**手机客户端用户登录接口**

**post url: http://120.24.101.50:8080/lwork/mobile/login**

**params: {uuid:uuid, pwd:pwd}**

**return: [{status,ok}] or [{status,failed},{reason,invalid\_pwd}]**

**二 手机客户端接口说明**

=================================SDK导入=================================

1.eclipse新建android工程后会自动生成一个libs的文件夹

2.在libs文件夹上点击右键，选择import，将webrtc.jar导入

3.在libs文件夹上点击右键，选择Build Path->Configure Build Path

4.在弹出的界面上左侧栏选择Java Build Path

5.在右侧的菜单中选择Libraries

6.选择Add JARS，添加webrtc.jar

7.将armeabi和armeabi-v7a文件夹拖入libs

8.在AndroidManifest.xml文件中增加如下三句话

<uses-permission android:name="android.permission.MODIFY\_AUDIO\_SETTINGS" />

<uses-permission android:name="android.permission.RECORD\_AUDIO" />

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

=================================SDK API=================================

提供了一个类CallMaker和一个interface UserCallback

import org.webrtc.videoengineapp.CallMaker;

import org.webrtc.videoengineapp.UserCallback;

UserCallback定义了三个回调函数 ie.

public class MyCallback implements UserCallback

{

public void ring(){};

public void talking(){};

public void hangup(){};

}

三个回调分别是在振铃，接通，对方挂机时被调用

CallMaker类提供如下API

CallMaker(Context ctxt,UserCallback api)

void release()

void setVolumeLevel(int vol)

void setLoudSpeaker(boolean bool)

String startCall(String phone,String userclass,String selfPhone)

String stopCall()

=============================================

CallMaker(Context ctxt,UserCallback api)

void release()

负责构造CallMaker类和释放，如果构造或者释放失败，会抛出RuntimeException

=============================================

设置本端听筒音量，默认值是204

void setVolumeLevel(int vol)

设置是否使用免提，默认值是不使用

void setLoudSpeaker(boolean bool)

=============================================

String startCall(String phone,String userclass,String selfPhone)

String stopCall()

发起呼叫和停止呼叫，phone为呼叫号码，selfPhone为用户号码，userclass为用户类型，目前是两个值有效分别是"registered","unregistered"

startCall和stopCall成功返回"ok"，失败则返回原因

此外，startCall和stopCall都调用了发送ajax请求的http接口，android的权限限制，要求UI类中不能调用这种接口，防止被吊死，所以如果想在UI类中调用

需要另起一个线程来执行startCall和stopCall

=============================================

demo代码

package com.example.test;

import android.os.Bundle;

import android.widget.Button;

import android.view.View;

import android.widget.EditText;

import android.app.Activity;

import android.view.Menu;

import android.os.AsyncTask;

import org.webrtc.videoengineapp.CallMaker;

import org.webrtc.videoengineapp.UserCallback;

public class MainActivity extends Activity implements View.OnClickListener,UserCallback

{

private CallMaker call = null;

private Button btCall;

private Button btStop;

private EditText txtPhone;

public void ring(){};

public void talking(){};

public void hangup(){};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

btCall = (Button) findViewById(R.id.call);

btStop = (Button) findViewById(R.id.stop);

txtPhone = (EditText) findViewById(R.id.phone);

btCall.setOnClickListener(this);

btStop.setOnClickListener(this);

call = new CallMaker(this,this);

}

@Override

protected void onDestroy() {

call.release();

call = null;

super.onDestroy();

}

public void onClick(View v){

switch (v.getId()){

case R.id.call:

new CallStarter().execute("");

break;

case R.id.stop:

new CallStoper().execute("");

break;

default:

break;

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

public class CallStarter extends AsyncTask<String, Integer, String>{

@Override

protected String doInBackground(String... data){

return call.startCall(txtPhone.getText().toString(),"registered","11111111111");

}

@Override

protected void onPostExecute(String rt){

super.onPostExecute(rt);

}

}

public class CallStoper extends AsyncTask<String, Integer, String>{

@Override

protected String doInBackground(String... data){

return call.stopCall();

}

@Override

protected void onPostExecute(String rt){

super.onPostExecute(rt);

}

}

}