

千寻见微 Android SDK 开发指南

V1.2.0

千寻位置网络有限公司

2016 年 12 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	8
1.1 产品简介	8
1.2 千寻见微服务结构图	9
第二章 开发指南	10
2.1 开发准备	10
2.1.1 注册为开发者	10
2.1.2 创建应用	10
2.1.3 下载 SDK	10
2.2 接入流程图	10
2.3 配置工程	11
2.3.1 Jar 包引入	11
2.3.2 添加权限	11
2.3.2 配置结束	12
第三章 SDK 接口说明	13
3.1 SDK 文件引用	13
3.2 获取 WzOrganizationManager	13
3.2.1 接口定义	13
3.2.2 示例代码	13
3.3 获取 WzBuildingManager	14
3.3.1 接口定义	14

3.3.2 示例代码	14
3.4 获取 WzMonitorPointManager	15
3.4.1 接口定义	15
3.4.2 示例代码	15
3.5 新增组织机构	15
3.5.1 接口定义	15
3.5.2 回调结果	16
3.5.3 示例代码	16
3.6 通过组织机构 ID 查询组织机构信息	17
3.6.1 接口定义	17
3.6.2 回调结果	17
3.6.3 示例代码	17
3.7 查询组织机构列表	18
3.7.1 接口定义	18
3.7.2 回调结果	18
3.7.3 示例代码	18
3.8 更新组织机构信息	19
3.8.1 接口定义	19
3.8.2 回调结果	19
3.8.3 示例代码	20
3.9 删除指定组织机构信息	20
3.9.1 接口定义	20

3.9.2 回调结果	21
3.9.3 示例代码	21
3.10 添加建筑物	22
3.10.1 接口定义	22
3.10.2 回调结果	22
3.10.3 示例代码	22
3.11 通过建筑物 id 查询建筑物	23
3.11.1 接口定义	23
3.11.2 回调结果	23
3.11.3 示例代码	23
3.12 根据组织机构 id 查询建筑物	24
3.12.1 接口定义	24
3.12.2 回调结果	24
3.12.3 示例代码	25
3.13 更新建筑物信息	25
3.13.1 接口定义	25
3.13.2 回调结果	26
3.13.3 示例代码	26
3.14 根据建筑物 id 删除建筑物	27
3.14.1 接口定义	27
3.14.2 回调结果	27
3.14.3 示例代码	27

3.15 新增监测点	28
3.15.1 接口定义	28
3.15.2 回调结果	28
3.15.3 示例代码	28
3.16 根据监测点 id 查询监测点信息	29
3.16.1 接口定义	29
3.16.2 回调结果	29
3.16.3 示例代码	30
3.17 根据建筑物 id 查询建筑物所拥有的监测点	30
3.17.1 接口定义	30
3.17.2 回调结果	31
3.17.3 示例代码	31
3.18 更新监测点信息	32
3.18.1 接口定义	32
3.18.2 回调结果	32
3.18.3 示例代码	32
3.19 根据监测点 ID 删除指定监测点	33
3.19.1 接口定义	33
3.19.2 回调结果	33
3.19.3 示例代码	33
3.20 按建筑物 ID 查询监测点记录	34
3.20.1 接口定义	34

3.20.2 回调结果	34
3.20.3 示例代码	35
3.21 按监测点查询监测点记录	35
3.21.1 接口定义	35
3.21.2 回调结果	36
3.21.3 示例代码	36
第四章 附 录	38
4.1 建筑物类型	38
4.2 监测点类型	38
4.3 监测点协议	38
4.4 行政区域示例	39

第一章 概述

1.1 产品简介

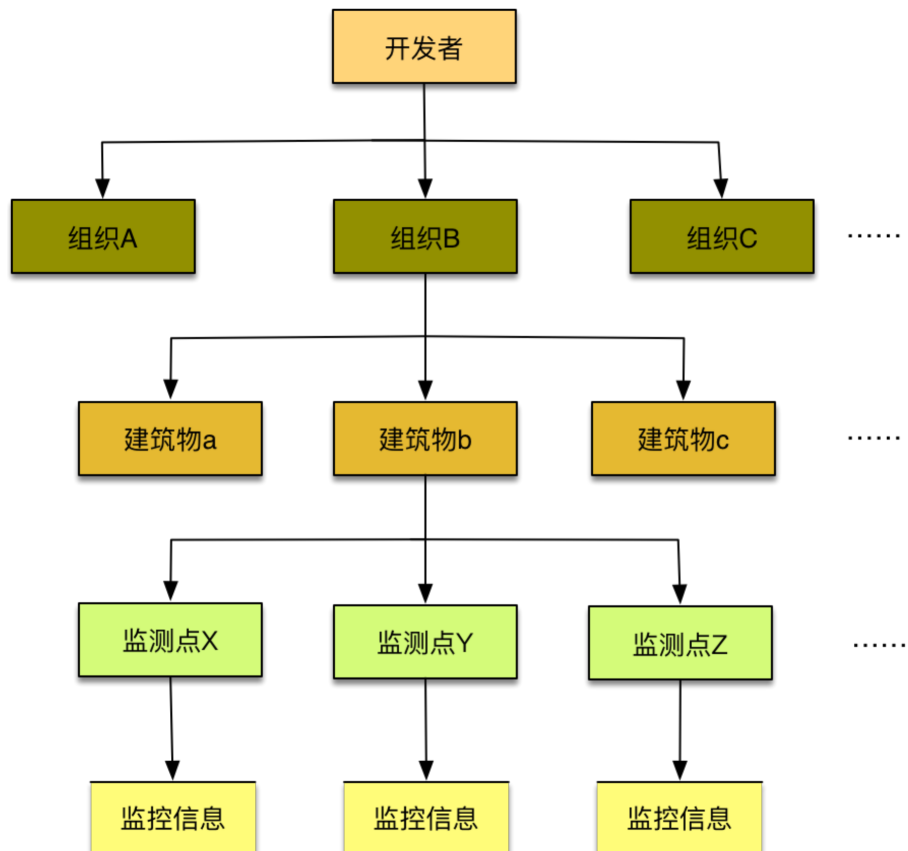
千寻见微 Android SDK 为基于 Android 平台的移动端应用提供组织、建筑物、监测点管理和监测点信息查询的服务接口，专注于为广大开发者提供灵活，方便的千寻见微相关数据管理和查询服务。

该 SDK 免费对外开放。使用该 SDK，可以创建和管理组织、建筑物及监测点对象，查询监控点毫米级监控信息，为开发者提供最大的灵活性。用户在获取这些服务的同时，不用考虑数据的存储、安全和繁琐的网络交互等细节。

千寻通用位置服务 SDK 主要提供以下功能：

用户根据自身业务类型，创建不同的组织、建筑物及监测点对象，并进行管理。用户可以查询不同监测点的毫米级监控信息。

1.2 千寻见微服务结构图



第二章 开发指南

2.1 开发准备

2.1.1 注册为开发者

开发者需要入驻千寻位置网，在千寻位置官网（qxwz.com）注册为千寻位置开发者，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。详细信息请见千寻官网帮助文档。

（<https://www.qxwz.com/help-document-settled-platform.html#link-0>）

2.1.2 创建应用

登录千寻位置网官网的开发者管理控制台创建应用，成功后会自动生成该应用的 AppKey 和 AppSecret，并确保有足够的差分账号配额；AppKey 和 AppSecret 是应用的凭证信息，请妥善保管。详细信息请见千寻官网帮助文档。

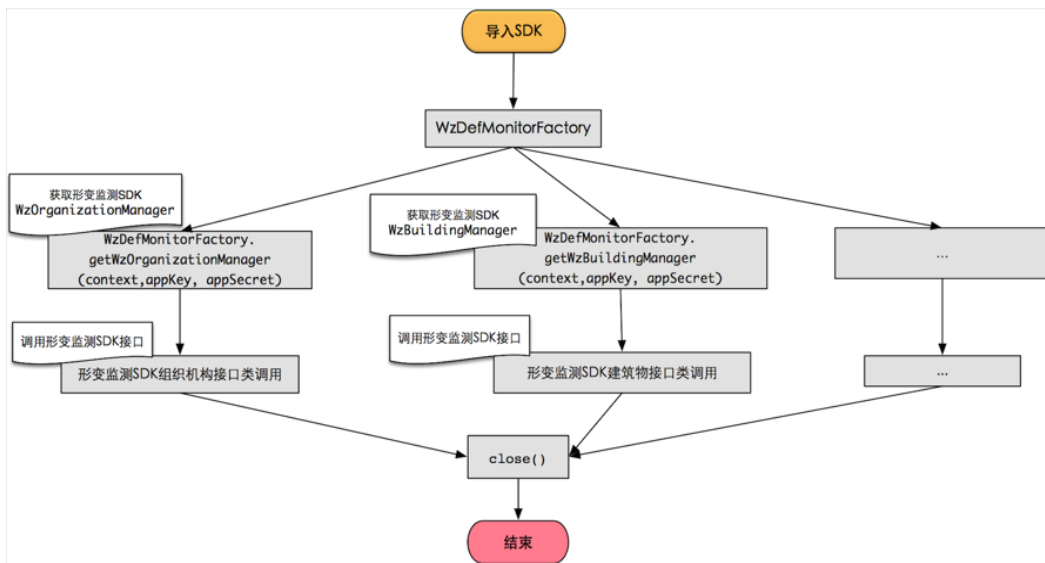
2.1.3 下载 SDK

在千寻位置网官网下载千寻见微 SDK（Android 版）。

2.2 接入流程图

SDK 使用流程大体如下：

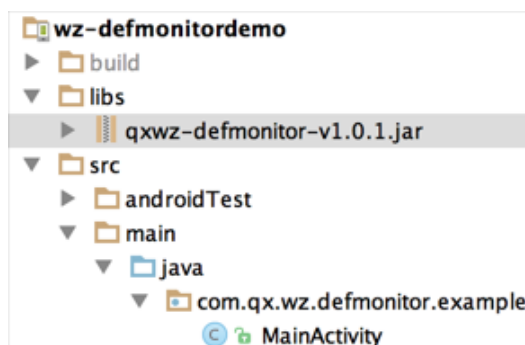
- 1，工程引入 SDK Jar 包，调用对应的 API 接口完成操作。
- 2，由于 API 接口有先后依赖关系，列出相关 API 的调用流程。如下：



2.3 配置工程

2.3.1 Jar 包引入

- 1, 准备千寻提供的 SDK jar 包. 例如: "qxwz-defmonitor-v1.0.1.jar"
- 2, 将千寻见微 SDK 拷贝到 libs 根目录下, 拷贝完成的工程目录如下:



2.3.2 添加权限

在工程的 "AndroidManifest.xml" 文件中进行添加, 建议直接拷贝。

```

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

```

2.3.2 配置结束

对整个工程进行 clean ，配置结束。

第三章 SDK 接口说明

3.1 SDK 文件引用

```
import com.qx.wz.defmonitor.WzDefMonitorFactory;
import com.qx.wz.defmonitor.WzDefMonitorRespondListener;
import com.qx.wz.defmonitor.dto.WzBuilding;
import com.qx.wz.defmonitor.dto.WzMonitorPoint;
import com.qx.wz.defmonitor.dto.WzOrganization;
import com.qx.wz.defmonitor.dto.WzPageResult;
import com.qx.wz.defmonitor.dto.WzRecord;
import com.qx.wz.defmonitor.dto.WzSDKIdResult;
import com.qx.wz.defmonitor.dto.WzSDKResult;
import com.qx.wz.defmonitor.manager.WzBuildingManager;
import com.qx.wz.defmonitor.manager.WzMonitorPointManager;
import com.qx.wz.defmonitor.manager.WzOrganizationManager;
import com.qx.wz.exception.WzException;
```

3.2 获取 WzOrganizationManager

3.2.1 接口定义

```
/**
 * 获取 WzOrganizationManager 的对象
 * @param context 应用上下文，不能为空
 * @param appKey 应用 Key，不能为空
 * @param appSecret 应用密钥，不能为空
 * @return WzOrganizationManager 对象实例
 * @throws WzException
 */

public static WzOrganizationManager getWzOrganizationManager (Context context,
                                                                String appKey, String appSecret) throws
WzException ;
```

3.2.2 示例代码

```
String appKey = "xx";
String appSecret = "xx";
```

```

WzOrganizationManager mWzOrganizationManager;
//得到 WzOrganizationManager
try {
mWzOrganizationManager = WzDefMonitorFactory. getWzOrganizationManager (this, appKey,
appSecret);
} catch (WzException e) {
e.printStackTrace();
}

```

3.3 获取 WzBuildingManager

3.3.1 接口定义

```

/**
 * 获取 WzBuildingManager 的对象
 * @param context 应用上下文，不能为空
 * @param appKey 应用 Key，不能为空
 * @param appSecret 应用密钥，不能为空
 * @return WzBuildingManager 对象实例
 * @throws WzException
 */
public static WzBuildingManager getWzBuildingManager (Context context,
String appKey, String appSecret) throws
WzException ;

```

3.3.2 示例代码

```

String appKey = "xx";
String appSecret = "xx";
WzBuildingManager mWzBuildingManager;
//得到 WzBuildingManager
try {
mWzBuildingManager = WzDefMonitorFactory. getWzBuildingManager (this, appKey,
appSecret);
} catch (WzException e) {
e.printStackTrace();
}

```

3.4 获取 WzMonitorPointManager

3.4.1 接口定义

```
/**
 * 获取 WzMonitorPointManager 的对象
 * @param context 应用上下文，不能为空
 * @param appKey 应用 Key，不能为空
 * @param appSecret 应用密钥，不能为空
 * @return WzMonitorPointManager 对象实例
 * @throws WzException
 */
public static WzMonitorPointManager getWzMonitorPointManager (Context context,
                                                             String appKey, String appSecret) throws
WzException ;
```

3.4.2 示例代码

```
String appKey = "xx";
String appSecret = "xx";
WzMonitorPointManager mWzMonitorPointManager;
//得到 WzMonitorPointManager
try {
    mWzMonitorPointManager = WzDefMonitorFactory. getWzMonitorPointManager (this, appKey,
    appSecret);
} catch (WzException e) {
    e.printStackTrace();
}
```

3.5 新增组织机构

3.5.1 接口定义

```
/**
 * 新增组织机构
 * @param wzOrganization 组织机构对象,不能为空
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象
返回,不能为空
 */
public void addOrganization(WzOrganization wzOrganization,
```

```
WzDefMonitorRespondListener<WzSDKIdResult>  
mDefMonitorRespondListener) throws WzException;
```

3.5.2 回调结果

onSuccess : 返回 WzSDKIdResult , 其中 code 为 0 , id 为新增组织机构 id
onError :
code=10001 , Invalid params
code=12001 , Internal error
code=20101 , Missing param
code=20201 , Entity has been created already

3.5.3 示例代码

```
// 示例：新增组织机构  
public void addOrganization () {  
try {  
    mWzOrganizationManager.addOrganization(wzOrganization,  
        new WzDefMonitorRespondListener<WzSDKIdResult>() {  
        @Override  
        public void onSuccess(WzSDKIdResult wzSDKIdResult) {  
            // TODO: do something when return successful result  
        }  
        @Override  
        public void onError(int code, String message, Throwable e) {  
            // TODO: do something when error happens  
        }  
        @Override  
        public void onCancel() {  
            // TODO: do something when cancel addOrganization  
        }  
    });  
} catch (WzException e) {  
    e.printStackTrace();  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}
```

3.6 通过组织机构 ID 查询组织机构信息

3.6.1 接口定义

```
/**
 * 通过组织机构 ID 查询组织机构信息
 * @param id 机构 ID,大于 0
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
public void queryOrganizationById(long id,
    WzDefMonitorRespondListener<WzOrganization> mDefMonitorRespondListener) throws
WzException;
```

3.6.2 回调结果

```
onSuccess : 返回 WzOrganization
onError :
code=10001 , Invalid params
code=10002 , No Permission
    code=12001 , Internal error
code=20101 , Missing param
code=20202 , Entity does not exist
```

3.6.3 示例代码

```
// 示例：通过组织机构 ID 查询组织机构信息
public void queryOrganizationById() {
    try {
        mWzOrganizationManager.queryOrganizationById(organizationById,
            new WzDefMonitorRespondListener<WzOrganization>() {
                @Override
                public void onSuccess(WzOrganization wzOrganization) {
                    // TODO: do something when return successful result
                }
                @Override
                public void onError(int code, String message, Throwable e) {
                    // TODO: do something when error happens
                }
                @Override
                public void onCancel() {
                    // TODO: do something when cancel queryOrganizationById
                }
            }
        );
    } catch (WzException e) {
        // TODO: do something when exception
    }
}
```

```

        }
    });
    } catch (WzException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

3.7 查询组织机构列表

3.7.1 接口定义

```

/**
 * 查询组织机构列表
 * @param pageNo 页码（未指定，则默认为 1；值小于等于 0，则置为 1）
 * @param pageSize 每页数据条数（未指定，则默认为 20；小于等于 0，则置为 20；值 > 2000，则置为 2000）
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回，不能为空
 */
public void getOrganizations(int pageNo, int pageSize,
    WzDefMonitorRespondListener<WzPageResult<WzOrganization>>
        mDefMonitorRespondListener) throws WzException;

```

3.7.2 回调结果

```

onSuccess : 返回 WzPageResult，其中 code 为 0
onError :
    code=10001，Invalid params
    code=10002，No Permission
    code=12001，Internal error
    code=20101，Missing param
    code=20202，Entity does not exist

```

3.7.3 示例代码

```

// 示例：查询组织机构列表
public void getOrganizations() {
    try {
        mWzOrganizationManager.getOrganizations(pageNo, pageSize,

```

```

new WzDefMonitorRespondListener<WzPageResult<WzOrganization>>() {
    @Override
    public void onSuccess(WzPageResult<WzOrganization>
wzOrganizationWzPageResult) {
        // TODO: do something when return successful result
    }
    @Override
    public void onError(int code, String message, Throwable e) {
        // TODO: do something when error happens
    }
    @Override
    public void onCancel() {
        // TODO: do something when cancel getOrganizations
    }
    });
} catch (WzException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

3.8 更新组织机构信息

3.8.1 接口定义

```

/**
 * 更新组织机构信息
 * @param wzOrganization 组织机构,不能为空
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回;
不能为空*/
public void updateOrganization(WzOrganization wzOrganization,
WzDefMonitorRespondListener<WzSDKResult>
mDefMonitorRespondListener) throws WzException;

```

3.8.2 回调结果

onSuccess : 返回 WzSDKResult , 其中 code 为 0

onError :

- code=10001 , Invalid params
- code=10002 , No Permission
- code=12001 , Internal error

```
code=20101 , Missing param  
code=20202 , Entity does not exist
```

3.8.3 示例代码

```
// 示例：更新组织机构信息  
public void updateOrganization () {  
    try {  
        mWzOrganizationManager.updateOrganization(wzOrganization,  
            new WzDefMonitorRespondListener<WzSDKResult>() {  
                @Override  
                public void onSuccess(WzSDKResult wzSDKResult) {  
                    // TODO: do something when return successful result  
                }  
                @Override  
                public void onError(int code, String message, Throwable e) {  
                    // TODO: do something when error happens  
                }  
                @Override  
                public void onCancel() {  
                    // TODO: do something when cancel updateOrganization  
                }  
            });  
    } catch (WzException e) {  
        e.printStackTrace();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

3.9 删除指定组织机构信息

3.9.1 接口定义

```
/**  
 * 删除指定组织机构信息  
 * @param id 组织机构 ID,大于 0  
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,  
 不能为空*/  
public void deleteOrganizationById(long id,  
    WzDefMonitorRespondListener<WzSDKResult> mDefMonitorRespondListener) throws  
WzException;
```

3.9.2 回调结果

```
onSuccess : 返回 WzSDKResult , 其中 code 为 0
onError :
code=10001 , Invalid params
code=10002 , No Permission
    code=12001 , Internal error
code=20101 , Missing param
    code=20202 , Entity does not exist
```

3.9.3 示例代码

```
// 示例：删除指定组织机构信息
public void deleteOrganizationById () {
try {
mWzOrganizationManager.deleteOrganizationById(organizationById,
    new WzDefMonitorRespondListener<WzSDKResult>() {
        @Override
        public void onSuccess(WzSDKResult wzSDKResult) {
            // TODO: do something when return successful result
        }
        @Override
        public void onError(int code, String message, Throwable e) {
            // TODO: do something when error happens
        }
        @Override
        public void onCancel() {
            // TODO: do something when cancel deleteOrganizationById
        }
    });
} catch (WzException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
```

3.10 添加建筑物

3.10.1 接口定义

```
/**
 * 添加建筑物
 * @param wzBuilding 建筑物对象,不能为空
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
public void addBuilding(WzBuilding wzBuilding,
        WzDefMonitorRespondListener<WzSDKIdResult> mDefMonitorRespondListener) throws
WzException;
```

3.10.2 回调结果

```
onSuccess : 返回 WzSDKIdResult , 其中 code 为 0, id 为新增建筑物 ID
onError :
    code=10001 , Invalid params
    code=12001 , Internal error
    code=20101 , Missing param
    code=20201 , Entity has been created already
```

3.10.3 示例代码

```
// 示例：添加建筑物
public void addBuilding () {
try {
mWzBuildingManager.addBuilding(wzBuilding,
    new WzDefMonitorRespondListener<WzSDKIdResult>() {
        @Override
        public void onSuccess(WzSDKIdResult wzSDKIdResult) {
            // TODO: do something when return successful result
        }
        @Override
        public void onError(int code, String message, Throwable e) {
            // TODO: do something when error happens
        }
        @Override
        public void onCancel() {
            // TODO: do something when cancel addBuilding
        }
    }
}
```

```

        });
    } catch (WzException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

3.11 通过建筑物 id 查询建筑物

3.11.1 接口定义

```

/**
 * 通过建筑物 id 查询建筑物
 * @param id 建筑物 ID,大于 0
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
public void queryBuildingById(long id,
                               WzDefMonitorRespondListener<WzBuilding> mDefMonitorRespondListener) throws
WzException;

```

3.11.2 回调结果

```

onSuccess : 返回 WzBuilding
onError :
code=10001 , Invalid params
code=10002 , No Permission
code=12001 , Internal error
code=20101 , Missing param
code=20202 , Entity does not exist

```

3.11.3 示例代码

```

// 示例：通过建筑物 id 查询建筑物
public void queryBuildingById() {
    try {
        mWzBuildingManager.queryBuildingById(buildingId,
            new WzDefMonitorRespondListener<WzBuilding>() {
                @Override
                public void onSuccess(WzBuilding wzBuilding) {
                    // TODO: do something when return successful result
                }
            }
        );
    }
}

```

```

    }
    @Override
    public void onError(int code, String message, Throwable e) {
        // TODO: do something when error happens
    }
    @Override
    public void onCancel() {
        // TODO: do something when cancel queryBuildingById
    }
});
} catch (WzException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

3.12 根据组织机构 id 查询建筑物

3.12.1 接口定义

```

/**
 * 根据组织机构 id 查询建筑物
 * @param wzOrganizationId 组织机构对象 ID,大于 0
 * @param pageNo 页码 (未指定,则默认为 1;值小于等于 0,则置为 1)
 * @param pageSize 每页数据条数 (未指定,则默认为 20;小于等于 0,则置为 20,size 大于 2000,则置为 2000)
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,不能为空*/
    public void queryBuildingsByOrganizationId(long wzOrganizationId,int pageNo, int pageSize,
        WzDefMonitorRespondListener<WzPageResult<WzBuilding>>
        mDefMonitorRespondListener)
        throws WzException;

```

3.12.2 回调结果

```

onSuccess : 返回 WzPageResult
onError :
code=10001 , Invalid params
code=10002 , No Permission
    code=12001 , Internal error
code=20101 , Missing param

```

```
code=20202 , Entity does not exist
```

3.12.3 示例代码

```
// 示例：根据组织机构 id 查询建筑物
public void queryBuildingsByOrganizationId() {
    try {
        mWzBuildingManager.queryBuildingsByOrganizationId(organizationId,          pageNo,
        pageSize, new WzDefMonitorRespondListener<WzPageResult<WzBuilding>>() {
            @Override
            public void onSuccess(WzPageResult<WzBuilding> wzBuildingWzPageResult) {
                // TODO: do something when return successful result
            }
            @Override
            public void onError(int code, String message, Throwable e) {
                // TODO: do something when error happens
            }
            @Override
            public void onCancel() {
                // TODO: do something when cancel queryBuildingsByOrganizationId
            }
        });
    } catch (WzException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

3.13 更新建筑物信息

3.13.1 接口定义

```
/**
 * 更新建筑物信息
 * @param wzBuilding 建筑物对象；不能为空
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
public void updateBuilding(WzBuilding wzBuilding,WzDefMonitorRespondListener<WzSDKResult>
mDefMonitorRespondListener) throws WzException;
```

3.13.2 回调结果

```
onSuccess : 返回 WzSDKResult , 其中 code 为 0
onError :
code=10001 , Invalid params
code=10002 , No Permission
    code=12001 , Internal error
code=20101 , Missing param
    code=20202 , Entity does not exist
```

3.13.3 示例代码

```
// 示例：更新建筑物信息
public void updateBuilding() {
    try {
mWzBuildingManager.updateBuilding(wzBuilding,
    new WzDefMonitorRespondListener<WzSDKResult>() {
        @Override
        public void onSuccess(WzSDKResult wzSDKResult) {
            // TODO: do something when return successful result
        }
        @Override
        public void onError(int code, String message, Throwable e) {
            // TODO: do something when error happens
        }
        @Override
        public void onCancel() {
            // TODO: do something when cancel updateBuilding
        }
    });
    } catch (WzException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

3.14 根据建筑物 id 删除建筑物

3.14.1 接口定义

```
/**
 * 根据建筑物 id 删除建筑物
 * @param id 建筑物 ID,大于 0
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,不能为空
 */
public void deleteBuildingById(long id, WzDefMonitorRespondListener<WzSDKResult>
                                mDefMonitorRespondListener) throws
WzException;
```

3.14.2 回调结果

```
onSuccess : 返回 WzSDKResult , 其中 code 为 0
onError :
    code=10001 , Invalid params
    code=10002 , No Permission
    code=12001 , Internal error
    code=20101 , Missing param
    code=20202 , Entity does not exist
```

3.14.3 示例代码

```
// 示例：根据建筑物 id 删除建筑物
public void deleteBuildingById() {
    try {
        mWzBuildingManager.deleteBuildingById(buildingId,
            new WzDefMonitorRespondListener<WzSDKResult>() {
                @Override
                public void onSuccess(WzSDKResult wzSDKResult) {
                    // TODO: do something when return successful result
                }
                @Override
                public void onError(int code, String message, Throwable e) {
                    // TODO: do something when error happens
                }
                @Override
                public void onCancel() {
```

```

        // TODO: do something when cancel deleteBuildingById
    }
    });
    } catch (WzException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

3.15 新增监测点

3.15.1 接口定义

```

/**
 * 新增监测点
 * @param wzMonitorPoint 监测点对象,不能为空
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
public void addMonitorPoint(WzMonitorPoint wzMonitorPoint,
                            WzDefMonitorRespondListener<WzSDKIdResult>
                            mDefMonitorRespondListener) throws
WzException;

```

3.15.2 回调结果

```

onSuccess : 返回 WzSDKIdResult , 其中 code 为 0 , id 为新增监测点 ID
onError :
    code=10001 , Invalid params
    code=12001 , Internal error
    code=20101 , Missing param
    code=20201 , Entity has been created already

```

3.15.3 示例代码

```

// 示例：新增监测点
public void addMonitorPoint () {
try {
mWzMonitorPointManager.addMonitorPoint(wzMonitorPoint,
    new WzDefMonitorRespondListener<WzSDKIdResult>() {
        @Override

```

```

        public void onSuccess(WzSDKIdResult wzSDKIdResult) {
            // TODO: do something when return successful result
        }
        @Override
        public void onError(int code, String message, Throwable e) {
            // TODO: do something when error happens
        }
        @Override
        public void onCancel() {
            // TODO: do something when cancel addMonitorPoint
        }
    });
    } catch (WzException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

3.16 根据监测点 id 查询监测点信息

3.16.1 接口定义

```

/**
 * 根据监测点 id 查询监测点信息
 * @param id 监测点 ID,大于 0
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
public void getMonitorPointById(long id,WzDefMonitorRespondListener<WzMonitorPoint>
mDefMonitorRespondListener) throws WzException;

```

3.16.2 回调结果

```

onSuccess : 返回 WzMonitorPoint
onError :
code=10001 , Invalid params
code=10002 , No Permission
code=12001 , Internal error
code=20101 , Missing param
code=20202 , Entity does not exist

```

3.16.3 示例代码

```
// 示例：根据监测点 id 查询监测点信息
public void getMonitorPointById() {
    try {
        mWzMonitorPointManager.getMonitorPointById(buildingId,
            new WzDefMonitorRespondListener<WzMonitorPoint>() {
                @Override
                public void onSuccess(WzMonitorPoint wzMonitorPoint) {
                    // TODO: do something when return successful result
                }
                @Override
                public void onError(int code, String message, Throwable throwable) {
                    // TODO: do something when error happens
                }
                @Override
                public void onCancel() {
                    // TODO: do something when cancel getMonitorPointById
                }
            });
    } catch (WzException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

3.17 根据建筑物 id 查询建筑物所拥有的监测点

3.17.1 接口定义

```
/**
 * 根据建筑物 id 查询建筑物所拥有的监测点
 * @param buildingId 建筑物 ID,大于 0
 * @param pageNo 页码 (未指定,则默认为 1;值小于等于 0,则置为 1)
 * @param pageSize 每页数据条数 (未指定,则默认为 20;小于等于 0,则置为 20;值>2000,则置为 2000)
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
 * 不能为空
 */
public void getMonitorPointsByBuildingId(long buildingId,int pageNo, int pageSize,
    WzDefMonitorRespondListener<WzPageResult<WzMonitorPoint>>
```

```
mDefMonitorRespondListener) throws WzException;
```

3.17.2 回调结果

```
onSuccess : 返回 WzPageResult  
onError :  
code=10001 , Invalid params  
code=10002 , No Permission  
    code=12001 , Internal error  
code=20101 , Missing param  
    code=20202 , Entity does not exist
```

3.17.3 示例代码

```
// 示例：根据建筑物 id 查询建筑物所拥有的监测点  
public void getMonitorPointsByBuildingId() {  
    try {  
        mWzMonitorPointManager.getMonitorPointsByBuildingId(buildingId, pageNo, pageSize,  
new WzDefMonitorRespondListener<WzPageResult<WzMonitorPoint>>() {  
            @Override  
            public void onSuccess(WzPageResult<WzMonitorPoint>  
wzMonitorPointWzPageResult) {  
                // TODO: do something when return successful result  
            }  
            @Override  
            public void onError(int code, String message, Throwable throwable) {  
                // TODO: do something when error happens  
            }  
            @Override  
            public void onCancel() {  
                // TODO: do something when cancel getMonitorPointsByBuildingId  
            }  
        });  
    } catch (WzException e) {  
        e.printStackTrace();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

3.18 更新监测点信息

3.18.1 接口定义

```
/**
 * 更新监测点信息
 * @param wzMonitorPoint 监测点对象,不能为空
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
public void updateMonitorPoint(WzMonitorPoint wzMonitorPoint,
                               WzDefMonitorRespondListener<WzSDKResult> mDefMonitorRespondListener) throws
WzException;
```

3.18.2 回调结果

```
onSuccess : 返回 WzSDKResult , code 为 0
onError :
code=10001 , Invalid params
code=10002 , No Permission
code=12001 , Internal error
code=20101 , Missing param
code=20202 , Entity does not exist
```

3.18.3 示例代码

```
// 示例 : 更新监测点信息
public void updateMonitorPoint () {
try {
mWzMonitorPointManager.updateMonitorPoint(wzMonitorPoint,
new WzDefMonitorRespondListener<WzSDKResult>() {
@Override
public void onSuccess(WzSDKResult wzSDKResult) {
// TODO: do something when return successful result
}
@Override
public void onError(int code, String message, Throwable throwable) {
// TODO: do something when error happens
}
@Override
public void onCancel() {
// TODO: do something when cancel updateMonitorPoint
}
```

```

        }
    });
} catch (WzException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

3.19 根据监测点 ID 删除指定监测点

3.19.1 接口定义

```

/**
 * 根据监测点 ID 删除指定监测点
 * @param pointId 监测点 ID,大于 0
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
public void deleteMonitorPointById(long pointId,
    WzDefMonitorRespondListener<WzSDKResult> mDefMonitorRespondListener) throws
WzException;

```

3.19.2 回调结果

```

onSuccess : 返回 WzSDKResult , code 为 0
onError :
code=10001 , Invalid params
code=10002 , No Permission
    code=12001 , Internal error
code=20101 , Missing param
    code=20202 , Entity does not exist

```

3.19.3 示例代码

```

// 示例：根据监测点 ID 删除指定监测点
public void deleteMonitorPointById() {
    try {
        mWzMonitorPointManager.deleteMonitorPointById(monitorPointId,
            new WzDefMonitorRespondListener<WzSDKResult>() {
                @Override
                public void onSuccess(WzSDKResult wzSDKResult) {

```

```

        // TODO: do something when return successful result
    }
    @Override
    public void onError(int code, String message, Throwable throwable) {
        // TODO: do something when error happens
    }
    @Override
    public void onCancel() {
        // TODO: do something when cancel deleteMonitorPointById
    }
    });
    } catch (WzException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

3.20 按建筑物 ID 查询监测点记录

3.20.1 接口定义

```

/**
 * 按建筑物 ID 查询监测点记录
 * @param buildingId 建筑物 ID,大于 0
 * @param beginTime 开始时间
 * @param endTime 结束时间
 * @param pageNo 页码 (未指定,则默认为 1;值小于等于 0,则置为 1)
 * @param pageSize 每页数据条数 (未指定,则默认为 20;小于等于 0,则置为 20;值>2000,则置为
2000)
 * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回,
不能为空*/
    public void getRecordsByBuildingId(long buildingId,long beginTime, long endTime, int pageNo, int
pageSize,
        WzDefMonitorRespondListener<WzPageResult<WzRecord>>
mDefMonitorRespondListener)
                                                                    throws
WzException;

```

3.20.2 回调结果

onSuccess : 返回 WzPageResult

```
onError :  
code=10001 , Invalid params  
code=10002 , No Permission  
    code=12001 , Internal error  
code=20101 , Missing param  
    code=20202 , Entity does not exist
```

3.20.3 示例代码

```
// 示例：按建筑物 ID 查询监测点记录  
public void getRecordsByBuildingId() {  
    try {  
        mWzMonitorPointManager.getRecordsByBuildingId(buildingId, beginTime, endTime,  
pageNo, pageSize, new WzDefMonitorRespondListener<WzPageResult<WzRecord>>() {  
            @Override  
            public void onSuccess(WzPageResult<WzRecord> wzRecordWzPageResult) {  
                // TODO: do something when return successful result  
            }  
            @Override  
            public void onError(int code, String message, Throwable throwable) {  
                // TODO: do something when error happens  
            }  
            @Override  
            public void onCancel() {  
                // TODO: do something when cancel getRecordsByBuildingId  
            }  
        });  
    } catch (WzException e) {  
        e.printStackTrace();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

3.21 按监测点查询监测点记录

3.21.1 接口定义

```
/**  
 * 按监测点查询监测点记录  
 * @param pointId 监测点 ID,大于 0  
 * @param beginTime 开始时间
```

```

    * @param endTime 结束时间
    * @param pageNo 页码 (未指定, 则默认为 1; 值小于等于 0, 则置为 1)
    * @param pageSize 每页数据条数 (未指定, 则默认为 20; 小于等于 0, 则置为 20; 值>2000, 则置为 2000)
    * @param mDefMonitorRespondListener 接口调用结果将通过 DefMonitorRespondListener 对象返回, 不能为空*/
    public void getRecordsByPointId(long pointId, long beginTime, long endTime, int pageNo, int pageSize,
        WzDefMonitorRespondListener<WzPageResult<WzRecord>>
        mDefMonitorRespondListener)
        throws WzException;

```

3.21.2 回调结果

```

onSuccess : 返回 WzPageResult
onError :
code=10001 , Invalid params
code=10002 , No Permission
    code=12001 , Internal error
code=20101 , Missing param
    code=20202 , Entity does not exist

```

3.21.3 示例代码

```

// 示例：按监测点查询监测点记录
public void getRecordsByPointId() {
    try {
        mWzMonitorPointManager.getRecordsByPointId(monitorPointId, beginTime, endTime,
            pageNo, pageSize, new WzDefMonitorRespondListener<WzPageResult<WzRecord>>() {
            @Override
            public void onSuccess(WzPageResult<WzRecord> wzRecordWzPageResult) {
                // TODO: do something when return successful result
            }
            @Override
            public void onError(int code, String message, Throwable throwable) {
                // TODO: do something when error happens
            }
            @Override
            public void onCancel() {
                // TODO: do something when cancel getRecordsByPointId
            }
        });
    } catch (WzException e) {

```

```
        e.printStackTrace();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

第四章 附录

4.1 建筑物类型

建筑物类型属性(type)取值如下：

1. HOUSE:房屋
2. LANDSLIP:滑坡
3. SIDESLOP:边坡
4. IRONTOWER:铁塔
5. ELECTRICTOWER:电力塔
6. SIGNALTOWER:信号塔
7. DIKE:大坝
8. FACTORYBUILDING:厂房
9. ROLLINGSTONE:滚石

4.2 监测点类型

监测点状态属性(status)取值范围如下：

1. ENABLED:启用(默认)
2. SUSPENDED:暂停服务
3. DISABLED:弃用

4.3 监测点协议

监测点协议属性(protocol)取值范围如下：

1. TCP:TCP (默认)

2. NTRIP:NTRIP

4.4 行政区域示例

接口中出现省、市、区/县属性时，只接受合法的行政区域名称。

直辖市如：上海市，上海，长宁区；

普通省份如：河南省，郑州市，中牟县。

示例：直辖市

```
WzBuilding wzBuilding = new WzBuilding();  
wzBuilding.setProvince("上海市");  
wzBuilding.setCity("上海");  
wzBuilding.setDistrict("长宁区");
```

示例：普通省份

```
WzBuilding wzBuilding = new WzBuilding();  
wzBuilding.setProvince("河南省");  
wzBuilding.setCity("郑州市");  
wzBuilding.setDistrict("中牟县");
```

示例：自治区

```
WzBuilding wzBuilding = new WzBuilding();  
wzBuilding.setProvince("西藏自治区");  
wzBuilding.setCity("拉萨市");  
wzBuilding.setDistrict("尼木县");
```