目　录

首页

更新履历-2020-08月

介绍

- 开放平台是什么
- 智慧教育云平台、开放平台用户说明
- 开放平台提供了哪些API

整体对接流程【工作步骤】

- 一.注册账号
- 二.登录
- 三.注册开发者【仅首次】
- 四.创建应用
- 五.申请Access Key 、 Secret Access
- 六.申请接口权限
- 七.接入授权
- 八.示例代码
- 常见错误

应用对接

- 接入准备【必读】
- 【授权】开放平台授权机制【帐号中心】
- 【授权】第三方登录（oauth）接入指南
- 【授权】通过code获取access_token示例
- 【认证】开放平台应用的开发者、服务提供者认证简要说明
- 【认证】开放平台API的认证说明
- 如何获取sdp-app-id

常用技术说明

- 理解什么是 OAuth 2.0 协议
- 网龙开放平台相关术语说明
- 开放平台上下文关系简要说明
- AK/SK简介
- 认证、授权和凭证

FOR 网龙内部

根据open_id+access_token获取uc mac token
根据open_id获取account_id（uc userid）
平台规范
开放平台-接口规范【建议】
开放平台api错误规范
【重要】对接厂家开发规范
服务接入
如何接入服务
接入案例
湖北教育云集成多家三方应用案例
湖北教育云与阳光评价应用集成
教育基础数据接口
根据open_id获取用户详细信息
根据openid及accesstoken获取用户信息
学科查询
学段查询
用户增量数据
学校管理
查询学校列表
查询用户有权限的学校列表
查询单个学校
根据学校编码查询学校
查询学校类型列表
根据帐号查询学校下user_id
年级管理
查询年级列表
查询单个年级信息
班级管理
查询年级下的班级列表
查询学校下单个班级
部门管理
查询学校部门列表
教师管理

获取单个教师信息
获取教师所在班级
获取班级的班主任列表
获取班级的任课教师列表
获取班主任管理的班级列表
查询学校教师
获取班级的教师列表
获取教师所在部门
学生管理
获取班级学生列表
获取单个学生信息
获取学生所在班级列表
根据学籍号查询学生user_id
家长管理
获取我的孩子信息
获取家长列表
设备管理
设置用户设备
根据设备编码查询用户
课程表
获取班级的课程表
获取全校各年级课程表
获取学校课程表
获取场地的课程表
获取学生的课程表
获取老师的课程表
学期信息
获取学校所有学期
获取单个学期
行政区划
根据IP定位当前的地区的行政区划
获取子行政区划列表
父级行政区划列表

教务需求
组织结构
任课管理
应用、数据融合
应用集成
数据融合
知识库
理解SSL证书
What is the SSL Certificate Chain?
HTTPS证书分类
移动端对接说明
移动端（IOS）接入
移动端（Android）接入
学习活动数据接口

首页

开放平台帮助文档

什么是开放平台？

开放平台是一个教育应用工厂，是一个专业的在线教育SaaS服务平台，
它为网龙华渔教育业务提供所有教学活动能力，通过开放平台可以拼装出各个教育产品。

介绍

开放平台是为网龙华渔教育业务提供所有教学活动能力的SaaS服务平台，它以“标准化 + 服务化 + 组件化”的方式全方位打造教、学、练、测、评、推荐等核心教学活动体系；通过构建丰富的教学方法库，涵盖教学设计、教学实施、教学评测、教学管理等教学全过程；同时通过快速组装教育业务应用，结合海量优质资源，开创全新教育模式与业务形态，构筑终身、开放、便捷、易用的个性化移动互联网学习平台，贯穿学前教育、基础教育、高等教育、职业教育、企业培训、非学历及终身教育等各个阶段，满足不同教育业务需求。

开放平台可支撑ToC、ToB、ToG业务，除直接的目标用户之外，政府机关、院校学校、培训机构、企业等也将作为重要服务对象，
开放平台将为各类业务提供教育的整体解决方案。

网龙愿景

网龙华渔教育的目标是帮助更多人获得有效的教育，为世界教育的发展和进步做贡献，培养21世纪未来社会的建设者和接班人。
网龙要做的是面向未来的学习，用技术创造教育的未来。为了达成这个目标，我们统一规划，将网龙在互联网技术上的优势，全方位应用在教育上，
以平台化的方式真正实现个性化的学习、有效果的学习、情景式的学习、人性化的学习、全民终身式的学习。

更新履历-2020-08月

重要提示：2020-08-06 登录注册方式变更

- 已由原网龙分配开发者账号，切换为开发者自行注册，注册入口在 open.101.com 顶部导航
- 开发者注册分为两步骤 1：注册登录账号（手机号） 2：申请开发者身份【个人开发者、企业开发者】
- 由网龙接口人通知原分配账号重新使用手机注册的厂家，请通过 open.101.com 首页进入，激活【申请开发者身份】

介绍

- 整体性介绍开放平台和如何注册账号、如何创建应用

开放平台是什么

提供应用接入、管理

为“智慧教育云平台”提供的统一的应用接入、管理能力；提供标准的应用接入申请、管理、上下架流程。提供标准oauth2授权协议，为接入的第三方应用提供统一授权支持。

场景举例：A公司提供一个产品应用，可通过开放平台对接集成到网龙公司的
平台，智慧教育云平台的用户登录后，可在智慧教育云平台上直接使用A公司应用，可一键免登录进入。

提供服务（API）接入、管理

为“智慧教育云平台”的内部、外部第三方厂家提供了“服务”的注册、管理、统一鉴权能力支持。

本部分的“服务”通常是指某厂家提供给开发人员使用的功能，例如用于两个厂家的数据交换api，两个厂家产品的必须的数据获取、交换等。具体表现形式是一个一个的API

智慧教育云平台开放的接口（API）

智慧教育云平台自身对第三方厂家提供了必要的开发接口

- 用户相关：可获取用户基本信息、组织信息等
- 用户行为数据：可获取用户的教、学、评、测、练等教学相关的行为数据，供第三方厂家应用开发使用

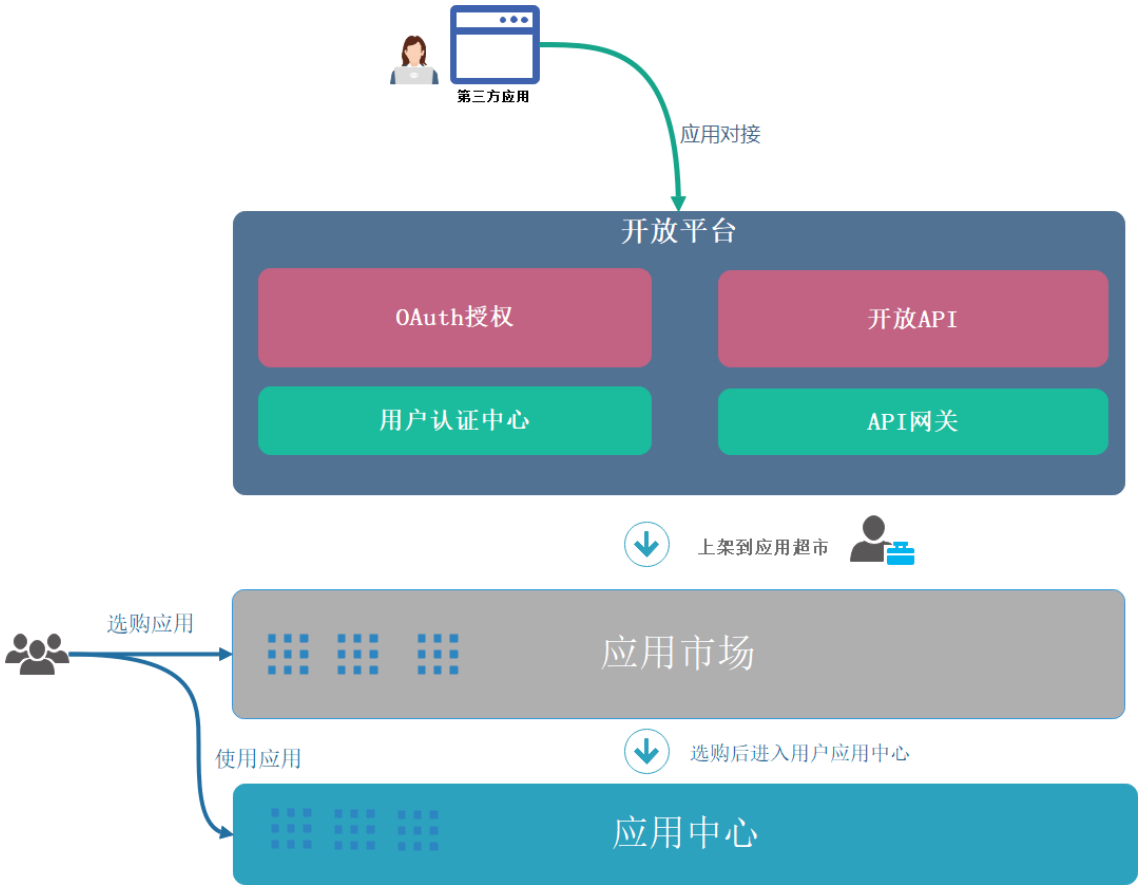
应用市场

应用市场对接入的应用提供了售卖能力

应用中心

应用中心是用户从应用市场购买后，可以在应用中心看见自己可用的应用入口

平台示意图


第三方应用
应用对接
开放平台
OAuth授权
开放API
用户认证中心
API网关
上架到应用超市
选购应用
应用市场
使用应用
选购后进入用户应用中心
应用中心

智慧教育云平台、开放平台用户说明

名词

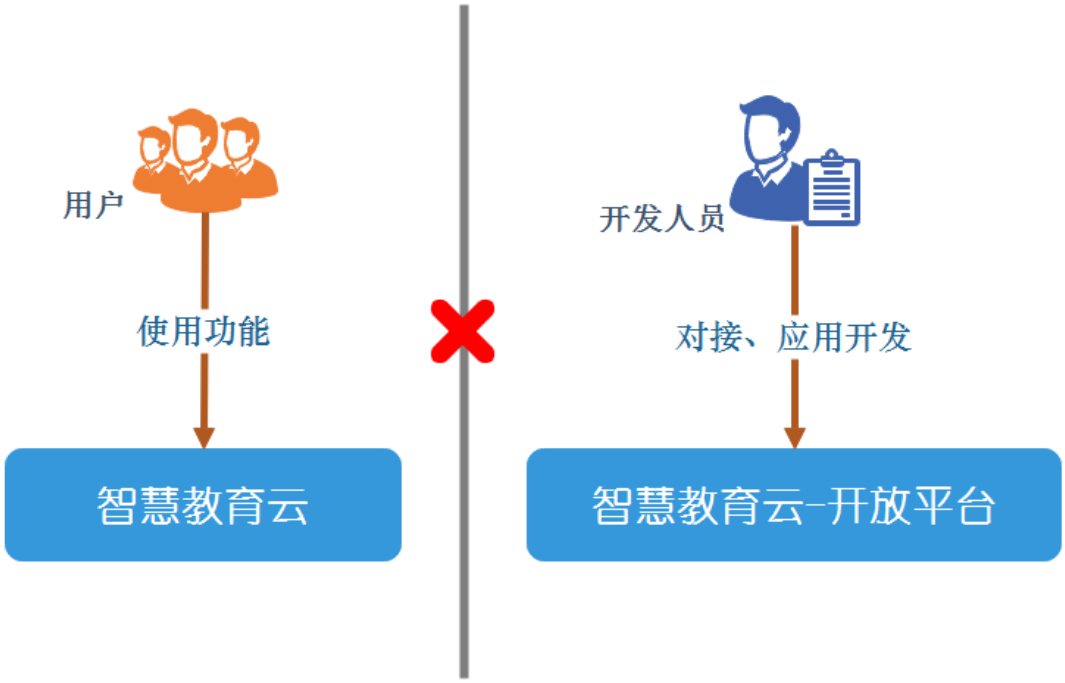
- 智慧教育云平台：指网龙公司开发的教育产品，例如湖北教育云、福建网教通平台、陕西教育云，现代化监测平台等
- 开放平台：指智慧教育云平台产品提供的为第三方开发者提供的开发平台，主要提供oauth2.0协议支持、统一api网关支持。

用户区别

- 智慧教育云平台用户：通常是智慧教育云平台产品的终端使用用户，例如教师、学生、教育主管领导、家长等。
- 开放平台用户：通常是基于开放平台进行应用开发、对接的程序开发人员。

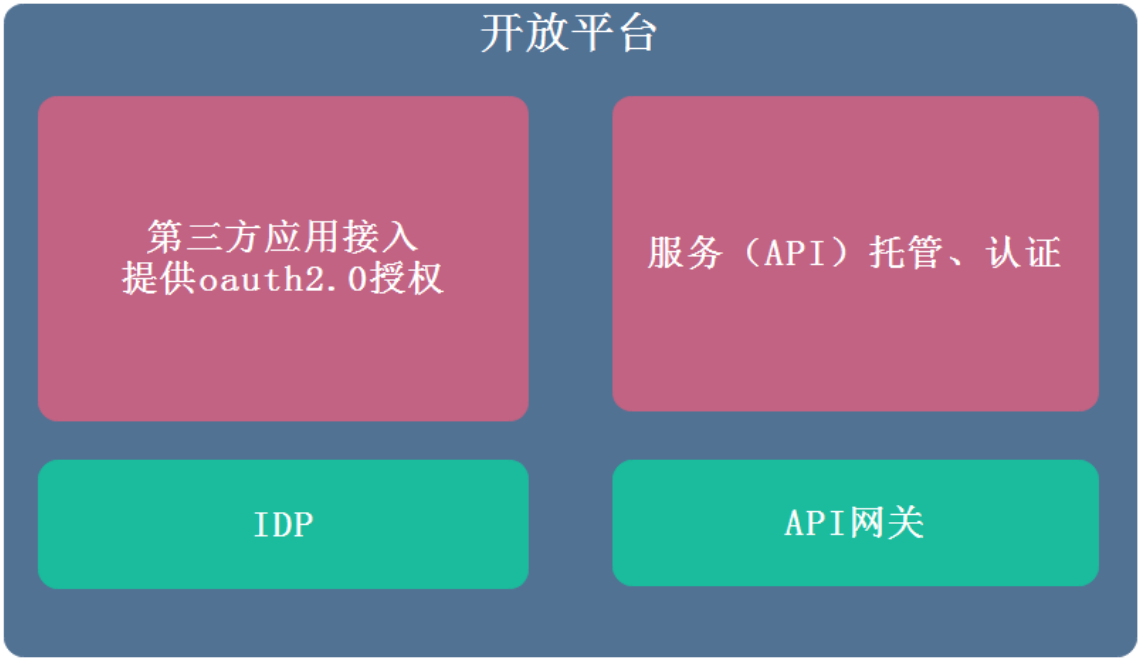
智慧教育云平台用户与开放平台用户为独立体系，两者不互通。在开放平台使用需要独立注册和登录。

开发人员账号，需要更高的审核验证流程。



开放平台提供了哪些API

开放平台功能分为应用接入（开放oauth标准第三方登录）、和服务（api）统一托管授权即开放API网关。



开放平台的API共分两部分：IDP(oauth)用户相关api、通过开放平台托管的网龙自有api或第三方合作公司提供的api。

一、与IDP（oauth）用户有关的api

- /v1.1/oauth2/get_user_info

根据open_id、access_token获取开放平台IDP公开的用户基本信息

响应体：nick_name(昵称)、avatar_url（头像）、gender（性别）

详细接口调用说明可参见 帮助文档与IDP相关的说明

二、开放平台API网关提供的接口

这部分主要为网龙公司、第三方合作公司提供的教育业务相关的API；通常应用的开发人员更多面向本部分做开发、调试工作。

通常用于智慧教育云平台与合作企业，或者合作企业之间的数据交换、服务调用、必要的元数据获取、制定和发布统一的行业数据规范和标准等。

例如：智慧教育云平台平台发布教师用户的编制信息（任职学校、地区信息、任课信息、教龄、级别），各合作企业可通过开放平台API网关的合法授权来获取该数据。

> 注：本部分的API认证协议，不同于oauth，使用 HMAC-SHA256签名认证方式。具体可参见 开放平台API的认证说明

整体对接流程【工作步骤】

一.注册账号

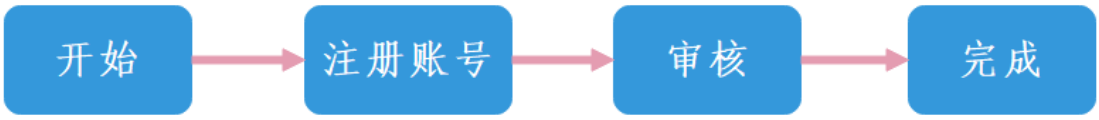
平台注册流程

重要提示：2020-08-06 登录注册方式变更

- 已由原网龙分配开发者账号，切换为开发者自行注册，注册入口在 open.101.com 顶部导航
- 开发者注册分为两步骤 1：注册登录账号（手机号） 2：申请开发者身份【个人开发者、企业开发者】
- 由网龙接口人通知原分配账号重新使用手机注册的厂家，请通过 open.101.com 首页进入，激活【申请开发者身份】

开发者首次注册时，进入http://open.101.com 通过右上角的进入开发者中心(新唯一入口，手机号)入口进入注册流程。

注册过程包括4步，如下图所示：



账号注册



填写手机号后，平台会给开发者发送验证码，如实填写验证码即可。

注册完成



注册完成后，可在登录页面登录开放平台

二.登录

在开放平台的登录页面



三.注册开发者【仅首次】

新厂家应用接入，首次需要申请开发者身份【个人开发者、企业开发者】

1. 选择开发者角色类型

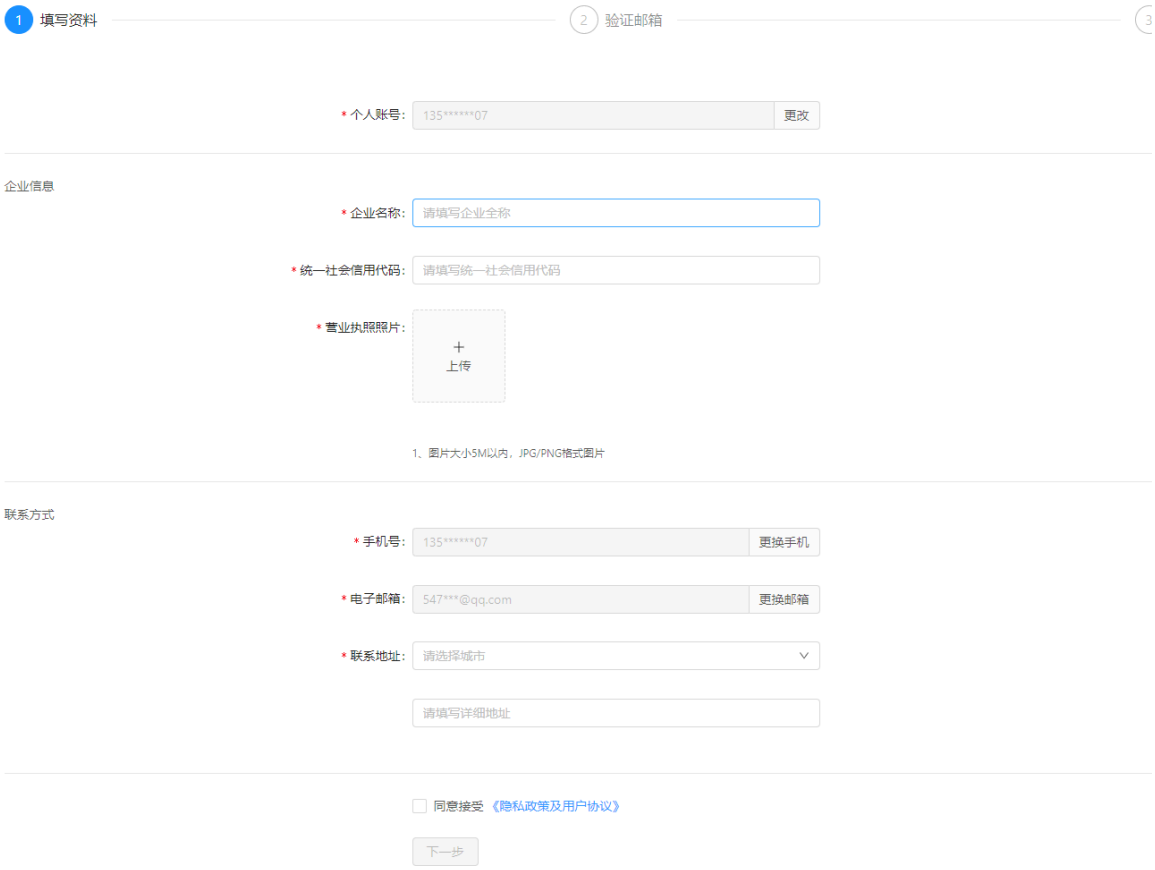


2. 填写资料

- 个人开发者


1 填写资料
2 验证邮箱
* 个人账号: 135******07 更改
个人信息
* 姓名: 请填写真实姓名
* 个人证件: 身份证 护照
* 证件号码: 请填写身份证号码!
* 身份证照片: + 上传
1、图片大小5M以内，JPG/PNG格式图片
2、请上传身份证人像面、国徽面以及手持身份证共计三张照片
3、开发者手持身份证/护照正面进行拍照，要求五官可见，证件信息清晰无遮挡
联系方式
* 手机号: 135******07 更换手机
* 电子邮箱: 547***@qq.com 更换邮箱
* 联系地址: 请选择城市
请填写详细地址

- 企业开发者



3. 验证邮箱



4. 完成注册，等待审核



5. 审核通过，进入开放平台


应用管理
接口查看
业务管理
开发文档
全部应用
已上架
审核中
审核未通过
未提交审核
已下架
暂无数据

四.创建应用

1. 创建应用



2. 填写应用名称、上传应用图标



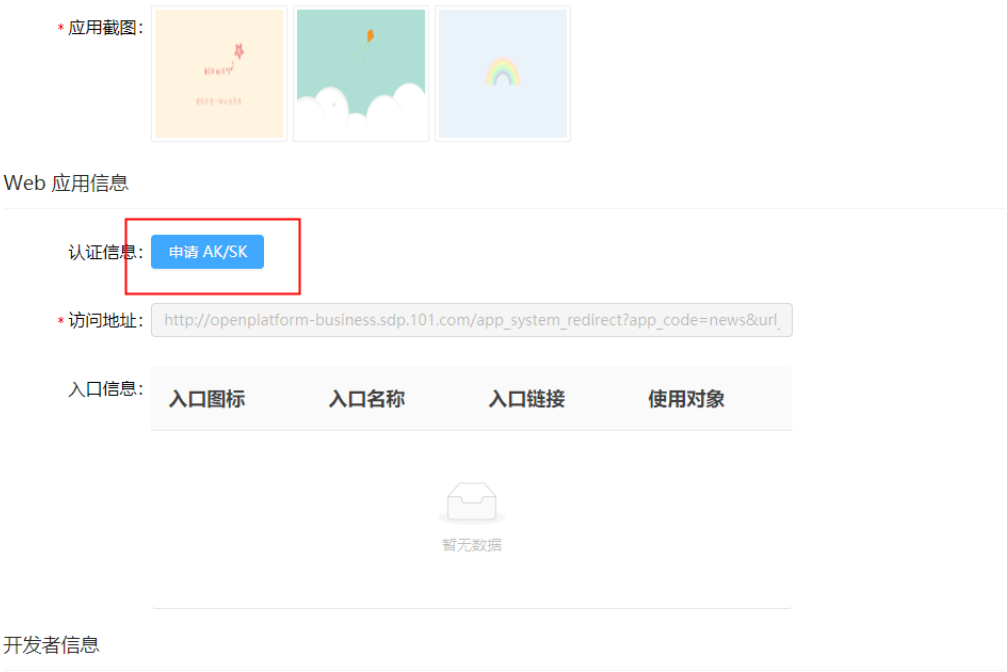
3. 填写应用详细信息

应用官网、应用简介、应用案例、应用截图等


添加应用
创建后，会为您分配一个APP ID和APP KEY
应用信息
我是应用
App ID : 11dfd562-f3d0-4b16-93fe-423227f24bc5
完善应用信息
应用基本信息
* 应用类型: 标准应用 虚拟应用
应用官网: http://www.101.com
管理地址: 请填写应用的管理地址
订购通知地址: 请填写应用的订购通知 POST 地址
* 应用简介: 应用简介应用简介应用简介

五.申请Access Key 、 Secret Access

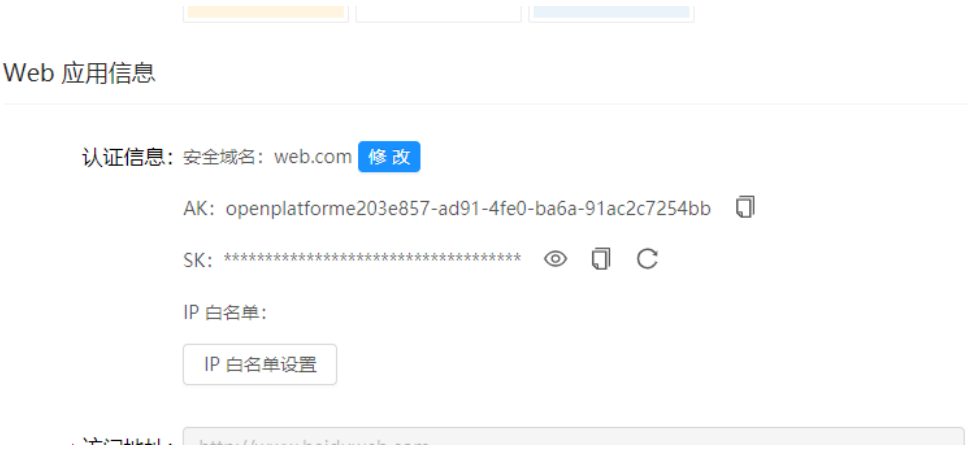
1. 在应用信息中，点击“申请”可直接申请Access Key信息



2. 点击申请后



3. 管理人员审核通过后


Web 应用信息
认证信息：安全域名：web.com
修 改
AK：openplatforme203e857-ad91-4fe0-ba6a-91ac2c7254bb
SK：**********
IP 白名单：
IP 白名单设置

六.申请接口权限

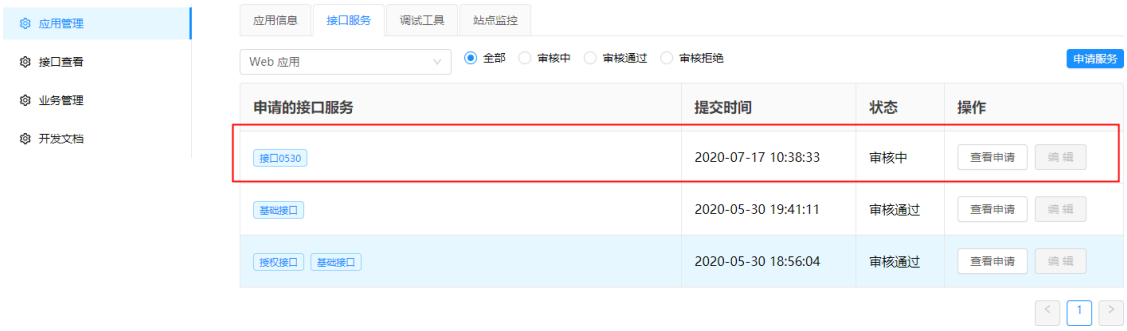
1. 申请接口权限





按需要申请接口

2. 等待审核


应用管理
接口查看
业务管理
开发文档
应用信息
接口服务
调试工具
站点监控
Web 应用
全部
审核中
审核通过
审核拒绝
申请服务
申请的接口服务
提交时间
状态
操作
接口0530
2020-07-17 10:38:33
审核中
查看申请
编辑
基础接口
2020-05-30 19:41:11
审核通过
查看申请
编辑
接权接口
基础接口
2020-05-30 18:56:04
审核通过
查看申请
编辑
1

七.接入授权

1. 定义三种参数：

- {api-gateway-host}



- {unified_domain}
  统一域名，即开放平台前台页面的域名
  如：
  湖北网教通：http://new.hbeducloud.com
  获取方式：



- {sdp-app-id}

  查看位置：开放平台-开发文档-接口调用，如图：



2.审核通过后，便可以开始接入。有两种授权流程：

- 授权码接入

1） 拼接授权码页面

```
http://{unified_domain}/openplatform/connect?sdp-app-id={sdp-app-id}&response_type={response_type}&access_key_id={access_key_id}&redirect_uri={redirect_uri}&scope={scope}&state={state}&type={type}
```

参数名	是否必须	类型	说明
sdp-app-id	是	string	开放平台租户标识（获取方式如上）
access_key_id	是	string	应用唯一标识,应用注册时由开放平台分配
redirect_uri	是	string	授权成功的回调地址, 其中域名必须符合应用注册时填写的安全域名
response_type	是	string	响应类型，目前仅支持传code
scope	是	string	scope_base（获取你的公开信息） 与scope_userinfo（获取你的详细信息）
state	是	string	第三方应用自行产生，可以使用随机字符串
type	否	string	h5 页面，不传为web页面，传‘h5’为h5页面

授权成功开放平台帐号中心会返回302 重定向到应用提供的{redirect_uri}页面并且带上code和state参数
{redirect_uri}?code={code}&state={state}

2） 根据code获取access_token

```
http://{api-gateway-host}/oauth/access_token
```

将code传过去，获取返回body中的access_token和open_id。Java代码示例：

```
public Object auth(@RequestParam("code") String code) throws IOException {
    if (StringUtils.isBlank(code)) {
        return "code is null";
    }
    HttpClient httpClient = NdHttpClientBuilder.getHttpClient(ACCESS_KEY_ID, SECRET_ACCESS_ID);//此类由sdk提供
    JSONObject obj = new JSONObject();
    obj.put("access_key_id", ACCESS_KEY_ID);
    obj.put("secret_access_key", SECRET_ACCESS_ID);
    obj.put("code", code);
    obj.put("grant_type", "authorization_code");
    HttpPost request = new HttpPost(API_GATEWAY_HOST + "/oauth/access_token");
    request.setHeader("Content-Type", "application/json");
    StringEntity entity = new StringEntity(obj.toString());
    request.setHeader("sdp-app-id", SDP_APP_ID);
    request.setEntity(entity);
    HttpHost httpHost = HttpHost.create(API_GATEWAY_HOST);
    HttpResponse httpResponse = httpClient.execute(httpHost, request);
    if (httpResponse.getStatusLine().getStatusCode() == 200) {
        String result = EntityUtils.toString(httpResponse.getEntity());
        return JSONObject.parseObject(result);
    }
    return EntityUtils.toString(httpResponse.getEntity());
}
```

返回参数：

参数	含义
access_token	访问令牌，接口调用凭证，有效期 7 天
expires_at	access_token 的过期时间
refresh_token	用于刷新 access_token，有效期 60 天
open_id	授权用户唯一标识
server_time	服务器时间

- ticket接入

1） 用户登录智慧教育云平台，进入第三方应用

用户进入第三方后，智慧教育云平台将会302到第三方提供的http地址，并附带ticket参数，如：{redirect_uri}?ticket={ticket}

2） 根据ticket获取access_token

```
http://{api-gateway-host}/oauth/access_token
```

将ticket传过去，获取返回body中的access_token和open_id。Java代码示例：

```
public Object authTicket(String ticket) throws IOException {
    if (StringUtils.isBlank(ticket)) {
        return "ticket is null";
    }
    HttpClient httpClient = NdHttpClientBuilder.getHttpClient(ACCESS_KEY_ID, SECRET_ACCESS_ID);//此类由sdk提供
    JSONObject obj = new JSONObject();
    obj.put("access_key_id", ACCESS_KEY_ID);
    obj.put("secret_access_key", SECRET_ACCESS_ID);
    obj.put("ticket", ticket);
    obj.put("grant_type", "ticket");
    HttpPost request = new HttpPost(API_GATEWAY_HOST + "/oauth/access_token");
    request.setHeader("Content-Type", "application/json");
    StringEntity entity = new StringEntity(obj.toString());
    request.setHeader("sdp-app-id", SDP_APP_ID);
    request.setEntity(entity);
    HttpHost httpHost = HttpHost.create(API_GATEWAY_HOST);
    HttpResponse httpResponse = httpClient.execute(httpHost, request);
    if (httpResponse.getStatusLine().getStatusCode() == 200) {
        String result = EntityUtils.toString(httpResponse.getEntity());
        return JSONObject.parseObject(result);
    }
    return EntityUtils.toString(httpResponse.getEntity());
}
```

返回参数：

参数	含义
access_token	访问令牌，接口调用凭证，有效期 7 天
expires_at	access_token 的过期时间
refresh_token	用于刷新 access_token，有效期 60 天
open_id	授权用户唯一标识
server_time	服务器时间

3.授权完成后可获取到access_token和open_id。获取用户的基本信息

```
http://{api-gateway-host}/base/get_user_info
```

Java代码示例：

```
public Object getUserInfo(String openid, String accessToken) throws IOException {
    HttpClient httpClient = NdHttpClientBuilder.getHttpClient(ACCESS_KEY_ID, SECRET_ACCESS_ID);//此类由sdk提供
    JSONObject obj = new JSONObject();
    obj.put("open_id", openid);
    obj.put("access_token", accessToken);
    HttpPost request = new HttpPost(API_GATEWAY_HOST + "/base/get_user_info");
    request.setHeader("Content-Type", "application/json");
    StringEntity entity = new StringEntity(obj.toString());
    request.setHeader("sdp-app-id", SDP_APP_ID);
    request.setEntity(entity);
    HttpHost httpHost = HttpHost.create(API_GATEWAY_HOST);
    HttpResponse httpResponse = httpClient.execute(httpHost, request);
    if (httpResponse.getStatusLine().getStatusCode() == 200) {
        String result = EntityUtils.toString(httpResponse.getEntity());
        return JSONObject.parseObject(result);
    }
    return EntityUtils.toString(httpResponse.getEntity());
}
```

返回参数：

参数	含义
open_id	开放平台帐号中心 颁发给应用的授权用户的唯一标识， 用来标记一个具体的用户
nick_name	昵称
avatar_url	头像

参数	含义
gender	性别 1：男，2：女，0：未知

4.令牌可定时续约，防止令牌失效

```
http://{api-gateway-host}/oauth/refresh_token
```

将授权后获取的refresh_token外加其他参数传入，Java代码示例：

```
public Object refresh(String refreshToken) throws IOException {
    HttpClient httpClient = NdHttpClientBuilder.getHttpClient(ACCESS_KEY_ID, SECRET_ACCESS_ID);//此类由sdk提供
    JSONObject obj = new JSONObject();
    obj.put("access_key_id", ACCESS_KEY_ID);
    obj.put("refresh_token", refreshToken);
    obj.put("grant_type", "refresh_token");
    HttpPost request = new HttpPost(API_GATEWAY_HOST + "/oauth/refresh_token");
    request.setHeader("Content-Type", "application/json");
    StringEntity entity = new StringEntity(obj.toString());
    request.setHeader("sdp-app-id", SDP_APP_ID);
    request.setEntity(entity);
    HttpHost httpHost = HttpHost.create(API_GATEWAY_HOST);
    HttpResponse httpResponse = httpClient.execute(httpHost, request);
    if (httpResponse.getStatusLine().getStatusCode() == 200) {
        String result = EntityUtils.toString(httpResponse.getEntity());
        return JSONObject.parseObject(result);
    }
    return EntityUtils.toString(httpResponse.getEntity());
}
```

返回参数：

参数	含义
access_token	访问令牌，接口调用凭证，有效期 7 天
expires_at	access_token 的过期时间
refresh_token	用于刷新 access_token，有效期 60 天
open_id	授权用户唯一标识
server_time	服务器时间

返回的access_token更新原来的令牌，refresh_token可用来下次续约

八.示例代码

Java client sdk下载

示例工程下载

业务接口根据自己需要选择接口对接，下面举两个例：

- 学科查询

```
http://{api-gateway-host}/base/subjects
```

该接口和用户无关，无需带用户相关的open_id和access_token，Java代码示例：

```
public Object subjects() throws IOException {
    HttpClient httpClient = NdHttpClientBuilder.getHttpClient(ACCESS_KEY_ID, SECRET_ACCESS_ID);//此类由sdk提供
    HttpGet request = new HttpGet(API_GATEWAY_HOST + "/base/subjects");
    request.setHeader("Content-Type", "application/json");
    request.setHeader("sdp-app-id", SDP_APP_ID);
    HttpHost httpHost = HttpHost.create(API_GATEWAY_HOST);
    HttpResponse httpResponse = httpClient.execute(httpHost, request);
    if (httpResponse.getStatusLine().getStatusCode() == 200) {
        String result = EntityUtils.toString(httpResponse.getEntity());
        return JSONObject.parseArray(result);
    }
    return EntityUtils.toString(httpResponse.getEntity());
}
```

- 查询用户有权限的学校列表

  ```
  http://{api-gateway-host}/base/schools/actions/search
  ```

  该接口和用户相关，需要额外传入idp-open-id和idp-access-token，放到header里，也就是open_id和access_token。Java代码示例：

```
public Object schoolsPrivate(
        Integer offset,
        Integer limit,
        String name,
        String openid,
        String accessToken) throws IOException {
    HttpClient httpClient = NdHttpClientBuilder.getHttpClient(ACCESS_KEY_ID, SECRET_ACCESS_ID);//此类由sdk提供
    HttpPost request = new HttpPost(String.format(API_GATEWAY_HOST + "/base/schools/actions/search?$offset=%s&$limit=%s", offset, limit));
    request.setHeader("Content-Type", "application/json");
    request.setHeader("sdp-app-id", SDP_APP_ID);
```

```
    request.setHeader("idp-open-id", openid);
    request.setHeader("idp-access-token", accessToken);
    JSONObject obj = new JSONObject();
    obj.put("name", name);
    StringEntity entity = new StringEntity(obj.toString());
    request.setEntity(entity);
    HttpHost httpHost = HttpHost.create(API_GATEWAY_HOST);
    HttpResponse httpResponse = httpClient.execute(httpHost, request);
    if (httpResponse.getStatusLine().getStatusCode() == 200) {
        String result = EntityUtils.toString(httpResponse.getEntity());
        return JSONObject.parseArray(result);
    }
    return EntityUtils.toString(httpResponse.getEntity());
}
```

常见错误

- 报错：HMAC signature cannot be verified, a valid date or x-date header is required for HMAC Authentication。

  解析：

  在对接 sdk 情况下，通常是两地时间偏差较大，然后服务在对签名中的时间进行校验时报错，可以将时间校正为北京时间即可解决。

  没有对接 sdk 情况下，通常是用户自己签名加密出错。推荐直接下载并使用如下的 java client sdk，其他语言可以参考 Java 代码编写自己的签名生成算法。

  Java client sdk下载链接

应用对接

- 如何以第三方应用的方式接入 智慧教育云平台
- 如何使用oauth（含部分代码片段）
- 第三方应用如何使用开放平台提供的已有接口（ak/sk）
- 开放平台的鉴权机制

接入准备【必读】

接入方必须的输入：

- 应用名称:【应用的实际名称】
- 应用介绍:【一段介绍应用的说明文本】
- 应用域名:【应用的主域名】
- 应用 LOGO:【三个好看的应用logo图片】
- 应用负责人:负责人姓名
- 应用负责人联系方式:负责人手机号

第三方厂家开发人员必读

【授权】开放平台授权机制【帐号中心】

- 开放平台-帐号中心提供基于OAuth2.0的授权协议支持；OAuth的技术说明可参见 理解什么是 OAuth 2.0 协议
- 通常，帐号中心提供的API均为与用户有关的接口；该部分接口调用均需要提供oauth的access_token

> 关于授权、认证的区别可参见 认证、授权和凭证

【授权】第三方登录（oauth）接入指南

1. 概述

本文档主要面向智慧教育云平台的第三方应用开发者，旨在通过接入准备，集成时序，接口定义等内容指导开发者实现第三方应用与网龙智慧教育云平台就用户认证授权的集成对接。

2. 术语说明

名词	含义
open_id	开放平台帐号中心给应用的授权用户的唯一标识， 用来标记一个具体的用户
access_token	开放平台帐号中心颁发给应用的用于调用资源接口的访问令牌
code	授权码
oauth2	一种标准授权协议 https://oauth.net/2/

3. 接入准备

第三方应用接入本开放平台需要先提交申请,提交的信息包括应用名称、应用介绍、应用域名、应用 LOGO、应用负责人等信息（提交途径另行沟通），

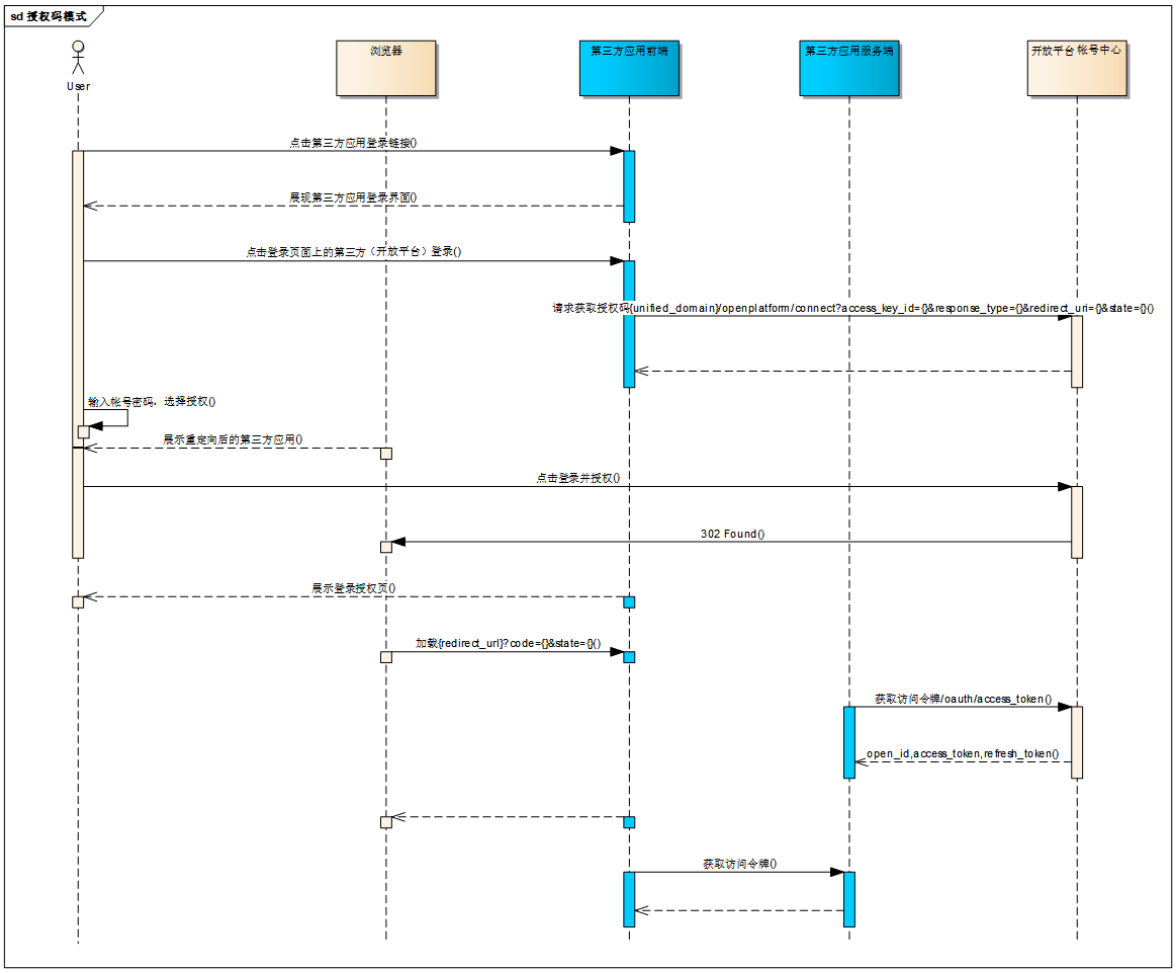
开放平台审核通过后颁发Access Key ID(AK)与Secret Access Key(SK)， 其中应用秘钥要求保存在应用的服务端。

4. 应用集成 网龙帐号登录

指第三方应用把智慧教育云平台帐号作为应用的第三方登录来对接，应用需要做的步骤如下

（1）在第三方登录(如微信、微博等)页面集成智慧教育云平台帐号的登录链接
（2）采用授权码模式与智慧教育云平台 开放平台帐号中心 对接，即先向 开放平台帐号中心![]获取授权码
（3）通过授权码获取访问令牌
（4）通过访问令牌获取资源信息，如用户信息。

4.1 集成总体时序



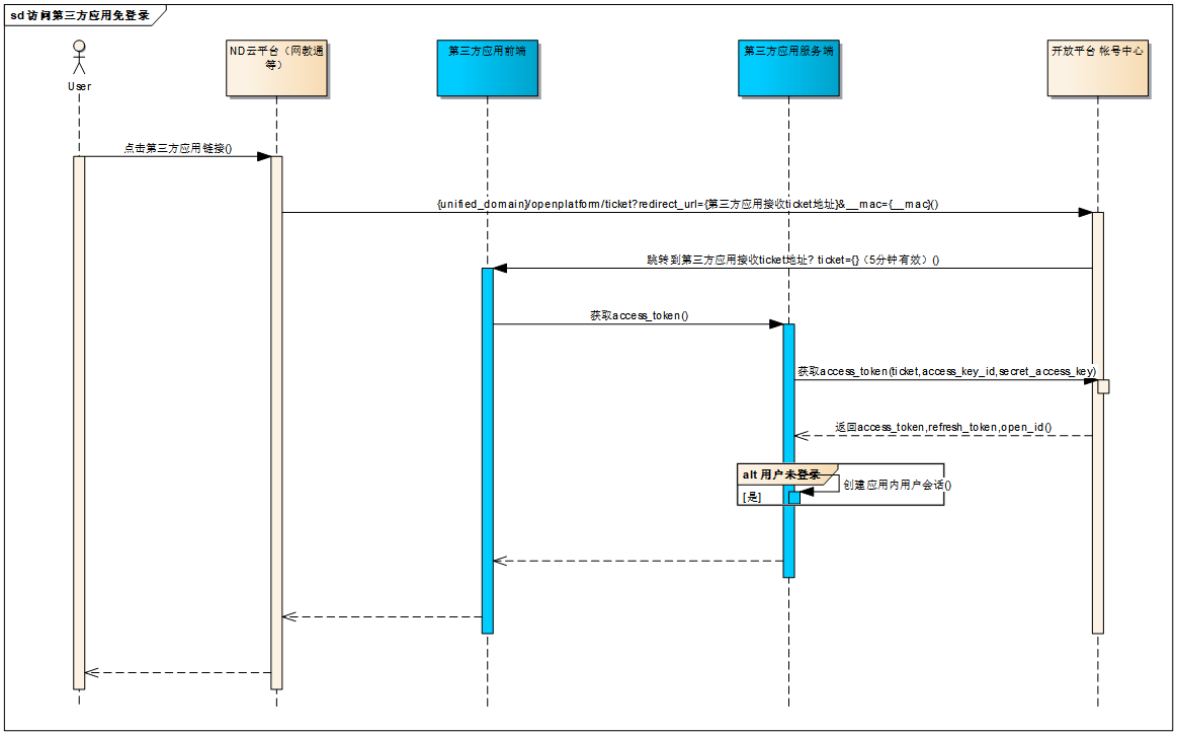
上述时序需要使用 6.1，6.2，6.3，6.4 接口

5. 应用集成 网龙智慧教育云平台（如：福建网教通、湖北省平台、陕西教育云，现代化监测平台）

指第三方应用通过 开放平台帐号中心 的能力实现与智慧教育云平台单点登录，即用户通过智慧教育云平台登录后，访问第三方应用均无需二次认证,第三方应用需要支持的能力如下

（1）接收智慧教育云平台跳转第三方应用时 URL 传递的临时会话票据 ticket
（2）通过 ticket 向 开放平台帐号 换取访问令牌 access_token
（3）通过 access_token 请求资源 API，如获取用户信息等
（4）通过令牌续约来延续 access_token 有效期

5.1 集成总体时序



上述时序需要使用 6.2，6.3，6.4 接口

6. 开放平台帐号中心 接口定义

默认约定

- 没有特殊说明的情况下，POST方法提交的请求 默认 ：http header Content-Type: application/json; 数据以json格式提交

例：

> HTTP HEADER：

```
Accept: application/json
Content-Type: application/json
```

> BODY DATA:

```
{
  "access_key_id": "9xxxad82-XXXX-0000-XXXX-02d26bc31db6",
  "secret_access_key": "XX56aab7c3d642458f2af8067XXXXXXX",
  "ticket": "XXXXAAAA-XXXX-4e7e-0000-0b9a92ca2b3d",
  "grant_type": "ticket"
}
```

- 所有接口请求都需要HMac签名 详见【认证】开放平台API的认证说明

6.1 获取授权码页面

请求URL:

- http://{unified_domain}/openplatform/connect?sdp-app-id={sdp-app-id}response_type={response_type}&access_key_id={access_key_id}&redirect_uri={redirect_uri}&scope={scope}&state={state}&type={type}

请求方式：

- GET

请求参数：

参数名	是否必须	类型	说明
sdp-app-id	是	string	开放平台租户标识（获取方式如下）
access_key_id	是	string	应用唯一标识,应用注册时由开放平台分配
redirect_uri	是	string	授权成功的回调地址, 其中域名必须符合应用注册时填写的安全域名
response_type	是	string	响应类型，目前仅支持传code
scope	是	string	scope_base（获取你的公开信息） 与scope_userinfo（获取你的详细信息）
state	是	string	第三方应用自行产生，可以使用随机字符串
type	否	string	h5 页面，不传为web页面，传‘h5’为h5页面

返回参数：

- 授权成功开放平台帐号中心会返回302 重定向到应用提供的{redirect_uri}页面并且带上code和state参数{redirect_uri}?code={code}&state={state}
- 若用户禁止授权，则重定向后不会带上code参数，仅会带上state参数{redirect_uri}?state={state}

关于scope

- scope标识应用授权作用域，拥有多个作用域用空格分隔，目前可组合的项有
- scope_base: 该作用域需显式地向用户询问是否允许某某应用访问您的基本信息(昵称，头像，性别)

6.2 获取访问令牌

请求URL:

- http://{api-gateway-host}/oauth/access_token

请求方式：

- POST

请求参数：

参数名	是否必须	类型	说明
access_key_id	是	string	应用唯一标识,应用注册时由开放平台分配

secret_access_key	是	string	应用密钥,应用注册时由开放平台分配
ticket	二选一	string	填写智慧教育云平台跳转到第三方应用目标页时
code	二选一	string	授权码，grant_type 传 authorization_code 时必
grant_type	是	string	授权类型，固定传 authorization_code 或 ticket

请求示例[ticket方式，code类似]（body）

HTTP HEADER

```
Accept: application/json
Content-Type: application/json
```

HTTP BODY

```
{
  "access_key_id": "9xxxad82-0000-4b83-aca7-02d26bc31db6",
  "secret_access_key": "XX56aab7c3d642458f2af8067ab0869e",
  "ticket": "191fc3b5-DDDD-XXXX-OOOO-0b9a92ca2b3d",
  "grant_type": "ticket"
}
```

返回参数：

参数	含义
access_token	访问令牌，接口调用凭证，有效期 7 天
expires_at	access_token 的过期时间
refresh_token	用于刷新 access_token，有效期 60 天
open_id	授权用户唯一标识
server_time	服务器时间

6.3令牌续约

请求URL:

- http://{api-gateway-host}/oauth/refresh_token

请求方式：

- POST

请求参数：

```
{
    "access_key_id":"1d1fbd2d-8d12-4c89-9979-7418fb41b362",
    "refresh_token":"28f705cd-ba7e-467b-a596-507fcd452eeb",
    "grant_type":"refresh_token"
}
```

参数名	是否必须	类型	说明
access_key_id	是	string	应用唯一标识,应用注册时由开放平台分配
refresh_token	是	string	填写通过 access_token 获取到的 refresh_token 参数
grant_type	是	string	固定填 refresh_token 这个字符串

返回参数：

参数	含义
access_token	访问令牌，接口调用凭证，有效期 7 天
expires_at	access_token 的过期时间
refresh_token	用于刷新 access_token，有效期 60 天
open_id	授权用户唯一标识
server_time	服务器时间

6.4 获取用户公开信息

请求URL:

- http://{api-gateway-host}/base/get_user_info

请求方式：

- POST

请求参数：

```
{
    "open_id":"66d37f73-be4b-4225-a1e5-aea24c4035ab",
    "access_token":"28f705cd-ba7e-467b-a596-507fcd452eeb"
}
```

参数名	是否必须	类型	说明
open_id	是	string	授权用户唯一标识
access_token	是	string	访问令牌，接口调用凭证

返回参数：

参数	含义
open_id	开放平台帐号中心 颁发给应用的授权用户的唯一标识， 用来标记一个具体的用户
nick_name	昵称
avatar_url	头像
gender	性别 1：男，2：女，0：未知

6.5 获取ticket页面

请求URL:

- http://{unified_domain}/openplatform/ticket?redirect_url={redirect_url}&__mac={__mac}

请求方式：

- GET

请求参数：

参数名	是否必须	类型	说明
__mac	是	string	网龙内部认证协议
redirect_url	是	string	第三方应用的完整http 地址

返回参数：

- 授权成功开放平台帐号中心会返回302 重定向到应用提供的{redirect_url}页面并且带上ticket参数
  {redirect_uri}?ticket={ticket}

  示例

- http://new.hbeducloud.com/openplatform/ticket?redirect_url=http://阳光评价/course/&__mac={__mac}

- http 302 跳转至阳光评价应用并携带真实ticket信息

- http://阳光评价/course/?ticket={real-ticket}

7. 接入地址

{api-gateway-host}

获取方式：



{unified_domain}

统一域名，即开放平台前台页面的域名

如：

湖北网教通：http://new.hbeducloud.com

获取方式：



8. 授权范围的说明(scope)

> 开放平台帐号中心可提供的授权范围分为scope_base（获取你的公开信息）与scope_userinfo（获取你的详细信息），其中scope_base为默认授权范围

属性	属性2	scope	备注
昵称	-	scope_base	
头像	-	scope_base	
性别	-	scope_base	
姓名	-	scope_userinfo	

属性	属性2	scope	备注
身份类型	-	scope_userinfo	如老师，学生等， 真实反映用户在客观组织中承担的身份， 并非指用户在软件系统 (如UC，RBAC，DAC) 里所承担的角色; 即用户身份类型不同于UC的组织管理员， 机构管理员也不同于RBAC或DAC中定义的角色
组织	统一社会信用代码	scope_userinfo	
组织	组织名称	scope_userinfo	
组织	所在区域 （省市区县）	scope_userinfo	
组织	组织类型（学校、企业、主管机构）	scope_userinfo	
所在节点路径	节点名称	scope_userinfo	
所在节点路径	节点类型 （如班级，年级）	scope_userinfo	
所在节点路径	节点路径 （格式如： 父节点名称/ 祖先节点名称/… /根组织名称）	scope_userinfo	.

9.sdp-app-id查看位置：开放平台-开发文档-接口调用，如图：


应用管理
接口查看
调试工具
业务管理
开发文档
接口调用
API调用方法详解
开放平台的API是基于HTTP协议来调用的，开发者可以直接根据协议来封装HTTP请求进行调用，以下主要是针对自行封装HTTP请求进行API调用的原理进行说明。
调用入口
调用API的服务URL地址，开放平台目前提供了1个环境：正式环境。
API 网关域名：http://mvp.gw.101.com/
sdp-app-id：9c7f56a3-d956-4014-8ffc-4c512c077f38

【授权】通过code获取access_token示例

```java
@RequestMapping(value = "/home", method = RequestMethod.GET)//需要设置的IDP的回调地址redirect_uri
public Object get(@RequestParam("code")String code) throws IOException {
    if(StringUtils.isNotBlank(code)){
        String accessKeyId = "xxxx";
        String secretAccessId = "xxxx";
        HttpClient httpClient = NdHttpClientBuilder.getHttpClient(accessKeyId, secretAccessId);//此类由sdk提供
        JSONObject obj = new JSONObject();
        obj.put("access_key_id", accessKeyId);
        obj.put("secret_access_key", secretAccessId);
        obj.put("code", code);
        obj.put("grant_type", "authorization_code");
        HttpPost request = new HttpPost(API_GATEWAY_HOST+"/oauth/access_token");
        request.setHeader("Content-Type","application/json");
        StringEntity entity = new StringEntity(obj.toString());
        request.setHeader("sdp-app-id",SDP_APP_ID);
        request.setEntity(entity);
        HttpHost httpHost = HttpHost.create(API_GATEWAY_HOST);
        HttpResponse httpResponse = httpClient.execute(httpHost,request);
        if(httpResponse.getStatusLine().getStatusCode()==200){
            String result = EntityUtils.toString(httpResponse.getEntity());
            JSONObject token = JSONObject.parseObject(result,JSONObject.class);
            return token;
        }
        return EntityUtils.toString(httpResponse.getEntity());
    }
    return "code is null";
}
```

【认证】开放平台应用的开发者、服务提供者认证简要说明

一、概念

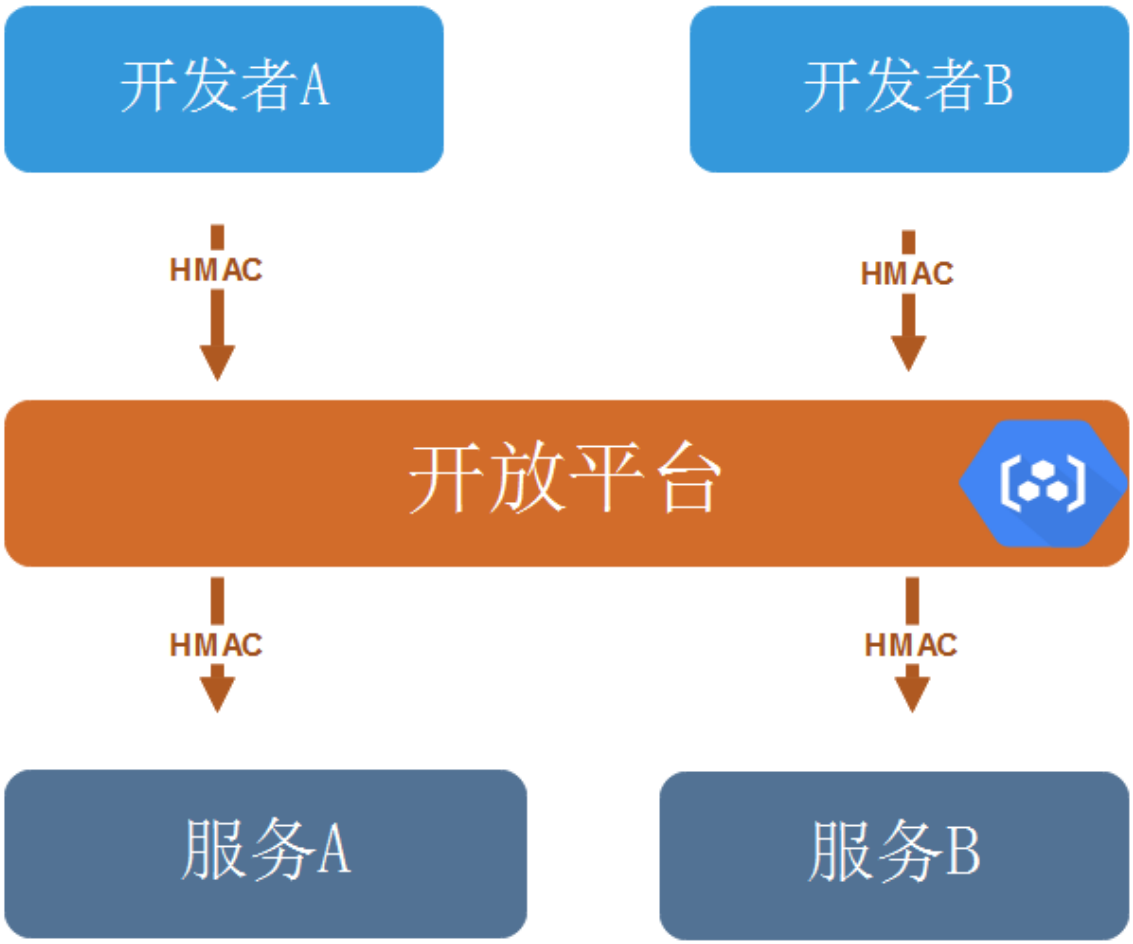
应用开发者：应用的开发者，调用方，它通常需要基于开放平台提供的api进行二次开发。

开放平台：智慧教育云平台提供的面向开发者的技术平台，提供应用接入、服务接入等管理能力。

服务提供者：服务的开发者、提供者，他提供了一定的技术能力，并通过开放平台进行“托管”，以供其他开发者来调用和使用。甚至可基于开放平台收取一定的服务费。

二、认证

开发者调用开放平台api、服务提供者在开放平台托管服务，均需要支持统一的HMAC认证方式



具体认证说明可参看文章：开放平台API的认证说明

【认证】开放平台API的认证说明

API认证简介

开发者可以通过两种方式与开放平台api进行交互，包括认证方式和匿名方式。

对于认证方式，需要通过使用Access Key Id / Secret Access Key加密的方法来验证某个请求的发送者身份。Access Key Id（AK）用于标示用户，Secret Access Key（SK）是用户用于加密认证字符串和开放平台用来验证认证字符串的密钥，其中SK必须保密，只有用户和开放平台知道。

当开放平台接收到用户的请求后，系统将使用相同的SK和同样的认证机制生成认证字符串，并与用户请求中包含的认证字符串进行比对。如果认证字符串相同，系统认为用户拥有指定的操作权限，并执行相关操作；如果认证字符串不同，系统将忽略该操作并返回错误码。

Java client sdk下载

示例工程下载

名词解释

API认证方式

- 开放平台API使用基于认证字符串的HTTP请求签名机制来验证用户身份。对于每个HTTP请求，都需要携带一个认证字符串然后通过http header将这个认证字符串包含在请求中：
- 在HTTP Header中包含认证字符串
- 通常使用的方法是在HTTP请求的Authorization头域中包含认证字符串，除了匿名请求之外，所有与开放平台的交互都应该包含该字段。

步骤（必读）

- 4.1 申请认证账号和密钥（AK/SK），可参见帮助文章 已创建应用，如何申请Access Key 、Secret Access
- 4.2 对每个将要发起的请求，携带必要的http header信息

```
Date: Thu, 22 Jun 2017 17:15:21 GMT
Authorization: hmac username="myUserName", algorithm="hmac-sha256", headers="date request-line host", signature="ujWCGHeec9Xd6UD2zlyxiNMCiXnDOWeVFMu5VeRUxtw="
```

其中：Date、Authorization均为http协议中的通用头部信息

Date规范可参见 https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Headers/Date

Authorization 为本平台指定的签名算法

4.2.1 username=”myUserName” 中的myUserName为认证账号，即AK

4.2.2 digest=HMAC-SHA256(, “mySecret”) 中的mySecret为认证秘钥，即SK

4.2.3 signature的签名信息包含了date信息，服务端会对date信息进行过期校验，有效期为5分钟

signature的计算逻辑如下：

```
signing_string="date: Thu, 22 Jun 2017 17:15:21 GMT\nGET /requests HTTP/1.1\nhost: xxx.xxx.com"
digest=HMAC-SHA256(<signing_string>, "mySecret")
base64_digest=base64(<digest>)
signature = base64_digest
```

signing_string说明：时间 请求的request-line 请求的host，如：需要请求接口 https://mvp.gw.101.com/oauth/access_token POST请求的，则signing_string 为”date: Thu, 22 Jun 2017 17:15:21 GMT\nPOST /oauth/access_token HTTP/1.1\nhost: mvp.gw.101.com”，这里的顺序是date request-line host，与上面的Authorization: hmac username=”myUserName”, algorithm=” hmac-sha256”, headers=”date request-line host”, signature=”ujWCGHeec9Xd6UD2zlyxiNMCiXnDOWeVFMu5VeRUxtw=” 中 headers的顺序一致就行，无需强制要求这个顺序，校验时也是根据headers中顺序解密匹配

使用SDK发送请求（必读）

开发人员可以使用开放平台提供的SDK来帮助发送接口请求，SDK自动在请求发起前补充认证信息

```
//sdk使用样例, 基于apache http client
public Object execute(JSONObject token) throws IOException {
      String GATEWAY_HOST = "https://mvp.gw.101.com";//网关地址
      String GATEWAY_GET_URL = "https://mvp.gw.101.com/base/subjects";//具体的请求地址, 域名与上面一致
      HttpClient httpClient = NdHttpClientBuilder.getHttpClient("YOUR REAL AK","YOUR REAL SK");//此类由sdk
提供
      HttpRequest request = new HttpGet(GATEWAY_GET_URL);
      request.setHeader("idp-open-id",token.getString("open_id"));//需要用户信息必传
      request.setHeader("idp-access-token",token.getString("access_token"));//需要用户信息必传
      request.setHeader("sdp-app-id",SDP_APP_ID);
      request.setHeader("OPF-APP-ID",OPEN_APP_ID);//开放平台的应用id, 为特殊接口需要, 非必须
      request.setHeader("Content-Type","application/json");
      HttpHost httpHost = HttpHost.create(GATEWAY_HOST);
      HttpResponse httpResponse = httpClient.execute(httpHost,request);
      return EntityUtils.toString(httpResponse.getEntity());
}
```

header字段说明

HEADER 字段	是否必选	用途	说明
idp-open-id	可选	用户openid	若请求的接口需要用户相关信息时必传
idp-access-token	可选	用户access_token	若请求的接口需要用户相关信息时必传
sdp-app-id	必选	租户ID	

HEADER 字段	是否必选	用途	说明
sdp-biz-type	可选	业务类型	如果有颁发biz-tyupe，必传
sdp-org-id	可选	组织ID	如果有提供org-id，必传
Content-Type	必选	标准头	application/json
OPF-APP-ID	可选	开放平台应用ID	特定接口需要必传， 具体看接口说明文档

重要说明

- sdp-app-id信息为创建应用后，系统自动生成的固定值

查看位置：开放平台-开发文档-接口调用，如图：



如何获取sdp-app-id

应用的sdp-app-id参数获取方式

路径：

- 进入开放平台
- 左侧菜单：开发文档-> 接口调用->调用入口部分

图示：



常用技术说明

使用本开放平台必须要了解的技术和名词

理解什么是 OAuth 2.0 协议

什么是 OAuth 2.0

OAuth 2.0 是一个行业的标准授权协议。OAuth 2.0 专注于简化客户端开发人员，同时为 Web 应用程序，桌面应用程序，手机和客厅设备提供特定的授权流程。

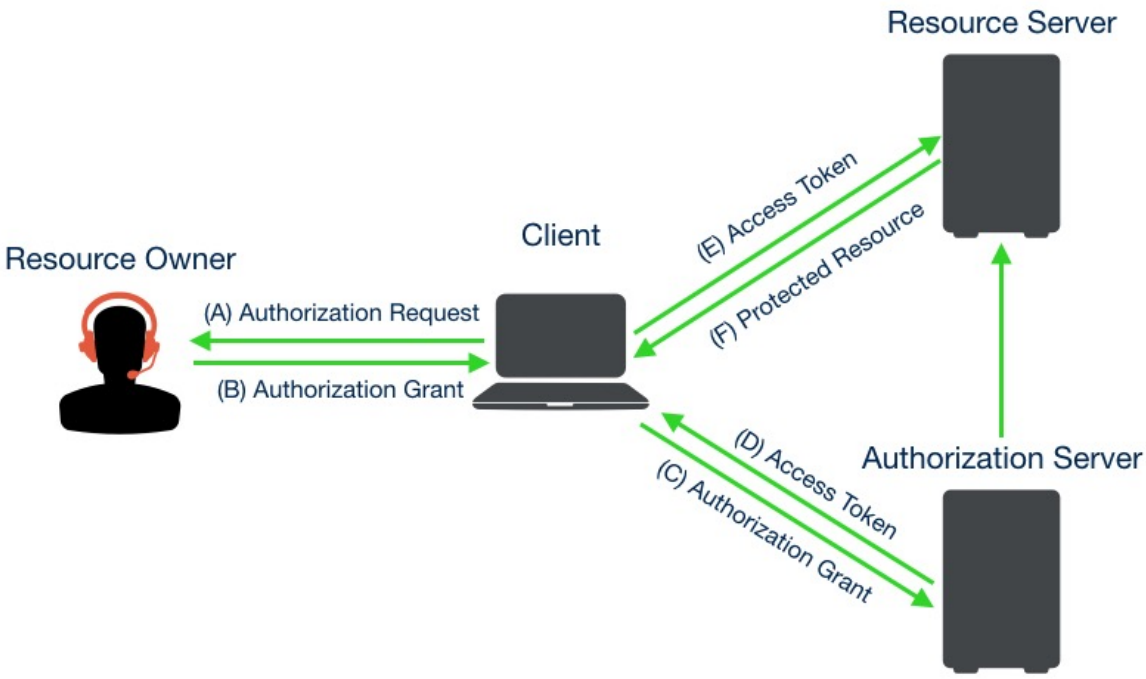
它的最终目的是为第三方应用颁发一个有时效性的令牌 token。使得第三方应用能够通过该令牌获取相关的资源。常见的场景就是：第三方登录。当你想要登录某个论坛，但没有账号，而这个论坛接入了如 QQ、Facebook 等登录功能，在你使用 QQ 登录的过程中就使用的 OAuth 2.0 协议。

角色

首先需要介绍的是 OAuth 2.0 协议中的一些角色，整个授权协议的流程都将围绕着这些角色：

- resource owner，资源所有者，能够允许访问受保护资源的实体。如果是个人，被称为 end-user。
- resource server，资源服务器，托管受保护资源的服务器。
- client，客户端，使用资源所有者的授权代表资源所有者发起对受保护资源的请求的应用程序。如：web网站，移动应用等。
- authorization server，授权服务器，能够向客户端颁发令牌。
- user-agent，用户代理，帮助资源所有者与客户端沟通的工具，一般为 web 浏览器，移动 APP 等。

协议流程



上图详细的描述了这四个角色之间的步骤流程：

(A) Client 请求 Resource Owner 的授权。授权请求可以直接向 Resource Owner 请求，也可以通过 Authorization Server 间接的进行。

(B) Client 获得授权许可。

(C) Client 向 Authorization Server 请求访问令牌。

(D) Authorization Server 验证授权许可，如果有效则颁发访问令牌。

(E) Client 通过访问令牌从 Resource Server 请求受保护资源。

(F) Resource Server 验证访问令牌，有效则响应请求。

授权

一个客户端想要获得授权，就需要先到服务商那注册你的应用。一般需要你提供下面这些信息：

- 应用名称
- 应用网站
- 重定向 URI 或回调 URL（redirect_uri）
- 重定向 URI 是服务商在用户授权（或拒绝）应用程序之后重定向用户的地址，因此也是用于处理授权代码或访问令牌的应用程序的一部分。

在你注册成功之后，你会从服务商那获取到你的应用相关的信息：

- 客户端标识 client_id
- 客户端密钥 client_secret
- client_id用来表识客户端（公开），client_secret用来验证客户端身份（保密）。

授权类型

OAuth 2.0 列举了四种授权类型，分别用于不同的场景：

- Authorization Code（授权码 code）：服务器与客户端配合使用。
- Implicit（隐式 token）：用于移动应用程序或 Web 应用程序（在用户设备上运行的应用程序）。
- Resource Owner Password Credentials（资源所有者密码凭证 password）：资源所有者和客户端之间具有高度信任时（例如，客户端是设备的操作系统的一部分，或者是一个高度特权应用程序），以及当其他授权许可类型（例如授权码）不可用时被使用。
- Client Credentials（客户端证书 client_credentials）：当客户端代表自己表演（客户端也是资源所有者）或者基于与授权服务器事先商定的授权请求对受保护资源的访问权限时，客户端凭据被用作为授权许可。

下面来具体说说这四种授权。注意重定向一定要用 302。

授权码模式

该方式需要资源服务器的参与，应用场景大概是：

- 资源拥有者（用户）需要登录客户端（APP），他选择了第三方登录。

- 客户端（APP）重定向到第三方授权服务器。此时客户端携带了客户端标识（client_id），那么第三方就知道这是哪个客户端，资源拥有者确定（拒绝）授权后需要重定向到哪里。
- 用户确认授权，客户端（APP）被重定向到注册时给定的 URI，并携带了第三方给定的 code。
- 在重定向的过程中，客户端拿到 code 与client_id、client_secret去授权服务器请求令牌，如果成功，直接请求资源服务器获取资源，整个过程，用户代理是不会拿到令牌 token 的。
- 客户端（APP）拿到令牌 token 后就可以向第三方的资源服务器请求资源了。

隐式模式

该方式一般用于移动客户端或网页客户端。隐式授权类似于授权码授权，但 token 被返回给用户代理再转发到客户端（APP），因此它可能会暴露给用户和用户设备上的其它客户端（APP）。此外，此流程不会对客户端（APP）的身份进行身份验证，并且依赖重定向 URI（已在服务商中注册）来实现此目的。

基本原理：要求用户授权应用程序，然后授权服务器将访问令牌传回给用户代理，用户代理将其传递给客户端。

资源所有者密码模式

用户将其服务凭证（用户名和密码）直接提供给客户端，该客户端使用凭据从服务获取访问令牌。如果其它方式不可行，则只应在授权服务器上启用该授权类型。此外，只有在客户端受到用户信任时才能使用它（例如，它由服务商自有，或用户的桌面操作系统）。

客户端模式

这种模式只需要提供client_id和client_secret即可获取授权。一般用于后端 API 的相关操作。

POST 请求客户端凭证流程：

- https://oauth.example.com/token?grant_type=client_credentials&client_id=CLIENT_ID&client_secret=CLIENT_SECRET

字段	描述
grant_type	必须，必须设置到客户端证书中。
scope	可选，授权的作用域。

返回值

```
{
  "access_token"  : "...",  "token_type"    : "...",  "expires_in"    : "...",
}
```

如果授权服务器验证成功，那么将直接返回令牌 token，改客户端已被授权。

网龙开放平台相关术语说明

一、第三方登录、用户授权相关

- oauth2：一种标准授权协议https://oauth.net/2/
- code：授权码（Authorization Code）授权码模式下的响应；使用idp提供的登录地址登录成功后，会根据约定协议返回给调用方，它用来获取access_token；
- access_token：IDP颁发给应用的用于调用资源接口的访问令牌，用于调用IDP提供的API.
- open_id：IDP颁发给应用的授权用户的唯一标识，用来标记一个具体的用户，基于IDP授权过的用户标识。代表一个用户。

二、平台概念

- 网龙智慧教育云平台：泛指提供第三方应用上架能力的产品，例如湖北教育云、福建网教通等由网龙公司提供的云品台产品。在某些文档中简称“云平台”
- APP ID:开发者在开放平台上创建的应用标识
- 应用：第三方厂家开发供用户使用的应用，可以是web端应用、手机端应用、pc端应用
- 服务:第三方厂家提供的服务，他通常提供给开发人员使用，表现形式为可调用的HTTP协议API

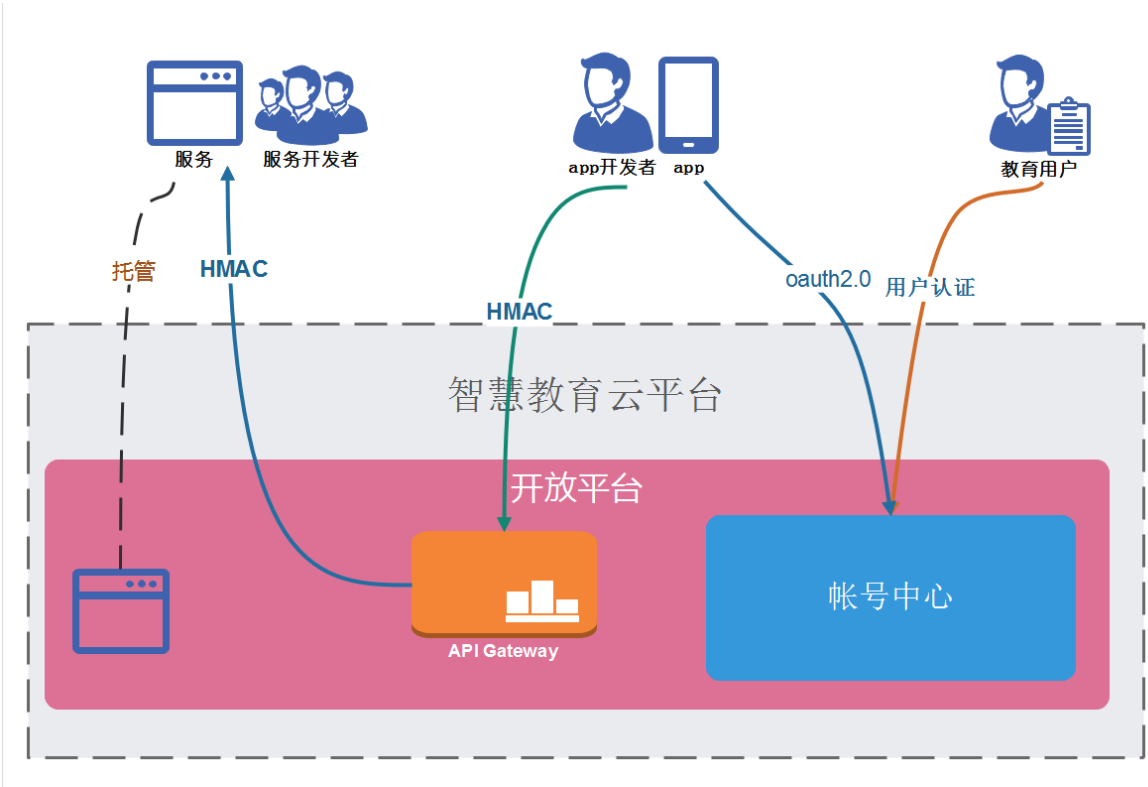
三、开放平台API网关、调用服务、服务开相关

- Access Key ID：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- Access Key Key：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。
- HMAC：是密钥相关的哈希运算消息认证码，HMAC运算利用哈希算法，以一个密钥和一个消息为输入，生成一个消息摘要作为输出。参见https://en.wikipedia.org/wiki/HMAC

开放平台上下文关系简要说明

名词说明

- 智慧教育云平台用户：智慧教育云平台的目标用户
- app开发者：通常是第三方厂家
- 服务开发者：通常是智慧教育云平台合作的第三方厂家和网龙公司
- app（应用）：第三方厂家开发供用户使用的应用，可以是web端应用、手机端应用、pc端应用
- 服务：第三方厂家提供的服务，他通常提供给开发人员使用



AK/SK简介

- AK（Access Key ID）/SK（Secret Access Key），主要用于对开发者的调用行为进行鉴权和认证，相当于开放平台的API专用的用户名及密码。
- 通过API网关向下层服务发送请求时，必须使用AK(Access Key ID)、SK(Secret Access Key)对请求进行签名。

用途

一般用于后台程序执行API调用时的服务端认证

原理

客户端：

1. 构建http请求（包含 access key）；
2. 使用请求内容和 使用secret access key计算的签名(signature)；
3. 发送请求到服务端。

服务端：

1. 根据发送的access key 查找数据库得到对应的secret-key；
2. 使用同样的算法将请求内容和 secret-key一起计算签名（signature），和步骤2一样；
3. 对比用户发送的签名和服务端计算的签名，两者相同则认证通过，否则失败。

认证、授权和凭证

认证、授权、凭证

首先，认证和授权是两个不同的概念，为了让我们的 API 更加安全和具有清晰的设计，理解认证和授权的不同就非常有必要了，它们在英文中也是不同的单词。

认证是 authentication，指的是当前用户的身份，当用户登陆过后系统便能追踪到他的身份做出符合相应业务逻辑的操作。即使用户没有登录，大多数系统也会追踪他的身份，只是当做来宾或者匿名用户来处理。

认证技术解决的是 “我是谁？”的问题。

授权则不同，授权是 authorization，指的是什么样的身份被允许访问某些资源，在获取到用户身份后继续检查用户的权限。单一的系统授权往往是伴随认证来完成的，但是在开放 API 的多系统结构下，授权可以由不同的系统来完成，例如 OAuth。

授权技术是解决“我能做什么？”的问题。

实现认证和授权的基础是需要一种媒介（credentials）来标记访问者的身份或权利，在现实生活中每个人都需要一张身份证才能访问自己的银行账户、结婚和办理养老保险等，这就是认证的凭证；在古代军事活动中，皇帝会给出战的将军颁发兵符，下级将领不关心持有兵符的人，只需要执行兵符对应的命令即可。在互联网世界中，服务器为每一个访问者颁发 session ID 存放到 cookie，这就是一种凭证技术。数字凭证还表现在方方面面，SSH 登录的密匙、JWT 令牌、一次性密码等。

用户账户也不一定是存放在数据库中的一张表，在一些企业 IT 系统中，对账户管理和权限有了更多的要求。所以账户技术（accounting）可以帮助我们使用不同的方式管理用户账户，同时具有不同系统之间共享账户的能力。例如微软的活动目录（AD），以及简单目录访问协议（LDAP），甚至区块链技术。

还有一个重要的概念是访问控制策略（AC）。如果我们需要把资源的权限划分到一个很细的粒度，就不得不考虑用户以何种身份来访问受限的资源，选择基于访问控制列表（ACL）还是基于用户角色的访问控制（RBAC）或者其他访问控制策略。

在流行的技术和框架中，这些概念都无法孤立的被实现，因此在现实中使用这些技术时，大家往往为一个 OAuth2 是认证还是授权这种概念争论不休。为了容易理解，我在文末附上了一份常见技术和概念的术语表。下面我会介绍在API开发中常常使用的几种认证和授权技术：HAMC、OAuth2，以及凭证技术JWT token。

HMAC（AK/SK）认证

在我们对接一些 PASS 平台和支付平台时，会要求我们预先生成一个 access key（AK） 和 secure key（SK），然后通过签名的方式完成认证请求，这种方式可以避免传输 secure key，且大多数情况下签名只允许使用一次，避免了重放攻击。

这种基于 AK/SK 的认证方式主要是利用散列的消息认证码 (Hash-based MessageAuthentication Code) 来实现的，因此有很多地方叫 HMAC 认证，实际上不是非常准确。HMAC 只是利用带有 key 值的哈希算法生成消息摘要，在设计 API 时有具体不同的实现。

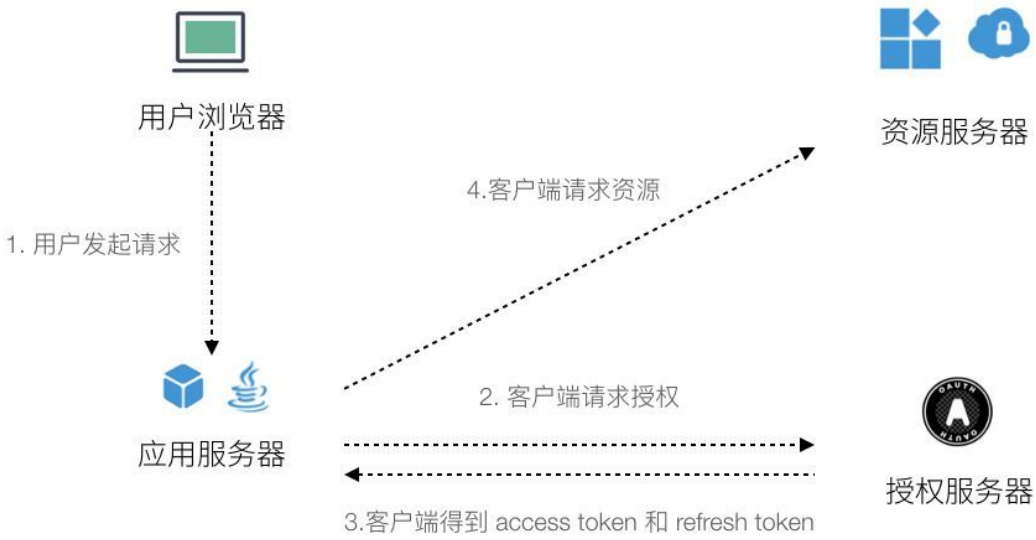
HMAC 在作为网络通信的认证设计中作为凭证生成算法使用，避免了口令等敏感信息在网络中传输。基本过程如下：

1. 客户端需要在认证服务器中预先设置 access key（AK 或叫 app ID） 和 secure key（SK）
2. 在调用 API 时，客户端需要对参数和 access key 进行自然排序后并使用 secure key 进行签名生成一个额外的参数 digest
3. 服务器根据预先设置的 secure key 进行同样的摘要计算，并要求结果完全一致
4. 注意 secure key 不能在网络中传输，以及在不受信任的位置存放（浏览器等）

OAuth2 和 Open ID

OAuth（开放授权）是一个开放标准，允许用户授权第三方网站访问他们存储在另外的服务提供者上的信息，而不需要将用户名和密码提供给第三方网站或分享他们数据的所有内容。

OAuth 是一个授权标准，而不是认证标准。提供资源的服务器不需要知道确切的用户身份（session），只需要验证授权服务器授予的权限（token）即可。



上图只是 OAuth 的一个简化流程，OAuth 的基本思路就是通过授权服务器获取 access token 和 refresh token（refresh token 用于重新刷新access token），然后通过 access token 从资源服务器获取数据 。在特定的场景下还有下面几种模式：

- 授权码模式（authorization code）
- 简化模式（implicit）
- 密码模式（resource owner password credentials）
- 客户端模式（client credentials）

如果需要获取用户的认证信息，OAuth 本身没有定义这部分内容，如果需要识别用户信息，则需要借助另外的认证层，例如 OpenID Connect。

JWT

在 OAuth 等分布式的认证、授权体系下，对凭证技术有了更多的要求，比如包含用户 ID、过期等信息，不需要再外部存储中关联。因此业界对 token 做了进一步优化，设计了一种自包含令牌，令牌签发后无需从服务器存储中检查是否合法，通过解析令牌就能获取令牌的过期、有效等信息，这就是JWT （JSON Web Token）。

JWT 是一种包含令牌（self-contained token），或者叫值令牌 （value token），我们以前使用关联到 session 上的 hash 值被叫做引用令牌（reference token）。

JWT token 在微服务的系统中优势特别突出。多层调用的 API 中可以直接传递 JWT token，利用自包含的能力，可以减少用户信息查询次数；更重要的是，使用非对称的加密方式可以通过在系统中分发密匙的方式验证 JWT token。

当然 OAuth 对 access token 等凭证所选用的技术并没有做出限制，OAuth 并不强制使用 JWT，在使用 JWT 自包含特性的优势时，必须考虑到 JWT 撤回困难的问题。在一些对撤回 token 要求很高的项目中不适合使用JWT，即使采用了一些方案实现（whitelist 和 blacklist）也违背了设计 JWT 的初衷。

OAuth、Open ID、OpenID Connect

- OAuth 负责解决分布式系统之间的授权问题，即使有时候客户端和资源服务器或者认证服务器存在同一台机器上。OAuth 没有解决认证的问题，但提供了良好的设计利于和现有的认证系统对接。

- Open ID 解决的问题是分布式系统之间身份认证问题，使用Open ID token 能在多个系统之间验证用户，以及返回用户信息，可以独立使用，与 OAuth 没有关联。
- OpenID Connect 解决的是在 OAuth 这套体系下的用户认证问题，实现的基本原理是将用户的认证信息（ID token）当做资源处理。在 OAuth 框架下完成授权后，再通过 access token 获取用户的身份。

Cookie 、Token in Cookie、Session Token 依然被使用

在构建 API 时，开发者会发现我们的认证方式和网页应用有一些不同，除了像 ajax 这种典型的 web 技术外，如果我们希望 API 是无状态的，不推荐使用 Cookie。

使用 Cookie 的本质是用户第一次访问时服务器会分配一个 Session ID，后面的请求中客户端都会带上这个 ID 作为当前用户的标志，因为 HTTP 本身是无状态的，Cookie 属于一种内建于浏览器中实现状态的方式。如果我们的 API 是用来给客户端使用的，强行要求 API 的调用者管理Cookie 也可以完成任务。

在一些遗留或者不是标准的认证实现的项目中，我们依然可以看到这些做法，快速地实现认证。

- 使用 cookie，例如 web 项目中 ajax 的方式
- 使用 session ID 或 hash 作为 token，但将 token 放入 header 中传递
- 将生成的 token （可能是JWT）放入 cookie 传递，利用 HTTPonly 和 Secure 标签保护 token

FOR 网龙内部

本章节仅面向网龙内部开发人员

根据open_id+access_token获取uc mac token

根据open_id+access_token获取uc mac token

请求URL:

- https://{uc-gateway}/v1.1/idp/get_mactoken

请求方式：

- POST

请求参数：

参数名	是否必须	类型	说明
open_id	是	string	授权用户唯一标识
access_token	是	string	IDP颁发的访问令牌
session_id	是	string	通过UC的创建会话获取

返回参数：

```
{
    "account_type":"person",  -- 帐户类型: person 个人帐户, org 组织帐户;
    "account_id":"123", -- 个人帐户id 注:该字段返回的是密文, 客户端用户创建会话时返回的会话密钥(session_key)进行DES解密
    "user_id"":"123", -- 组织帐户id 注:该字段返回的是密文, 客户端用户创建会话时返回的会话密钥(session_key)进行DES解密
    "region":"wx", --归属IDC, wx:无锡, sg：新加坡,ca:加州(全球化后生效),
    "access_token" : "2YotnFZFEj r1zCsicMWpAA" ,--UC Token
    "expires_at" : "2015-06-30T14:17:21.275+0800", --Token过期时间
    "refresh_token" : "tGzv3J OkF0XG5Qx2TlKWIA" , --刷新用Token
    "mac_key" : "adij q39j dlaska9asud" , --Mac key值  注:该字段返回的是密文, 客户端用户创建会话时返回的会话密钥(session_key)进行DES解密
    "mac_algorithm" : "hmac- sha- 256" , --Mac算法
    "server_time": "2019-06-30T18:18:20.777+0800" -- 服务器时间
}
```

根据open_id获取account_id（uc userid）

根据open_id获取account_id

如果调用的是非IDP开放的接口，需要经过智慧教育云平台中转的，智慧教育云平台可以通过如下接口来获取到UC内部的帐户ID

请求URL:

- https://{uc-gateway}/v1.1/idp/{access_token}

请求方式：

- POST

认证方式:

- BTS

请求参数：

参数名	是否必须	类型	说明
open_id	是	string	授权用户唯一标识

返回参数：

参数名	说明
account_id	UC内部的帐户ID
account_type	帐户类型(person/org)

平台规范

开放平台的建议性规范

开放平台-接口规范【建议】

一、基本规范

- 按照 Richardson Maturity Mode 对于REST评价的模型， 本规范基于 level2 来设计

二、性能要求

（1）接口TPS(每秒请求次数)要求大于2000。

（2）接口时延要求，在最大TPS情况下，99%的请求时延小于150毫秒。

（3）接口稳定性要求，调用成功率> 99.99%。全年服务中断不超过20分钟。

（4）可靠性要求，系统需要支持水平扩容，满足业务发展。

三、安全要求

（1）接口协议：HTTPS。

（2）HTTPS证书要求：使用合法CA颁发的证书，不允许使用自签名证书。

（3）请求方法：GET、POST方式。

（4）接口响应大小限制，接口响应总大小不超过24KB，各字段大小限制参考接口文档定义。

（5）接口中的资源链接（如图片链接、音频链接等）要求是HTTPS协议。

四、认证要求

（1）接口需要有认证机制，提供标准OAuth2.0 或者AK/SK的身份认证方式。

开放平台api错误规范

一、错误规范

当接口出现非2xx、3xx的HTTP响应时，采用返回统一HTTP响应信息：

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
{
    "code": "INVALID_ARGUMENT",
    "message": "{error message}",
    "request_id": "01234567-89ab-cdef-0123-456789abcdef",
    "host_id": "{server identity}",
    "server_time": "2014-01-01T12:00:00Z"
}
```

错误信息结构
错误规范中包含 3 个层级的信息

```
└ http status code
    └ code
    └ message
```

- http status code

符合 HTTP 协议的响应状态码

- code

用来表示某类的错误，如缺少参数、类型不匹配等等，用来对 http status code 进行扩展，开发人员可以据此进行错误的细节处理。详见#code

- message

为错误的摘要信息，并且应该包含对用户处理该错误有指导意义的信息

【重要】对接厂家开发规范

本文介绍华渔网教通平台的重要对接信息

多租户

- 网教通（华渔网教通，以下简称网教通）是一个SAAS模式的多租户教育平台；为不同地区、重要客户等单独提供了各自隔离的软件产品；如湖北省平台、福建网教通、网教通公共版等
- 每个版本的网教通，都拥有一个独立的开放平台，与三方厂家对接
- 不同的租户，拥有完全独立的域名和用于对接三方厂家的开放平台；即：某一个厂家的某款应用，只能对接到一个特定版本的网教通（例：阳光评价已对接湖北省平台）

对第三方厂家的开发建议

- 希望厂家应用本身也支持多租户，以便后期与其他版本的网教通进行对接【建议】
- 首次对接前，请考虑网教通的多租户特性，以及可能存在的后期更多合作可能性；需要设计上能支持以配置的方式来快速对接其他版本的网教通
- 理想情况下，某厂家对接一次网教通开放平台后；需要再次对接其他版本的网教通的时候，应该只需要在对应的开放平台上创建应用、获取AK\SK，配置到自己服务端即可完成对接。

服务接入

- 如何将自己的服务托管到开放平台，以供其他开发人员调用使用

如何接入服务

目前第三方合作企业需要进行在开放平台服务（API)托管的，暂时需要通过线下沟通确认接口信息和颁发认证口令，进行托管，暂未开放自助托管API。

如有实际需要，请联系对口商务负责人、或项目接口人，由后台代为添加接口信息。

https://wiki.doc.101.com/index.php?title=%E6%95%99%E8%82%B2%E4%B8%9A%E5%8A%A1%E8%83%BD%E5%8A%9B-%E5%BC%80%E6%94%BE%E5%B9%B3%E5%8F%B0-%E8%AE%BE%E8%AE%A1#.E6.9C.8D.E5.8A.A1.E7.AB.AF.EF.BC.88.E6.89.98.E7.AE.A1.E6.8E.A5.E5.8F.A3.E6.9C.8D.E5.8A.A1.EF.BC.89.E6.8E.A5.E5.85.A5

接入案例

- 部分对接案例

湖北教育云集成多家三方应用案例

湖北教育云与北京极课大数据平台无缝对接，平台用户一键直达第三方应用，无需二次登陆、认证。



湖北教育云与阳光评价应用集成

阳光评价是面向中小学校，对全体在校学生进行综合素质评估的在线评价系统。学生成长档案记录为评价工作提供了实证依据；小组评价、班级评价、任务和成就等特色功能为学校德育工作提供便利；多角色共同参与的方式让学校教育和家庭教育有机结合。通过该系统，有效地实现了学生评价的自我诊断功能，提升了评价结果的“含金量”，提高了德育管理工作效率，同时加强了家校沟通、实现家校共育。

教育基础数据接口

开放教育相关的基础数据

- 根据openid获取用户详细信息
- 学科查询
- 学段查询
- 学校管理相关（学校列表、学校详情、根据学校编码查询学校）
- 年级管理（查询年级列表、查询单个年级信息）
- 班级管理（查询年级下班级列表、查询学校下单个班级信息）
- 部门管理：查询学校部门列表
- 教师管理（获取班级的班主任列表、获取班主任管理的班级列表、获取班级的任课教师列表、查询学校教师、获取班级的教师列表、获取教师所在部门）
- 学生管理（获取班级学生列表、获取单个学生信息、获取学生所在班级列表、设置学生设备、根据设备编码查询学生、根据学籍号查询学生user_id）
- 家长管理 （获取我的孩子信息、获取家长列表）
- 设备管理（设置用户设备、根据设备编码查询用户）
- 课程表 （获取班级的课程表、获取全校各年级课程表、获取学校课程表、获取场地的课程表、获取学生的课程表、获取老师的课程表）
- 学期信息 （获取学校所有学期、获取单个学期）
- 行政区划（根据IP定位当前的地区的行政区划，行政区划-获取子行政区划列表、行政区划-父级行政区划列表）

根据open_id获取用户详细信息

简要描述：

- 根据openid获取用户详细信息（网教通特有接口，非网教通不可用）

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/wjt/v1/users/extends

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
idp-open-id	是	string	用户openid
idp-access-token	是	string	用户access_token
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

返回示例:

正确时返回:

```
{
    "user_id",7771234567,
    "area": "string",
    "identity_code":"string",
    "is_master": "string",
    "subject": "string",
    "area_code": "string",
    "school_name": "string",
    "grade_name": "string",
    "class_name": "string",
    "card": "string"
    "user_type":"string",
    "org_id":"long",
    "org_name":"string",
    "section_list": [
        {
            "id": "string",
            "name": "string"
        }
    ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
user_id	long	网龙真实用户ID
area	string	所属区县
identity_code	string	学生 学籍号，教师身份证号（工号）
~~is_master~~	string	~~是否班主任(无数据，废弃，待下架)~~
~~subject~~	string	~~学科 (无数据，废弃，待下架)~~
area_code	string	区县代码
~~school_name~~	string	~~学校名称(无数据，废弃，待下架)~~
~~grade_name~~	string	~~年级名称 (无数据，废弃，待下架)~~
~~class_name~~	string	~~班级名称(无数据，废弃，待下架)~~
~~card~~	string	~~学生学籍号/身份证号(无数据，废弃，待下架)~~

参数名	类型	说明
user_type	string	用户类型/用户当前角色(‘STUDENT’: ‘学生’, ‘TEACHER’: ‘老师’, ‘GUARDIAN’: ‘家长’, ‘PRE_SCHOOL_MANAGE’: ‘学校管理者’, ‘PROVINCE_LEADER’: ‘区域管理者’)
org_id	long	所在组织id
org_name	string	所在组织名称
section_list	SectionVo	学校的学段，用户没有学校就没有学段
section_list.id	string	学段编码
section_list.name	string	学段名称

根据openid及accesstoken获取用户信息

简要描述：

- 根据openid及accesstoken获取用户信息

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-12-05	init

请求URL:

- http://{api-gateway-host}/base/user

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
idp-open-id	是	string	用户openid
idp-access-token	是	string	用户access_token
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

返回示例:

正确时返回:

```
{
    "user_id",7771234567,
    "nick_name": "string",
    "real_name": "string",
    "gender": "int"
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
user_id	long	网龙真实用户ID
nick_name	string	昵称
real_name	string	真实姓名
gender	string	性别 1：男，2：女，0：未知

学科查询

简要描述：

- 学科查询

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-17	init

请求URL:

- http://{api-gateway-host}/base/subjects

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

返回示例:

正确时返回:

```
[
  {
```

```
    "id": "string",
    "name": "string"
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	编码
name	string	学科名称

学段查询

简要描述：

- 学段查询

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-17	init

请求URL:

- http://{api-gateway-host}/base/sections

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

返回示例:

正确时返回:

```
[
  {
```

```
    "id": "string",
    "name": "string"
 }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	编码
name	string	学段名称

用户增量数据

简要描述：

- 用户增量数据

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-03-23	init

请求URL:

- http://{api-gateway-host}/base/modify_users/action/search

请求方式：

- POST

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

|参数名|是否必须|类型|默认值|说明|
|body |是|ModifyUserParam||搜索条件|

SchoolVoForSearch

参数名	是否必须	类型	默认值	说明
offset	是	long		分页索引
limit	是	long		每页数量
org_id	是	long		组织id（学校id）
operate	否	string		操作类型：CREATE、UPDATE、DELETE 不传为全部操作
stat_period_start_date	是	long		操作开始时间： 格式yyyyMMdd
stat_period_end_date	是	long		操作结束时间： 格式yyyyMMdd

返回示例:

正确时返回:

```
{
    "total": 0,
    "items": [
        {
            "user_id": 0,
            "real_name": "string",
            "nick_name": "string",
            "org_user_code": "string",
            "gender": 0,
            "icon": "string",
            "org_id": 0,
            "org_name": "string",
            "node_id": 0,
            "node_name": "string",
            "status": "string",
            "identities": [
                {
                    "code": "string",
                    "name": "string"
                }
            ],
            "stations": [
                {
                    "station_id": "string",
                    "station_name": "string"
                }
            ],
            "ic_card": "string",
            "subjects": [
                {
                    "subject_id": "string",
                    "subject_name": "string"
```

```
            }
        ],
        "operate": "string",
        "tenant_id": "string",
        "stat_time_dimension": 0,
        "stat_period_code": "string",
        "stat_period_start_date": 0,
        "stat_period_end_date": 0
    }
  ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
total	long	总数
items	EduUserVo	用户记录

EduUserVo

参数名	类型	说明
user_id	long	用户id（网龙真实的用户id）
real_name	string	用户姓名
nick_name	string	昵称
org_user_code	string	工号（学生是学号，教师是工号）
gender	int	性别：1男 2女
icon	string	头像
org_id	long	组织id（学校id）
org_name	string	组织名称（学校名称）
node_id	long	节点标识（有可能是学校、班级、部门，一般学生是班级、教师是部门）
node_name	string	节点名称 （有可能是学校、班级、部门，一般学生是班级、教师是部门）
status	int	状态：0不可用 1可用

参数名	类型	说明
identities	array[IdentityVo]	身份（学生、教师等）
stations	array[StationVo]	岗位
ic_card	string	IC卡号
subjects	array[SubjectVo]	任教学科
operate	string	操作类型：CREATE、UPDATE、DELETE
stat_period_start_date	int	统计开始日期
stat_period_end_date	int	统计结束日期

IdentityVo

参数名	类型	说明
code	string	身份编码
name	string	身份名称

StationVo

参数名	类型	说明
station_id	string	岗位编码
station_name	string	岗位名称

SubjectVo

参数名	类型	说明
subject_id	string	学科标识
subject_name	string	学科名称

学校管理

学校管理

-查询学校列表
-查询用户有权限的学校列表
-查询单个学校
-根据学校编码查询学校
-查询学校类型列表
-根据帐号查询学校下user_id

查询学校列表

简要描述：

- 查询学校列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-11-26	init

请求URL:

- http://{api-gateway-host}/base/schools/actions/search_all

请求方式：

- POST

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
$offset	是	int	0	偏移量

参数名	是否必须	类型	默认值	说明
$limit	是	int	20	最大笔数
body	是	SchoolVOForSearch		搜索条件

SchoolVoForSearch

参数名	是否必须	类型	默认值	说明
name	否	string		学校名称

返回示例:

正确时返回:

```
{
    "total": 0,
    "items": [
        {
            "id": 0,
            "name": "string",
            "edu_type_value": "string",
            "edu_type_name": "string"
            "address": "string",
            "org_code": "string"
        }
    ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	学校ID
name	string	学校名称
edu_type_value	string	学校教育类型
edu_type_name	string	学校教育类型名称
address	string	地址
org_code	string	学校组织名称

查询用户有权限的学校列表

简要描述：

- 查询用户有权限的学校列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/actions/search

请求方式：

- POST

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
idp-open-id	是	string	用户openid
idp-access-token	是	string	用户access_token
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
$offset	是	int	0	偏移量
$limit	是	int	20	最大笔数
body	是	SchoolVOForSearch		搜索条件

SchoolVoForSearch

参数名	是否必须	类型	默认值	说明
name	否	string		学校名称

返回示例:

正确时返回:

```
{
    "total": 0,
    "items": [
        {
            "id": 0,
            "name": "string",
            "edu_type_value": "string",
            "edu_type_name": "string"
            "address": "string",
            "org_code": "string"
        }
    ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	学校ID
name	string	学校名称
edu_type_value	string	学校教育类型
edu_type_name	string	学校教育类型名称
address	string	地址

参数名	类型	说明
org_code	string	学校组织名称

查询单个学校

简要描述：

- 根据学校id查询学校信息

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

返回示例:

正确时返回:

```
{
    "id": 0,
    "name": "string",
    "org_code": "string",
    "icon": "string",
    "school_number": "string"
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	学校ID
name	string	学校名称
icon	string	学校图片
school_number	string	学校编码
org_code	string	学校组织名称

根据学校编码查询学校

简要描述：

- 根据学校编码查询学校

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/actions/query_by_school_number

请求方式：

- POST

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
idp-open-id	是	string	用户openid
idp-access-token	是	string	用户access_token
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
$offset	是	int	0	偏移量
$limit	是	int	20	最大笔数
school_number	是	string		学校编码

返回示例:

正确时返回:

```
{
    "total": 0,
    "items": [
        {
            "id": 0,
            "name": "string",
            "org_code": "string",
            "icon": "string",
            "school_number": "string"
        }
    ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	学校ID
name	string	学校名称
icon	string	学校图片
school_number	string	学校编码
org_code	string	学校组织名称

查询学校类型列表

简要描述：

- 查询学校类型列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-11-06	init

请求URL:

- http://{api-gateway-host}/base/school_edu_types

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

返回示例:

正确时返回:

```
{
    "items": [
```

```
        {
            "name": "string",
            "value": "string"
        }
    ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
name	string	学校类型名称
value	string	学校类型

根据帐号查询学校下user_id

简要描述：

- 根据帐号查询学校下user_id

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-12-23	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/users/actions/find_by_account

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
account	是	string		帐号（教师工号或学生学号）

返回示例:

正确时返回:

```
[
  {
    "user_id":0
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
user_id	long	user_id

年级管理

年级管理

-查询年级列表
-查询单个年级信息

查询年级列表

简要描述：

- 查询学校年级列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/grades

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
section	否	string		学段编码

返回示例:

正确时返回:

```
[
    {
        "id": 0,
        "name": "string",
        "section": "string",
        "begin_year": 0
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	年级ID
name	string	年级名称
icon	string	学校图片
section	string	学段
begin_year	long	入学年份

查询单个年级信息

简要描述：

- 根据学校id及年级id查询单个年级信息

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/grades/{grade_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
grade_id	是	long		年级id

返回示例:

正确时返回:

```
{
    "id": 0,
    "name": "string",
    "section": "string",
    "begin_year": 0
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	年级ID
name	string	年级名称
icon	string	学校图片
section	string	学段
begin_year	long	入学年份

班级管理

班级管理

-查询年级下班级列表
-查询学校下单个班级

查询年级下的班级列表

简要描述：

- 查询年级下班级列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/grades/{grade_id}/classes

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
grade_id	是	long		年级id
with_detail	否	bool		是否查询班级信息
with_grade	否	bool		是否查询年级信息
with_school	否	bool		是否查询学校信息
class_type	否	string		班级类型 class-行政班，teaching_class-教学班，network_class-网络班

返回示例:

正确时返回:

```
[
  {
    "id": 0,
    "name": "string",
    "class_type": "string",
    "cover_url": "string",
    "grade_id": 0,
    "grade_name": "string",
    "school_id": 0,
    "school_name": "string"
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	班级ID
name	string	班级名称 仅在with_detail为true时返回
cover_url	string	班级封面 仅在with_detail为true时返回
class_type	string	班级类型 class-行政班，teaching_class-教学班，network_class-网络班
grade_id	long	年级ID 仅在with_detail为true时返回
grade_name	string	年级名称 仅在with_grade为true时返回

参数名	类型	说明
school_id	string	学校ID 仅在with_detail为true时返回
school_name	string	学校名称 仅在with_school为true时返回

查询学校下单个班级

简要描述：

- 查询学校下班级信息

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/classes/{class_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
class_id	是	long		年级id
with_detail	否	bool		是否查询班级信息
with_grade	否	bool		是否查询年级信息
with_school	否	bool		是否查询学校信息

返回示例:

正确时返回:

```
{
    "id": 0,
    "name": "string",
    "class_type": "string",
    "class_type_name": "string",
    "cover_url": "string",
    "grade_id": 0,
    "grade_name": "string",
    "school_id": 0,
    "school_name": "string"
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	班级ID
name	string	班级名称 仅在with_detail为true时返回
class_type	string	班级类型 目前支持类型：class-行政班, teaching_class-教学班；以后可能支持更多类型， 请自行过滤不需要的班级类型。
class_type_name	string	班级类型名称
cover_url	string	班级封面 仅在with_detail为true时返回
grade_id	long	年级ID 仅在with_detail为true时返回
grade_name	string	年级名称 仅在with_grade为true时返回
school_id	string	学校ID 仅在with_detail为true时返回

参数名	类型	说明
school_name	string	学校名称 仅在with_school为true时返回

部门管理

部门管理

-查询学校部门列表

查询学校部门列表

简要描述：

- 查询学校下部门列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/departments

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

返回示例:

正确时返回:

```
[
  {
    "id": 0,
    "name": "string"
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	部门ID
name	string	部门名称

教师管理

教师管理

-获取单个教师信息
-获取教师所在班级
-获取班级的班主任列表
-获取班主任管理的班级列表
-获取班级的任课老师列表
-查询学校教师
-获取班级的教师列表

获取单个教师信息

简要描述：

- 获取单个教师信息

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-12-23	init

请求URL:

- http://{api-gateway-host}/base/teachers/{teacher_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
teacher_id	是	long		教师id

返回示例:

正确时返回:

```
{
    "id": 0,
    "org_user_code": "string",
    "real_name": "string",
    "nick_name": "string"
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	教师user_id
org_user_code	string	教师工号
real_name	string	教师真实姓名
nick_name	string	昵称

获取教师所在班级

简要描述：

- 获取教师所在班级

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-12-23	init

请求URL:

- http://{api-gateway-host}/base/teachers/{teacher_id}/classes

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
teacher_id	是	long		教师id

参数名	是否必须	类型	默认值	说明
with_detail	否	bool	false	是否查询班级信息
with_grade_detail	否	bool	false	是否查询年级信息
class_type	否	string		班级类型 class-行政班，teaching_class-教学班，network_class-网络班

返回示例:

正确时返回:

```
[
  {
    "class_id": 0,
    "class_name": "string",
    "class_type": "string",
    "grade_id": 0,
    "grade_name": "string",
    "join_time": "2015-01-01T00:00:00"
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
class_id	long	班级ID
class_name	string	班级名称
class_type	string	班级类型 class-行政班，teaching_class-教学班，network_class-网络班
grade_id	long	年级ID
grade_name	string	年级名称
join_time	date	班级加入时间

获取班级的班主任列表

简要描述：

- 查询班级班主任列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/classes/{class_id}/head_teachers

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
class_id	是	long		班级id

返回示例:

正确时返回:

```
[
  {
    "teacher_id":0
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
teacher_id	long	班主任user_id

获取班级的任课教师列表

简要描述：

- 获取班级的任课教师列表(带任课科目，无教师详情)

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/classes/{class_id}/subject_teachers

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
class_id	是	long		班级id

参数名	是否必须	类型	默认值	说明
term_id	否	string		学期标识(如果有设置当前学期， 不传的话默认取当前学期)

返回示例:

正确时返回:

```
[
  {
    "class_id": 0,
    "teacher_id": 0,
    "class_type": "string",
    "subjects": [
        {
            "id": "string",
            "name": "string"
        }
    ]
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
class_id	long	班级ID
teacher_id	long	教师id
class_type	string	班级类型 目前支持类型：class-行政班, teaching_class-教学班；以后可能支持更多类型， 请自行过滤不需要的班级类型。
subjects	array[SubjectVO]	学科列表

SubjectVo

参数名	类型	说明
id	long	学科id
name	long	学科名称

获取班主任管理的班级列表

简要描述：

- 获取班主任管理的班级列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/head_teachers/{teacher_id}/classes

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
teacher_id	是	long		班级id

返回示例:

正确时返回:

```
[
    {
        "class_id": 495360212322
    },
    {
        "class_id": 495360194954
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
class_id	long	班级ID

查询学校教师

简要描述：

- 查询学校教师

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/teachers

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
$limit	否	int	20	记录数
$offset	否	int	0	偏移量

返回示例:

正确时返回:

```
[
  {
    "id": 0,
    "org_user_code": "string",
    "real_name": "string",
    "nick_name": "string"
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	教师ID
org_user_code	string	教师工号
real_name	string	教师真实姓名
nick_name	string	昵称 无昵称的情况下会自动将真实姓名写入昵称

获取班级的教师列表

简要描述：

- 获取班级的教师列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/classes/{class_id}/teachers

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
class_id	是	long		班级ID
with_detail	否	bool	false	附带详细信息

返回示例:

正确时返回:

```
[
 {
    "id": 0,
    "org_user_code": "string",
    "real_name": "string",
    "nick_name": "string"
 }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	教师ID
org_user_code	string	教师工号
real_name	string	教师真实姓名
nick_name	string	昵称 仅在with_detail=true时返回， 无昵称的情况下会自动将真实姓名写入昵称

获取教师所在部门

简要描述：

- 获取教师所在部门

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-02-28	init

请求URL:

- http://{api-gateway-host}/base/teachers/{teacher_id}/departments

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
teacher_id	是	long		教师id

返回示例:

正确时返回:

```
{
    "items": [
        {
            "id": 0,
            "name": "string"
        }
    ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	部门id
name	string	部门名称

学生管理

学生管理

-获取班级学生列表
-获取单个学生信息
-获取学生所在班级列表
-根据学籍号查询学生user_id

获取班级学生列表

简要描述：

- 获取班级学生列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/classes/{class_id}/students

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
class_id	是	long		班级ID
with_detail	否	bool	false	附带详细信息

返回示例:

正确时返回:

```
[
  {
    "id": 0,
    "stu_num": "string",
    "real_name": "string",
    "nick_name": "string",
    "gender": 0,
    "student_code": "string",
    "seat_number": 0
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	学生ID
stu_num	string	学生学号 仅在with_detail为true时返回
real_name	string	真实姓名 仅在with_detail为true时返回
nick_name	string	昵称 仅在with_detail为true时返回
gender	int	性别 0-未知 1-男 2-女 仅在with_detail为true时返回
student_code	string	学籍号 仅在with_detail为true时返回
seat_number	long	座号 仅在with_detail为true时返回

获取单个学生信息

简要描述：

- 查询学生信息

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/students/{student_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
student_id	是	long		学生id

返回示例:

正确时返回:

```
{
    "id": 0,
    "stu_num": "string",
    "real_name": "string",
    "nick_name": "string",
    "gender": 0,
    "student_code": "string",
    "seat_number": 0
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	学生ID
stu_num	string	学生学号
real_name	string	真实姓名
nick_name	string	昵称
gender	int	性别 0-未知 1-男 2-女
student_code	string	学籍号
seat_number	long	座号

获取学生所在班级列表

简要描述：

- 获取学生所在班级

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/students/{student_id}/classes

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
student_id	是	long		学生id

参数名	是否必须	类型	默认值	说明
with_detail	否	bool		是否查询班级信息
with_grade_detail	否	bool		是否查询年级信息
class_type	否	string		班级类型 class-行政班, teaching_class-教学班, network_class-网络班

返回示例:

正确时返回:

```
[
  {
    "class_id": 0,
    "class_type": "string",
    "class_name": "string",
    "grade_id": 0,
    "grade_name": "string"
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
class_id	long	班级ID
class_name	string	班级名称 仅在with_detail为true时返回
class_type	string	班级类型 class-行政班, teaching_class-教学班, network_class-网络班
grade_id	long	年级ID 仅在with_detail为true时返回
grade_name	string	年级名称 仅在with_grade_detail为true时返回

根据学籍号查询学生user_id

简要描述：

- 根据学籍号查询学生user_id

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-12-23	init

请求URL:

- http://{api-gateway-host}/base/students/actions/query_by_student_code

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
student_code	是	string		学籍号

返回示例:

正确时返回:

```
100000
```

错误时返回:

返回参数说明:

参数名	类型	说明
	long	学生ID

家长管理

家长管理

-获取我的孩子信息
-获取家长列表

获取我的孩子信息

简要描述：

- 根据openid跟accesstoken获取我的孩子列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/guardians/actions/get_my_children

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
idp-open-id	是	string	用户openid
idp-access-token	是	string	用户access_token
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

返回示例:

正确时返回:

```
[
  {
    "user_id": 0,
    "display_name": "string",
    "school_id": 0,
    "school_name": "string",
    "class_id": 0,
    "class_name": "string",
    "is_current": true
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
user_id	long	网龙真实用户ID
display_name	string	用户名
school_id	long	学校id
school_name	string	学校名称
class_id	long	班级ID
class_name	string	班级名称
is_current	bool	是否是当前展示用户

获取家长列表

简要描述：

- 根据学校id及班级id获取家长列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2019-09-02	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/classes/{class_id}/guardians

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
idp-open-id	是	string	用户openid
idp-access-token	是	string	用户access_token
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id
class_id	是	long		年级id
with_detail	否	bool	false	附带详细信息

返回示例:

正确时返回:

```
[
  {
    "id": 0,
    "account": "string",
    "real_name": "string",
    "nick_name": "string",
    "relations": [
        {
            "id": "string",
            "display_name": "string",
            "relation": "string",
            "relation_name": "string"
        }
    ]
  }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	long	家长ID
account	string	帐号
real_name	string	真实姓名
nick_name	string	昵称 无昵称的情况下会自动将真实姓名写入昵称
relations	array[GuardianChildrenVO]	关系列表

GuardianChildrenVO

参数名	类型	说明

参数名	类型	说明
id	long	孩子ID
display_name	string	孩子名字
relation	string	关系 mother father other grandfather grandmother grandpa_m grandma_m
relation_name	string	关系名称 母亲 父亲 其他 爷爷 奶奶 姥爷 姥姥

设备管理

设备管理

-设置用户设备
-根据设备编码查询用户

设置用户设备

简要描述：

- 设置用户设备(卡号)

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-02-28	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/users/{user_id}/devices/{device_id}

请求方式：

- PUT

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
user_id	是	long		用户id
device_id	是	long		设备类型ID（使用5为卡号）
type	是	string		用户类型（student或teacher）
body	是	DeviceVo		卡号信息

DeviceVo

参数名	是否必须	类型	默认值	说明
device_code	是	string		设备编码

返回示例:

正确时返回:

错误时返回:

返回参数说明:

根据设备编码查询用户

简要描述：

- 根据设备编码查询学生

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-02-28	init

请求URL:

- http://{api-gateway-host}/base/schools/{school_id}/users/actions/get_by_device_code

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校id

参数名	是否必须	类型	默认值	说明
device_code	是	string		设备编码（卡号）
type	是	string		用户类型（student或teacher）
device_id	是	long		设备类型ID（默认5）

返回示例:

正确时返回:

```
{
    "user_id": [
        0
    ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
user_id	long	用户id

课程表

课程表

-获取班级的课程表
-获取全校各年级课程表
-获取学校课程表
-获取场地的课程表
-获取学生的课程表
-获取老师的课程表

获取班级的课程表

简要描述：

- 获取班级的课程表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-04-29	init

请求URL:

- http://{api-gateway-host}/ct/timetables/classes/{class_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
class_id	是	long		班级ID(在path上)

参数名	是否必须	类型	默认值	说明
term	是	string		所在学期
week	是	int		第几周
day	否	int		第几天

返回示例:

正确时返回:

```
[
    {
        "id": "string",
        "type": 0,
        "week_info": 0,
        "days": 0,
        "lesson": 0,
        "begin_time": "2015-01-01T00:00:00",
        "end_time": "2015-01-01T00:00:00",
        "class_id": 0,
        "class_name": "string",
        "class_type": 0,
        "grade_id": 0,
        "grade_name": "string",
        "course_id": "string",
        "course_name": "string",
        "subject_id": "string",
        "subject_name": "string",
        "teacher_id": 0,
        "teacher_name": "string",
        "site_id": "string",
        "site_name": "string",
        "building_id": "string",
        "building_name": "string",
        "update_time": "2015-01-01T00:00:00"
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	课表条目Id
type	int	上课类型：1-早读，2-上课，3-晚自习
week_info	long	周信息(二进制表示)

参数名	类型	说明
days	int	周几
lesson	int	第几节
begin_time	string	开始时间：时间格式为HH:mm:ss
end_time	string	结束时间：时间格式为HH:mm:ss
class_id	long	班级ID
class_name	string	班级名称
class_type	int	班级类型：1-行政班，2-网络班，3-教学班
grade_id	long	年级ID
grade_name	string	年级名称
course_id	string	课程ID
course_name	string	课程名称
subject_id	string	学科Id
subject_name	string	学科名称
teacher_id	long	教师ID
teacher_name	string	教师名
site_id	string	场地Id
site_name	string	场地名称
building_id	string	建筑ID
building_name	string	建筑名称
update_time	string	修改时间

获取全校各年级课程表

简要描述：

- 获取全校各年级课程表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-04-29	init

请求URL:

- http://{api-gateway-host}/ct/timetables/grades

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校Id

参数名	是否必须	类型	默认值	说明
term	是	string		所在学期
week	是	int		第几周
grade	否	long		年级
type	否	long		返回类型：1-早读，2-上课，3-晚自习，[空]-全部

返回示例:

正确时返回:

```
[
    {
    "grade": "string",
    "grade_name": "string",
    "class_id": "string",
    "class_name": "string",
    "timetables": [
        {
            "id": "string",
            "type": 0,
            "days": 0,
            "lesson": 0,
            "course_id": "string",
            "course_name": "string",
            "teacher_id": 0,
            "teacher_name": "string",
            "site_id": "string",
            "site_name": "string",
            "building_id": "string",
            "building_name": "string"
        }
      ]
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
grade	string	年级
grade_name	string	年级名称
class_id	string	班级ID

参数名	类型	说明
class_name	string	班级名称
timetables	array	课程表单元格
timetables.id	string	
timetables.type	int	上课类型：1-早读，2-上课，3-晚自习
timetables.days	int	周几
timetables.lesson	int	第几节
timetables.course_id	string	课程ID
timetables.course_name	string	课程名称
timetables.teacher_id	long	教师ID
timetables.teacher_name	string	教师名
timetables.site_id	string	场地Id
timetables.site_name	string	场地名称
timetables.building_id	string	建筑ID
timetables.building_name	string	建筑名称

获取学校课程表

简要描述：

- 获取学校课程表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-04-29	init

请求URL:

- http://{api-gateway-host}/ct/timetables/schools/{school_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校Id(在path上)

参数名	是否必须	类型	默认值	说明
term	否	string		所在学期
week	否	int		第几周
day	否	int		第几天

返回示例:

正确时返回:

```
[
    {
        "id": "string",
        "type": 0,
        "week_info": 0,
        "days": 0,
        "lesson": 0,
        "begin_time": "2015-01-01T00:00:00",
        "end_time": "2015-01-01T00:00:00",
        "class_id": 0,
        "class_name": "string",
        "class_type": 0,
        "grade_id": 0,
        "grade_name": "string",
        "course_id": "string",
        "course_name": "string",
        "subject_id": "string",
        "subject_name": "string",
        "teacher_id": 0,
        "teacher_name": "string",
        "site_id": "string",
        "site_name": "string",
        "building_id": "string",
        "building_name": "string",
        "update_time": "2015-01-01T00:00:00"
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	课表条目Id
type	int	上课类型：1-早读，2-上课，3-晚自习
week_info	long	周信息(二进制表示)

参数名	类型	说明
days	int	周几
lesson	int	第几节
begin_time	string	开始时间：时间格式为HH:mm:ss
end_time	string	结束时间：时间格式为HH:mm:ss
class_id	long	班级ID
class_name	string	班级名称
class_type	int	班级类型：1-行政班，2-网络班，3-教学班
grade_id	long	年级ID
grade_name	string	年级名称
course_id	string	课程ID
course_name	string	课程名称
subject_id	string	学科Id
subject_name	string	学科名称
teacher_id	long	教师ID
teacher_name	string	教师名
site_id	string	场地Id
site_name	string	场地名称
building_id	string	建筑ID
building_name	string	建筑名称
update_time	string	修改时间

获取场地的课程表

简要描述：

- 获取场地的课程表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-04-29	init

请求URL:

- http://{api-gateway-host}/ct/timetables/sites/{site_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
site_id	是	string		场地Id(在path上)

参数名	是否必须	类型	默认值	说明
term	是	string		所在学期
week	是	int		第几周
day	否	int		第几天

返回示例:

正确时返回:

```
[
    {
        "id": "string",
        "type": 0,
        "week_info": 0,
        "days": 0,
        "lesson": 0,
        "begin_time": "2015-01-01T00:00:00",
        "end_time": "2015-01-01T00:00:00",
        "class_id": 0,
        "class_name": "string",
        "class_type": 0,
        "grade_id": 0,
        "grade_name": "string",
        "course_id": "string",
        "course_name": "string",
        "subject_id": "string",
        "subject_name": "string",
        "teacher_id": 0,
        "teacher_name": "string",
        "site_id": "string",
        "site_name": "string",
        "building_id": "string",
        "building_name": "string",
        "update_time": "2015-01-01T00:00:00"
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	课表条目Id
type	int	上课类型：1-早读，2-上课，3-晚自习
week_info	long	周信息(二进制表示)

参数名	类型	说明
days	int	周几
lesson	int	第几节
begin_time	string	开始时间：时间格式为HH:mm:ss
end_time	string	结束时间：时间格式为HH:mm:ss
class_id	long	班级ID
class_name	string	班级名称
class_type	int	班级类型：1-行政班，2-网络班，3-教学班
grade_id	long	年级ID
grade_name	string	年级名称
course_id	string	课程ID
course_name	string	课程名称
subject_id	string	学科Id
subject_name	string	学科名称
teacher_id	long	教师ID
teacher_name	string	教师名
site_id	string	场地Id
site_name	string	场地名称
building_id	string	建筑ID
building_name	string	建筑名称
update_time	string	修改时间

获取学生的课程表

简要描述：

- 获取学生的课程表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-04-29	init

请求URL:

- http://{api-gateway-host}/ct/timetables/students/{student_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
student_id	是	string		学生ID(在path上)

参数名	是否必须	类型	默认值	说明
term	是	string		所在学期
week	是	int		第几周
day	否	int		第几天

返回示例:

正确时返回:

```
[
    {
        "id": "string",
        "type": 0,
        "week_info": 0,
        "days": 0,
        "lesson": 0,
        "begin_time": "2015-01-01T00:00:00",
        "end_time": "2015-01-01T00:00:00",
        "class_id": 0,
        "class_name": "string",
        "class_type": 0,
        "grade_id": 0,
        "grade_name": "string",
        "course_id": "string",
        "course_name": "string",
        "subject_id": "string",
        "subject_name": "string",
        "teacher_id": 0,
        "teacher_name": "string",
        "site_id": "string",
        "site_name": "string",
        "building_id": "string",
        "building_name": "string",
        "update_time": "2015-01-01T00:00:00"
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	课表条目Id
type	int	上课类型：1-早读，2-上课，3-晚自习
week_info	long	周信息(二进制表示)

参数名	类型	说明
days	int	周几
lesson	int	第几节
begin_time	string	开始时间：时间格式为HH:mm:ss
end_time	string	结束时间：时间格式为HH:mm:ss
class_id	long	班级ID
class_name	string	班级名称
class_type	int	班级类型：1-行政班，2-网络班，3-教学班
grade_id	long	年级ID
grade_name	string	年级名称
course_id	string	课程ID
course_name	string	课程名称
subject_id	string	学科Id
subject_name	string	学科名称
teacher_id	long	教师ID
teacher_name	string	教师名
site_id	string	场地Id
site_name	string	场地名称
building_id	string	建筑ID
building_name	string	建筑名称
update_time	string	修改时间

获取老师的课程表

简要描述：

- 获取老师的课程表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-04-29	init

请求URL:

- http://{api-gateway-host}/ct/timetables/teachers/{teacher_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
teacher_id	是	string		教师ID(在path上)

参数名	是否必须	类型	默认值	说明
term	是	string		所在学期
week	是	int		第几周
day	否	int		第几天

返回示例:

正确时返回:

```
[
    {
        "id": "string",
        "type": 0,
        "week_info": 0,
        "days": 0,
        "lesson": 0,
        "begin_time": "2015-01-01T00:00:00",
        "end_time": "2015-01-01T00:00:00",
        "class_id": 0,
        "class_name": "string",
        "class_type": 0,
        "grade_id": 0,
        "grade_name": "string",
        "course_id": "string",
        "course_name": "string",
        "subject_id": "string",
        "subject_name": "string",
        "teacher_id": 0,
        "teacher_name": "string",
        "site_id": "string",
        "site_name": "string",
        "building_id": "string",
        "building_name": "string",
        "update_time": "2015-01-01T00:00:00"
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	课表条目Id
type	int	上课类型：1-早读，2-上课，3-晚自习
week_info	long	周信息(二进制表示)

参数名	类型	说明
days	int	周几
lesson	int	第几节
begin_time	string	开始时间：时间格式为HH:mm:ss
end_time	string	结束时间：时间格式为HH:mm:ss
class_id	long	班级ID
class_name	string	班级名称
class_type	int	班级类型：1-行政班，2-网络班，3-教学班
grade_id	long	年级ID
grade_name	string	年级名称
course_id	string	课程ID
course_name	string	课程名称
subject_id	string	学科Id
subject_name	string	学科名称
teacher_id	long	教师ID
teacher_name	string	教师名
site_id	string	场地Id
site_name	string	场地名称
building_id	string	建筑ID
building_name	string	建筑名称
update_time	string	修改时间

学期信息

学期信息

-获取学校所有学期
-获取单个学期

获取学校所有学期

简要描述：

- 获取学校所有学期

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-09-15	init

请求URL:

- http://{api-gateway-host}/edu/schools/{school_id}/terms/actions/get_all

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校ID(在path上)

返回示例:

正确时返回:

```
[
    {
        "id": "string",
        "name": "string",
        "academic_year": 0,
        "academic_year_name": "string",
        "start_time": "2015-01-01T00:00:00",
        "end_time": "2015-01-01T00:00:00",
        "is_act": 0,
        "is_apply": 0,
        "school_id": 0,
        "num": 0
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	标识
name	string	名称
academic_year	number	学年
academic_year_name	string	学年名称
start_time	string	开始时间
end_time	string	结束时间
is_act	number	是否当前学期[1是0否]
is_apply	number	是否当前招生年级[1是0否]
school_id	number	学校id
num	number	序号(指某一学年第N学期)

获取单个学期

简要描述：

- 获取单个学期

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-09-15	init

请求URL:

- http://{api-gateway-host}/edu/schools/{school_id}/terms/{term_id}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
school_id	是	long		学校ID(在path上)

参数名	是否必须	类型	默认值	说明
term_id	是	string		学期ID(在path上)

返回示例:

正确时返回:

```
    {
        "id": "string",
        "name": "string",
        "academic_year": 0,
        "academic_year_name": "string",
        "start_time": "2015-01-01T00:00:00",
        "end_time": "2015-01-01T00:00:00",
        "is_act": 0,
        "is_apply": 0,
        "school_id": 0,
        "num": 0
    }
```

错误时返回:

返回参数说明:

参数名	类型	说明
id	string	标识
name	string	名称
academic_year	number	学年
academic_year_name	string	学年名称
start_time	string	开始时间
end_time	string	结束时间
is_act	number	是否当前学期[1是0否]
is_apply	number	是否当前招生年级[1是0否]
school_id	number	学校id
num	number	序号(指某一学年第N学期)

行政区划

-根据IP定位当前的地区的行政区划
-行政区划-获取子行政区划列表
-行政区划-父级行政区划列表

根据IP定位当前的地区的行政区划

简要描述：

- 根据IP定位当前的地区的行政区划

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2021-04-09	init

请求URL:

- http://{api-gateway-host}/k12/areas/actions/get_current_areacode

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

返回示例:

正确时返回:

```
[
    {
```

```
        "area_code": "string"
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
area_code	string	行政区划编码

获取子行政区划列表

简要描述：

- 获取子行政区划列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2021-04-09	init

请求URL:

- http://{api-gateway-host}/k12/areas/{area_code}

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
area_code	是	string		行政区划编码(在path上)

返回示例:

正确时返回:

```
[
    {
        "area_code": "string",
        "area_name": "string",
        "level": 0,
        "id": "string"
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
area_code	string	行政区划编码
area_name	string	行政区划名称
level	int	行政区划层级
id	string	区域id（组织id）

父级行政区划列表

简要描述：

- 父级行政区划列表

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2021-04-09	init

请求URL:

- http://{api-gateway-host}/k12/areas/{area_code}/parents

请求方式：

- GET

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

参数名	是否必须	类型	默认值	说明
area_code	是	string		行政区划编码(在path上)

返回示例:

正确时返回:

```
[
    {
        "area_code": "string",
        "area_name": "string",
        "level": 0,
        "id": "string"
    }
]
```

错误时返回:

返回参数说明:

参数名	类型	说明
area_code	string	行政区划编码
area_name	string	行政区划名称
level	int	行政区划层级
id	string	区域id（组织id）

教务需求

重要说明

本部分主要用于网龙网教通应用对接第三方厂家，对方厂家需要提供的必要数据需求

组织结构

简要描述：

学校组织结构部分数据说明

学段 枚举数据

枚举内容
幼儿园
小学
初中
高中
中等职业学校
高等学校（大学）
培训机构
其他

学校

描述	类型
学校id	long
编码	string
名称	string
学段 使用以上枚举数据	list<string>

学校管理员

描述	类型
学校id	long
管理员id	long

描述	类型
昵称	string
姓名	string
帐号	string

> 一个学校拥有多个管理员

部门

描述	类型
学校id	long
部门父节点id	long
部门id	long
名称	string

> 一个学校有多个部门

教师

描述	类型
教师id	long
工号	string
昵称	string
性别	gender 枚举 男，女，未知
所在部门id	list<long>
真实姓名	string

> 一个部门有0个或多个教师

年级

描述	类型
学校id	long
年级id	long
入学年份	int
毕业年份	int
名称	string
年制	int
学段	枚举

> 一个学校有多个年级

班级

描述	类型
年级id	long
班级id	long
班级名称	string
班主任ID列表 最多三个 默认显示第一个	list<long>

> 年级下有多个班级

学生

描述	类型
班级id	long
学生id	long
昵称	string
性别	gender 枚举 男，女，未知
姓名	string

描述	类型
学号	string
学籍号	string
座号	int

> 班级里有多个学生

家长

描述	类型
家长id	long
性别	gender 枚举 男，女，未知
昵称	string
姓名	string

家长学生关系

描述	类型
家长id	long
学生id	long
监控关系	枚举 母亲 父亲 其他 爷爷 奶奶 姥爷 姥姥

任课管理

简要描述：

任课部分数据结构说明

学科

描述	类型
编码	string
名称	string

教材

描述	类型
id	string
名称	string

课程

描述	类型
学校id	long
课程id	long
学科id	long
教材id	long
名称	string

学期

描述	类型
学校id	long
学期id	long
季节	int

描述	类型
开始时间	date
结束时间	date
名称	string
年份	int
是否当前学期	bool
是否招生学期	bool

教师课程关系表

描述	类型
教师id	long
课程id	long

教师班级课程关系 先有教师课程关系才能加入到班级的学期中

描述	类型
教师id	long
课程id	long
班级id	long
学期id	long

应用、数据融合

本部分主要介绍业主使用网教通作为教育信息化平台如何与第三方的应用集成以及应用数据如何融合的方案

- 第三方应用如何与网教通平台集成（统一账号）
- 第三方应用的数据如何“融合”汇集到网教通平台

应用集成

问题背景

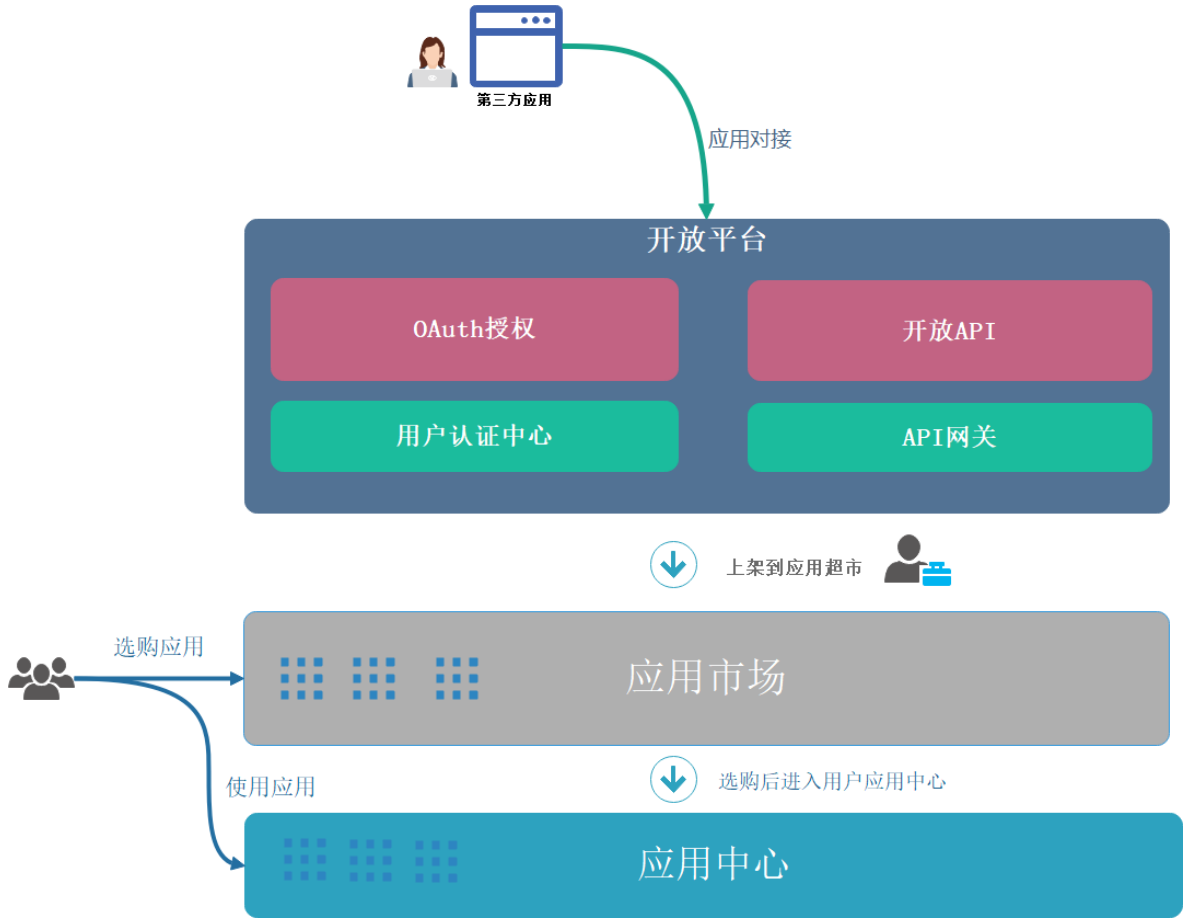
目前已知各级教育主管部门以及各校级信息化系统中，已自建、采购了各领域的应用子系统，但缺少相应的前期规划，“烟筒式”应用系统分布各处。各自系统独立运行提供功能，账号间不互通，无法做到统一的账号分配、权限授权等，对应用的监管、统计、分析困难。

在此背景下，需要新一代的教育信息化平台能统一对已存在应用和待建应用进行统一上架管理；并需要系统统一的账号授信中心，并开放行业标准授权协议要求各三方应用功能使用标准协议进行账号对接、统一授权。

解决方案

开放平台针对存在问题，规划、研发了适合教育领域的开放平台，旨在解决应用集成、账号统一等技术问题。

- 提供应用接入的整体解决方案；从应用创建、审核、接口认证、上架、下架、使用统计等提供完整声明周期管理。
- 提供标准的OAuth2.0授权协议，解决教育信息化平台与第三方应用功能的统一账号管理功能。
- 提供必要的教育基础数据，供第三方接入应用调用，以达成全平台应用共享和统一一份标准一致的基础数据、数据字典。
- 提供统一的服务托管能力，可为信息化平台、第三方应用开发厂家提供统一认证协议的接口调用平台。
- 提供应用的统一售卖
- 提供应用入口的统一管理，方便用户对第三方应用的查找及使用



详细资料

- 平台的介绍和创建应用说明可参见 介绍和注册
- 应用的接入说明可参见 应用对接
- 教育基础数据、数据字典部分的数据可参见 教育基础数据接口

数据融合

背景

已建设的教育信息化平台和相应的应用系统产生数据对业主是非常重要的资产，各级教育主管部门、各校级领导对该资产均有做集中存取、集中管理、分析的诉求。

但目前已知的各教育信息化平台中，多为停留在应用的集成、账号集成层面；数据方面的融合没有提供必要的解决方案。

解决方案

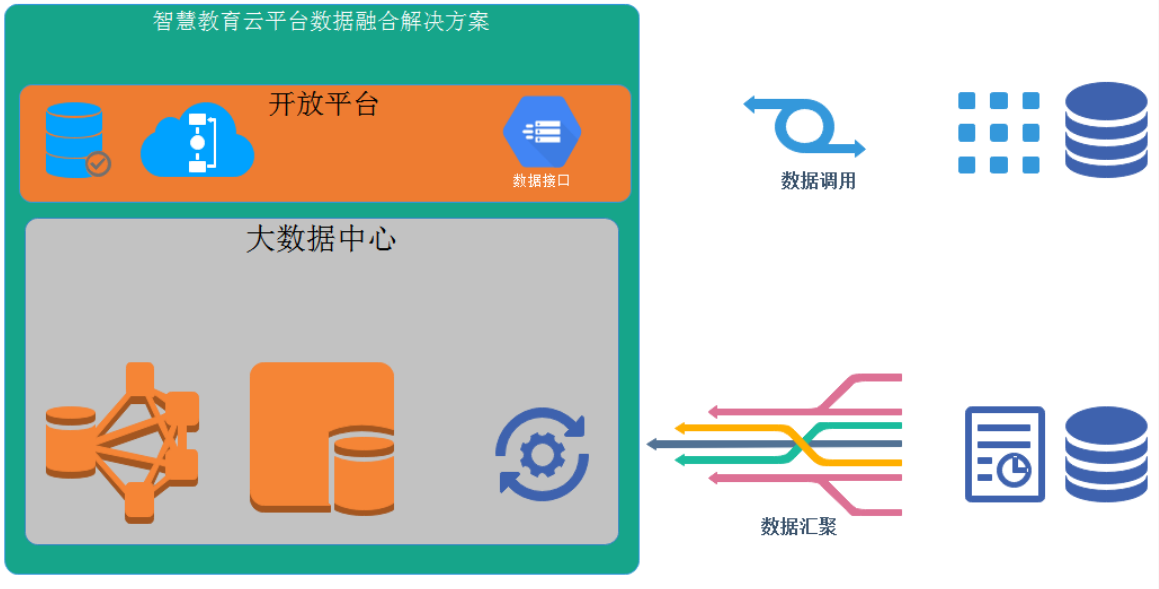
我们将数据融合方面的诉求从技术和业务上拆分为两类

- 毫秒级数据交互：通常解决第三方应用与信息化平台之间的实时数据交互。特点是单次通讯数据量小（KB级）、高并发、低延时需求。
- 分钟级、小时级数据交互（大数据）：通常解决各应用运行时产生的教学过程数据、日志数据、资源文件等数据的集中汇聚、存储、分析需求。特点是单次通讯数据量大（GB、TB级）、低频次发生、单次执行时间长（分钟级，甚至小时级）

毫秒级数据交互

- 通过开放平台的能力，提供在线API。（基础数据字典、学校基础数据、学科信息、年级班级信息、用户身份信息）
- 统一API认证鉴权能力，

分钟级、小时级数据交互（大数据）



知识库

- 提供基本的常见互联网知识

理解SSL证书

SSL或称Secure Sockets Layer（安全套接层）是一种加密技术，它在web服务器和允许传输私有数据的浏览器之间建立加密链接。为此，必须在web服务器上安装SSL证书，通常可以将其视为浏览器地址栏上的挂锁图标。

HTTP和HTTPS

HTTP全称Hypertext Transfer Protocol（超文本传输协议）通过以纯文本形式将数据从特定服务器发送到浏览器，同时允许用户浏览web页面，从而促进internet上的通信。HTTPS全称 Hypertext Transfer Protocol Secure（超文本传输安全协议）是一种加密连接，使用密文在服务器之间安全地传输数据。这两个协议之间的区别在于SSL证书的存在。

首先，让我们了解什么是SSL，什么是SSL证书？

SSL证书也称为TLS证书，是一种数字证书，它可以通过加密数据来帮助你避免数据泄露的情况，因此，当用户在其web浏览器的页面上输入信用卡详细信息、联系电话、地址和社会安全号码等私人信息时，这些信息就会以HTTPS加密的方式安全地传输到服务器上，使得中间没有人可以入侵并窃取数据。从而为用户提供一个安全的环境来访问你的网站，且不用担心被数据被窃取。

CA机构是什么？

CA机构是什么？CA机构就是SSL证书审核签发机构，电子商务交易中备受信任的第三方，承担检验公钥体系合法性的责任。

在选择CA机构之前一定要考察核对该机构是否有可信资质，SSL证书选择也要选择权威机构的证书，CA机构在信息安全行业担任着一个十分重要的角色，因此国内数字证书机构并不多。与此同时，通过WEBTRUST国际认证更加少之又少，我国CA机构仅有3家通过。

> 网龙公司CA颁发机构：GlobalSign

选择HTTPS的原因

为了避免遭受网络攻击，使用HTTPS协议是相对正确的选择，因为它可以安全地加密你的所有数据。谷歌Chrome也启动了一项标准的安全检查，如果一个站点没有SSL证书，它将立即被标记为“不安全”。

HTTPS等于稳定的引线。访问者显然会信任由谷歌或其他浏览器批准的网站。然而，他们在访问一个被浏览器标记为“不安全”的网页前，会立即产生不信任，或者三思而后行。

SSL的缺点

虽然SSL证书有多个优点，但是仍有大部分的网站并没有部署SSL证书。其中SSL证书的价格成本就使很多的站长们望而却步。因为申请SSL证书，CA机构需要建立一个值得信赖的基础设施并验证申请者的身份，需要花费较大的金额，从而增加了SSL证书的成本。但近年来，为了推动SSL证书的使用，部分的CA机构推出免费SSL证书。因而引发了另一个缺点，因为免费，安全等级相对较低，并且导致某些CA机构并没有严格审核申请人的身份，导致部分钓鱼网站也可申请免费的SSL证书。为了避免类似事件再次发生，浏览器厂商纷纷对相关CA机构作出不受信任的决策。

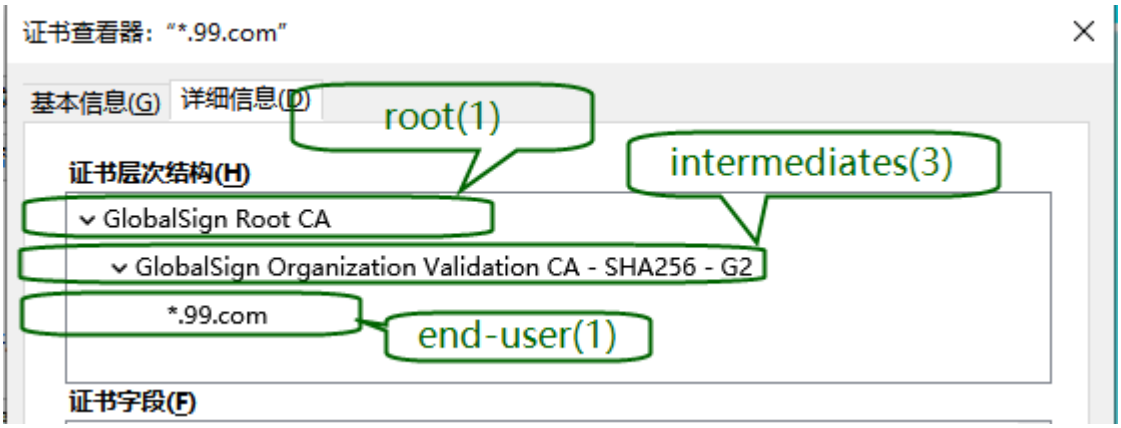
网络安全无小事，任何一个细小的疏忽都有可能带来岌岌可危的后果。当网络数据正越来越成为具有重要商业价值的资产，无论个人还是企业都应重视。我们不仅希望像数安时代GDCA这样的服务商能够从技术和服务上将网络风险降到最低，同时更要企业增强自我网络安全意识，选择有服务品质保障的电子认证服务商进行合作，只有多方共同携手，才能撑起隐私和数据安全的“保护伞”。

What is the SSL Certificate Chain?

There are two types of certificate authorities (CAs): root CAs and intermediate CAs. In order for an SSL certificate to be trusted, that certificate must have been issued by a CA that is included in the trusted store of the device that is connecting.

If the certificate was not issued by a trusted CA, the connecting device (eg. a web browser) will then check to see if the certificate of the issuing CA was issued by a trusted CA, and so on until either a trusted CA is found (at which point a trusted, secure connection will be established) or no trusted CA can be found (at which point the device will usually display an error).

The list of SSL certificates, from the root certificate to the end-user certificate, represents the SSL certificate chain.



Example of an SSL Certificate chain

Here's a practical example. Let's suppose that you purchase a certificate from the Awesome Authority for the domain example.awesome.

Awesome Authority is not a root certificate authority. In other words, its certificate is not directly embedded in your web browser and therefore it can't be explicitly trusted.

- Awesome Authority utilizes a certificate issued by Intermediate Awesome CA Alpha.
- Intermediate Awesome CA Alpha utilizes a certificate issued by Intermediate Awesome CA Beta.
- Intermediate Awesome CA Beta utilizes a certificate issued by Intermediate Awesome CA Gamma.
- Intermediate Awesome CA Gamma utilizes a certificate issued by The King of Awesomeness.
- The King of Awesomeness is a Root CA. Its certificate is directly embedded in your web browser, therefore it can be explicitly trusted.

In our example, the SSL certificate chain is represented by 6 certificates:

1. End-user Certificate - Issued to: example.com; Issued By: Awesome Authority
2. Intermediate Certificate 1 - Issued to: Awesome Authority; Issued By: Intermediate Awesome CA Alpha
3. Intermediate Certificate 2 - Issued to: Intermediate Awesome CA Alpha; Issued By: Intermediate Awesome CA Beta
4. Intermediate Certificate 3 - Issued to: Intermediate Awesome CA Beta; Issued By: Intermediate Awesome CA Gamma
5. Intermediate Certificate 4 - Issued to: Intermediate Awesome CA Gamma; Issued By: The King of Awesomeness
6. Root certificate - Issued by and to: The King of Awesomeness

Certificate 1 is your end-user certificate, the one you purchase from the CA. The certificates from 2 to 5 are called intermediate certificates. Certificate 6, the one at the top of the chain (or at the end, depending on how you read the chain), is called root certificate.

When you install your end-user certificate for example.awesome, you must bundle all the intermediate certificates and install them along with your end-user certificate. If the SSL certificate chain is invalid or broken, your certificate will not be trusted by

some devices.

HTTPS证书分类

HTTPS证书分类

按验证等级

- 域名型SSL证书，DVSSL，Domain Validation SSL Certificate

信任等级普通，只需验证网站的真实性便可颁发证书保护网站。无需人工审核，快速颁发。

- 企业组织型SSL证书，OVSSL，Organization Validation SSL Certificate

信任等级强，须要验证企业的身份，审核严格，安全性更高。需要人工审核，证书包含企业认证信息。

- 增强扩展型SSL证书，EVSSL，Extended Validation SSL Certificate

信任等级最高，一般用于银行证券等金融机构，人工审核，审核严格，颁发之后可以使网站域名在浏览器地址栏变成绿色。

DV	OV	EV
信任等级普通， 只需验证网站的真实性便 可颁发证书保护网站	信任等级强， 须要验证企业的身份， 审核严格，安全性更高	信任等级最高， 一般用于银行证券等金融机 构
无需人工审核，快速颁发	需要人工审核， 证书包含企业认证信息	人工严格审核， 颁发之后可以使网站域名在 浏览器地址栏变成绿色

按保护域名数量

- 单域名型证书, Single Domain SSL Certificates

一张SSL证书保护一个域名，一个网站的安全。

- 多域名型证书 Multi Domain SSL Certificates

一张SSL证书可以保护多个域名，多个网站的安全，多域型SSL证书。支持任何域名，可以是不同的顶级域名（如：hello.com, tv.com, example.com）。

- 通配符型证书, Wildcard SSL Certificates

一张SSL证书可以保护相同主域名下所有的子域名的安全。通配型证书只支持*.domain.com通配符子域名（如：a.hello.com, b.hello.com、c.hello.com）。

其中，单域名型证书和多域名证书分为三种验证等级：DV, OV, EV。 而通配符型证书只有DV和OV两种。

移动端对接说明

目前支持的集成方式

- 三方厂家提供H5 集成
- APP拉起：在网教通app中携带授权信息（ticket）拉起三方厂家app
- *小程序对接【规划中】*

移动端（IOS）接入

概述

开放平台,Android应用接入指南

接入说明

1.使用scheme方式启动

```
scheme方式配置launch_url配置为：
[scheme]://[host]/[path]?[query]
如：thirdappdemo://com.nd.sdp.openplatform.ThirdAppDemo/localpath?pageName=ViewController&ticket=123
```

- scheme：判别启动的 scheme，必填；启动应用配置URL Schemes；
- host：应用BundleId，例如：host=com.nd.sdp.openplatform.ThirdAppDemo
- path：启动路径
- query：启动app传参,且需要URLEncoder编码

2.后台配置

完整配置说明，请参考：

在开放平台后台配置应用指南

3.单点登录支持

目前开放平台移动端提供临时会话票据ticket 给接入应用，完成单点登录。具体接入步骤：

1. “开放平台”通过mac token获取到临时会话票据ticket
2. 第三方应用接收云平台跳转第三方应用时URL传递的临时会话票据ticket
3. 第三方应用通过ticket向IDP换取访问令牌access_token
4. 第三方应用通过access_token请求资源API，如获取用户信息等

参考文档：IDP

参数

- 当启动应用launch_url 含有ticket=${user_ticket}占位时，表示需要获取ticket并将其传递到启动app。
- 如何获ticket—-调用{uc-gateway}/v1.1/get_ticket获取临时会话票据ticket(post 请求，参数为空，但是还是需要传递一个空的{}，不然报错)。
- 获取到ticket之后将通过query传递到启动的app。

```
thirdappdemo://com.nd.sdp.openplatform.ThirdAppDemo/localpath?pageName=ViewController&ticket=123
```

- app启动后可从启动地址的query中取出ticket
- 通过ticket去获取access_token。（通常建议是这个操作放在App云端服务器，由云端中转接口调用请求，因为应用的密钥client_secret和获取到的access_token为了安全都应保存在服务端）

```
1、我们因为现在没有app服务端，就本地直接调用接口获取access_token。
2、首先向开放平台申请注册了app，获取client_id和client_secret
  应用ID：d11bd323-3d76-47eb-968c-36851184372a
  密钥：acdb9f6339814436995d39babf873960
3、调用接口 {uc-gateway}/v1.1/oauth2/access_token通过ticket获取access_token
4、获取到了access_token之后就完成认证，同时也提供了刷新access_token接口。
5、后续可以通过access_token去获取用户信息。 {uc-gateway}/v1.1/oauth2/get_user_info获取用户基本信息
```

参考DEMO

https://github.com/zhengzd/iOSOpenPlatformDemo

参考链接

- 在开放平台后台配置应用指南
- 开放平台帮助文档

移动端（Android）接入

概述

开放平台,Android应用接入指南

接入说明

1.接入应用访问协议

目前提供接入应用的启动方式有以下几种：

使用包名启动

```
直接配置包名即可。
op://包名?type=pkname
如：op://com.nd.hy.ele.startb?type=pkname
```

使用scheme方式启动（推荐使用）

```
scheme方式配置launch_url配置为：
op://scheme_url?type=scheme
如：op://example://com.nd.hy.ele.startb/ActivityDemo5?type=scheme
```

显式启动，启动app的指定页面。

```
显式配置launch_url配置为： op://包名?type=explicit&page=页面名
如：op://com.nd.hy.ele.startb?type=explicit&intent=com.nd.hy.ele.startb.MainActivity2
```

隐式启动，启动app的指定页面。

```
隐式配置launch_url配置为：
op://包名?type=implicit&intent=页面的AndroidManifest.xml中对应的action名
如：op://com.nd.hy.ele.startb?type=implicit&intent=android.intent.action.qidong3
```

备注：……

2.后台配置

完整配置说明，请参考：

在开放平台后台配置应用指南

3.单点登录支持

目前开放平台移动端提供临时会话票据ticket 给接入应用，完成单点登录。具体接入步骤：

1. “开放平台”通过mac token获取到临时会话票据ticket
2. 第三方应用接收云平台跳转第三方应用时URL传递的临时会话票据ticket
3. 第三方应用通过ticket向IDP换取访问令牌access_token
4. 第三方应用通过access_token请求资源API，如获取用户信息等

参考文档 ：获取oauth的ticket相关文档

参数

- 当启动应用launch_url 含有ticket=${user_ticket}占位时，表示需要获取ticket并将其传递到启动app。
- 如何获ticket—调用{uc-gateway}/v1.1/get_ticket获取临时会话票据ticket(post 请求，参数为空，但是还是需要传递一个空的{}，不然报错)。
- 获取到ticket之后将标签通过Bundlec传递到启动的app。

```
Bundle bundle = new Bundle();
bundle.putString(ticket, "user_ticket");
intent.putExtras(bundle);
```

- app启动页面中判断bundle是否为空，同时读取bundle中是否带有ticket

```
Bundle bundle=intent.getExtras();
if(bundle!=null) {
    String ticket = bundle.getString("ticket");
}
```

- 通过ticket去获取access_token。（通常建议是这个操作放在App云端服务器，由云端中转接口调用请求，因为应用的密钥client_secret和获取到的access_token为了安全都应保存在服务端）

```
1、我们因为现在没有app服务端，就本地直接调用接口获取access_token。
2、首先向开放平台申请注册了app，获取client_id和client_secret
  应用ID：d11bd323-3d76-47eb-968c-36851184372a
  密钥：acdb9f6339814436995d39babf873960
3、调用接口 {uc-gateway}/v1.1/oauth2/access_token通过ticket获取access_token
4、获取到了access_token之后就完成认证，同时也提供了刷新access_token接口。
5、后续可以通过access_token去获取用户信息。 {uc-gateway}/v1.1/oauth2/get_user_info获取用户基本信息
```

参考DEMO

https://github.com/kinglong123/appStartup

参考链接

- 在开放平台后台配置应用指南
- 开放平台帮助文档

学习活动数据接口

1. 学习对象类型简要描述：

- 用户学习数据查询

接口版本：

版本号	制定人	制定日期	修订日期
v0.1	dev	2020-12-31	init

请求URL:

- http://{api-gateway-host}/learning-activity/user_datas/user_study/search/user

请求方式：

- POST

认证方式：

- HMAC

> 详细可参见 开放平台API的认证说明

请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json
Date	是	string	http协议标准头
Authorization	是	string	hmac 签名
sdp-app-id	是	string	应用租户ID，具体信息获取方式参见应用的sdp-app-id参数获取方式

请求参数:

|参数名|是否必须|类型|默认值|说明|
|body |是|UserStudyDataSearchParam||搜索条件|

UserStudyDataSearchParam

参数名	是否必须	类型	默认值	说明
offset	否	int	0	分页索引 默认0
limit	否	int	10	每页数量 默认10条
object_types	是	array		学习对象类型列表：[100102,100103]
name	否	string		学习对象名称，支持模糊查询
update_begin_time	否	date		更新的开始时间：格式yyyy-MM-dd HH:mm:ss
update_end_time	否	date		更新的结束时间：格式yyyy-MM-dd HH:mm:ss
sort	否	string		排序，\"x desc\"
tenant_id	否	long		租户标识
project_id	否	long		业务租户标识
context_id	否	string		上下文
object_ids	否	array		学习对象标识
status	否	array		状态
required	否	int		是否必修
user_id	是	long		用户id

返回示例:

正确时返回:

```
{
    "total": 0,
    "items": [
        {
            "id": "string",
            "user_id": 0,
            "sdp_app_id": "string",
            "timestamp": "string",
            "domain": {
                "object_id": "string",
                "object_type": 0,
                "start_time": "2015-01-01T00:00:00",
                "end_time": "2015-01-01T00:00:00",
```

```
                "complete": 0,
                "complete_time": "2015-01-01T00:00:00",
                "enrolled_time": "2015-01-01T00:00:00",
                "passed": 0,
                "passed_time": "2015-01-01T00:00:00",
                "progress": 0,
                "duration": 0,
                "total_period": 0,
                "extra": "string"
            }
        }
    ]
}
```

错误时返回:

返回参数说明:

参数名	类型	说明
total	long	总数
items	UserDataVo	个人领域数据记录

UserDataVo

参数名	类型	说明
id	string	结果标识
tenant_id	long	租户标识
user_id	long	用户标识
sdp_app_id	string	应用标识
sdp_biz_type	string	业务类型
sdp_org_id	long	组织标识
timestamp	string	结果发生时间
domain	UserStudyDataVo	领域数据
project_id	long	业务租户标识（projectId）

UserStudyDataVo

参数名	类型	说明

参数名	类型	说明
object_id	string	学习对象标识
object_type	int	学习对象类型
startTime	date	开始学习时间
endTime	date	最后学习时间
complete	int	完成状态
completeTime	date	完成时间
enrolledTime	date	报名时间
passed	int	是否通过
passedTime	date	通过时间
progress	int	总进度-放大10000倍
duration	int	学习时间(秒)
totalPeriod	int	累计学时(秒)-处理后的时长
extra	string	扩展信息

#