**广东省全民健康信息综合管理平台**

**公共服务系统互联网医院单点接入指引（试行）**

广东省卫生健康委员会

2025年4月

**目录**

[1. 对接内容 3](#_Toc195622507)

[1.1. 工作目标与任务 3](#_Toc195622508)

[1.2. 对接前置条件 3](#_Toc195622509)

[1.2.1. 业务系统接入注册 3](#_Toc195622510)

[1.3. 单点登录 3](#_Toc195622511)

[1.3.1. 跳转链接 3](#_Toc195622512)

[2. 接口定义 5](#_Toc195622513)

[2.1. 协议 5](#_Toc195622514)

[2.2. 请求响应报文格式 5](#_Toc195622515)

[2.3. 错误码定义 5](#_Toc195622516)

[2.4. 接口文档 6](#_Toc195622517)

[2.4.1. code换取用户认证信息 6](#_Toc195622518)

[3. 数据字典 8](#_Toc195622519)

[3.1. 证件类型 8](#_Toc195622520)

[3.2. 性别 9](#_Toc195622521)

[3.3. 民族代码 9](#_Toc195622522)

[3.4. 国籍 10](#_Toc195622523)

[4. 数据加密方法 11](#_Toc195622524)

[4.1. 加密描述 11](#_Toc195622525)

[4.2. 加密使用方法 11](#_Toc195622526)

# 对接内容

## 工作目标与任务

广东省全民健康信息综合管理平台公共服务系统实现全省健康公共服务管理，为全省居民提供如预约挂号、健康档案查询、家庭医生签约、免疫接种、智能导诊、移动支付、随访等各类基于电子健康码的医疗健康互联网服务。

本指引主要针对互联网医院与省卫生健康委健康公共服务小程序“粤健通”的单点对接。

## 对接前置条件

### 业务系统接入注册

由项目组完成相关配置后，分配业务系统应用标识(appCode)和敏感信息加密秘钥(appKey)等相关信息。

## 单点登录

### 跳转链接

#### 业务场景

用户在“粤健通”小程序内点击业务系统链接并完成刷脸登录后，跳转到业务系统相关页面时，业务系统通过用户认证相关信息实现单点登录功能。

#### 业务流程



业务系统与“粤健通”小程序通过电子完成单点登录认证的流程如下：

1. 用户点击业务系统链接后，由“粤健通”小程序完成刷脸认证登录流程；

2. 刷脸认证后，“粤健通”小程序调用服务端，获取此次会话的code(5分钟有效)；

3.“粤健通”小程序code传参
 （1）跳转到h5:

小程序在业务系统链接url后拼接上code参数，例如业务系统链接为“http://www.baidu.com”,则拼接code后的链接“http://www.baidu.com?code=afadsgfaar”

（2）跳转到小程序：

 // 发送方

wx.navigateToMiniProgram({

 appId: '目标AppID',

 extraData: { code: 'value' }

});

// 接收方（目标小程序）

App({

 onLaunch(options) {

 console.log('来自跳转小程序的参数:', options.referrerInfo.extraData);

 }

});

4. 业务系统前端获取code值并传入业务系统服务端，业务系统服务端调用code换取用户认证信息接口，获取用户认证类型和认证代码；

5. 业务系统服务端解密用户认证信息后，完成系统内部的账号认证流程。

# 接口定义

## 协议

与“粤健通”小程序系统通信协议，使用HTTPS协议。

## 请求响应报文格式

接口正常响应的报文，规定以标准JSON格式字符串返回。其中，

1. 字符串编码格式统一使用UTF-8；
2. 按照API网关接入规范要求，请求“粤健通”接口的请求头，均需增加网关相关请求参数，网关接入参数如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名** | **必填** | **数据格式** | **类型范围** | **说明** |
| x-tsp-nonce | 是 | {nonce} | String | 请求参数随机字符串，用于校验请求签名 |
| x-tsp-signature | 是 | {singature} | String | 请求参数签名值 |
| x-tsp-appSecret | 是 | {appSecret} | String | paasid对应的Secret |
| x-tsp-paasid | 是 | {paasid} | String | 当前业务系统的网关标识 |
| Content-type | 是 | application/json;charset=utf-8 | String | 文件拓展名，默认application/json |
| x-tsp-timestamp | 是 | {timestamp} | String | 应用程序发出请求的客户端时间 |

## 错误码定义

|  |  |
| --- | --- |
| 错误码 | 描述 |
| 0 | 成功 |
| 1 | 失败 |
| 2 | 参数不合法 |
| 3 | 验签失败 |

## 接口文档

### code换取用户认证信息

#### 描述

业务系统通过此接口，以code换取用户健康码信息后，完成系统内的账号认证流程。

#### URL

（1）政务外网

https://sjzt.wjw.gdgov.cn/data\_center/gateway/common/api

生产环境：

service ID：DGSP\_1911981103558230017

passid： （由项目组按医疗机构提供）

token： （由项目组按医疗机构提供）

测试环境：

service ID：DGSP\_1911964810465574914

passid：（由项目组按医疗机构提供）

token： （由项目组按医疗机构提供）

#### 协议类型

GET协议

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 必填 | 类型 | 描述 |
| appCode | 是 | String(10) | 系统编码，业务系统接入时由”粤健通”系统分配 |
| command | 是 | String(10) | 固定传01 |
| code | 是 | String(32) | 认证code，5分钟内有效 |
| businessType | 是 | Integer | 业务类型：1：预约挂号；（互联网医院接入按此业务类型）2：健康档案查询；3：家庭医生签约；4：免疫接种；5：双向转诊；6：母子E手册；7：智慧结控；8：医疗废弃物；9：中医药局10：奖励扶助11：托育12：健康深圳《新冠预约》…… |

#### 响应报文内容

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 必填 | 类型 | 描述 |
| errcode | 是 | Integer | 响应编码。响应成功为0；响应错误详见本文2.3错误码定义列表 |
| errmsg | 是 | String(100) | 响应结果描述，详见本文2.3错误码定义列表 |
| data | 否 | Object | 响应成功时，返回的业务数据 |

data

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| certificateNo | String(64) | 证件号码，SM4算法加密 |
| certificateType | String(5) | 证件类型，详见3.1数据字典 |
| healthCardNo | String(128) | 电子健康码 |
| phone | String(50) | 手机号码，SM4加密 |
| name | String(128) | 用户姓名 |
| gender | Integer | 性别，详见3.2数据字典 |
| nation | String(5) | 民族 详见章节“3.3民族代码 |
| userId | String(20) | 粤健通用户ID |
| channel | String(5) | 渠道，1内地、2港澳、3台湾、4外籍 |

#### 报文示例

响应报文：

{

    "errcode": 0,

    "errmsg": "SUCCESS",

    "data": {

        "certificateNo": "xxxxx",

        "certificateType": "xxxxx

        "healthCardNo": "xxxx",

        "phone": "xxxx",

        "gender": xxx,

        "name": "xxx",

"nation": "xxx",

"userId": "xxx",

"channel": "xxx",

    }

}

# 数据字典

## 证件类型

|  |  |  |
| --- | --- | --- |
| **值含义** | **值** | **说明** |
| 居民身份证 | 01 |  |
| 居民户口簿 | 02 |  |
| 护照 | 03 |  |
| 军官证 | 04 |  |
| 驾驶证 | 05 |  |
| 港澳居民来往内地通行证 | 06 |  |
| 台湾居民来往内地通行证 | 07 |  |
| 出生医学证明 | 08 |  |
| 其他法定有效证件 | 99 |  |

## 性别

|  |  |  |
| --- | --- | --- |
| **值含义** | **值** | **说明** |
| 男 | 1 |  |
| 女 | 2 |  |
| 保密 | 3 |  |
| 未知 | 4 |  |

## 民族代码

|  |  |  |
| --- | --- | --- |
| **值含义** | **值** | **说明** |
| 汉族 | 01 |  |
| 蒙古族 | 02 |  |
| 回族 | 03 |  |
| 藏族 | 04 |  |
| 维吾尔族 | 05 |  |
| 苗族 | 06 |  |
| 彝族 | 07 |  |
| 壮族 | 08 |  |
| 布依族 | 09 |  |
| 朝鲜族 | 10 |  |
| 满族 | 11 |  |
| 侗族 | 12 |  |
| 瑶族 | 13 |  |
| 白族 | 14 |  |
| 土家族 | 15 |  |
| 哈尼族 | 16 |  |
| 哈萨克族 | 17 |  |
| 傣族 | 18 |  |
| 黎族 | 19 |  |
| 傈僳族 | 20 |  |
| 佤族 | 21 |  |
| 畲族 | 22 |  |
| 高山族 | 23 |  |
| 拉祜族 | 24 |  |
| 水族 | 25 |  |
| 东乡族 | 26 |  |
| 纳西族 | 27 |  |
| 景颇族 | 28 |  |
| 柯尔柯孜族 | 29 |  |
| 土族 | 30 |  |
| 达尔族 | 31 |  |
| 仫佬族 | 32 |  |
| 羌族 | 33 |  |
| 布朗族 | 34 |  |
| 撒拉族 | 35 |  |
| 毛南族 | 36 |  |
| 仡佬族 | 37 |  |
| 锡伯族 | 38 |  |
| 阿昌族 | 39 |  |
| 普米族 | 40 |  |
| 塔吉克族 | 41 |  |
| 怒族 | 42 |  |
| 乌孜别克族 | 43 |  |
| 俄罗斯族 | 44 |  |
| 鄂温克族 | 45 |  |
| 德昂族 | 46 |  |
| 保安族 | 47 |  |
| 裕固族 | 48 |  |
| 京族 | 49 |  |
| 塔塔尔族 | 50 |  |
| 独龙族 | 51 |  |
| 鄂伦春族 | 52 |  |
| 赫哲族 | 53 |  |
| 门巴族 | 54 |  |
| 珞巴族 | 55 |  |
| 基诺族 | 56 |  |
| 其它民族或外国人 | 99 |  |

## 国籍

|  |  |  |
| --- | --- | --- |
| **值含义** | **值** | **说明** |
| 汉族 | 01 |  |

# 数据加密方法

## 加密描述

1. 本文中所用到的数据加密采用SM4加密算法，加密密钥由省公共服务系统向应用分配，应用系统使用分配的密钥进行加密后传输数据，省公共服务系统收到数据后使用相同的密钥进行解密，反之省公共服务系统向应用系统返回数据时使用同样的方法加密
2. 加密Key，分配至地市/医院的 appKey做为密钥
3. 敏感信息加密 例：
	1. 姓名
	2. 电话
	3. 微信号
	4. 邮箱号
	5. 电子健康码ID

## 加密使用方法

**以下为加密使用方法（SM4加密）**

public static void main(String[] args) {

String params = "小明";

 String key = "A7E2D00B4058086CBD3D4B8A5E0B2697";

 try {

 //加密

 String cipher = Sm4Utils.encryptEcb(key,params);

 System.out.println("加密后数据："+cipher);

 //效验加密前后的字符串是否为同一数据

 System.out.println(Sm4Utils.verifyEcb(key,cipher,params));

 //解密

 params = Sm4Utils.decrptEcb(key,cipher);

 System.out.println("解密后数据："+params);

 }catch (Exception e){

 e.printStackTrace();

 }

 }