

使用 Microsoft Graph

- Microsoft Graph 概述。
- 通过查询参数优化数据使用。
- 优化网络流量。
- 使用 Microsoft Graph 访问用户数据。
- 使用 Microsoft Graph 访问文件。
- 在 Microsoft Graph 中管理组生命周期。



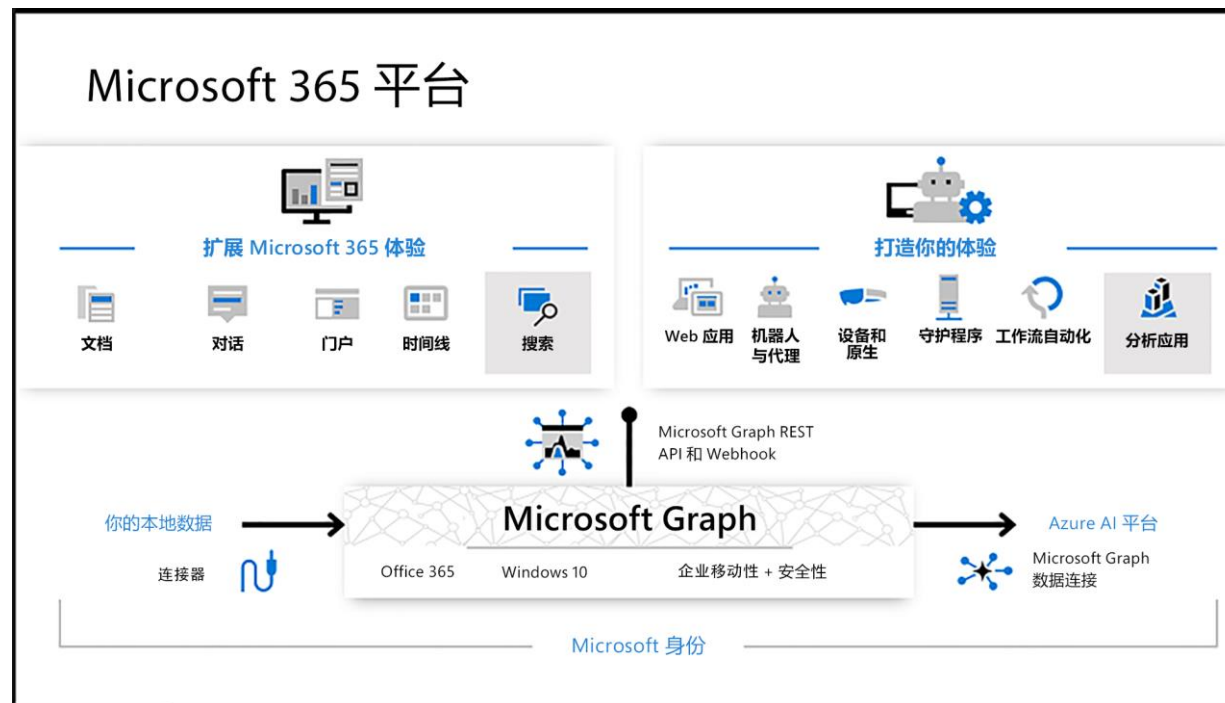
Microsoft Graph 概述

什么是 Microsoft Graph?

Microsoft Graph 是一种 RESTful Web API, 使你能够访问 Microsoft 云服务资源。

三个主要组成部分:

- Microsoft Graph API。
- Microsoft Graph 数据连接。
- Microsoft Graph 连接器。



访问 Microsoft Graph

- **使用 Microsoft Graph REST API**：使用你最熟悉的平台、框架和编程语言来发出常见的 HTTP 请求并处理 HTTP 响应。
- **使用 Microsoft Graph 原生 SDK**：将构造和处理 HTTP 请求的详细信息进行抽象。
- 身份验证选项：
 - **Azure AD 帐户**：M365 和 O365 资源
 - **Microsoft 帐户**：使用者资源

使用 Microsoft Graph API

- 使用 Microsoft Graph Explorer 测试

- 请求:

{HTTP method}https://graph.microsoft.com/{**version**}/{**resource**?{**query-parameters**}

- 响应:

- 状态代码
- 响应消息
- nextLink

通过查询参数优化数据使用

查询参数简介

- Microsoft Graph API 实现了许多 OData 协议的查询参数。
- 可以使用查询参数：
 - 对结果排序
 - 限制结果数
 - 搜索资源
 - 选择特定的数据属性
 - 筛选数据

\$select 查询参数

使用 \$select 可：

- 仅获取所需数据
- 获取默认情况下不会返回的属性

请求登录用户的 *from* 和 *subject* 属性：

[https://graph.microsoft.com/v1.0/me/messages?\\$select=from,subject](https://graph.microsoft.com/v1.0/me/messages?$select=from,subject)

\$orderby 查询参数

使用 \$orderby 指定结果的排序顺序

比在客户端上获取所有结果并排序更加高效

获取组织中的用户，按显示名称排序：

[https://graph.microsoft.com/v1.0/users?\\$orderby=displayName](https://graph.microsoft.com/v1.0/users?$orderby=displayName)

\$skip 和 \$top 查询参数

- 使用 \$top 查询参数指定结果集的页面大小。

https://graph.microsoft.com/v1.0/users?\$top=5

- 将 @odata:nextLink 属性的 URL 值发送到 Microsoft Graph, 检索下一页结果。

"@odata.nextLink": "https://graph.microsoft.com/v1.0/users?\$top=5&\$skiptoken=X%274453707...6633B90000000000000000000000%27"

- 使用 \$skip 查询参数设置要从集合起始位置跳过的项目数。

https://graph.microsoft.com/v1.0/me/events?\$orderby=createdDateTime&\$skip=20

\$expand 查询参数

- 使用 \$expand 获取与请求的资源相关的其他资源。
- 提供要扩展并包含在结果中的属性的逗号分隔列表。

获取根驱动器信息，以及驱动器中的顶级子项：

GET [https://graph.microsoft.com/v1.0/me/drive/root?\\$expand=children](https://graph.microsoft.com/v1.0/me/drive/root?$expand=children)

\$count 查询参数

添加 \$count=true 作为查询字符串参数，以显示结果集合中的项目总数

要获取当前用户的联系人集合，以及联系人集合中的项目数，请执行以下操作：

[https://graph.microsoft.com/v1.0/me/contacts?\\$count=true](https://graph.microsoft.com/v1.0/me/contacts?$count=true)

\$search 查询参数

- 使用 \$search 限制请求的结果，以匹配搜索条件。
- 支持关键字查询语言 (KQL) 结构化查询，可将搜索目标锁定于特定字段。
- 你可以搜索消息和人员集合。
 - 消息搜索功能默认情况下会搜索 *from*、*subject* 和 *body* 属性
 - 人员搜索功能会搜索 *displayName* 和 *emailAddress* 属性

在登录用户的收件箱中提取 *from*、*subject* 或 *body* 属性包含 “pizza” 的邮件

GET https://graph.microsoft.com/v1.0/me/messages?\$search="pizza"

\$filter 查询参数

- 使用 \$filter 查找与指定查询匹配的资源。
- 用于仅检索集合的子集。
- 不同 Microsoft Graph API 对 \$filter 运算符的支持不同。

使用 *equals* 逻辑运算符获取登录用户收件箱中的所有未读邮件：

[https://graph.microsoft.com/v1.0/me/mailFolders/inbox/messages?\\$filter=isRead eq false](https://graph.microsoft.com/v1.0/me/mailFolders/inbox/messages?$filter=isRead eq false)

演示

在通过 HTTP 查询 Microsoft Graph 时
使用查询参数



优化网络流量

接收变更通知

构建应用程序，响应 Microsoft Graph 中数据的变化：

- 订阅对所支持资源的更改
- 在创建通知时，客户端会在所提供的 URL 收到通知
- 发送 HTTP 202 响应，以确认收到通知
- 验证 clientState 属性
- 根据业务逻辑更新应用程序

执行批量请求

使用 \$batch 将多个 Graph 请求合并为单个 HTTP 请求，从而减少 HTTP 调用的次数。

- 创建一个 JSON 请求数组，然后 POST 到 \$batch。
- 使用 **dependsOn** 指定一个批次中请求间的依赖关系。

```
{
  "requests": [
    {
      "id": "1",
      "method": "GET",
      "url": "... "
    },
    {
      "id": "2",
      "dependsOn": [ "1" ],
      "method": "GET",
      "url": "... "
    },
    {
      "id": "3",
      "method": "GET",
      "url": "... "
    },
    {
      "id": "4",
      "dependsOn": [ "2" ],
      "method": "GET",
      "url": "... "
    }
  ]
}
```

使用 delta 查询获取更改

高效地轮询 Microsoft Graph 以了解数据集的更改。

- 对所需资源发出 GET 请求，使用 delta 函数。
- 如果有其他页面，Graph 响应将包括 nextLink URL；如果没有有关资源现有状态的更多数据，将包括 deltaLink URL。
- 对于未来的请求，请使用 deltaLink URL 检索自上次请求以来资源的更改。
- 必须在初始请求中指定可选查询参数。然后，它们将被编码在 nextLink 或 deltaLink 令牌中。

处理节流

节流会限制对服务的并发调用次数，以防止过度使用资源。

发生节流时，Microsoft Graph 会返回 HTTP 状态代码 429，请求也会失败

- 等待 **Retry-After** 标头中指定的秒数
- 重试请求

演示

通过批处理请求减少流量



使用 Microsoft Graph 访问用户数据

在 Microsoft Graph 中执行用户相关操作

访问和操作用户资源，而无需执行其他调用，查找特定身份验证信息，以及直接针对其他 Microsoft Graph 资源发出查询。

通过 Microsoft Graph 访问用户：

- `/users/{id | userPrincipalName}`
- `/me`

获取用户的相关信息

任务	HTTP 请求
获取组织中的用户列表	GET /users
获取用户的个人资料照片	GET /me/photo/\$value GET /users/{id userPrincipalName}/photo/\$value
获取有关个人资料照片的元数据	GET /me/photo GET /users/{id userPrincipalName}/photo
获取用户的经理的个人资料	GET /me/manager GET /users/{id userPrincipalName}/manager

演示

从 Microsoft Graph
查询用户数据



使用 Microsoft Graph 访问文件

在 Microsoft Graph 中处理文件

- Microsoft Graph 公开了两种处理文件的资源类型：
 - Drive 资源
 - DriveItem 资源
- Drive 和 DriveItem 资源以三种不同的方式公开数据：
 - 属性公开简单值。
 - Facet 公开复杂值。
 - 引用指向其他资源的集合。

获取有关驱动器的信息

驱动器是文件系统的顶级容器，例如 OneDrive 或 SharePoint 文档库。开发人员可以使用 Microsoft Graph 来检索驱动器资源的**属性**和**关系**。

任务	HTTP 请求
获取有关驱动器的信息	GET /me/drive GET /users/{id UserPrincipalName}/drive
获取驱动器中的项目列表	GET /me/drive/root/children
获取登录用户相关的趋势项目列表	GET /me/drive/recent

下载文件

开发人员可以使用 Microsoft Graph 下载 Driveltem 的主流（文件）的内容。

只能下载具有 **file** 属性的 Driveltem。

HTTP 请求

```
GET /me/drive/items/{item-id}/content
```

上传大文件

使用上传会话上传文件分两个步骤：

1. 创建上传会话。
2. 将字节上传到上传会话。

HTTP 请求：

```
POST /me/drive/root:{item-path}:/creatUploadSession
```

从一个组中的所有者列表中获取用户对象，然后检索该用户的文件

根据本主题的场景，需要按以下顺序遍历 Microsoft Graph：

- 枚举组的所有者，以获取用户 ID。
- 如有必要，授予适当的权限，进行列表所有者的 API 调用。
- 获取用户的驱动器 ID。
- 使用之前的 HTTP 响应返回的用户 ID 和驱动器 ID，列出用户驱动器中的项目。



在 Microsoft Graph 中管理组生命周期

在 Microsoft Graph 中执行与组有关的操作

Microsoft Graph 中的组资源可表示多种内容，因为存在不同类型的组。可通过 Microsoft Graph 访问的组类型包括：

- Office 365 组
- 安全组

获取组信息

开发人员可以使用 Microsoft Graph API 或某个 Microsoft Graph SDK 来获取组的列表或特定组。

HTTP 请求

```
GET https://graph.microsoft.com/v1.0/groups  
GET https://graph.microsoft.com/v1.0/groups/{ID}
```

获取组信息

任务	HTTP 请求
获取组负责人	GET /groups/{id}/owners
获取组成员	GET /groups/{id}/members
获取某用户为负责人的组列表	GET https://graph.microsoft.com/v1.0/me/ownedObjects
获取某用户为成员的组列表	GET https://graph.microsoft.com/v1.0/me/memberOf GET https://graph.microsoft.com/v1.0/users/{ID userPrincipalName}/memberOf

管理组

任务	HTTP 请求
创建组	POST /groups
利用组创建团队	PUT /groups/{id}/team
删除组	DELETE https://graph.microsoft.com/v1.0/groups/{ID}