

# 纸纹核验SDK对接文档V0.0.2

本文档主要阐述 纸纹核验SDK 与品牌商小程序的业务交互及开发对接事宜:

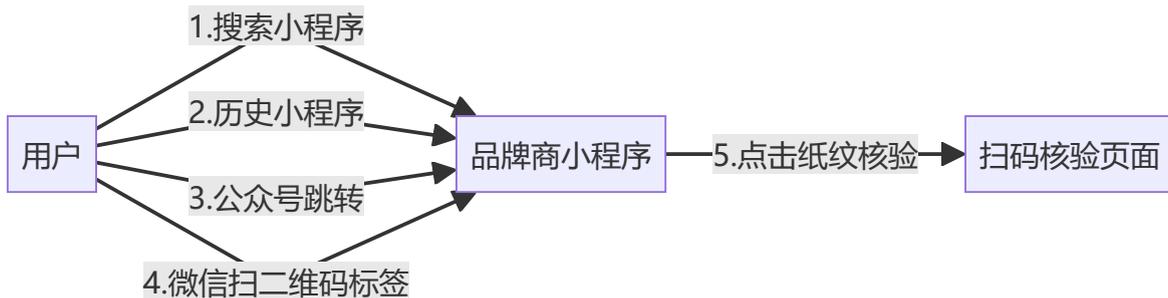
- 1、[品牌商小程序进入 纸纹核验 业务交互场景](#)
- 2、[品牌商小程序与 纸纹核验SDK 的交互时序](#)
- 3、品牌商小程序与 纸纹核验SDK 需要做的开发对接
- 4、授权密钥的管理
- 5、品牌商小程序使用 纸纹核验SDK 需要的后台配置
- 6、品牌商小程序如何更新升级 纸纹核验SDK 的新功能

术语:

- **二维码标签**: 现实场景中贴在商品或外包装上的 印有二维码图案的标签纸, 使用纸纹核验功能扫码可鉴别该标签真伪

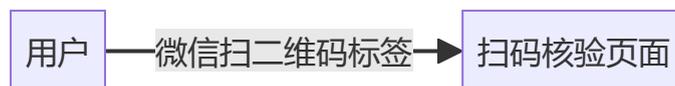
## 1、品牌商小程序进入 纸纹核验 业务交互场景

### 1.1、从品牌商小程序手动点击进入 扫码核验页面



- 1. 用户在微信中搜索品牌商小程序进入
- 2. 用户在微信历史小程序中点击进入
- 3. 用户从品牌商的关联公众号入口进入
- 4. 用户使用微信扫一扫, 扫描商品二维码标签, 直接打开品牌商小程序。需配置关联二维码,见后台配置章节
- 5. 在品牌商小程序中点击 纸纹核验 按钮跳转到 扫码核验页面

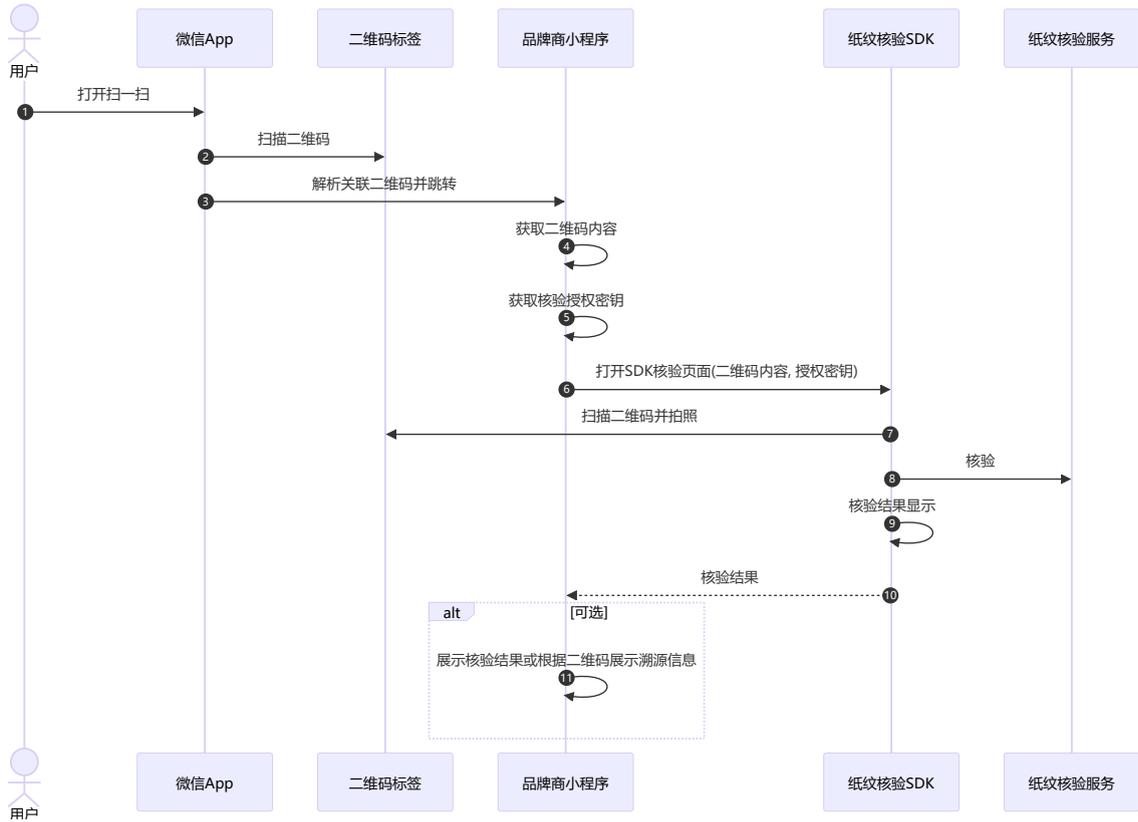
### 1.2 使用微信扫一扫直接跳转到 扫码核验页面



- 用户使用微信扫一扫, 扫描商品二维码标签, 直接打开 扫码核验页面。需配置关联二维码,见后台配置章节

## 2、品牌商小程序与 纸纹核验SDK 的交互时序

品牌商小程序需先集成 纸纹核验SDK



1. 用户打开微信扫一扫
2. 扫描二维码标签。二维码需在小程序配置关联打开小程序
3. 微信打开品牌商小程序
4. 品牌商小程序打开扫码核验页面
5. 品牌商小程序打开扫码核验页面
6. 品牌商小程序打开扫码核验页面
7. 扫描标签并拍照
8. 提交纸纹核验
9. 展示核验结果
10. SDK将核验结果返回给品牌商小程序
11. 品牌商小程序可以使用二维码内容和结果展示溯源信息(可选)

## 3、品牌商小程序与 纸纹核验SDK 需要做的开发对接

纸纹核验SDK 以微信小程序插件形式提供，品牌商需在小程序管理后台申请使用该插件并在代码中集成

- 1. 打开插件详情页并点击添加插件，等待审核通过。  
详情页地址：<https://mp.weixin.qq.com/wxopen/plugindevdoc?appid=wx79ea23ba34f928a4>
- 2. 在宿主小程序的app.json中声明插件：

```
{
  "plugins": {
    "paperprint-plugin": {
      "version": "根据实际版本号填写，如1.0.0",
      "provider": "wx79ea23ba34f928a4"
    }
  }
}
```

- 3.集成sdk配套的页面和工具类，并放在对应目录下，在宿主app.json中声明新加的页面  
 页面：pages/paperprint/resultH5  
 工具类：utils/paperprint.js

```
//app.json
{
  "pages": [
    ....
    "pages/paperprint/resultH5"
  ]
}
```

- 4.在需要唤起sdk进行核验的时候调用paperprint工具类提供的verify方法，如在index.js中

```
const paperprint = require('../utils/paperprint')
Page({
  ...
  startVerification() {
    /**
     * scanQrCode: 微信扫码进入小程序时获取的二维码值，如果不是微信扫码进入则传"",
     * clientId: sdk分配的clientId
     * clientSecret: sdk分配的clientSecret
     * callback: sdk核验后返回的数据
     */
    paperprint.verify(scanQrCode, clientId, clientSecret, (verifyQrCode,
    verifyResult) => {
      //verifyQrCode: string类型，核验检测到的二维码内容
      //verifyResult: object类型，核验的结果，详情见下表
    });
  },
})
```

### verifyResult详情

字段	类型	说明
status	int	核验结果0-原厂标签，1-非原厂标签，-1-标签未注册，-2-核验失败
msg	string	status的描述信息
brandMerchant	string	品牌商名称
productInfo	string	商品信息

## 4、授权密钥的管理

- 1.可以直接把授权密钥方便快捷的内置在页面程序中

```

const paperprint = require('../..//utils/paperprint')
const CLIENT_ID = "XXXXXXXXXX";
const CLIENT_SECRET = "XXXXXXXXXX";
Page({
  ...
  startVerification() {
    paperprint.verify("", CLIENT_ID, CLIENT_SECRET, (verifyQrCode,
verifyResult) => {
      ...
    });
  },
})

```

- 2.为了增加安全性, 可以额外添加配套的配置获取类并放在对应目录下, 从预编译的wasm中获取授权密钥  
配置获取类: wasm/paperprintLib.js, wasm/paperprintLib.wasm

```

const paperprint = require('../..//utils/paperprint')
const paperprintLib = require('../..//wasm/paperprintLib.js');
Page({
  ...
  startVerification() {
    paperprintLib().then(Module => {

      let paperprintConfig = JSON.parse(Module.getConfig());
      paperprint.verify("", paperprintConfig.clientId,
paperprintConfig.clientSecret, (qrCode, verifyResult) => {
        ...
      });

    }).catch(err => {
      console.error("Failed to load WASM:", err);
    });
  },
})

```

## 5、品牌商小程序使用 纸纹核验SDK 需要的后台配置

后台地址: <https://mp.weixin.qq.com/wxamp/index/index>

- 1.配置业务域名  
配置位置: 管理->开发管理->开发设置->业务域名  
添加配置业务域名: <https://ppsys.sinosun.com>  
同时需要提供微信后台下载的校验文件
- 2.如果需要微信扫一扫, 扫描商品二维码标签进入小程序, 需要在小程序后台配置关联内容  
配置位置: 管理->开发管理->开发设置->扫普通链接二维码打开小程序  
文档说明: <https://developers.weixin.qq.com/miniprogram/introduction/qrcode.html#%E6%B5%8B%E8%AF%95%E8%B0%83%E8%AF%95>
- 3.如果需要微信扫一扫, 扫描商品二维码标签, 直接打开 扫码核验页面, 则需要完成上述2中配置, 然后再在扫码打开的页面的onLoad方法中, 直接调用sdk方法打开 扫码核验页面

```
const paperprint = require('../..//utils/paperprint')
const CLIENT_ID = "XXXXXXXXXX";
const CLIENT_SECRET = "XXXXXXXXXX";
Page({
  ...
  onLoad(options) {
    paperprint.verify("", CLIENT_ID, CLIENT_SECRET, (verifyQrCode,
verifyResult) => {
      ...
    });
  },
})
```

## 6、品牌商小程序如何更新升级 [纸纹核验SDK](#) 的新功能

- 1.在宿主小程序的app.json中更新sdk插件的版本，然后再重新发布小程序

```
{
  "plugins": {
    "paperprint-plugin": {
      "version": "填写更新后的插件版本号，如2.0.0",
      "provider": "wx79ea23ba34f928a4"
    }
  }
}
```