**中移在线SDK接入文档**

V1.0

**中移在线服务有限公司版权所有**

**文档修订记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **日期** | **版本号** | **描述** | **著者** | **审阅者** | **日期** |
| 2019/05/31 | V1.0 | 文档创建 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1. 前言 1](#_Toc10468733)

[2. 接入准备 1](#_Toc10468734)

[3. 开发指南 4](#_Toc10468735)

[(一) SDK初始化 4](#_Toc10468736)

[(二) 读取身份证 4](#_Toc10468737)

[a) 无UI读取身份证 4](#_Toc10468738)

[b) 使用Activity套用Fragment方式读取身份证 5](#_Toc10468739)

[c) 调用Activity读取身份证 6](#_Toc10468740)

[(三) 人像比对 6](#_Toc10468741)

[a) 无UI人像比对 6](#_Toc10468742)

[b) 使用Activity套用Fragment方式人像比对 7](#_Toc10468745)

[c) 调用人像比对Activity 7](#_Toc10468749)

[(四) 无证比对 8](#_Toc10468750)

[a) 调用无证比对Activity 8](#_Toc10468751)

[b) 调用接口比对 9](#_Toc10468752)

[(五) 图片比对 11](#_Toc10468753)

[a) 本地比对 11](#_Toc10468754)

[b) 后台比对 11](#_Toc10468755)

[(六) 标准比对 12](#_Toc10468756)

[(七) 拍照 15](#_Toc10468757)

[a) 无UI普通拍照 15](#_Toc10468758)

[b) 无UI 自动抓拍 16](#_Toc10468759)

[c) Activity 嵌套Fragment普通拍照 16](#_Toc10468760)

[d) Activity 嵌套 Fragment 自动抓拍 16](#_Toc10468761)

[e) 调用拍照Activity 16](#_Toc10468762)

[(八) 替他接口 18](#_Toc10468763)

# 前言

文中所使用的tid为客户流水号，不能为空。

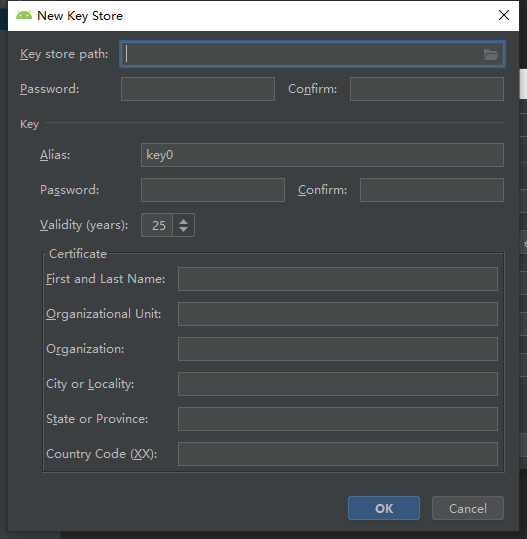
如果需要无UI SDK只需要使用sdk-release.aar文件。

如果需要带UI SDK 需要使用sdk-release.aar和ui-release.aar文件。

# 接入准备

第一步：创建Android项目（项目需要支持kotlin环境）

第二步：生成Android APP的签名文件



第三步：修改app 运行使用的签名文件

android {

...

signingConfigs {

release {

keyAlias 'g3yyt'

keyPassword '123456789'

storeFile file('../key')

storePassword '123456789'

}

}

buildTypes {

debug {

minifyEnabled false

signingConfig signingConfigs.release

}

release {

minifyEnabled true

proguardFilesget DefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

signingConfig signingConfigs.release

}

}

//需要使用java1.8环境

compileOptions {

sourceCompatibility = '1.8'

targetCompatibility = '1.8'

}

}

第四步：添加需要的依赖

implementation 'com.google.android.agera:content:1.3.0'

implementation 'com.alibaba:fastjson:1.2.44'

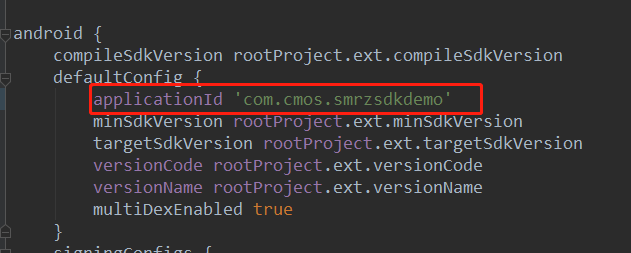
implementation 'org.ligboy.retrofit2:converter-fastjson-android:2.1.0'

implementation 'me.drakeet.retrofit2:adapter-agera:2.1.0'

implementation 'com.squareup.okhttp3:okhttp:3.10.0'

implementation 'com.squareup.retrofit2:retrofit:2.3.0'

第五步：得到应用的applicationId



第六步：通过Android Studio获取签名文件的sha1值

打开 Android Studio 的 Terminal 工具。

输入命令：keytool -v -list -keystore  keystore文件路径。

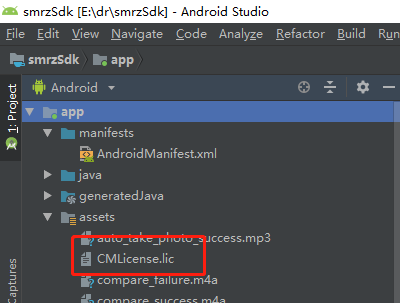
输入 Keystore 密码



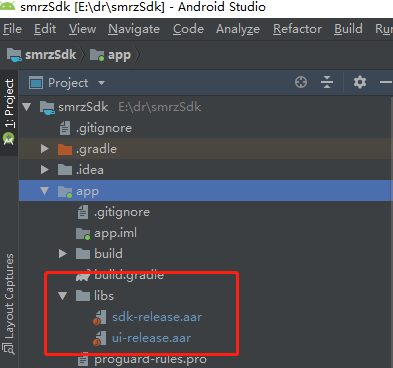
第七步：注册应用（会使用到applicationId和sh1值），并下载授权文件*CMLicense.lic，绑定设备序列号*

第八步：把授权文件*CMLicense.lic放在Android 的assets目录*

*例如：*



第九步：下载SDK并复制到libs目录



第十步：使用SDK

implementation fileTree(include: ['\*.jar', '\*.aar'], dir: 'libs')

第十一步：添加权限（请注意Android 6.0及以上的系统动态权限申请，在调用能力之前确保权限已经获取）

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

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

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

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

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

# 开发指南

## SDK初始化

建议在Application中进行调用初始化，如果初始化失败了，可以再次调用该方法进行初始化

SmrzSdk.getInstance().initEnv(getApplicationContext(), new Callback() {

@Override

public void onSuccess() {

Log.e("init", "success");

}

@Override

public void onFail(String msg) {

Log.e("init", "fail:" + msg);

}

});

## 读取身份证

### 无UI读取身份证

请参考demo NoUIReadCardActivity.java文件

第一步：获取读取身份证对象

IDCardReader reader = SmrzSdk.getInstance().getIDCardReader();

第二步：打开读取设备和设置结果回调

/\*\*

\* 打开读取模块

\*

\* @param context 上下文

\* @param callback 身份证读取回调

\*/

void open(@NotNull Activity context, IDCardReadCallback callback)

第三步：在Android 生命周期方法中调用对象方法

在onResume()生命周期方法中调用：

/\*\*

\* 开始读取，sam方式可以重复调用（读取成功或失败后），nfc方式不支持重复调用

\* </br>sam:R1、P1、D1、D2;nfc:R2、P2

\*

\* @param transactionId 流水号，必传，不能为空

\*/

void startRead(String transactionId);

在onPause()生命周期方法中调用：

/\*\*

\* 暂停读取

\*/

void pauseRead();

在onDestroy()生命周期方法中调用：

/\*\*

\* 关闭设备

\*/

void close();

在Activity 的onNewIntent()方法中调用：

/\*\*

\* 在Activity的onNewIntent回调中调用此方法，用于nfc方式读取身份证

\*

\* @param intent Activity的onNewIntent回调的参数intent

\*/

void onNewIntent(Intent intent);

### 使用Activity套用Fragment方式读取身份证

请参考demo FragmentUIReadCardActivity.java 文件

第一步 创建对象：readCardFragment = new ReadCardFragment();

第二步 设置结果回调：readCardFragment.setIDCardReadCallback(this);

第三步 设置流水号：

readCardFragment.setTransactionId(tid);

第四步 在Activity的onNewIntent(Intent intent)方法中调用：

readCardFragment.onNewIntent(intent);

### 调用Activity读取身份证

请参考 demo UIReadCardActivity.java文件

直接调用Activity读取身份证，在onActivityResult方法中接收结果：

例如：

调用Activity

Intent intent = new ReadCardActivity.Builder(“tid”

.build(UIReadCardActivity.this);

startActivityForResult(intent, request\_code);

接收结果：

private void handlerReadCardUIResult(int resultCode, Intent data) {

if (resultCode == RESULT\_CANCELED) {

idcardInfo.setText("取消读取身份证");

} else if (resultCode == RESULT\_OK) {

int code = data.getIntExtra(ReadCardActivity.CODE, -1);

if (code == -1) {

//出现未知错误，请反馈

return;

}

//0:读取成功；1：读取失败

if (code == 0) {

IdCard idCard = (IdCard) data.getSerializableExtra(ReadCardActivity.RESULT\_KEY\_NAME);

} else {

Toast.makeText(this, data.getStringExtra(ReadCardActivity.ERR\_MESSAGE), Toast.LENGTH\_SHORT).show();

}

}

}

## 人像比对

人像比对建议使用全屏方式，比对倒计时建议为15秒

### 无UI人像比对

请参考 demo NoUICompareActivity.java文件

第一步　在布局文件中使用xml：

<com.cmos.smrz.compare.view.FaceCompareView

android:id="@+id/faceCompare"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

第二步 在onCreate()生命周期方法中调用：

得到View：faceCompareView = findViewById(R.id.faceCompare);

设置数据：faceCompareView.setData(avatar, liveness,tid);

设置回调：faceCompareView.setCallback(this);

第三步 在其他生命周期方法中调用：

在onResume方法中调用：faceCompareView.onResume();

在onPause方法中调用：faceCompareView.onPause();

在onDestroy方法中调用：faceCompareView.onDestroy();

第四步 比对倒计时结束后调用：

faceCompareView.terminate();



### 使用Activity套用Fragment方式人像比对

请参考 demo FragmentUIFaceCompareActivity.java 文件

第一步 创建对象：fragment = FaceCompareFragment.create(avatar, true,tid);

第二步 设置回调：fragment.setCompareCallback(this);



### 调用人像比对Activity

请参考 demo UIFaceCompareActivity.java文件

直接调用Activity 进行比对，在onActivityResult方法中接收结果：

例如：

调用Activity：

Intent intent = new FaceCompareActivity.Builder(tid)

.setContext(this)

.setAvatar(App.avatar)

.setLiveness(true)

.setCountTime(20)

.build();

startActivityForResult(intent, request\_code);

接收结果：

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (resultCode == RESULT\_OK) {

int code = data.getIntExtra(FaceCompareActivity.CODE, -1);

if (code == -1) {

Toast.makeText(this, "出现位置错误，请上报", Toast.LENGTH\_SHORT).show();

return;

}

String imagePath = null;

switch (code) {

case FaceCompareActivity.COMPARE\_SUCCESS:

imagePath=data.getStringExtra(FaceCompareActivity.IMAGE\_PATH);

compare\_bitmap.setImageBitmap(BitmapFactory.decodeFile(imagePath));

resultAlert.setText("人脸比对成功\r\n比对分数：" + data.getFloatExtra(FaceCompareActivity.COMPARE\_SCORE, -1) + "\r\n活体检测分数" + data.getFloatExtra(FaceCompareActivity.LIVENESS\_SCORE, -1));

break;

case FaceCompareActivity.COMPARE\_FAILURE:

imagePath=data.getStringExtra(FaceCompareActivity.IMAGE\_PATH);

compare\_bitmap.setImageBitmap(BitmapFactory.decodeFile(imagePath));

resultAlert.setText("人脸比对失败\r\n比对分数：" + data.getFloatExtra(FaceCompareActivity.COMPARE\_SCORE, -1) + "\r\n活体检测分数" + data.getFloatExtra(FaceCompareActivity.LIVENESS\_SCORE, -1));

break;

case FaceCompareActivity.LIVENESS\_FAILURE:

imagePath = data.getStringExtra(FaceCompareActivity.IMAGE\_PATH);

compare\_bitmap.setImageBitmap(BitmapFactory.decodeFile(imagePath));

resultAlert.setText("活体检测失败\r\n活体检测分数" + data.getFloatExtra(FaceCompareActivity.LIVENESS\_SCORE, -1));

break;

case FaceCompareActivity.NO\_FACE:

compare\_bitmap.setImageBitmap(null);

resultAlert.setText("没有检测到人脸");

break;

}

}

}

## 无证比对

离线模式不能使用

### 调用无证比对Activity

请参考 demo MyNoCardCompareActivity.java 文件

例如：

调用Activity:

Intent build = new NoCardCompareActivity.Builder(tid).build(this);

startActivityForResult(build, request\_code);

接收结果:

NoCardCompareActivity.SRC\_RESULT 的内容参考调用接口比对的回调结果

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == request\_code && resultCode == RESULT\_OK) {

int code = data.getIntExtra(NoCardCompareActivity.CODE, -1);

switch (code) {

case NoCardCompareActivity.COMPARE\_SUCCESS:

Toast.makeText(this, "无证比对通过", Toast.LENGTH\_SHORT).show();

LogUtils.e(this, data.getStringExtra(NoCardCompareActivity.SRC\_RESULT));

break;

case NoCardCompareActivity.COMPARE\_FAIL:

Toast.makeText(this, "无证比对失败", Toast.LENGTH\_SHORT).show();

LogUtils.e(this, data.getStringExtra(NoCardCompareActivity.SRC\_RESULT));

break;

case NoCardCompareActivity.ERROR:

Toast.makeText(this, "无证比对出现错误", Toast.LENGTH\_SHORT).show();

LogUtils.e(this, data.getStringExtra(NoCardCompareActivity.ERROR\_MESSAGE));

break;

}

}

}

### 调用接口比对

SmrzSdk.getInstance().noCardCompare(tid, id, name, img, new ResultCallback() {

@Override

public void onSuccess(String result) {

result.setText(result);

}

@Override

public void onError(String s) {

result.setText(s);

}

});

onSuccess(String result)回调结果：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 父元素名称 | 约束 | 类型 | 宽度 | 描述 | 取值说明 |
| returnCode |  | 非空 | String | 4 | 返回编码 | 0000 |
| returnMessage |  | 非空 | String | 200 | 返回描述 | Ok |
| bean |  | 非空 | Map |  |  |  |
| result | bean | 非空 | String | 10 | 对比结果 | 1：通过  0：未通过 |
| faceResult | bean | ? | String |  | 静默比对结果 | 1：通过  0：未通过 |
| faceScore |  | ? | String |  | 静默得分 | 89.01 |

json内容为：

{

"returnCode": "0000",

"returnMessage": "ok",

"bean": {

"result": "0",

},

"beans":[],

"ext1":[]

}

返回编码

返回值 描述

0000 处理成功

2991 Token验证失败

2994 必填参数为空

2998 与服务器系统时间不符，相差不能超过5分钟

2999 其他错误，根据消息进行提示即可

2500 服务器发生错误

## 图片比对

请参考 demo TwoImageCompareActivity.java文件。离线模式不支持后台比对。

### 本地比对

SmrzSdk.getInstance().compare(bitmap1,

bitmap2,

new CompareCallbackForBitmap() {

@Override

public void onSuccess(float score) {

result.setText("比对分数：" + score);

}

@Override

public void onFail(int code, String message) {

result.setText("比对失败->code:" + code + ",错误信息：" + message);

}

});

### 后台比对

SmrzSdk.getInstance().compare(tid,

imgBytes1,

imgBytes2, new ResultCallback() {

@Override

public void onSuccess(String jsonStr) {

result.setText(jsonStr);

}

@Override

public void onError(String message) {

result.setText(message);

}

});

onSuccess(String jsonStr)回调结果：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **元素名称** | **父元素名称** | **约束** | **类型** | **宽度** | **描述** | **取值说明** |
| returnCode |  | 非空 | String | 4 | 返回编码 | 0000 |
| returnMessage |  | 非空 | String | 200 | 返回描述 | Ok |
| bean |  | 非空 | Map |  |  |  |
| idntifResult | bean | 非空 | String | 10 | 对比结果 | 0：通过  1：有比分，但是分值比对失败  2：未检测到人脸  3：其他失败原因 |
| resultScore | bean | 非空 | String | 10 | 比对分值 |  |
| transactionID | bean | 非空 | String | 20 | 流水号 |  |

Json内容为：

{

"returnCode": "0000",

"returnMessage": "ok",

"bean": {

"idntifResult": "0",

"resultScore": "90",

"transactionId": "xxxxxxxxxxxxxxxxxxx"

},

"beans":[],

"ext1":[]

}

返回编码

返回值 描述

0000 处理成功

2990 未检测到真人

2991 Token验证失败

2992 未开通两两比对，使用前需申请开通

2994 必填参数为空

2997 图片保存失败

2998 与服务器系统时间不符，相差不能超过5分钟

2999 其他错误，根据消息进行提示即可

2500 服务器发生错误

## 标准比对

调用Activity 读取身份证成功后直接进行比对（请参考demo OneKeyCompareActivity.java 文件）。

调用Activity：

Intent intent = new StandardCompareActivity.Builder(tid)

.setCountTime(20)

.setLiveness(true)

.build(this);

startActivityForResult(intent, request\_code);

接收结果：

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

String errMsg, path;

Bitmap avatar;

IdCard idCard;

float score, livenessScore;

if (requestCode == request\_code && resultCode == RESULT\_OK) {

int code = data.getIntExtra(StandardCompareActivity.CODE, -1);

if (code == -1) {

Toast.makeText(this, "出现未知错误，请上报", Toast.LENGTH\_SHORT).show();

return;

}

switch (code) {

case StandardCompareActivity.READ\_IDCARD\_ERROR:

errMsg = data.getStringExtra(StandardCompareActivity.ERROR\_MESSAGE);

Toast.makeText(this, "读取身份证错误：" + errMsg, Toast.LENGTH\_SHORT).show();

alertMsg.setText(errMsg);

break;

case StandardCompareActivity.COMPARE\_SUCCESS:

idCard = (IdCard) data.getSerializableExtra(StandardCompareActivity.IDCARD\_RESULT);

score = data.getFloatExtra(StandardCompareActivity.COMPARE\_SCORE, 0);

livenessScore = data.getFloatExtra(StandardCompareActivity.LIVENESS\_SCORE, 0);

path = data.getStringExtra(StandardCompareActivity.IMAGE\_PATH);

alertMsg.setText("比对成功\r\n姓名:" + idCard.getName() + "\r\n比对分数:" + score + "\r\n活体检测分数:" + livenessScore);

avatar = BitmapFactory.decodeByteArray(idCard.getAvatar(), 0, idCard.getAvatar().length);

if (avatar == null) {

//TODO 如果需要芯片照请申请

LogUtils.e(this, "如果需要芯片照请申请");

} else {

idAvatar.setImageBitmap(avatar);

}

if (!TextUtils.isEmpty(path)) {

compareAvatar.setImageBitmap(BitmapFactory.decodeFile(path));

}

break;

case StandardCompareActivity.COMPARE\_FAILURE:

idCard = (IdCard) data.getSerializableExtra(StandardCompareActivity.IDCARD\_RESULT);

score = data.getFloatExtra(StandardCompareActivity.COMPARE\_SCORE, 0);

livenessScore = data.getFloatExtra(StandardCompareActivity.LIVENESS\_SCORE, 0);

path = data.getStringExtra(StandardCompareActivity.IMAGE\_PATH);

alertMsg.setText("比对失败\r\n姓名:" + idCard.getName() + "\r\n比对分数:" + score + "\r\n活体检测分数:" + livenessScore);

avatar = BitmapFactory.decodeByteArray(idCard.getAvatar(), 0, idCard.getAvatar().length);

if (avatar == null) {

LogUtils.e(this, "如果需要芯片照请申请");

} else {

idAvatar.setImageBitmap(avatar);

}

if (!TextUtils.isEmpty(path)) {

compareAvatar.setImageBitmap(BitmapFactory.decodeFile(path));

}

break;

case StandardCompareActivity.COMPARE\_NO\_FACE:

idCard = (IdCard) data.getSerializableExtra(StandardCompareActivity.IDCARD\_RESULT);

alertMsg.setText("没有检测到人脸\r\n姓名:" + idCard.getName());

avatar = BitmapFactory.decodeByteArray(idCard.getAvatar(), 0, idCard.getAvatar().length);

if (avatar == null) {

LogUtils.e(this, "如果需要芯片照请申请");

} else {

idAvatar.setImageBitmap(avatar);

}

break;

case StandardCompareActivity.COMPARE\_LIVENESS\_FAILURE:

idCard = (IdCard) data.getSerializableExtra(StandardCompareActivity.IDCARD\_RESULT);

score = data.getFloatExtra(StandardCompareActivity.COMPARE\_SCORE, 0);

livenessScore = data.getFloatExtra(StandardCompareActivity.LIVENESS\_SCORE, 0);

path = data.getStringExtra(StandardCompareActivity.IMAGE\_PATH);

alertMsg.setText("活体检测失败\r\n姓名:" + idCard.getName() + "\r\n活体检测分数:" + livenessScore);

avatar = BitmapFactory.decodeByteArray(idCard.getAvatar(), 0, idCard.getAvatar().length);

if (avatar == null) {

LogUtils.e(this, "如果需要芯片照请申请");

} else {

idAvatar.setImageBitmap(avatar);

}

if (!TextUtils.isEmpty(path)) {

compareAvatar.setImageBitmap(BitmapFactory.decodeFile(path));

}

break;

}

}

}

## 拍照

拍照分为普通拍照和自动抓拍人脸。自动抓拍人脸支持设置活体检测、显示人脸轮廓、提醒显示。推荐使用全屏方式。

### 无UI普通拍照

请参考demo NoUINormalTakePhotoActivity.java文件。

第一步 布局文件使用xml:

<com.cmos.smrz.takephoto.view.CameraView

android:id="@+id/camera"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

第二步 调用CameraView 的生命周期方法：

cameraView.onResume();

cameraView.onPause();

cameraView.onDestroy();

第三步 拍照：

cameraView.takePicture(tid, (data, camera) -> {

})

### 无UI 自动抓拍

请参考demo NoUIAutoTakePhotoActivity.java 文件。

第一步 布局文件使用xml:

<com.cmos.smrz.takephoto.view.CameraView

android:id="@+id/camera"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

第二步 调用CameraView 的生命周期方法：

cameraView.onResume();

cameraView.onPause();

cameraView.onDestroy();

第三步 设置拍照参数和回调：

/\*\*

\* 使用自动抓拍

\*

\* @param transactionId 流水号，不能为空

\* @param callback 回调

\* @param liveness 活体检测开关

\* @param showFace 是否显示人脸轮廓

\*/

cameraView.autoTakePhoto(transactionId, callback, liveness, showFace);

### Activity 嵌套Fragment普通拍照

请参考demo FragmentUINormalTakePhotoActivity.java 文件

第一步 创建Fragment:

fragment = TakePhotoFragment.createFragment();

第二步 拍照：

fragment.takePicture(tid, (data, camera) ->{})

### Activity 嵌套 Fragment 自动抓拍

请参考demo FragmentUIAutoTakePhotoActivity.java 文件

第一步 创建Fragment:

fragment = TakePhotoFragment.createFragment(tid, true, true);

第二步 设置回调：

fragment.setAutoTakeCallback(this);

第三步 抓拍超时调用：

fragment.terminate();

### 调用拍照Activity

请参考demo UITakePhotoActivity.java文件。

调用Activity ：

普通拍照：

Intent intent = new TakePhotoActivity.Builder(tid)

.useNormalTakePhoto(showOverlay)

.build(this);

startActivityForResult(intent, normal\_request\_code);

自动拍照：

Intent intent = new TakePhotoActivity.Builder(tid)

.useAutoTakePhoto(liveness, true, false, 15)

.build(this);

startActivityForResult(intent, auto\_request\_code);

接收结果：

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

srcImage.setImageBitmap(null);

avatarImage.setImageBitmap(null);

if (resultCode == RESULT\_OK) {

if (requestCode == auto\_request\_code) {

//success = 0, liveness\_fail = 1, no\_face = 2, error = 3;

int code = data.getIntExtra(TakePhotoActivity.CODE, -1);

switch (code) {

case 0:

String srcPath = data.getStringExtra(TakePhotoActivity.SRC\_IMAGE\_PATH\_KEY);

String avatarPath = data.getStringExtra(TakePhotoActivity.AVATAR\_IMAGE\_PATH\_KEY);

srcImage.setImageBitmap(BitmapFactory.decodeFile(srcPath));

avatarImage.setImageBitmap(BitmapFactory.decodeFile(avatarPath));

if (liveness)

Toast.makeText(this, "活体检测分值：" + data.getFloatExtra(TakePhotoActivity.LIVENESS\_SCORE, -1), Toast.LENGTH\_SHORT).show();

break;

case 1:

srcImage.setImageBitmap(BitmapFactory.decodeFile(data.getStringExtra(TakePhotoActivity.SRC\_IMAGE\_PATH\_KEY)));

Toast.makeText(this, "活体检测失败,分值：" + data.getFloatExtra(TakePhotoActivity.LIVENESS\_SCORE, -1), Toast.LENGTH\_SHORT).show();

break;

case 2:

Toast.makeText(this, "没有检测到人脸", Toast.LENGTH\_SHORT).show();

break;

case 3:

Toast.makeText(this, data.getStringExtra(TakePhotoActivity.ERROR\_MSG), Toast.LENGTH\_SHORT).show();

break;

}

} else if (requestCode == normal\_request\_code) {

//success = 0,error = 3;

int code = data.getIntExtra(TakePhotoActivity.CODE, -1);

switch (code) {

case 0:

String srcPath = data.getStringExtra(TakePhotoActivity.SRC\_IMAGE\_PATH\_KEY);

srcImage.setImageBitmap(BitmapFactory.decodeFile(srcPath));

if (showOverlay) {

String avatarPath = data.getStringExtra(TakePhotoActivity.AVATAR\_IMAGE\_PATH\_KEY);

avatarImage.setImageBitmap(BitmapFactory.decodeFile(avatarPath));

}

break;

case 3:

Toast.makeText(this, "错误：" + data.getStringExtra(TakePhotoActivity.ERROR\_MSG), Toast.LENGTH\_SHORT).show();

break;

}

}

}

}

## 替他接口

请参考Java doc 帮助手册。

