alpsCloud 统一认证中心使用说明

版本号:V1.0.1
@Date:2019/12/12

技术: springsecurity+oauth2 + RBAC

微服务 Provider:

1. **alps-system-admin --- RBAC 权限服务**
2. **alps-oauth-uaa-admin --- Oauth2 管理服务**
3. **alps-oauth-server ---Oauth2 基础服务**

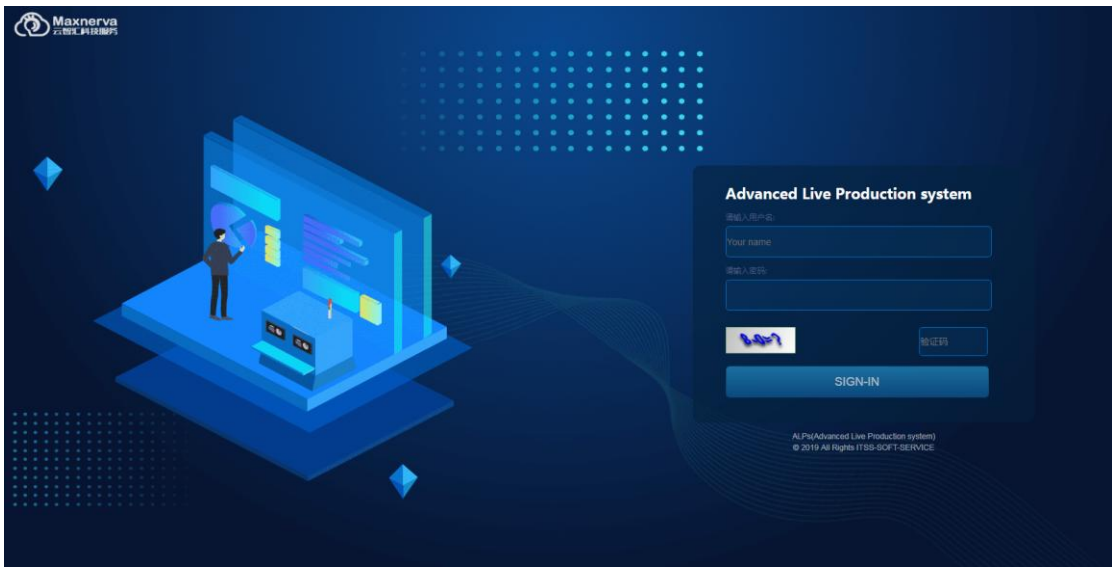
一: RBAC 权限服务

提供以下功能:

1. 用户管理 ---- 用户/账号 操作
2. 角色管理 ---- 角色操作,权限设定
3. 租户管理 ---- 租户管理
4. 部门管理 ---- 部门管理
5. 应用管理 ---- 应用操作,接口设定,权限管控
6. 岗位管理 ---- 岗伴=位管理
7. 数据字典 ---- 数据字典
8. 账号在线管理 ---- 账号在线管理

1. **启动 8001 端口的 sysPro 服务.**
2. **输入网址 http://10.202.86.138:8001/ 输入用户名和密码进入 RBAC 系统**

可在系统下面设定所有 RBAC 相关操作.



输入用户名和密码进入

进入到系统管理下面可进行所有的 RBAC 操作.

二:oauth2 使用

***暂无操作页面,如有 Application 先手动生成 app 的 key,cecrit.password,并手动添加到以下 Table.**

---- 执行 KeyRandomGen.java

```
/**
 * @author:Yujie.lee
 * Date:2019年12月11日
 * Todo 手动生成 appKey,appSecretKey以及password
 */
public class KeyRandomGen {

    /**
     *date:2019年12月11日
     *Author:Yujie.lee
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        String appKey = RandomValueUtils.randomAlphanumeric(32);
        String appSecretKey = RandomValueUtils.randomAlphanumeric(32);
        String password  = encryptPassword(appSecretKey);
        System.out.println("appKey === > " +appKey);
        System.out.println("appSecretKey === > " +appSecretKey);
        System.out.println("password === > " +password);


    }

      public static PasswordEncoder passwordEncoder() {
```

将生成的 ppKey,appSecretKey 以及 password 以及 app 信息添加到相应的 App 及 Oauth2 Table 中.

```
appKey === > f4tPX71UQoas4K5JJeLXyPINLXCFcCC7
appSecretKey === > ZCsYZUOYOwTj4qwGaKxBgxAWD4zODOrK
password === > $2a$10$Pj9AeOVH/Yf7nVYGC.cqieNLJED6r31B8.WPWR0ZYk0CfNE7finQO
```

Alps oauth2 目前支持:

I. **密码(passWord)授权模式**
II. **客户端(Client)授权模式**
III. **授权码模式**
IV. **隐式授权码模式**

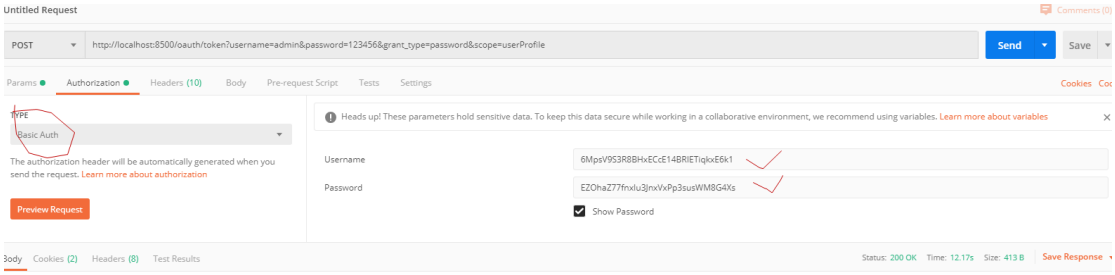
以下分别介绍 4 种 Oauth2 的 4 种使用方法

1. 密码(password)模式

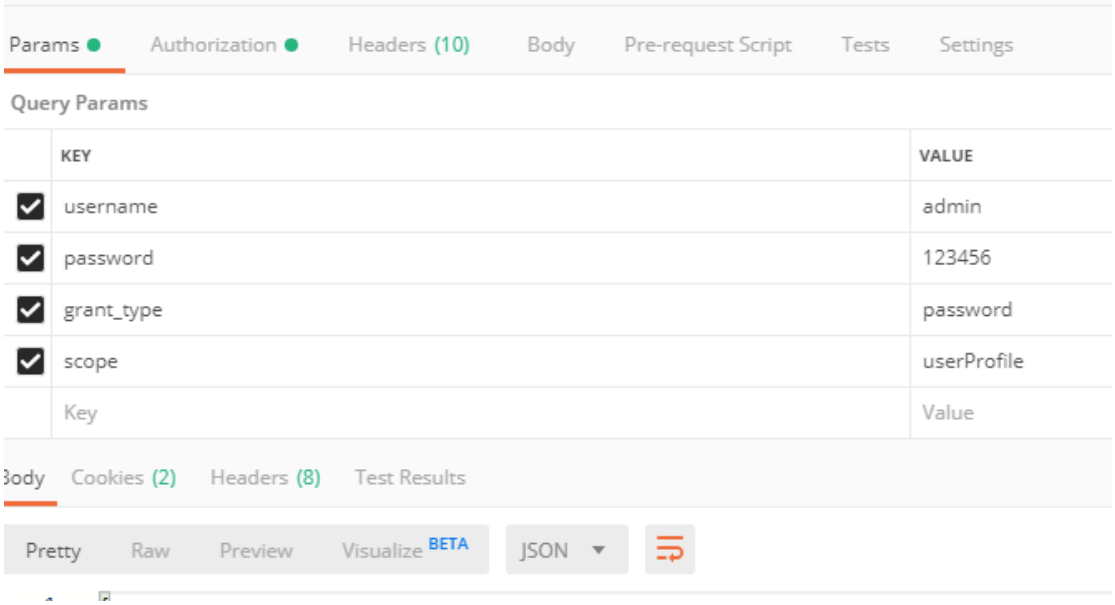
URI: http://localhost:8500/oauth/token

利用 postman

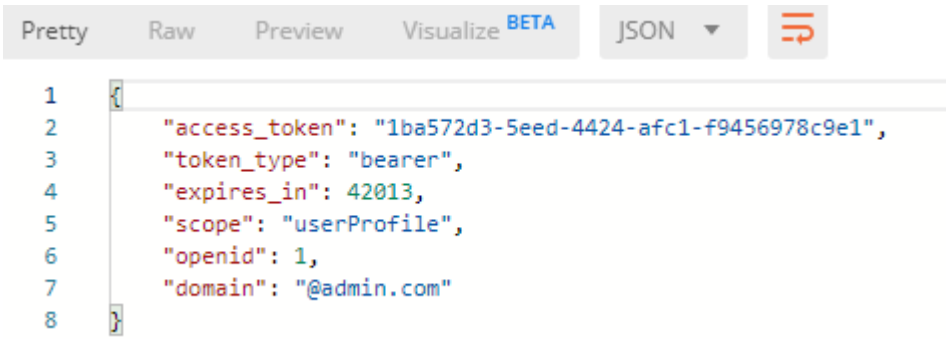
先设置请求头 basic 方式配置 client_id 和 client_secre



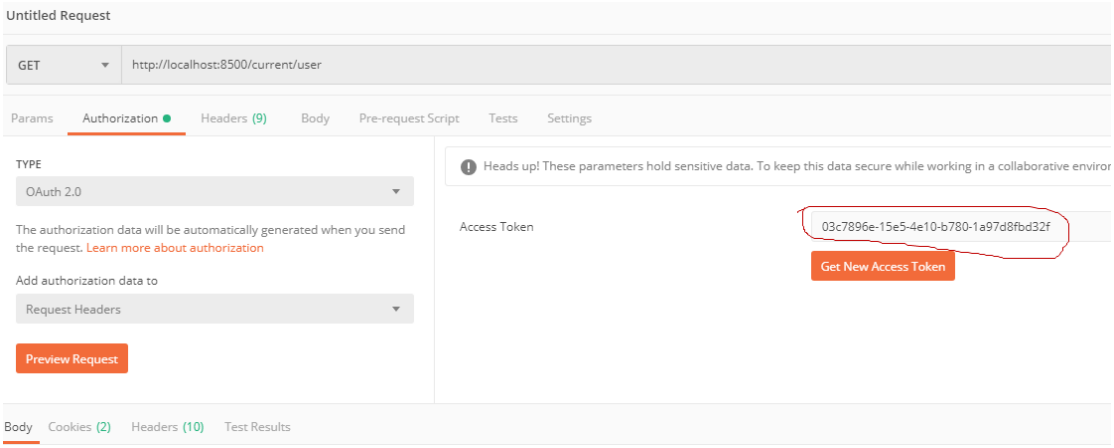
再设置 params



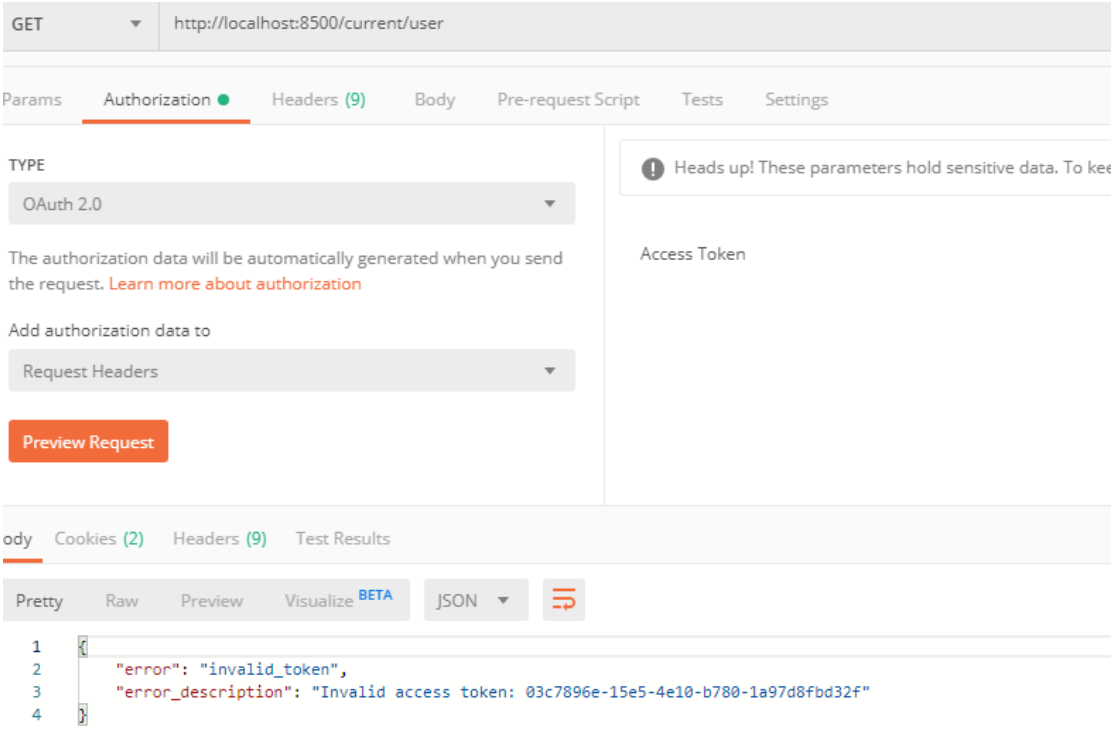
如果都正确认请求会返回所需要的 Token



拿到 Token 后,就可以访问相应的接口获取数据了.



如果访问没有授权的资源会返回权限不足.

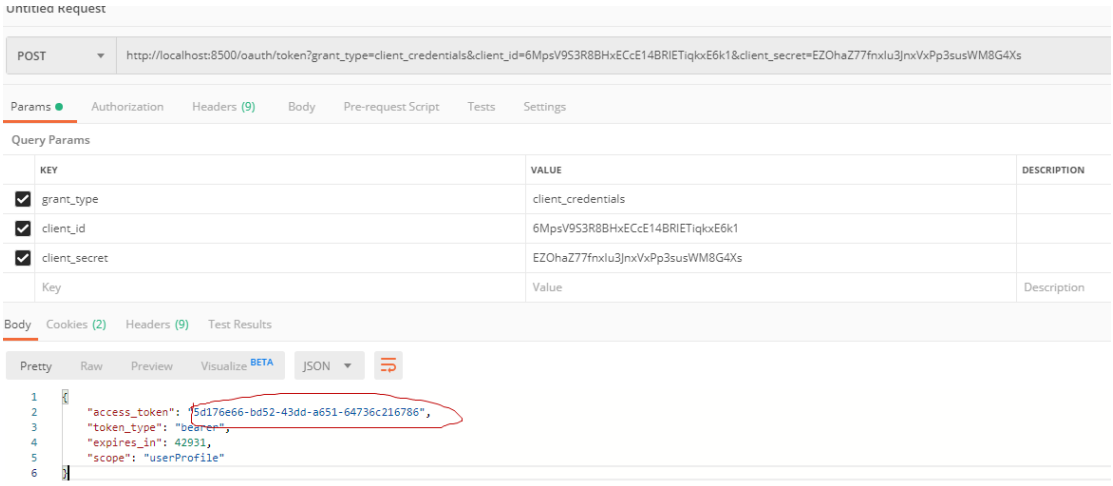


2. 客户端模式

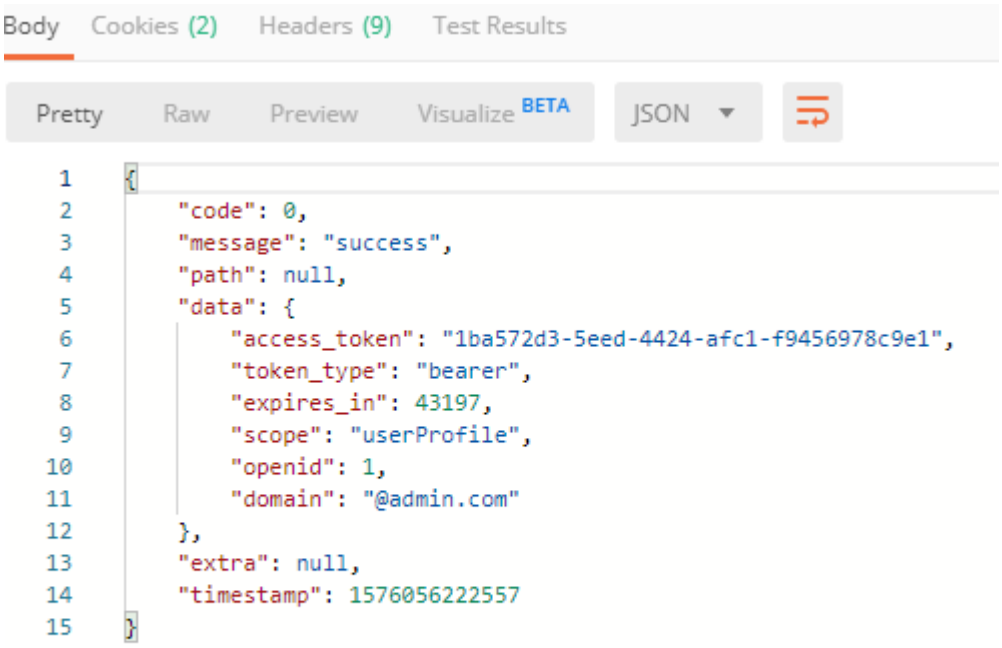
URI:http:///localhost:8500/oauth/token?grant_type=client_credentials&client_id=6MpsV9S3R8BHxECcE14BRIETiqkxE6k1&client_secret=EZOhaZ77fnxIu3JnxVxPp3susWM8G4Xs

通过 Client 的 key 与 secret 获取 Token 进行访问

如下操作



如果都正确认请求会返回所需要的 Token



3. **授权码模式**

给第三方授权,并支持 Token 刷新

URI:

http://localhost:8500/oauth/authorize?response_type=code&client_id=6MpsV9S3R8BHxECcE14BRIETiqkxE6k1&redirect_uri=http://hr.maxnerva.cn:9000

首先的 APP Table 增加相应的 APP 信息.

app_id	api_key	secret_key	app_name	app_name_en
123545625	I0lqybNA2VbbsFzmVqb6DQ6PksG1t2bK	NxW9mJb95pUiw6po0BSh6MCBqBnsigSJ	test	test
1552274783265	6MpsV9S3R8BHxECcE14BRIETiqkxE6k1	EZOhaZ77fnxIu3JnxVxPp3susWM8G4Xs	平台用户认证服务器	alps-uaa-admin-server
1552294656514	rOOM15Zbc3UFWgW2h71gRFvi	2Iw2B0UCMYd1cZjt8Fpr4KJUx75wQCwW	SSO单点登录	sso-demo
1552294656516	1eMFzjoVhSrhI5rpJPkKI7gOdyiIS5zT5678	f5mOf9RjVvV168IF7samrzgUBMAxJnmg	人事系统	HR
1552294656517	pJ6tNsKHclpT3NQYOU3WVaKG2o9L0wOn	S9cr8OCzEqYYWAH2hczflpKcXn9XvbBY	aplcloud官网	alps

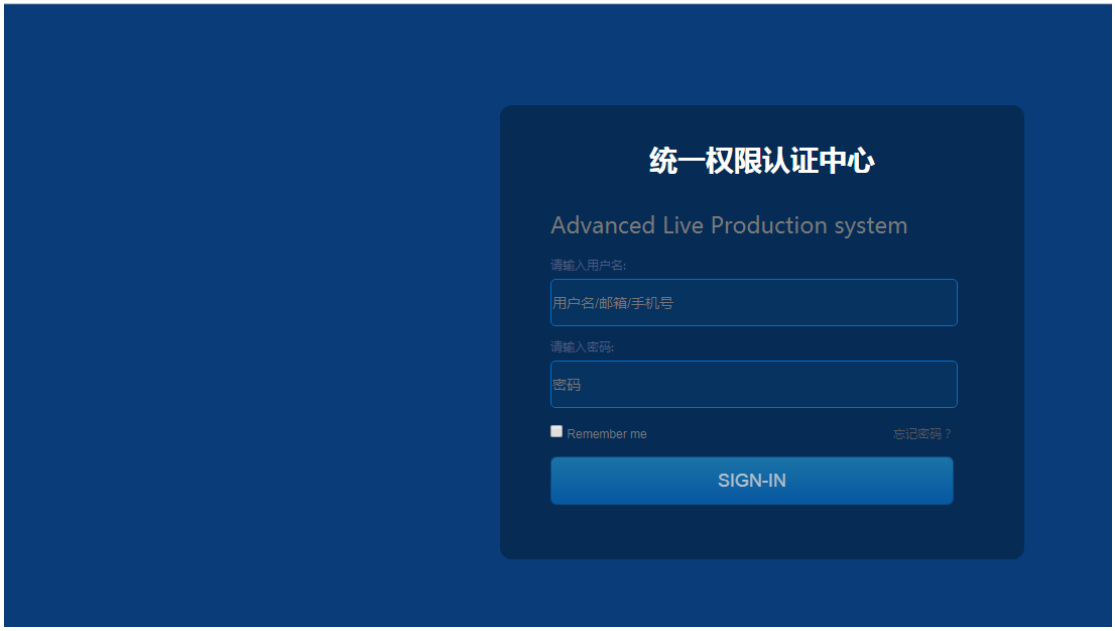
再在 oauth2 的 Table　oauth_client_details 中添加相关信息

开始事务　备注　筛选　排序　导入　导出

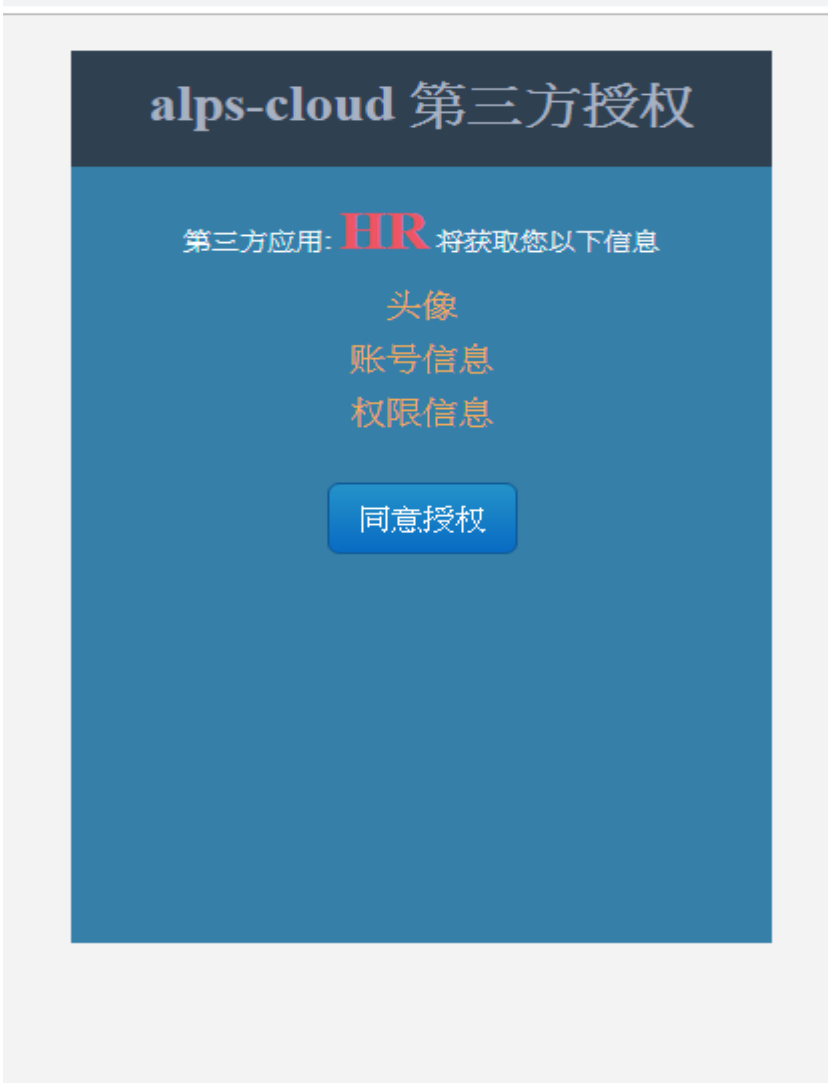
id	client_id	client_secret	resource_ids	scope	a
1	1eMFzjoVhSrhI5rpJPkKI7gOdyiIS5zT5678	$2a$10$tD4kI6Xpl5zs9.L6tKICb.GUoywjnxHXTXLSsDlvMuJZ.b7zHM4ZC	(Null)	userProfile	a
2	6MpsV9S3R8BHxECcE14BRIETiqkxE6k1	$2a$10$nB6OAgnhpNNor7w2FQlcF.ZpXfz9xMJGr.oeo6YX7qn/jQ29M9XG2		userProfile	a
3	I0lqybNA2VbbsFzmVqb6DQ6PksG1t2bK	$2a$10$HK2rswSOfEUSl6f7yv4zM.LRa.SCEPYmJZM72eqgpZlGKNO0FTmja	(Null)	userProfile	a
4	pJ6tNsKHclpT3NQYOU3WVsKG2o9L0wOn	$2a$10$1HiSmNBa8BE54TkqCAYXZ.2xg9Wkh0Pp1awz1u3o1RXiBu2vJ6pKy	(Null)	userProfile	a
5	rOOM15Zbc3UFWgW2h71gRFvi	$2a$10$NSb94sKA5i9kS/F0mUBehuBs9Gtvlv8wdxQOYMg3WxMRt0nyP2Xn.		userProfile	a

引导用户跳转到统一权限登录页面:

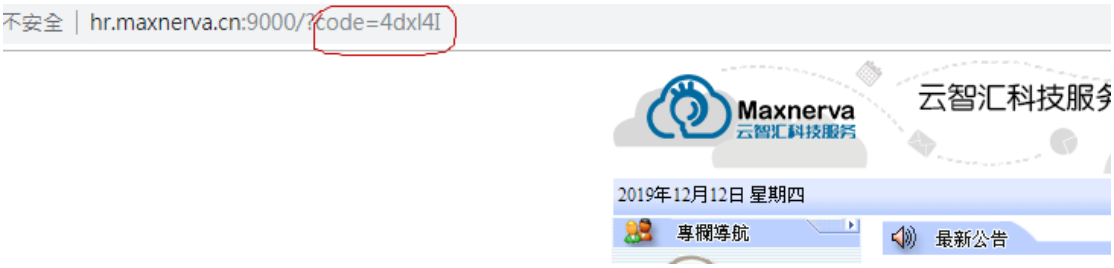
ost:8500/login



输入用户名和密码后,会跳转到授权页面

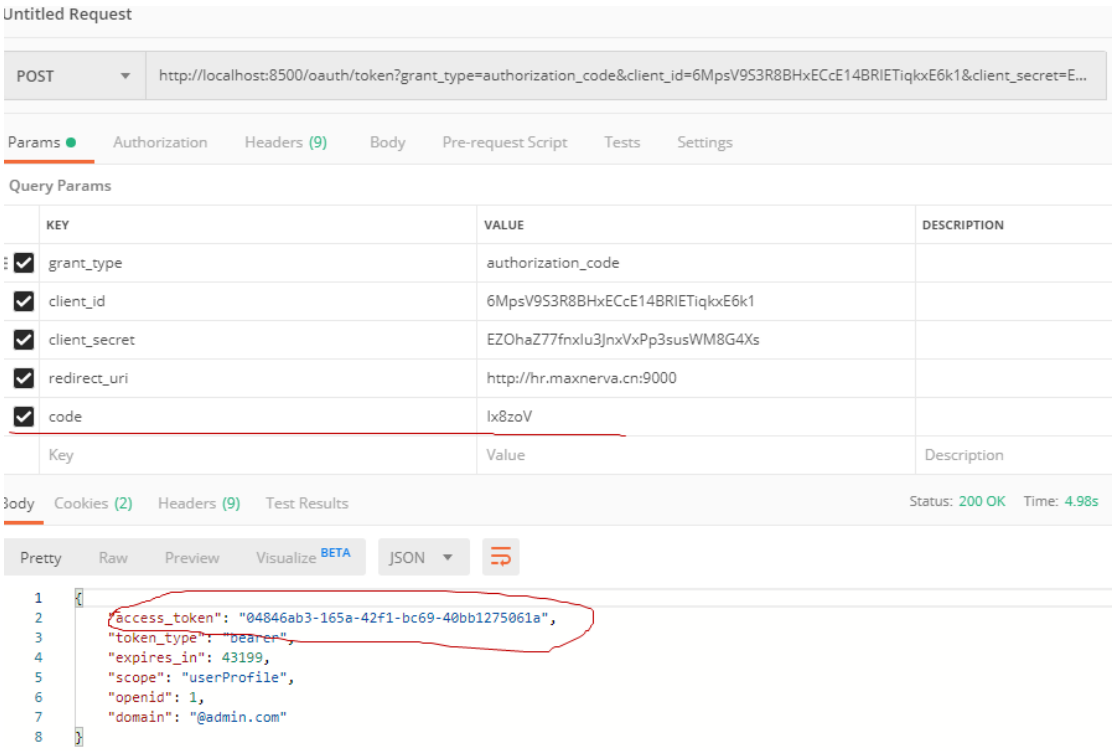


同意给第三方应用授权后就会中转到第三方应用,并携带上相应的授权码.



再利用获取的授权码 code 获取对应的 Token 进行访问

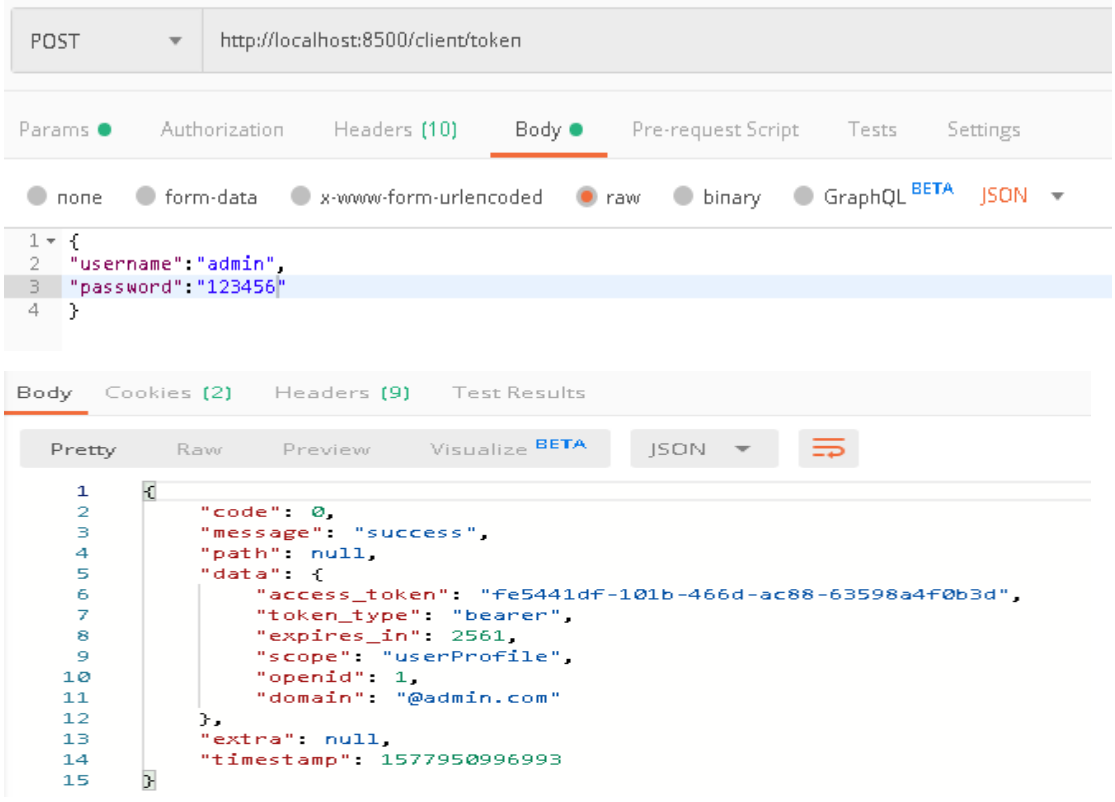
http://localhost:8500/oauth/token?grant_type=authorization_code&client_id=6MpsV9S3R8BHxECcE14BRIETiqkxE6k1&client_secret=EZOhaZ77fnxIu3JnxVxPp3susWM8G4Xs&redirect_uri=http://hr.maxnerva.cn:9000&code=Ix8zoV



4.隐式授权码获取 Token (参考授权码模式,略)

5. 开发者通过平台账号密码快速接口获取平台 Token.

URI: http://localhost:8500/client/token?username=admin&password=123456



6.oauth2 与网关结合

第一步,增加相应的 **APP** 信息.

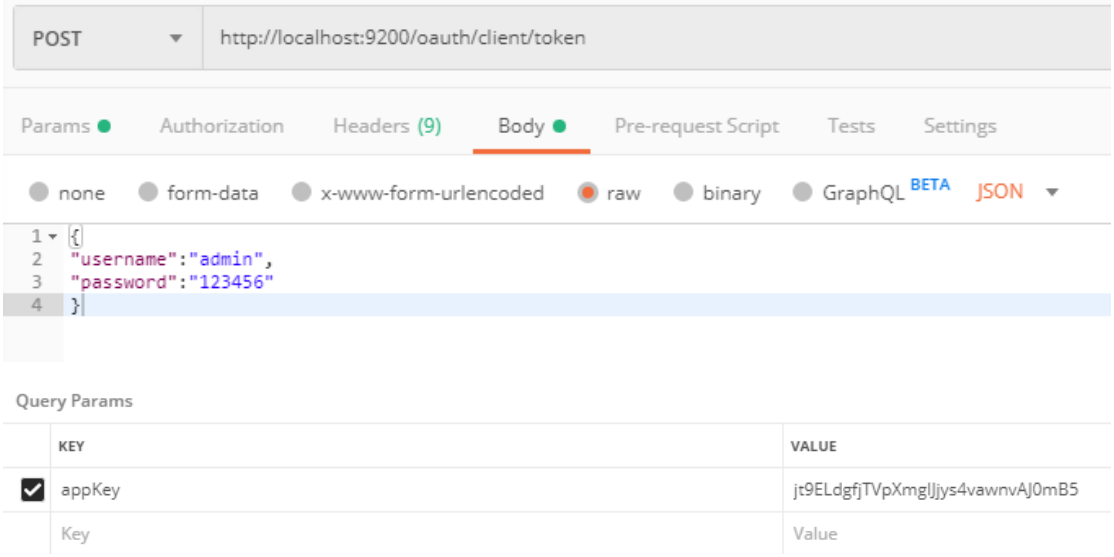
增加 **base_app** 与 **oauth_client_details** 两个 **Table** 中 **app** 信息.

第二步,配置 **gateway** 中路由:

其中路由 **ID** 必须与 **oauth_client_details** 中的 **Client_id** 相同

```
- id: jt9ELdgfjTVpXmgIJjys4vawnvAJ0mB5
  uri: lb://test
  order: 105
  predicates:
  - Path=/test/**
  filters:
  - AlpsSwaggerHeaderFilter
  - StripPrefix=1
  - name: Hystrix
    args:
```

第三步,在请求获取 **Token** 需将 **Client_id** 同用户名信息一起请求 **Token.**生成的 **Token** 后续只能作为访问皮 **client** 的凭证.



Alpscloud 授权使用方法及约定

1. 密码模式:

- 客户端(Client)要求用户登录,输入用户名和密码
- 将 Client 端的用户名,密码以及客户端约定的 client_id + client_secret 发送给授权服务器
- 授权服务器校验用户名、用户密码、client_id、client_secret，均通过返回 token 到客户端
- 客户端保存 token
- 客户请求携带 Token 访问资源服务器

使用范围: **遗留老旧系统**

支持令牌刷新: **是**

2.客户端模式:

- 客户端(Client)使用约定的 client_id + client_secret + 授权模式标识访问授权服务器
- 授权服务器校验通过返回 token 给客户端
- 客户端保存 token
- 客户请求携带 Token 访问资源服务器

使用范围: **平台内部服务/内部 app/内部高信任系统之间的 api 访问**

支持令牌刷新: **否**

3.授权码模式:

- 客户端发送 client_id + client_secret + 授权模式标识(grant_type) + 回调地址(redirect_uri)拼成 url 访问授权服务器
- 授权服务器引导用户跳转到登录界面，要求用户登录（此时用户提交的密码等直接发到授权服务器，进行校验）
- 授权服务器返回授权审批确认页面，要求用户授权
- 授权服务器接到授权后返回授权码到回调地址
- 客户端接收到授权码，并使用授权码 + client_id + client_secret 访授权服务器,并拿到返回的 Token
- 客户端保存 token
- 客户请求携带 Token 访问资源服务器

使用范围: **第三方 Web 服务器端应用与第三方原生 App**

支持令牌刷新: **是**

4.隐式授权码模式:

- 客户端发送 client_id + client_secret + 授权模式标识(grant_type) + 回调地址(redirect_uri)拼成 url 访问授权服务器
- 授权服务器引导用户跳转到登录界面，要求用户登录（此时用户提交的密码等直接发到授权服务器，进行校验）
- 授权服务器返回授权审批确认页面，要求用户授权
- ~~授权服务器接到授权后返回授权码到回调地址~~
- ~~客户端接收到授权码，并使用授权码 + client_id + client_secret 访授权服务器,并拿到返回的 Token~~
- 授权服务器接到授权后返回 Token 到回调地址
- 客户端保存 token
- 客户请求携带 Token 访问资源服务器

使用范围: 没有后台服务的纯 **Web** 页面

支持令牌刷新: 否