下，拷贝完成的工程目录如图所示：



如果您使用的是 Android Studio，请在您项目下的 build.gradle 中添加如下代码：

```
sourceSets {  
    main {  
        jni.srcDirs = []  
        jniLibs.srcDirs = ['libs']// include .so file into apk  
    }  
}
```

2.1.5 添加权限

在工程的"AndroidManifest.xml"文件中添加网络权限，

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

如需要上传位置等操作，请按需添加位置等相关权限。

2.1.6 结束配置

clean 工程，结束配置

2.1.7 设置 appkey 和 appSecret

您需要设置 appKey 和 appSecret 后，才有权限访问对应的 API 接口。appKey 和 appSecret 是访问应用的唯一凭证信息，请务必妥善保管，避免泄露。

您首先需要使用包 com.qx.wz.glocationservice 中的 WzGLocationFactory 类，调用 getWzConfigManager、getWzEntityManager、getWzEntityPointManager 方法，传入 appKey 和 appkeySecet，获取三个 Manager 类后，然后便可调用对应的接口。

2.1.8 查看类参考

您可以查看千寻云踪 Android SDK 类参考，详细了解每个接口详情，帮助你快速开发。

2.2 新手上路

Android 设备既是一种定位终端，同时也是一台管理设备。因此使用千寻云踪 Android SDK，一般应用于如下三种场景：

(1) 位置上传：仅通过 Android 设备上传位置信息，该位置信息既可来自于 Android 设备，也可以来自于 Android 设备的配套设备，例如通过蓝牙连接的定位设备。此类场景中，Android 仅上传位置信息，自身不承载其他业务。

(2) 业务管理：搭建业务管理系统，进行实体管理、位置管理和配置管理。此类场景中，Android 设备仅承载业务管理系统，实体位置的上传工作交由各个实体独立完成。

(3) C 端应用：一方面 Android 设备自身上传位置，同时承载 C 类的业务系统，例如运动健身等场景。开发此类应用，在千寻云踪的基础上，需要开发者自己配套搭建用户体系等。

新手上路为您提供最基本的使用流程，针对不同的场景，提供最简单的操作步骤，帮助你快速完成开发工作。

2.2.1 上传位置

千寻云踪提供两种位置上传方式，一种是自主上传方式，即仅仅提供上传接口，定位信息获取、上传频率、数据补传等设置需要您单独完成；另一种是半自动上传方式，千寻云踪支持您设置采集频率、上传频率、支持弱网数据补传。

2.2.1.1 自主上传

如果您需要通过 Android 设备上传位置，请务必提前完成实体登记工作，获取实体名称后，才可以调用接口上传到云端进行存储和处理。对于定位信息的获取、上传频率的设置等工作则需要您单独开发。

实体的登记工作既可以直接调用创建实体的接口自行创建，也可以在配套的业务系统中事先完成，然后将实体名称分配给指定的 Android 设备。

创建实体相关接口：

(1) 创建实体

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityManager	createEntity

上传位置相关接口：

(1) 上传单个轨迹点：

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityPointManager	addPoint

(2) 上传多个轨迹点

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityPointManager	addPoints

2.2.1.2 半自动上传

千寻云踪支持您配置采集频率和上传频率，自主采集数据，并支持弱网数据补传功能，减少您的开发工作量。

(1) 设置采集频率

支持用户自定义采集频率和上传频率，默认为 5s 采集一次数据，60s 上传一次数据；上传时间间隔需要大于等于采集间隔，且不得大于采集间隔的 200 倍。

相关接口：

包名	类名	方法
com.qx.wz.glocationservice.manager	WzConfigManager	setSDKDataCollectingConfig

(2) 信息采集

千寻云踪仅定时触发采集操作，实体的位置及属性信息的采集工作需要您单独完成。您可以直接调用程序获取手机的位置，也可集成高德或百度的定位 SDK，其他的属性信息可按需采集。

相关接口：

包名	类名	方法
com.qx.wz.glocationservice	WzGLocationManager	getWzEntityPointManager

示例代码：

您需要单独实现 WzEntityPointInfoProvider 类, 在 getEntityPoint 接口中完成位置及其属性信息的采集工作。

```
mWzEntityPointManager = WzGLocationFactory.getWzEntityPointManager(this,
    appKey, appSecret, entityName,
    new WzEntityPointInfoProvider() {
        @Override
        public EntityPoint getEntityPoint() {
            EntityPoint entityPoint = new EntityPoint();
            entityPoint.setEntityName(entityName);
            entityPoint.setGnssTime(new Date().getTime());
            if (mAMapLocation != null) {
                entityPoint.setLon(mAMapLocation.getLongitude());
                entityPoint.setLat(mAMapLocation.getLatitude());
            }
            return entityPoint;
        },
        new WzGLRespondListener<WzSDKResult>() {

            @Override
            public void onSuccess(WzSDKResult sdkResult) {
                // TODO: do something when return successful result
                Log.Toast("initial onSucess()" + sdkResult.getCode());
            }
            @Override
            public void onCancel() {
                // TODO: do something when cancel createEntity
                Log.d(TAG, "initial onCancel()");
            }
            @Override
            public void onError(int code, String message, Throwable e) {
                // TODO: do something when error happens
                Log.Toast("initial onError()" + code + " - " + message);
            }
        });
    };
```

(3) 弱网数据补传功能

千寻云踪支持弱网数据补传功能, 支持将数据保存在本地, 待网络恢复后重新启动上传。目前支持用户在本地存储 10000 条数据, 超过这个值后, 采集的新数据将替换本地最老的数据。

2.2.2 业务管理

如果您基于 Android 设备管理实体、查看实体位置和历史轨迹，那么您只需要简单完成如下三步，便可快速搭建一个完整的业务系统。

首先您可以进行实体的增、删、改、查工作，这个实体可以是任意的人、车、物，例如商用车、乘用车、可穿戴设备、GPS 跟踪器等。在同一个应用下面，实体名称全局唯一。

实体相关接口：

(1) 创建实体

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityManager	createEntity

(2) 删除实体

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityManager	deleteEntity

(3) 更新实体

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityManager	updateEntity

(4) 查询实体

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityManager	queryEntity

其次你需要在配套的业务后台中完成对应实体的位置上传工作，协议的解析、上传频率的设置需要您单独完成，然后调用千寻云踪的 Web API 接口上传到千寻后台。

上传位置相关接口：

(1) 上传单个轨迹点 :

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityPointManager	addPoint

(2) 上传多个轨迹点

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityPointManager	addPoints

最后 , 将实体的位置源源不断地上传到云平台上后 , 调用查看实时位置和历史轨迹的接口 , 配合地图便可显示实体的实时位置和历史轨迹。

相关接口 :

(1) 查看实时位置

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityPointManager	locate

(2) 查看历史轨迹

包名	类名	方法
com.qx.wz.glocationservice.manager	WzEntityPointManager	queryPoints

2.2.3 C 类应用

用户的 Android 设备既上传位置信息 , 同时作为 C 类的业务系统。上传位置请参见 2.2.1 和 2.2.2 章节 , 鉴于 C 类用户通常只能查看自身终端 , 因此需要开发者配套搭建用户体系 , 然后调用实时位置查看接口和历史轨迹等接口 , 查询 Android 设备的实时状态和历史信息。

位置查看相关接口 :

(1) 查看实时位置

包名	类名	方法
com.qx.wz.glocationserver.manager	WzEntityPointManager	Locate

(2) 查看历史轨迹

包名	类名	方法
com.qx.wz.glocationserver.manager	WzEntityPointManager	queryPoints

2.2.4 更多功能

如果您希望了解更多千寻云踪 Android SDK 的功能，或者需要查看接口的细节，请查看千寻云踪 Android SDK 类参考文档。

第三章 更新日志

3.1 千寻云踪 Android SDK V1.0 发布上线

3.1.1 发布时间

2016 年 6 月 21 日

3.1.2 版本描述

实现位置数据采集、存储和开放的主流程，支持实体管理、位置管理和配置管理，对应千寻云踪 Web API V1.1 版本。

3.1.3 详细描述

- (1) 实体管理：支持实体的创建、删除、更新和查询；实体支持自定义字段。
- (2) 位置管理：支持上传、批量上传轨迹点、查询历史轨迹；轨迹点支持自定义字段。
- (3) 配置管理：支持存储配置、抽稀配置，支持查询配置详情。

3.2 千寻云踪 Web API V1.1 发布上线

3.2.1 发布时间

2016 年 9 月 22 日

3.2.2 版本描述

支持自定义采集频率和上传频率，支持弱网数据补传。

3.2.3 详情描述

- (1) 采集频率和上传频率设置：支持用户自定义设置数据采集频率和上传频率。
- (2) 信息采集：支持用户自主获取设置位置及其属性信息，SDK 按设置的频率进行数据采集和上传。
- (3) 弱网数据补传：支持在网络情况不好或无网络的情况下，将数据保存在本地；待网络恢复后，将自动上传数据到服务器。

