# Microsoft Graph

A gateway to data, 通过Microsoft Graph，可以让你的自定义应用系统（不管是Web应用，还是桌面应用，抑或 是移动App）通过统一的、RESTful的接口访问到授权用户的Office 365的资源。稍微深入地 展开一点来说，

* 一方面你的应用可以使用Office 365提供的Identity服务，简化和统一身份验 证环节；
* 第二方面，你将直接将Office 365的能力无缝地集成到你的应用中去，免费享受到微 软强大的基础投资带来的好处。

要进行Microsoft Graph的开发，必须要在Azure AD中进行应用程序的注册，由于历史的原 因，目前我们有两个不同的版本。目前针对Microsoft Graph的应用程序注册，有两种途径：

1. Azure AD应用程序，它仅适合于开发人员希望用户授权访问工作或者学校的账号。
2. Azure AD 2.0应用程序，它适合于开发人员希望用户授权访问工作或者学校的账 号，也能授权访问个人账号。

**TODO**

Access Token

https://developer.microsoft.com/en-us/graph/docs/concepts/auth\_overview

# Azure AD

参考：<https://docs.microsoft.com/zh-cn/azure/active-directory/develop/azure-ad-developers-guide>

<https://portal.azure.com/>

针对开发人员的 Azure Active Directory, 是一个云标识服务(a cloud identity service)，开发人员可以使用它来生成应用.。 Azure AD 支持开发人员的应用包括:

* single-tenant apps
* line-of-business (LOB) apps
* multi-tenant apps

单租户业务线 (LOB) 应用和多租户应用。 除了基本的sign in，Azure AD 还可以让：

* 应用调用 [Microsoft Graph](https://developer.microsoft.com/en-us/graph/docs/concepts/overview) 等 Microsoft API
* 以及在 Azure AD 平台上生成的自定义 API

In the past, application developers who wanted to support both personal Microsoft accounts and work accounts from Azure Active Directory had to integrate with two separate systems. 想要支持个人 Microsoft 帐户和 Azure Active Directory 中的工作帐户的应用程序开发人员必须集成两个单独的系统。 Azure Active Directory (Azure AD) v2.0 终结点推出了新的身份验证 API 版本，可简化此过程。 使用 Azure AD v2.0 终结点可通过一次集成从这两种类型的帐户登录。

**Azure AD 1.0**

参考：https://docs.microsoft.com/en-us/azure/active-directory/develop/authentication-scenarios#application-types-and-scenarios.

* [Web 浏览器到 Web 应用程序](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/authentication-scenarios#web-browser-to-web-application)
* [单页面应用程序 (SPA)](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/authentication-scenarios#single-page-application-spa)
* [本机应用程序到 Web API](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/authentication-scenarios#native-application-to-web-api)
* [Web 应用程序到 Web API](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/authentication-scenarios#web-application-to-web-api)
* [后台或服务器应用程序到 Web API](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/authentication-scenarios#daemon-or-server-application-to-web-api)

**Azure AD 2.0**

The Azure AD v2.0 endpoint enables sign-in from both types of accounts by using a single integration.

只要是OAuth 2.0 或 OpenID connection, Azure AD 2.0 supports authentication.

参考：

https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-app-types

https://docs.microsoft.com/zh-cn/azure/active-directory/develop/v2-app-types

The types of apps:

* Web Apps: 对于通过浏览器访问的 Web 应用（.NET、PHP、Java、Ruby、Python、Node 等），可以使用 [OpenID Connect](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/active-directory-v2-protocols) 来执行用户登录。
* Web APIs: 可以使用 v2.0 终结点来保护 Web 服务，例如应用的 RESTful Web API。 Web API 使用 OAuth 2.0 访问令牌（而不是 ID 令牌和会话 Cookie）来保护数据并对传入的请求进行身份验证。
* Mobile and Native APP
* Single page app (Javascript): 新式应用都有一个单页应用前端（主要以 JavaScript 编写）。 通常，这通过 AngularJS、Ember.js 或 Durandal.js 等框架编写。 Azure AD v2.0 终结点通过使用 [OAuth 2.0 隐式流](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/v2-oauth2-implicit-grant-flow)支持这些应用
* Daemons and server-side apps: 守护程序和服务器端应用包含长时运行进程或无需用户交互便可操作的应用还需要通过其他方法访问受保护的资源，例如 Web API。

不同点：

* One app ID for all platforms
* Scopes, not resources
* Incremental and dynamic consent
* Well-known scopes 请参阅 [v2.0 范围参考](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/v2-permissions-and-consent)。
* Token Claims： v2.0 终结点颁发的令牌中的声明与正式发布的 Azure AD 终结点颁发的令牌不会完全相同。请参阅 [v2.0 令牌参考](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/v2-id-and-access-tokens)。
* Limitations: 使用 v2.0 终结点时有一些要注意的限制。请参阅 [v2.0 限制文档](https://docs.microsoft.com/zh-cn/azure/active-directory/develop/active-directory-v2-limitations)

## OAuth认证流程

三个步骤:

1. 客户端代表用户发起认证请求（通常是/authorize 这个地址），这个会跳转到Office 365 的登录页面，让用户输入账号和密码
2. 如果用户提供了正确的账号和密码，并确认授权，Azure AD会向我们在注册应用程序时 提供的回调地址（redirectUrl）POST一个请求过来，附上一个code，然后我们的应用需 要继续用这个code去发起一个请求，申请访问令牌（通常是/token这个地址）
3. 客户端得到令牌（Access\_Token），就可以代表用户访问Microsoft Graph的资源（通常 是放在请求的头部里面）。这里需要注意的是，通常令牌都是会一定时间过期的，Micrsoft Graph的令牌默认为1小时有效。过期前可以通过一定的方式刷新令牌

参考：<https://docs.microsoft.com/en-us/azure/active-directory/develop/v1-protocols-oauth-code>

Access tokens issued by Azure AD are base 64 encoded JSON Web Tokens (JWT)

包括：information (or claims) about your app and the permissions it has for the resources and APIs available through Microsoft Graph

在call MS Graph时，attach the access token as a Bearer token to the Authorization header in an HTTP request。

**Two types of permissions:**

1. Delegated permissions
2. Application permissions

**Where to get an access token?** Azure AD

1. 通过/authorize endpoint, where your app can send a user to authenticate with Azure AD and consent to the permissions your app needs.
2. 通过/token endpoint, where your app can get an access token once user consent has been granted.

**过程：**

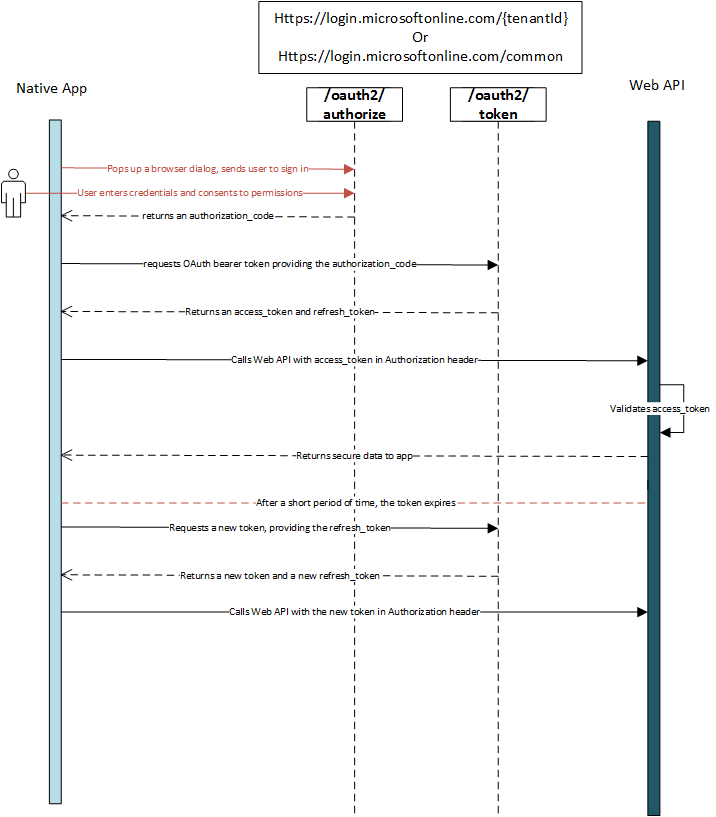
User 给予permissions –> oauth2/authorize 返回 authorization\_code

User 用authorization\_code 去请求Oauth bearer token -> oauth2/token 返回 access\_token 和 refresh\_token

User 用access\_token访问Web API （放在Header的Authorization）

一个小时过期后，

User 用 refresh\_token请求新的access\_token -> oauth2/token 返回access\_token 和新的refresh\_token



## 注册Azure AD 应用程序

Azure AD 从趋势上来说，2.0将逐渐全面取代，成为日后 主要的方式。但就目前而言，2.0中所提供的服务数量还没有1.0多。

Auzre AD应用程序有两种主要类型

* Web应用/API (需要给出登录Url, 这个需要填写对应的网站真正的登录路径)
* 本机应用 （redirected URL, 或随便给一个例如：<http://localhost>）

通常指的是网站或者服务站点，或者则指的是桌面应用或者移动应用。

最重要的就是定义该应用程序需要访问的资源，以及申请的权限了.

**两种权限：**

* delegated permission（为oauth权限，这是需要 用户授权，并且模拟用户的身份去进行操作，适合于一些有用户交互的应用程序）
* application permission(称为role权限, 则适合于一些在后台运行的服务或者自动运行的脚本)

1. **创建应用程序**

以第一步需要使用你的Office 365账号登录到Azure管理中心 <https://portal.azure.com>

* 1. Azure Active Directory
  2. App registrations
  3. New application registration
  4. Apply permission()
     1. Setting -> Required Permissions -> Add -> 1 Select an API –>委托权限
     2. 如果我们是为后台服务类应用程序进行注册，那么除了创建密钥外，还需要为应用程序申 请“应用程序权限”，而不是“委派权限”

1. **申请权限**

为这个应用程序申请必要的权限。

1. **创建密钥**（Optional）

一般情况下，我们调用Microsoft Graph之前，都会弹出（或者跳转） 到指定的页面要求用户去输入账号信息，并且亲自确认授权。**但是**如果我们的应用程序是一 个后台的服务，它需要一直运行，而且是没有交互界面的，这种情况下则需要另外创建一个密钥，同时采用Client Credential的方式来获取授权。

## 注册Azure AD 2.0 应用程序

目前功能还在不断完善，但Azure AD 2.0会逐渐成为主流，它有如下几个优势

1. Azure AD 2.0 应用程序，既支持访问工作或学校账号，也支持访问个人账号的资源
2. 注册Azure AD 2.0应用程序，不需要访问目标客户的Azure AD，是在一个独立的平台注 册，也就是说这种应用程序天生就是Multi Tenant模式的，有更高的复用性。
3. Azure AD 2.0应用程序的权限，是动态申请的，有利于应用程序升级，简化部署和管 理。
4. Azure AD 2.0应用程序，微软提供了更加高级的开发工具支持，针对大部分开发平台都 有SDK提供。

注册网站:

<https://apps.dev.microsoft.com/>

### 通过PowerShell来创建应用程序并且定义服务和权 限声

### Microsoft Graph 应用开发

三个步骤实现Microsoft Graph应用开发

* 第一步是注册应用程序
* 第二步是 实现身份认证
* 第三步就是该怎么调用你就怎么调用

### 桌面应用程序 实例

指在Windows桌面上面直接运行的.NET应用 程序，包括Console Application, WPF Application，Windows Forms Application, UWP Application

Azure AD 2.0 版：

1. 注册Azure AD 2.0 App`: 给出
   1. app name
   2. add platform
   3. add delegated permissions: file.read, Mail.read, Mail Send, and User.Read
2. 创建Console Application
   1. Create a new Console project
   2. Download and install NuGet
   3. install the client library via NuGet
      1. 在项目点击右键，选”管理NuGet程序包”：search for “Microsoft.Graph”
      2. Install it
      3. Search for “Microsoft.Identity.Client –IncludePrerelease” 如果没找到，可以用package Manager Console （**Tools > NuGet Package Manager > Package Manager Console）**

运行：Install-Package Microsoft.Identity.Client –IncludePrerelease

* + 1. Update-Package

NuGet使用：参考：https://docs.microsoft.com/en-us/nuget/quickstart/install-and-use-a-package-in-visual-studio