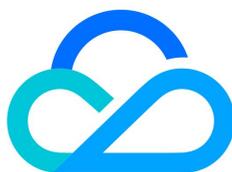


# 人脸核身 实践教程



腾讯云

**【 版权声明 】**

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分的内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

**【 商标声明 】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

**【 服务声明 】**

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

**【 联系我们 】**

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

## 文档目录

### 实践教程

用腾讯云 AI 意愿核身为电话合规保驾护航

在线考试场景身份核验实践

## 实践教程

# 用腾讯云 AI 意愿核身为电话合规保驾护航

最近更新时间：2024-05-16 15:21:01

腾讯云 AI 意愿核身功能是结合人脸核身与实时音视频技术打造的一款满足实名、实人、真实意愿的产品，还是首批通过国家信通院人脸识别评估的产品，并且荣获四级（优秀级）安全防护等级。不仅如此，腾讯云 AI 意愿核身也提供了微信小程序、微信 H5、SDK 等多种接入方式。

本文将详细介绍如何接入腾讯云 AI 意愿核身，降本增效、更加智能地实现对高危电话用户进行二次实名、实名、语音认证等功能，提醒用户使用的电话卡涉嫌法律风险等，用于预防电信诈骗，为电话合规保驾护航。

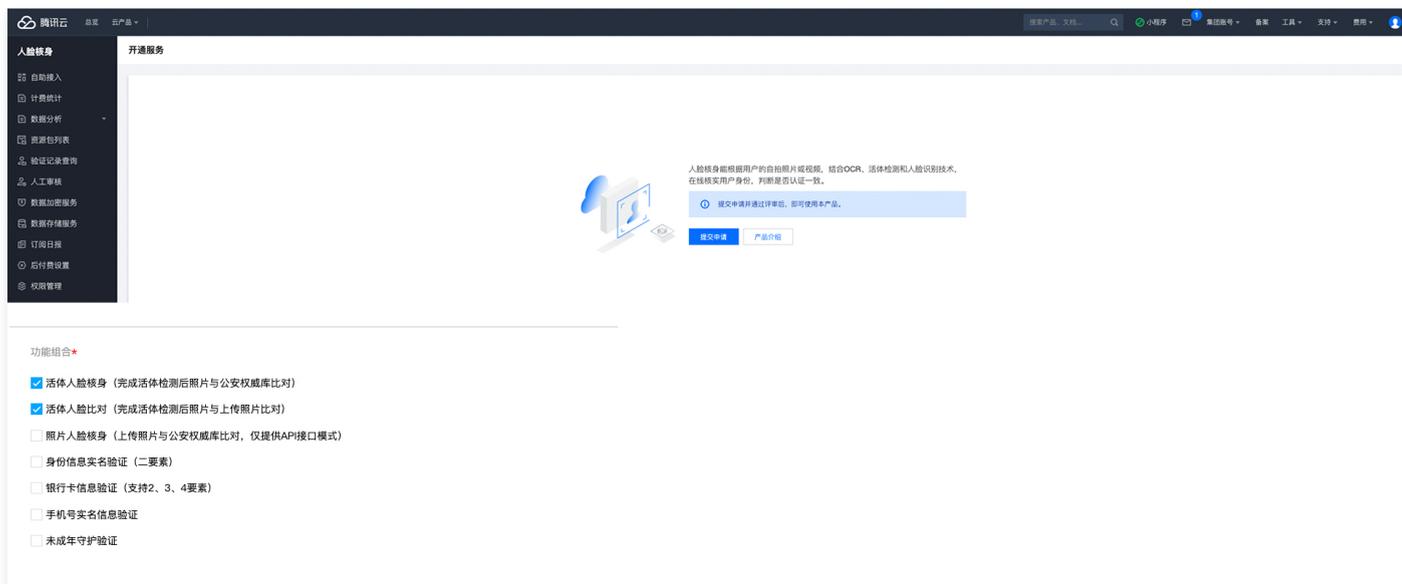
## 准备工作

### 1. 开通人脸核身

登录 [人脸核身控制台](#)，若首次使用人脸核身服务，需要先开通人脸核身服务。单击**提交申请**，按照实际情况填写信息，即可提交申请。

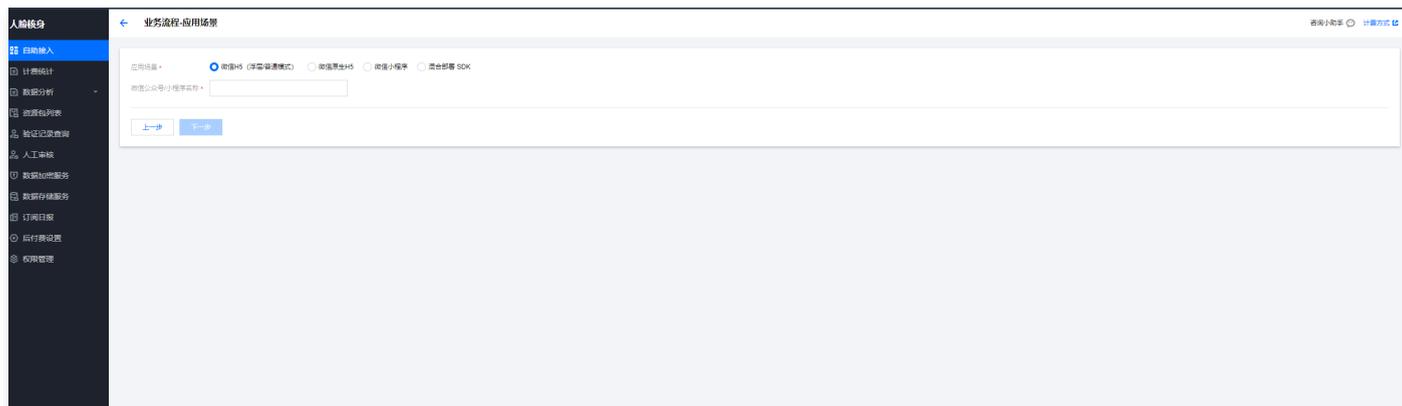
#### 说明

- 互联网行业和金融行业必须上传业务相关营业资质。
- 本次接入需要的功能组合为：活体人脸核身（完成活体检测后照片与权威库比对）和活体人脸比对（完成活体检测后照片与上传照片比对）。



### 2. 业务申请

根据业务需求，选择相对应的接入模式，**申请对应的业务**。



- 选择接入方式，微信原生 H5 的接入方式有行业限制，且资质文件中主体与需要接入公众号主体一致，详细行业类别和资质材料请查阅 [微信 HTML5 及小程序资质文件列表](#)。
- 如果需要申请微信原生 H5 和微信小程序，但是没有对应的微信小程序和公众号，可以前往 [微信公众平台](#) 进行申请。

3. 了解腾讯云 AI 意愿核身

意愿核身详细内容请参见 [意愿核身接入指引](#)、[计费错误码](#) 等。

4. 意愿核身配置

相关业务通过审核后，请联系 [人脸核身助手](#)，完成意愿核身业务的配置。

RuleID	业务名称	应用场景	需求状态	最近修改时间	可用流程配置	操作
		H5页面 (普通模式)	已通过	2022-02-23 17:58:04	<a href="#">查看配置</a> <a href="#">接口文档</a>	<a href="#">编辑</a> <a href="#">更多</a>
		H5页面 (普通模式)	已通过	2022-02-23 10:15:05	<a href="#">查看配置</a> <a href="#">接口文档</a>	<a href="#">编辑</a> <a href="#">更多</a>
		H5页面 (普通模式)	已通过	2021-12-06 15:56:49	<a href="#">查看配置</a> <a href="#">接口文档</a>	<a href="#">编辑</a> <a href="#">更多</a>
		H5页面 (普通模式)	已通过	2021-11-18 19:38:59	<a href="#">查看配置</a> <a href="#">接口文档</a>	<a href="#">编辑</a> <a href="#">更多</a>
		H5页面 (普通模式)	已通过	2021-09-14 14:46:53	<a href="#">查看配置</a> <a href="#">接口文档</a>	<a href="#">编辑</a> <a href="#">更多</a>
		H5页面 (普通模式)	已通过	2021-09-13 19:35:36	<a href="#">查看配置</a> <a href="#">接口文档</a>	<a href="#">编辑</a> <a href="#">更多</a>
		微信小程序	已通过	2021-09-09 17:10:07	<a href="#">下载小程序SDK</a> <a href="#">查看配置</a> <a href="#">接口文档</a>	<a href="#">编辑</a> <a href="#">更多</a>
		H5页面 (普通模式)	已停用	2021-12-06 16:51:59	<a href="#">查看配置</a> <a href="#">接口文档</a>	<a href="#">编辑</a> <a href="#">启用</a>

5. 获取 API 密钥

登录官网控制台 [创建 API 密钥](#) (SecretId 和 SecretKey)。

**注意**

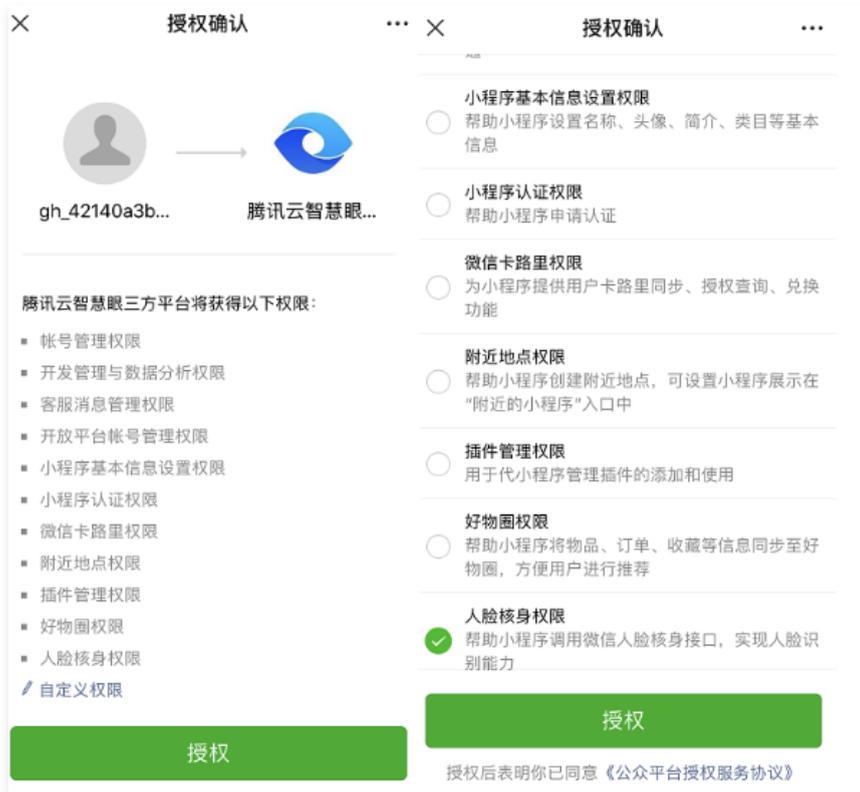
API 密钥需要妥善保管。

接入意愿核身

第一步：授予权限与设置白名单

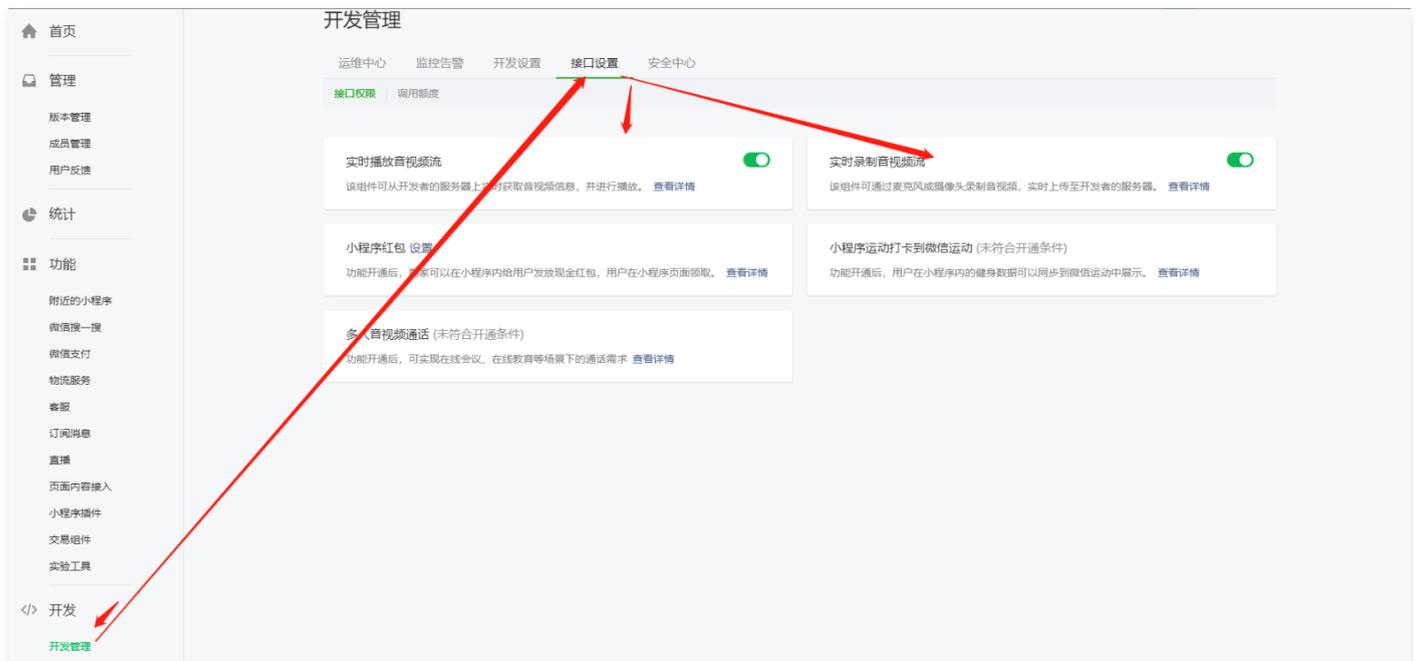
• 为人脸核身授权

小程序开发需要授权，打开 [二维码](#)，小程序管理员扫码后，单击自定义权限，只勾选人脸核身权限，将该权限授权给人脸核身第三方平台。



● 开启实时播放/录制音视频流权限

使用腾讯云AI 意愿核身需要开启实时播放音视频流 (live-player) 和实时录制音视频流 (live-pusher)，登录 [微信公众平台](#) 开启对应的权限。



● 相关域名添加白名单

登录 [微信公众平台](#)，给意愿核身需要的域名添加白名单权限，相关操作如下所示：

- 将以下域名添加至 request 合法域名：  
`https://events.tim.qq.com;https://faceid.qq.com;https://grouptalk.c2c.qq.com`。  
`https://pingtas.qq.com;https://web.sdk.qcloud.com;https://webim.tim.qq.com;https://yun.tim.qq.com`。
- 将以下域名添加至 socket 合法域名：`wss://wss.im.qcloud.com;wss://wss.tim.qq.com`。
- 将以下域名添加至 uploadFile 合法域名：`https://cos.ap-shanghai.myqcloud.com;https://faceid.qq.com`。

○ 将以下域名添加到 uploadFile 合法域名: `https://cos.ap-shanghai.myqcloud.com;https://faceid.qq.com`。



## 第二步：初始化意愿核身 SDK

下载 [意愿核身小程序 SDK](#)，将 verify\_mpsdk 文件夹放到小程序项目根目录，调用 init 方法，初始化意愿核身 SDK。

```
//app.js
App({
  onLaunch: function () {
    // 初始化意愿核身组件
    const Verify = require('/verify_mpsdk/main');
    Verify.init();
  }
})
// app.json
{
  "pages": [
    "verify_mpsdk/index/index"
  ]
}
```

微信小程序接入意愿核身，可参考 [意愿核身 demo](#)。

## 第三步：后端接入意愿核身实名核身鉴权接口

调用意愿核身 [实名核身鉴权接口](#)，获取意愿核身流程标识 BizToken。Java SDK 的引入可参考 [Java SDK 接入指引](#)。

### ⚠ 注意

需传入意愿核身所需字段 IntentionVerifyText。

意愿核身提供了多种主流语言接入，包含 Java、Python、Go 等，以 Java 为例：

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
import com.tencentcloudapi.faceid.v20180301.FaceidClient;
import com.tencentcloudapi.faceid.v20180301.models.*;

public class DetectAuth
```

```
{
    public static void main(String [] args) {
        try{
            // 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey,此处还请注意密钥对的保密
            // 密钥可前往https://console.cloud.tencent.com/cam/capi网站进行获取
            Credential cred = new Credential("SecretId", "SecretKey");
            // 实例化一个http选项，可选的，没有特殊需求可以跳过
            HttpProfile httpProfile = new HttpProfile();
            httpProfile.setEndpoint("faceid.tencentcloudapi.com");
            // 实例化一个client选项，可选的，没有特殊需求可以跳过
            ClientProfile clientProfile = new ClientProfile();
            clientProfile.setHttpProfile(httpProfile);
            // 实例化要请求产品的client对象,clientProfile是可选的
            FaceidClient client = new FaceidClient(cred, "", clientProfile);
            // 实例化一个请求对象,每个接口都会对应一个request对象
            DetectAuthRequest req = new DetectAuthRequest();

            // 返回的resp是一个DetectAuthResponse的实例，与请求对象对应
            DetectAuthResponse resp = client.DetectAuth(req);
            // 输出json格式的字符串回包
            System.out.println(DetectAuthResponse.toJsonString(resp));
        } catch (TencentCloudSDKException e) {
            System.out.println(e.toString());
        }
    }
}
```

#### 第四步：进入意愿核身流程

接入方服务端将 BizToken 返回给接入方小程序，然后小程序调用核身方法 startVerify 进入核身流程。

```
// 单击某个按钮时，触发该函数
gotoVerify: function () {
    // 去接入方服务端调用DetectAuth接口获取BizToken，需要接入方服务端自行实现
    let BizToken = getBizToken();
    // 调用实名核身功能
    wx.startVerify({
        data: {
            token: BizToken // BizToken
        },
        success: (res) => { // 验证成功后触发
            // res 包含验证成功的token
        },
        fail: (err) => { // 验证失败时触发
            // err 包含错误码，错误信息
        }
    });
}
```

#### 第五步：后端接入获取意愿核身结果接口

调用意愿核身 [获取意愿核身结果接口](#)，拉取意愿核身结果。

小程序在完成意愿核身之后，回调我们的后台，我们的后台通过[获取意愿核身结果接口](#)和 BizToken，拉取意愿核身结果。

以 Java 为例：

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
import com.tencentcloudapi.faceid.v20180301.FaceidClient;
```

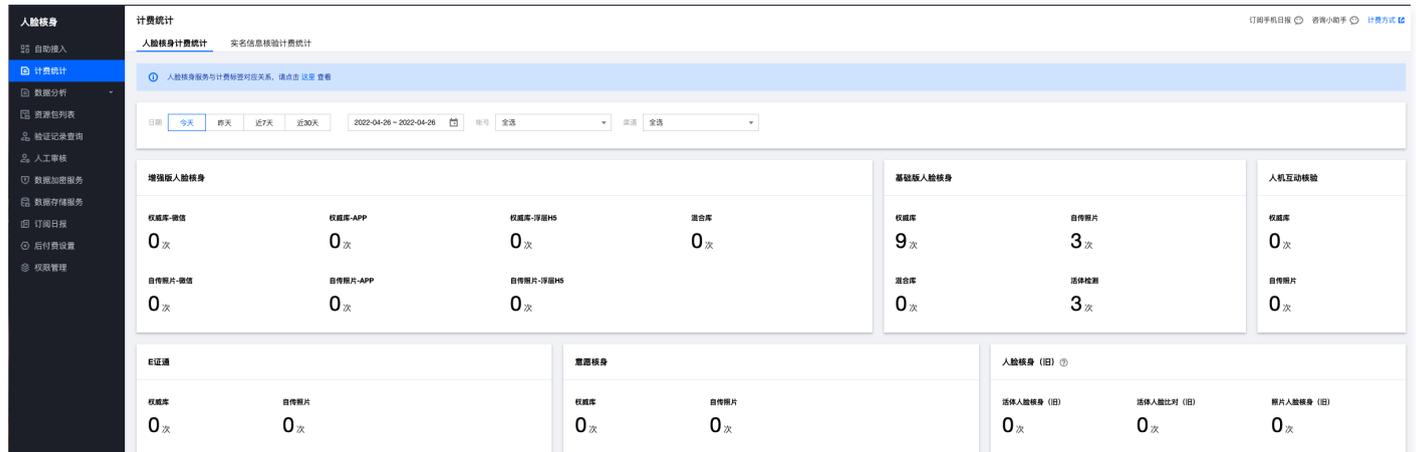
```
import com.tencentcloudapi.faceid.v20180301.models.*;

public class GetDetectInfoEnhanced
{
    public static void main(String [] args) {
        try{
            // 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey,此处还请注意密钥对的保密
            // 密钥可前往https://console.cloud.tencent.com/cam/capi网站进行获取
            Credential cred = new Credential("SecretId", "SecretKey");
            // 实例化一个http选项，可选的，没有特殊需求可以跳过
            HttpProfile httpProfile = new HttpProfile();
            httpProfile.setEndpoint("faceid.tencentcloudapi.com");
            // 实例化一个client选项，可选的，没有特殊需求可以跳过
            ClientProfile clientProfile = new ClientProfile();
            clientProfile.setHttpProfile(httpProfile);
            // 实例化要请求产品的client对象，clientProfile是可选的
            FaceidClient client = new FaceidClient(cred, "", clientProfile);
            // 实例化一个请求对象，每个接口都会对应一个request对象
            GetDetectInfoEnhancedRequest req = new GetDetectInfoEnhancedRequest();

            // 返回的resp是一个GetDetectInfoEnhancedResponse的实例，与请求对象对应
            GetDetectInfoEnhancedResponse resp = client.GetDetectInfoEnhanced(req);
            // 输出json格式的字符串回包
            System.out.println(GetDetectInfoEnhancedResponse.toJsonString(resp));
        } catch (TencentCloudSDKException e) {
            System.out.println(e.toString());
        }
    }
}
```

## 查询调用量

登录 [人脸核身控制台](#)，在计费统计页面，可查看意愿核身的计费量情况。



# 在线考试场景身份核验实践

最近更新时间：2022-08-24 15:54:25

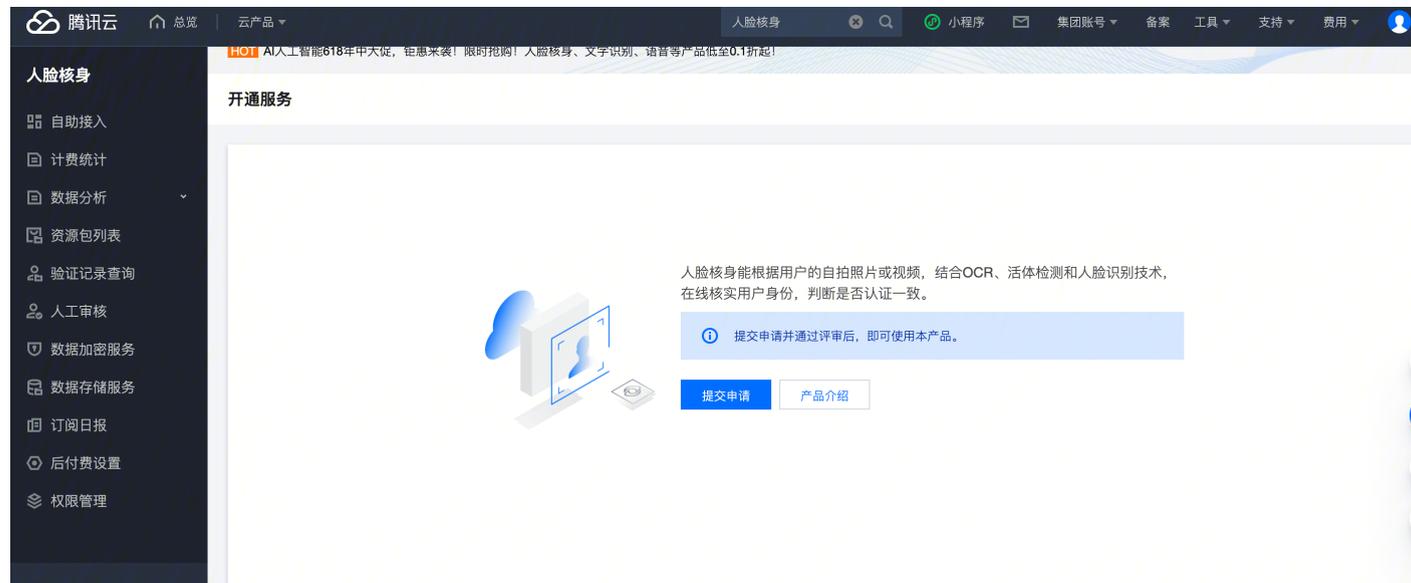
本文档介绍如何使用腾讯云慧眼人脸核身能力，完成在线考试过程中用户身份核验的。不仅满足考试用户身份核验，在考试过程中，如需随机抽选查看是否违规，还可结合人脸识别等能力达到防作弊的管理。

## 准备工作

准备工作包括：开通人脸核身服务、业务申请、了解腾讯云慧眼人脸核身配置、获取账号的云 API 密钥5个步骤。

## 开通人脸核身服务

登录 [人脸核身控制台](#)，首次使用人脸核身服务的用户，需要先开通人脸核身服务。单击**提交申请**，按照实际情况填写信息即可。



### 说明

- 互联网行业和金融行业必须上传业务相关营业资质。
- 腾讯云 E证通服务不仅支持权威库比对，还支持客户自传身份证照片，本实例需要的功能组合为：活体人脸核身（完成活体检测后照片与权威库比对）和身份信息实名验证（二要素）。

## 业务申请

成功开通人脸核身服务后，进入自助接入页面。E证通服务目前只向已完成企业实名认证的腾讯云用户开发，使用服务前需要通过企业资质审核。

1. 单击认证指引按钮进入企业账号认证指引页面，按照指引完成企业账号实名认证。



2. 企业认证完成后即可进行商户 ID 申请，开启 E 证通服务。



## 了解腾讯云 E 证通服务

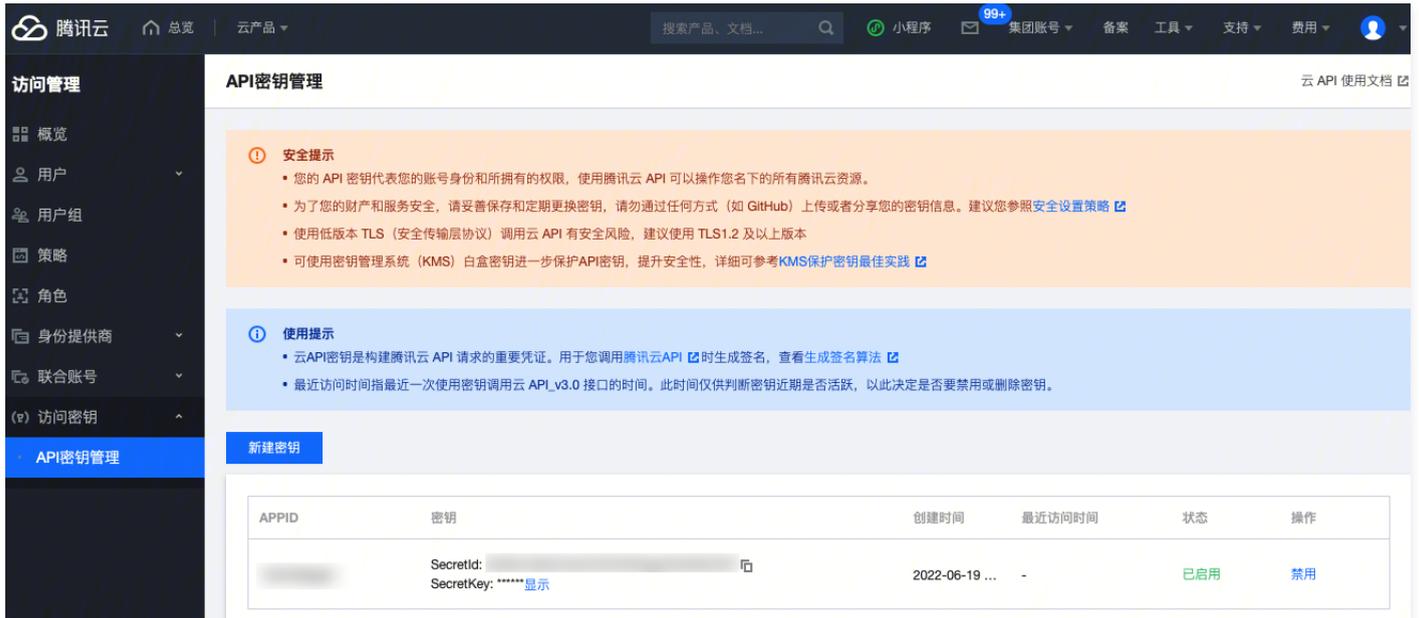
可在官网产品文档中了解 E 证通服务的功能和优势等，详情请参见 [E 证通](#)。

## 获取账号的云 API 密钥

在腾讯云访问管理的 [API 密钥管理](#) 页面，新建或获取个人密钥（SecretId 和 SecretKey）。

### 注意

获取的 API 密钥需要妥善保管。



## 接入 E证通人脸核身服务

### 查看接入文档并选择接入方式

首先查看 [E证通接入文档](#)，E证通支持的接入方式有三种：E证通小程序接入、E证通小程序接入（uni-app）和 E证通 H5 接入。

- E证通小程序接入和E证通小程序接入（uni-app）需要微信小程序支持，实现接入方小程序和 eID 小程序之间的交互逻辑。
- H5 接入方式只需要接入方后台调用 E证通服务接口，就可以得到一个用户可访问的 H5 页面。

结合在线考试身份核验应用场景，选择基于 H5 的 E证通服务进行接入。

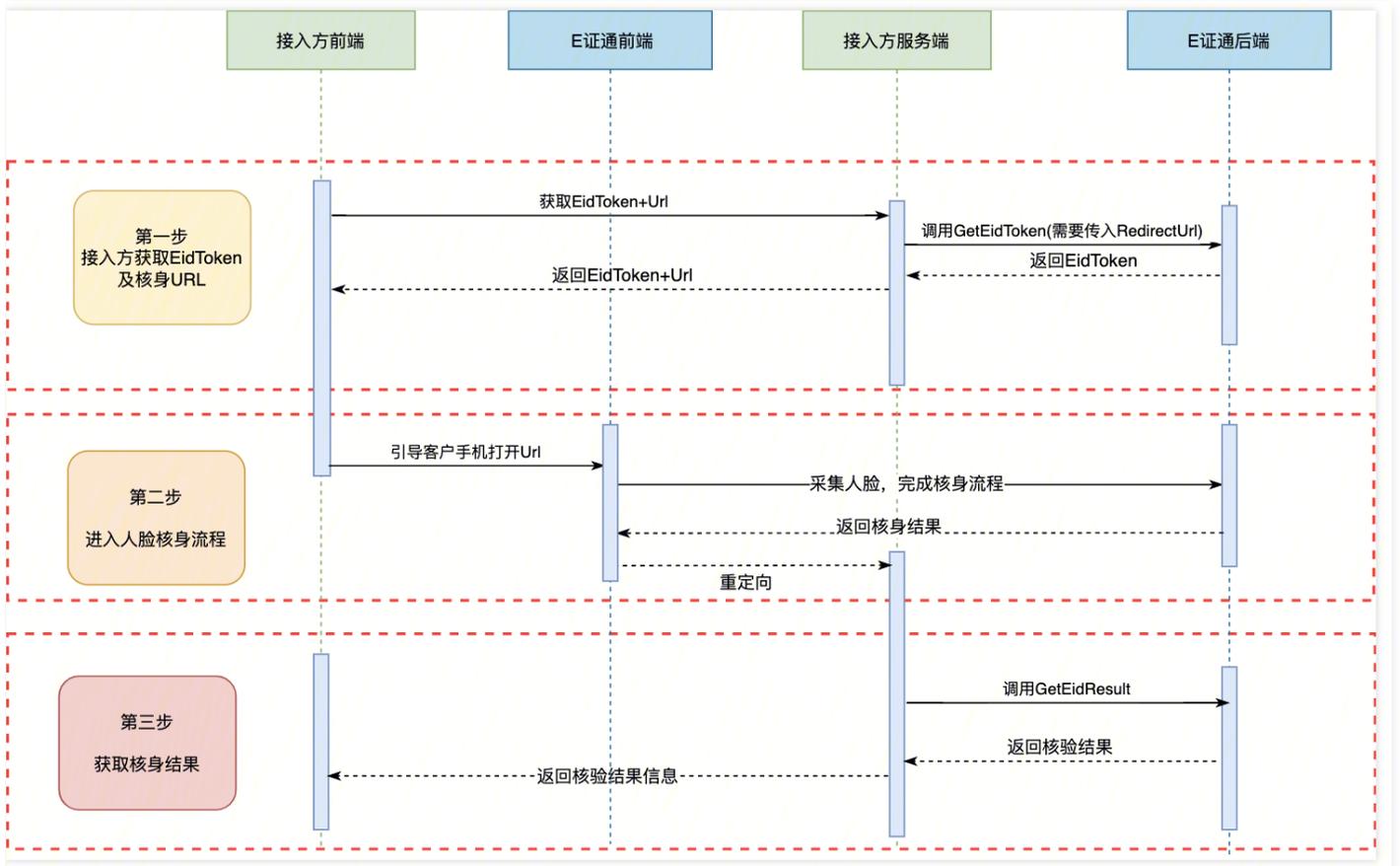
### 确定交互流程

E证通 H5 接入方式需要与 E证通后端接口交互两次：

- 初始时获取 EidToken 和核身 URL。
- 考生完成核身流程后获取核验结果。

获取核验结果有两种方式：轮询和重定向。

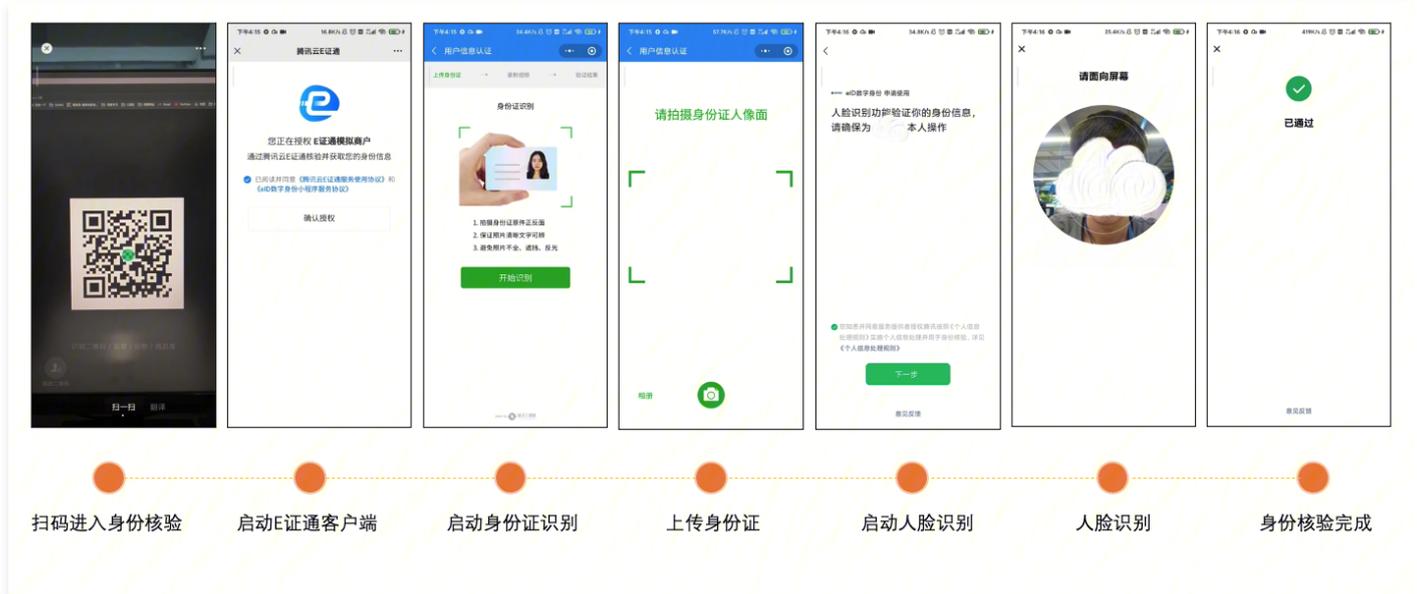
在这个场景下，选择重定向方式，时序图如下所示：



其中，接入方前端为考生个人微信，接入方服务端为我们实现的接入逻辑。整个实现流程为：

1. 给考生发放一个二维码，这个二维码指向接入方调用 GetEidToken 的接口。
2. 考生使用微信扫描二维码后跳转到核身 Url 指定的核身 H5 页面，进行身份证 OCR 识别以及人脸识别身份核验，核身完成后自动调用 RedirectUrl 指定的获取核验结果的接入方后端接口。
3. 由于 E证通的重定向方式会自动将 token 作为参数传入 RedirectUrl 指定的接口，所以接入方获取结果的接口能够拿到 token，调用 E证通后端提供的 GetEidResult 接口，获得核验结果。

通过下面这个实际使用页面中的流程图能够形象地说明这一过程。



### 接入实现

基于前面的分析，我们只需要实现两个接入方后端接口即可完成对 E证通 H5 方式的接入，同时为了实现通过扫描二维码启动身份核验，需要实现一个获取指向 getToken 接口的二维码的接口。

### 主函数中开放出三个接口

实现如下：

```
func main() {
    // http请求路径和接口名
    http.HandleFunc("/getToken", getToken)
    http.HandleFunc("/redirect", getResult)
    http.HandleFunc("/getqrcode", getQRCode)

    // 在IP:Port开启服务
    err := http.ListenAndServe("ServerIP:Port", nil)
    if err != nil {
        fmt.Println("ListenAndServe Error:", err.Error())
    }
}
```

### 三个 HTTP 接口的实现

1. getToken 接口：由考生触发，调用E证通后端 [GetEidToken](#)，获取到核身 Url，考生进行刷脸认证。基于 Go 的实现如下：

```
func getToken(w http.ResponseWriter, req *http.Request) {
    credential := common.NewCredential(
        "SecretId",
        "SecretKey",
    )
    cpf := profile.NewClientProfile()
    cpf.HttpProfile.Endpoint = "faceid.tencentcloudapi.com"
    client, _ := faceid.NewClient(credential, "", cpf)

    request := faceid.NewGetEidTokenRequest()

    // 装填参数
    merchantId := "MerchantId"
    redirectUrl := "http://ServerIP:Port/redirect"
    request.MerchantId = &merchantId
    request.RedirectUrl = &redirectUrl

    // 发起调用
    response, err := client.GetEidToken(request)
    if _, ok := err.(*errors.TencentCloudSDKError); ok {
        fmt.Printf("An API error has returned: %s", err)
        return
    }
    if err != nil {
        panic(err)
    }
    // 控制前端H5页面跳转
    http.Redirect(w, req, *response.Response.Url, 302)
}
```

2. getResult 接口：RedirectUrl 指定，考生在 H5 页面完成核身后，由E证通前端自动调用，传入本次核验的 token，获取核验结果。基于 Go 的实现如下：

```
func getResult(w http.ResponseWriter, req *http.Request) {
    credential := common.NewCredential(
        "SecretId",
```

```
    "SecretKey",
)

cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = "faceid.tencentcloudapi.com"
client, _ := faceid.NewClient(credential, "", cpf)

request := faceid.NewGetEidResultRequest()
// 装填参数
err := req.ParseForm()
if err != nil {
    fmt.Println(err.Error())
}
token, found := req.Form["token"]
if !found {
    fmt.Println("参数解析出错")
}
request.EidToken = &token[0]
// 发起结果查询
response, err := client.GetEidResult(request)
if _, ok := err.(*errors.TencentCloudSDKError); ok {
    fmt.Println("An API error has returned: %s", err)
    return
}
if err != nil {
    panic(err)
}
// 获取结果信息
code := response.Response.Text.ErrCode
name := response.Response.Text.Name
if *code == 0 {
    res := "认证通过, 该用户是" + *name
    fmt.Println(res)
    fmt.Fprint(w, res)
} else {
    res := "!!! 认证未通过, 该用户不是" + *name
    fmt.Println(res)
    fmt.Fprint(w, res)
}
}
```

### 3. 获取二维码的接口实现代码如下:

```
func getQRCode(w http.ResponseWriter, req *http.Request) {
    err1 := req.ParseForm()
    if err1 != nil {
        fmt.Println(err1.Error())
    }
    url := "http://ServerIP:Port/gettoken"
    // 设置http响应头首部行
    w.Header().Set("content-type", "image/png")
    w.Write(stringToQR(url))
}

// 字符串转二维码
func stringToQR(url string) []byte {
    res, err := qrcode.Encode(url, qrcode.Medium, 256)
    if err != nil {
        fmt.Println(err.Error())
    }
}
```

```
return res  
}
```

在接入过程中，可以利用文档的 API 接口范例，得到对接口的调用框架代码，只需要在里面完成自己的逻辑就行。例如获取 E证通 Token 请参见 [API 文档](#)。

推荐使用 API Explorer，详细操作及说明请参见云 API 文档 [使用 API Explorer](#)。