# 移动学习平台OAuth2.0概述

## 相关概念

OAuth2.0（开放授权）是一个开放标准，用户授权后，第三方应用无需获取用户的用户名和密码就可以访问该用户在某一网站上存储的私密的资源（如照片，视频，联系人列表）。

Access Token：用户身份验证和授权的凭证。第三方应用在调用开放平台开放API之前，首先需要获取Access Token。

## 使用授权前准备

您需要创建一个应用以获取 client\_id 和 client\_secret，申请方式请联系管理员

## 移动学习平台支持的OAuth授权

目前，移动学习平台OAuth2.0支持五种获取Access Token的流程和一种刷新获取AccessToken方式，第三方可根据需求选取合适的方式：

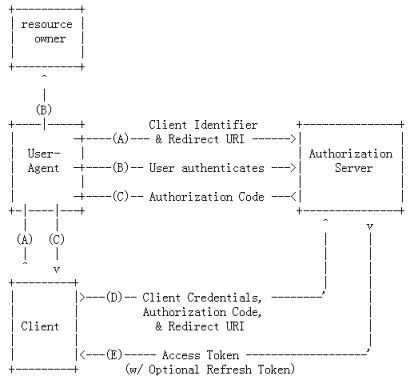
移动学习平台授权的Access Token是有有效期的，这样会影响用户的体验和增加开发者的工作。所以平台提供了一种方式可以保证授权有效期为永久。

* **实现方式**：返回给第三方30天有效期的Access Token + 近十年有效期的Refresh Token。
* **实现原理**：Refresh Token的作用就是在Token有效期截止前，刷新以获取新的Access Token。

|  |  |  |  |
| --- | --- | --- | --- |
| **获取途径** | **授权流程** | **介绍** | **有效期** |
| 新获取 | [Authorization Code](#_Authorization_Code授权) | 又称Web Server Flow，适用于所有有Server端配合的应用。 | 有效期30天的Access Token+有效期3650天的Refresh Token。 |
| 新获取 | [Resource Owner Password Credentials Grant](#_密码模式授权) | 用户向客户端提供自己的用户名和密码。客户端使用这些信息，向"服务商提供商"索要授权。 | 有效期30天的Access Token+有效期3650天的Refresh Token。 |
| 刷新 | [Refresh Token](#_Refresh_Token) | Access Token刷新方式，适用于所有有Server端配合的应用 。 | 3650天刷新期限。 |

# Authorization Code授权

## 简介

采用Authorization Code获取Access Token的授权验证流程又被称为Web Server Flow，适用于所有有Server端的应用，如Web/Wap站点、有Server端的手机/桌面客户端应用等。其流程示意图如下：  


 对于应用而言，其流程由获取Authorization Code和通过Authorization Code获取Access Token这2步组成。

## 获取Authorization Code

### 请求数据包格式

其获取方式是通过重定向用户浏览器（或手机/桌面应用中的浏览器组件）到“http://10.2.58.179:8080/educloudapi/oauth/authorize”地址上，并带上以下参数：

* **token**：必须参数，标识用户登录身份。
* **client\_id**：必须参数，申请应用时获得的client\_id。
* **response\_type**：必须参数，此值固定为“code”。
* **redirect\_uri**：必须参数，授权后要回调的URI，即接收Authorization Code的URI。如果用户在授权过程中取消授权，会回调该URI，并在URI末尾附上error=access\_denied参数。对于无Web Server的应用，其值可以是“oob”，此时用户同意授权后，授权服务会将Authorization Code直接显示在响应页面的页面中及页面title中。非“oob”值的redirect\_uri按照如下规则进行匹配：（1）如果开发者在“授权安全设置”中配置了“授权回调地址”，则redirect\_uri必须与“授权回调地址”中的某一个相匹配；（2）如果未配置“授权回调地址”，redirect\_uri所在域名必须与开发者注册应用时所提供的网站根域名列表或应用的站点地址（如果根域名列表没填写）的域名相匹配。
* **scope**：非必须参数，以空格分隔的权限列表，若不传递此参数，代表请求用户的默认权限。关于权限的具体信息请参考“权限列表”。
* **state**：非必须参数，用于保持请求和回调的状态，授权服务器在回调时（重定向用户浏览器到“redirect\_uri”时），会在Query Parameter中原样回传该参数。OAuth2.0标准协议建议，利用state参数来防止CSRF攻击。

例如：“client\_id”为“Va5yQRHlA4Fq4eR3LT0vuXV4”的应用要请求某个用户的默认权限和email访问权限，用户当前token为4ae2c3f1-2c27-48cf-88b7-b19751d33793，并在授权后需跳转到“http://www.example.com/oauth\_redirect”，则应用需要重定向用户的浏览器到如下URL：

http://10.2.58.179:8080/educloudapi/oauth/authorize?

response\_type=code&

client\_id=Va5yQRHlA4Fq4eR3LT0vuXV4&

redirect\_uri=http%3A%2F%2Fwww.example.com%2Foauth\_redirect&

scope=email&

token=4ae2c3f1-2c27-48cf-88b7-b19751d33793

### 响应数据包格式

此时授权服务会根据应用传递参数的不同，为用户展现不同的授权页面。如果用户在此页面同意授权，授权服务则将重定向用户浏览器到应用所指定的“redirect\_uri”，并附带上表示授权服务所分配的Authorization Code的code参数，以及state参数（如果请求authorization code时带了这个参数）。

例如：继续上面的例子，假设授权服务在用户同意授权后生成的Authorization Code为“mgO6j0gY8Hy1ptBl”，则授权服务将会返回如下响应包以重定向用户浏览器到“http://www.example.com/oauth\_redirect”地址上：

HTTP/1.1 302 Found

Location: http://www.example.com/oauth\_redirect?code=mgO6j0gY8Hy1ptBl

“code”参数可以在“redirect\_uri”对应的应用后端程序中获取。

|  |  |
| --- | --- |
| http://221.130.6.212:3282/images/icons/emoticons/warning.gif | **注意** 每一个Authorization Code的有效期为10分钟，并且只能使用一次，再次使用将无效。 |

## 通过Authorization Code获取Access Token

### 请求数据包格式

通过上面第一步获得Authorization Code后，便可以用其换取一个Access Token。获取方式是，应用在其服务端程序中发送请求（推荐使用POST）到 OAuth2.0授权服务的“http://10.2.58.179:8080/educloudapi/oauth/token”地址上，并带上以下5个必须参数：

* **grant\_type：**必须参数，此值固定为“authorization\_code”；
* **code：**必须参数，通过上面第一步所获得的Authorization Code；
* **client\_id：**必须参数，申请应用时获得的client\_id；
* **client\_secret：**必须参数，'申请应用时获得的client\_secret；
* **redirect\_uri：**必须参数，该值必须与获取Authorization Code时传递的“redirect\_uri”保持一致。

例如：

http://10.2.58.179:8080/educloudapi/oauth/token?

grant\_type=authorization\_code&

code=mgO6j0gY8Hy1ptBl&

client\_id=Va5yQRHlA4Fq4eR3LT0vuXV4&

client\_secret=0rDSjzQ20XUj5itV7WRtznPQSzr5pVw2&

redirect\_uri=http%3A%2F%2Fwww.example.com%2Foauth\_redirect

### 响应数据包格式

若参数无误，服务器将返回一段JSON文本，包含以下参数：

* **access\_token：**要获取的Access Token；
* **expires\_in：**Access Token的有效期，以秒为单位；请参考“[移动学习平台OAuth2.0概述](#_移动学习平台OAuth2.0概述)”
* **refresh\_token：**用于刷新Access Token 的 Refresh Token,所有应用都会返回该参数；（**10年的有效期**
* **scope：**Access Token最终的访问范围，即用户实际授予的权限列表（用户在授权页面时，有可能会取消掉某些请求的权限），关于权限的具体信息参考“[授权权限列表](#_授权权限列表)”一节；

例如：

HTTP/1.1 200 OK

Content-Type: application/json

Cache-Control: no-store

{

"access\_token": "d3fd32e7-688d-482b-943f-567877e8387f",

"token\_type": "bearer",

"refresh\_token": "b196f97f-749c-41ab-ba77-f9acef15ad43",

"expires\_in": 2330260,

"scope": "basic email"

}

若请求错误，服务器将返回一段JSON文本，包含以下参数：

* **error：**错误码；关于错误码的详细信息请参考“授权错误响应”。
* **error\_description：**错误描述信息，用来帮助理解和解决发生的错误。

例如：

HTTP/1.1 400 Bad Request

Content-Type: application/json

Cache-Control: no-store

{

"error": "invalid\_grant",

"error\_description": "Invalid authorization code: mgO6j0gY8Hy1ptBl"

}

## 开发者需要注意的事项

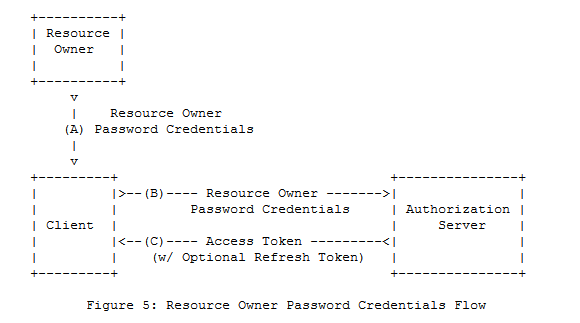
* 默认情况下，Access Token的**有效期为永久（一个月的Access Token + 10年的Refresh Token）**。

# 密码模式授权

## 密码模式

密码模式（Resource Owner Password Credentials Grant）中，用户向客户端提供自己的用户名和密码。客户端使用这些信息，向"服务商提供商"索要授权。  
在这种模式中，用户必须把自己的密码给客户端，但是客户端不得储存密码。这通常用在用户对客户端高度信任的情况下，比如客户端是操作系统的一部分，或者由一个著名公司出品。而认证服务器只有在其他授权模式无法执行的情况下，才能考虑使用这种模式。

## 密码授权模式步骤如下：



  （A）用户向客户端提供用户名和密码。

  （B）客户端将用户名和密码发给认证服务器，向后者请求令牌。

  （C）认证服务器确认无误后，向客户端提供访问令牌。

     B步骤中，客户端发出的HTTP请求，包含以下参数：

* client\_id：必须参数，申请应用时获得的client\_id
* client\_secret：必须参数，申请应用时获得的client\_secret
* grant\_type：必须参数，表示授权类型，此处的值固定为"password"。
* username：必须参数，表示用户名
* password：必须参数，表示用户的密码
* scope：可选参数，表示权限范围

  下面是一个例子。

http://10.2.58.179:8080/educloudapi/oauth/token?grant\_type=password&username=222222 &password=111111&client\_id=client2&client\_secret=secret2

  C步骤中，认证服务器向客户端发送访问令牌，下面是一个例子。

{

"access\_token": "bb588e7d-b4c0-49f9-b509-7d3969eafab0",

"token\_type": "bearer",

"refresh\_token": "99c38e17-ed28-461b-9d99-c67e5748e9a3",

"expires\_in": 2534649,

"scope": "basic"

}

      整个过程中，客户端不得保存用户的密码。

# Refresh Token

## 简介

        对于从开放平台申请到允许永久授权权限的应用，无论其采用Authorization Code、Reource Owner Password Credentials、Client Credentials

中的哪一个去获取Access Token，都会拿到一个有效期为一个月的Access Token和有效期为10年的Refresh Token对于这些应用，只要用户在10年内登陆应用，

应用就可以使用Refresh Token刷新以获得新的Access Token（新的Refresh Token也会同时下发），从而达到只要用户不连续10年未登陆过你的应用就不需要重新登陆的目的。

## 获取Access Token

请求数据包格式

        使用Refresh Token刷新以获得新的Access Token，需要应用在其服务端发送请求（推荐用POST方法）到开放平台授权服务的“http://10.2.58.179:8080/educloudapi/oauth/token”地址上，并带上以下参数：

* **grant\_type：**必须参数，固定为“refresh\_token”；
* **refresh\_token：**必须参数，用于刷新Access Token用的Refresh Token；
* **client\_id：**必须参数，申请应用时获得的client\_id；
* **client\_secret：**必须参数，申请应用时获得的client\_secret;
* **scope：**非必须参数。以空格分隔的权限列表，若不传递此参数，代表请求的数据访问操作权限与上次获取Access Token时一致。通过Refresh Token刷新Access Token时所要求的scope权限范围必须小于等于上次获取Access Token时授予的权限范围。

例如：

http://10.2.58.179:8080/educloudapi/oauth/token?

grant\_type=refresh\_token&

refresh\_token=b196f97f-749c-41ab-ba77-f9acef15ad43&

client\_id=Va5yQRHlA4Fq4eR3LT0vuXV4&

client\_secret= 0rDSjzQ20XUj5itV7WRtznPQSzr5pVw2&

scope=email

响应数据包格式

        若参数无误，服务器将返回一段JSON文本，包含以下参数：

* **access\_token：**要获取的Access Token；
* **expires\_in：**Access Token的有效期，以秒为单位；请参考“[移动学习平台OAuth2.0概述](#_移动学习平台OAuth2.0概述)”
* **refresh\_token：**用于刷新Access Token 的 Refresh Token,并不是所有应用都会返回该参数；（10年的有效期）
* **scope：**Access Token最终的访问范围

例如：

http://10.2.58.179:8080/educloudapi/oauth/token?

grant\_type=refresh\_token&

refresh\_token=b196f97f-749c-41ab-ba77-f9acef15ad43&

client\_id=Va5yQRHlA4Fq4eR3LT0vuXV4&

client\_secret= 0rDSjzQ20XUj5itV7WRtznPQSzr5pVw2&

scope=email

        若请求错误，服务器将返回一段JSON文本，包含以下参数：

* **error：**错误码；关于错误码的详细信息请参考“[授权错误响应](#_授权错误响应)”一节。
* **error\_description：**错误描述信息，用来帮助理解和解决发生的错误。

**例如：**

HTTP/1.1 400 Bad Request

Content-Type: application/json

Cache-Control: no-store

{

"error": "invalid\_grant",

"error\_description": "Invalid authorization code: ANXxSNjwQDugOnqeikRMu2bKaXCdlLxn"

}

# 授权权限列表

每一个Access Token代表“一个用户”或“开放平台”授予“一个应用”的“一系列数据访问操作权限”，请参考：

|  |  |
| --- | --- |
| **用户授权相关的权限** | **介绍** |
| basic | 用户基本权限，可以获取用户的基本信息 。 |

# 授权错误响应

移动学习平台OAuth2.0在接受验证授权请求时，授权服务器会按照OAuth2.0协议对本次请求参数、请求头部进行检验，若请求不合法或验证未通过，授权服务器会返回相应的错误信息。

## 错误码返回信息简介

错误码返回方式分两种：

* 在浏览器访问请求授权Endpoint"/oauth/authorize"时返回，直接在浏览器页面上显示错误信息。
* 在应用的服务端访问数据接口，如: 请求Access Token Endpoint "/oauth/token"时返回，返回方式是返回一段JSON文本，包含以下字段：

**error**：错误码；   
**error\_description**：错误描述信息，用来帮助理解和解决发生的错误。

例如：  
HTTP/1.1 400 Bad Request  
Content-Type: application/json  
Cache-Control: no-store

{

"error": "invalid\_grant",

"error\_description": "Invalid authorization code: ANXxSNjwQDugOnqeikRMu2bKaXCdlLxn"

}

**OAuth 2.0错误码列表**

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误信息** | **详细描述** |
| invalid\_request | invalid refresh token | 请求缺少某个必需参数，包含一个不支持的参数或参数值，或者格式不正确。 |
| invalid\_client | unknown client id | client\_id”、“client\_secret”参数无效。 |
| invalid\_grant | The provided authorization grant is revoked | 提供的Access Grant是无效的、过期的或已撤销的，例如，Authorization Code无效(一个授权码只能使用一次)、Refresh Token无效、redirect\_uri与获取Authorization Code时提供的不一致、Devie Code无效(一个设备授权码只能使用一次)等。 |
| unauthorized\_client | The client is not authorized to use this authorization grant type | 应用没有被授权，无法使用所指定的grant\_type。 |
| unsupported\_grant\_type | The authorization grant type is not supported | “grant\_type” 不支持该参数。 |
| invalid\_scope | The requested scope is exceeds the scope granted by the resource owner | 请求的“scope”参数是无效的、未知的、格式不正确的、或所请求的权限范围超过了数据拥有者所授予的权限范围。 |
| expired\_token | refresh token has been used | 提供的Refresh Token已过期 |
| redirect\_uri\_mismatch | Invalid redirect uri | “redirect\_uri”所在的根域与开发者注册应用时所填写的根域名不匹配。 |
| unsupported\_response\_type | The response type is not supported | “response\_type”参数值不为OAuth2.0服务所支持，或者应用已经主动禁用了对应的授权模式 |

# API列表

通过开放平台提供的REST API，第三方应用可以获取到用户的用户资料等基本信息，以及今后平台开放的其他任何数据，但前提是应用必须获得到开放平台和用户的授权。

## 开放API接口调用方式

开放平台提供的所有开放API的调用都是通过向开放平台的REST服务器发送HTTP请求来实现的。大多数API都支持使用POST和GET两种方法发送HTTP请求。

## 通过HTTP请求调用开放API

通过HTTP协议发送开放API调用请求时只需要在请求API对应的URL地址时，通过GET参数或POST参数传递具体开放API接口的业务级参数及下表中的几个系统级参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **参数类型** | **是否必需** | **描述** |
| access\_token | string | 是 | OAuth2.0验证授权后获得的token。 |

### 通过请求调用开放API示例

假设应用通过OAuth2.0协议获取Access Token时，授权服务器返回的JSON内容为：

{

"access\_token": "d3fd32e7-688d-482b-943f-567877e8387f",

"token\_type": "bearer",

"refresh\_token": "b196f97f-749c-41ab-ba77-f9acef15ad43",

"expires\_in": 2330260,

"scope": "basic email"

}

要求以json格式返回API调用结果，则应用既可以通过GET方法发送如下请求包来调用获取当前登录用户的基本资料的开放API接口：

GET http://10.2.58.179:8080/educloudapi/openapi/user/user\_info?access\_token=d3fd32e7-688d-482b-943f-567877e8387f HTTP/1.1

Host:10.2.58.179:8080

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36

Accept: \*/\*Accept-Encoding: gzip,deflate

Accept-Charset: utf-8

Connection: close

也可以通过POST方法发送如下请求包，来获取开放API调用的响应结果。

POST http://10.2.58.179:8080/educloudapi/openapi/user/user\_info HTTP/1.1

Host: 10.2.58.179:8080

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36

Accept:\*/\*

Accept-Encoding: gzip,deflate

Accept-Charset: utf-8

Connection: close

access\_token=d3fd32e7-688d-482b-943f-567877e8387f

## 目前开放平台提供的API列表

|  |  |  |
| --- | --- | --- |
| **用户信息类接口** | **访问相对路径** | **描述** |
| 获取当前登录用户的信息 | /openapi/user/user\_info | 返回当前登录用户的用户ID、真实姓名、昵称、账号、 手机号、性别、地址和校讯通账号等信息。 |

### 获取当前登录用户信息的正确示例

{

"ret": 0,

"msg": "获取用户成功",

"body": {

"user": {

"id": 214,

"name": "张老大",

"nickName": "zhangld",

"account": "zs1988",

"phone": "13512345678",

"sex": 1,

"address": null,

"xxtAccount": null

}

}

}

### 获取当前登录用户信息的错误示例

{

"error": "unauthorized",

"error\_description": "An Authentication object was not found in the SecurityContext"

}

{

"error": "invalid\_token",

"error\_description": "Invalid access token: 80a94a01-396d-4f2f-938b-0c97ed31e66d"

}