

功能介绍

通过 sdk 访问并拉取目录服务器配置

接口说明

接口创建和销毁

TCLSCreateTdir	创建 ITdir
参数	无
返回值	ITdir*

TCLSRelease	释放 ITdir
参数	ITdir** ppTdir
返回值	void

接口使用

Init	ITdir 初始化
参数	<div><div>1. const int appld 即通分配的业务 ID</div><div>2. const char *ipList IP 列表，多个使用“ ”连接，如“10.12.16.173 10.12.16.171”</div><div>3. const char* portList 端口列表</div><div>4. const char * lastSuccessIP 暂无用处，默认为 NULL</div><div>5. const char * lastSuccessPort 暂无用处，默认为 NULL</div><div>6. const char * openId</div></div>

	openId, 默认为 NULL
返回值	<ol style="list-style-type: none"> ITdir::TDIR_NO_ERROR 成功时返回 ITdir::PARAM_ERROR ipList 或者 portList 为 NULL 时返回 ITdir::ALLOCAT_MEMORY_FAILED 分配内存失败时返回

Status	查询内部状态
参数	无
返回值	<ol style="list-style-type: none"> ITdir:: UN_INIT 未初始化状态, 请调用 Init 接口进行初始化。初始化后进入 ITdir:: WAIT_FOR_QUERY 状态 ITdir:: WAIT_FOR_QUERY 等待发送请求包状态, 请调用 Query 接口驱动发包逻辑。非 UN_INIT 和 WAIT_FOR_QUERY 状态调用 Query, 都会进入此状态 ITdir:: WAIT_FOR_RECV 获取服务器数据状态: 请调用 Recv 接口驱动收包逻辑。使用 Query 接口成功发送请求包之后, 会进入此状态 ITdir:: RECV_DONE 获取目录树完毕状态: 可以使用 GetTreeNodes 接口获取目录树。使用 Recv 接口完成收包之后进入此状态

Query	驱动发包逻辑接口
参数	<ol style="list-style-type: none"> int timeout 分配给发包逻辑的时间片, 单位毫秒。超时后该接口返回
返回值	<ol style="list-style-type: none"> ITdir::NO_ERROR 无错误 ITdir::NEED_INIT_BEFORE_QUERY 在 ITdir:: UN_INIT 状态下调用 Query, 返回该错误 ITdir::WAIT_SERVER_REP_TIMEOUT 单个服务器超时(发起请求至完成目录树获取的时间超过时限), 会返回该错误

4. ITdir::INIT_TGCPAPI_FAILED
初始化内部的 tgcpapi 失败
5. ITdir::ALL_IP_CONNECT_FAILED
Init 接口传入 ipList 和 portList 的所有组合的服务器都超时
6. ITdir::TGCPAPI_ERROR
tgcpapi 库出错

Recv	驱动收包逻辑接口
参数	<ol style="list-style-type: none"> 1. int timeout 分配给收包逻辑的时间片，单位毫秒。超时后该接口返回
返回值	<ol style="list-style-type: none"> 1. ITdir::NO_ERROR 无错误 2. ITdir::NEED_WAIT_FOR_RECV_STATUS 非 ITdir:: WAIT_FOR_RECV 状态下调用 Recv 接口，返回该错误 3. ITdir::WAIT_SERVER_REP_TIMEOUT 单个服务器超时(发起请求至完成目录树获取的时间超过时限)，会返回该错误 4. ITdir::INIT_TGCPAPI_FAILED 初始化内部的 tgcpapi 失败 5. ITdir::ALL_IP_CONNECT_FAILED Init 接口传入 ipList 和 portList 的所有组合的服务器都超时 6. ITdir::TGCPAPI_ERROR tgcpapi 库出错 7. ITdir::ALLOCAT_MEMORY_FAILED 分配内存失败 8. ITdir::UNPACK_FAILED 解析服务器回包失败

GetTreeNodes

当状态 ITdir:: RECV_DONE 时，通过 GetTreeNodes 获

取目录树	
参数	1. std::list<TreeNode> &list 目录树
返回值	1. ITdir::NO_ERROR 无错误 2. 目录服务器内部状态 非 ITdir:: RECV_DONE 时，返回内部状态

SetSvrTimeout	
设置单个服务器的超时时间，默认 5000ms	
参数	1. int timeout 设置请求单个服务器的超时时间，单位毫秒
返回值	ITdir::NO_ERROR

EnableLog	
打开内部 log，默认关闭	
参数	无
返回值	ITdir::NO_ERROR

DisableLog	
关闭内部 log	
参数	无
返回值	ITdir::NO_ERROR

使用样例

1、 引入头文件

```
#include "ITdir.h"
```

2、 创建 ITdir 接口

```
mTdir = NApollo::TCLSCreateTdir();
```

3、 帧循环中驱动

```
// 获取目录服务器内部状态
NApollo::ITdir::Result status = mTdir->Status();
```

```

// 根据目录服务器状态执行不同的逻辑
switch (status) {

    //未初始化
    case NApollo::ITdir::UN_INIT:
    {
        // ip列表, 多个使用"|"线隔开
        char ipList[] = "113.108.0.75|113.108.0.76|game.qq.com";
        // port列表, 多个使用"|"线隔开
        char portList[] = "8856|8889";
        // 即通分配的serviceID
        int appID = 715013506;
        // 初始化目录服务器
        result = mTdir->Init(appID, ipList, portList);

        CCLOG("TdirScene:: init result: %d", result);
        break;
    }

    // 等待发送数据
    case NApollo::ITdir::WAIT_FOR_QUERY:
    {
        // 发送数据
        result = mTdir->Query();
        CCLOG("TdirScene:: query result: %d", result);
        break;
    }

    // 等待收取数据
    case NApollo::ITdir::WAIT_FOR_RECV:
    {
        // 收取数据
        result = mTdir->Recv();
        CCLOG("TdirScene:: recv result: %d", result);
        break;
    }

    // 完成获取目录树
    case NApollo::ITdir::RECV_DONE:
    {
        //mTreeNodeList为最终的目录树
        result = mTdir->GetTreeNodes(mTreeNodeList);
        CCLOG("TdirScene:: get tree node result: %d", result);
    }
}

```

```
        //销毁目录服务器对象。帧循环中无需再调用目录服务器相关代码
        NApollo::TCLSRelease(&mTdir);
        break;
    }
    default:
    {
        CCLOG("TdirScene:: error status: %d", status);
        break;
    }
}
```