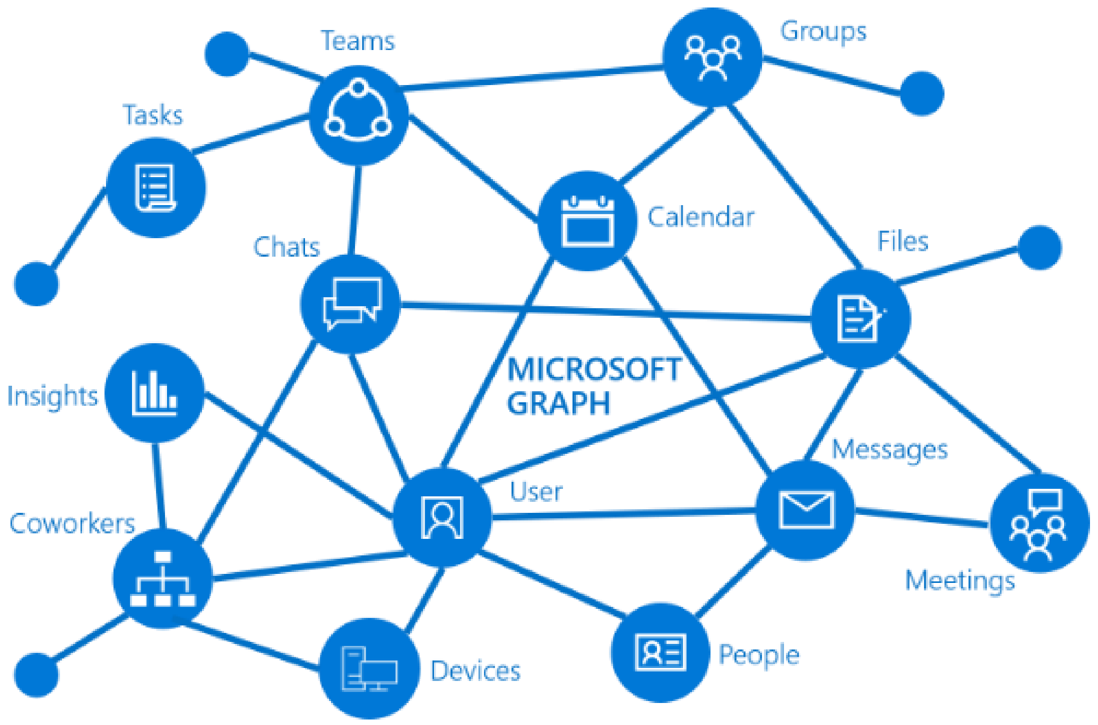


Maneuvering Your Way to Microsoft Graph API

TODAY'S

Speaker

Jocel Sabellano

- Sr. Systems Engineer
- I write automations in Powershell
- github.com/theitrx

TODAY'S

Agenda

What is Microsoft Graph API

- Overview of Microsoft Graph API
- Benefits of knowing Graph
- What applications or automations can you make with Graph

Authenticate to Graph API

- Grant Types
- Requesting a token
- Sending queries to Graph

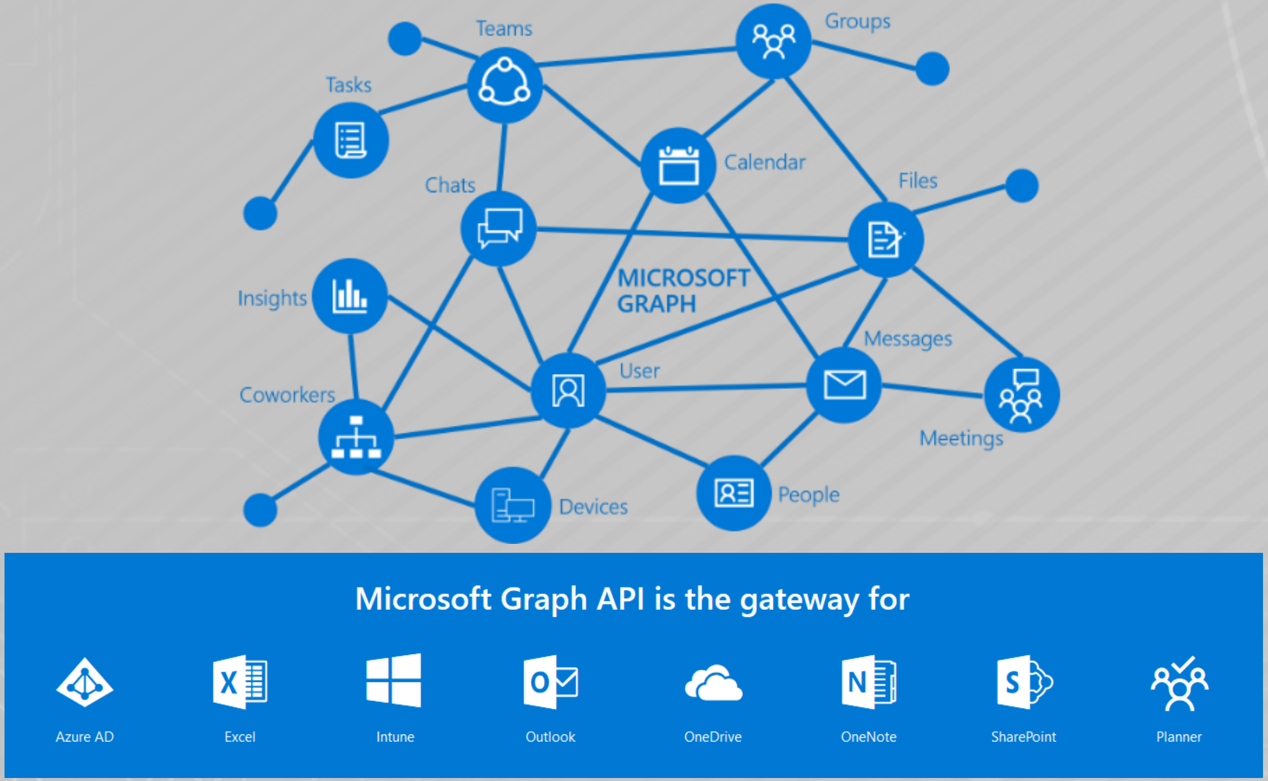
OVERVIEW OF MICROSOFT GRAPH API

WHAT IS MICROSOFT GRAPH

Is a REST API, a unified endpoint

That gives you access to
- Azure AD
- Excel
- Intune
- Outlook
- Onedrive
- Sharepoint
- Planner
- ..and More!

And you can
- Generate reports and audit data
- Perform hundreds of tasks programmatically
- Build automations or apps
- ..and more!



Why is Microsoft Graph Important

What can you do with Microsoft Graph?

Onboard users
Automate user onboarding workflows and manage changes to user roles within an organization.

Integrate with Excel
Tap into Excel data to build powerful workflows that automate data collection.

Manage employee profiles
Manage employee information stored across various locations in the enterprise.

Manage Intune and AutoPilot
Backup your tenant's entire Intune and AutoPilot configuration, compare backup sets and restore from an older configuration

Correlate security alerts
Use one API to access 9 Microsoft security services, Provide visibility into security data to enable proactive risk management, Invoke actions to in response to new threats and review your tenants Secure Scores

Reporting, Insights and Analytics
Integrate Office 365 usage reporting into your organization's existing reporting solution. View user analytics and enterprise insights

Why is Microsoft Graph Important

Microsoft Graph Stats

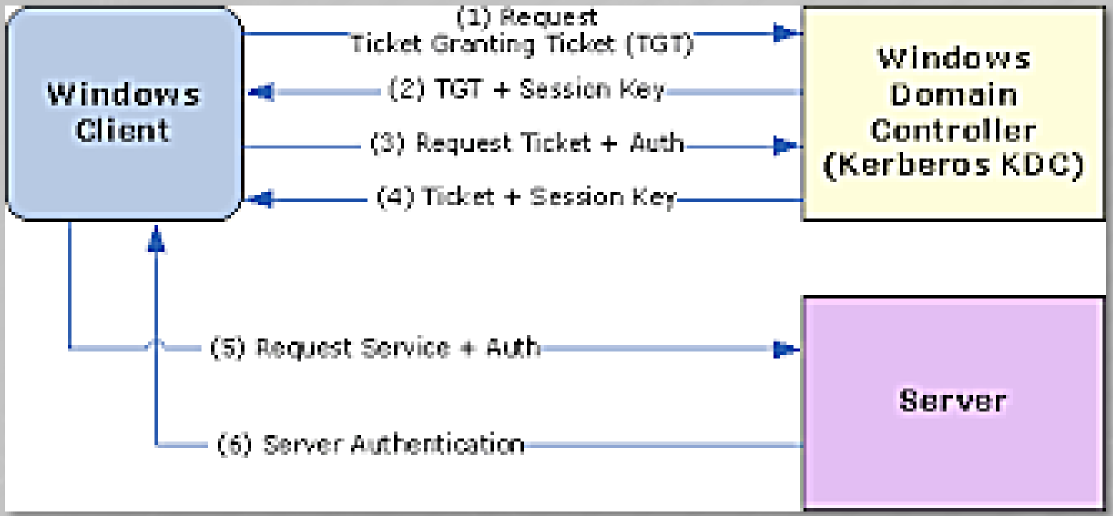
90%	of all Fortune 500 companies have data in Microsoft Graph
100B	Microsoft Graph requests every month
18T	resources (emails, events, users, files, groups, and more) in Microsoft Graph
180M	monthly active users of Office 365 commercial

GRAPH VS PSSESSION

AUTHENTICATION

PSSESSION

- Uses Active Directory or Local Account Creds
  - Username
  - Password
- Types
  - CredSSP
  - Kerberos
  - Digest
- PSCredential Object

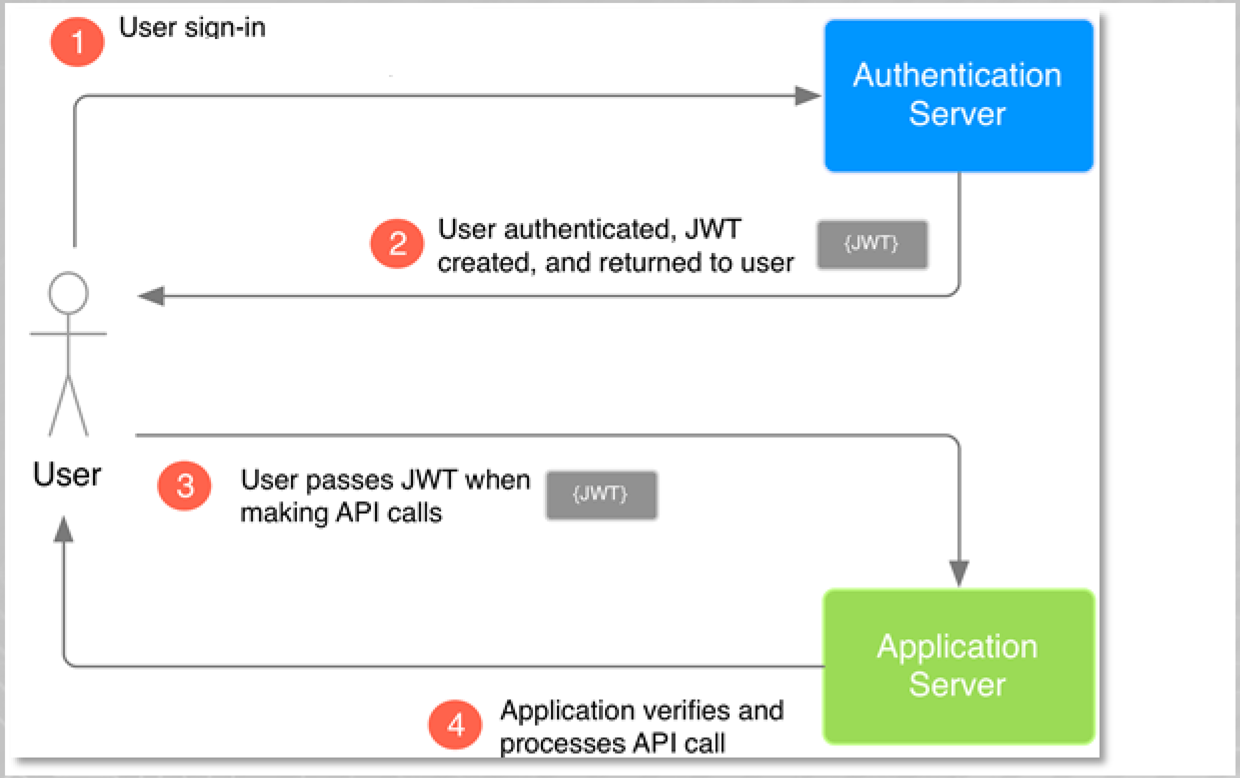

Windows Client
(1) Request Ticket Granting Ticket (TGT)
(2) TGT + Session Key
(3) Request Ticket + Auth
(4) Ticket + Session Key
Windows Domain Controller (Kerberos KDC)
(5) Request Service + Auth
(6) Server Authentication
Server

GRAPH VS PSSESSION

AUTHENTICATION

GRAPH API

- HTTP
  - Azure AD Application
    - Application ID
    - Application Secret
  - Microsoft / Azure AD Account
    - Username
    - Password
  - JSON Web Token
  - Authentication code
  - Device code


1 User sign-in
Authentication Server
2 User authenticated, JWT created, and returned to user
{JWT}
User
3 User passes JWT when making API calls
{JWT}
Application Server
4 Application verifies and processes API call

AUTHENTICATE TO GRAPH API

OBJECTS YOU NEED

Service Principal

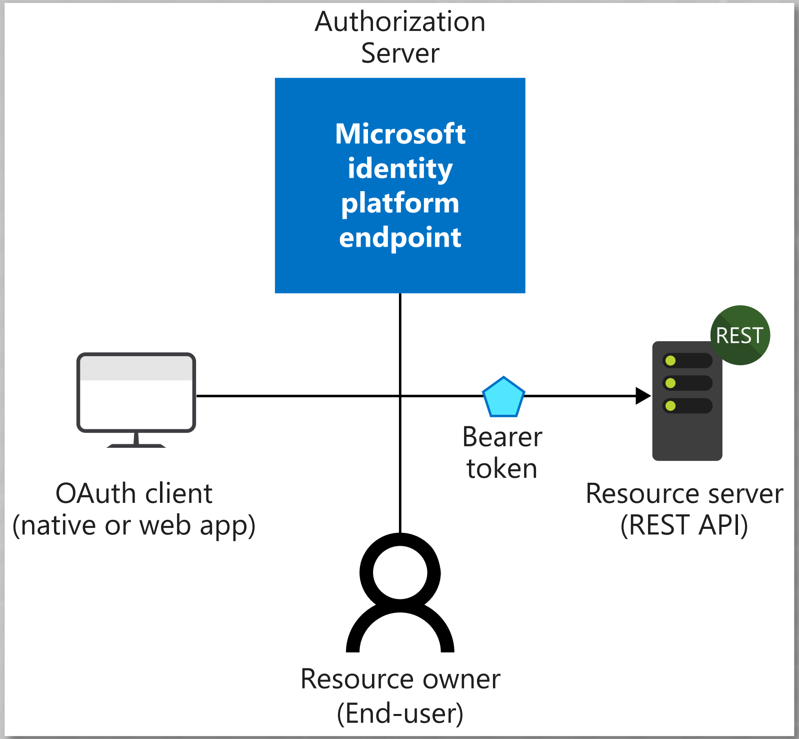
- Application ID/Secret
- Directory (tenant) ID
- Permissions
- User or Admin Consent
- AAD User Account Credentials

Access Token

- Needed to Access Graph API
- JSON Web Token (JWT)
- URL - https://login.microsoftonline.com
  /{TenantID}
  /oauth2/v2.0/token

Graph API Endpoint

- Where you will send your request
- Root URL - https://graph.microsoft.com/v1.0/
- Root URL - https://graph.microsoft.com/beta/
- https://graph.Microsoft.com
  /{version}
  /{resource}
  /{id}
  /{property}
  /{query-parameters}

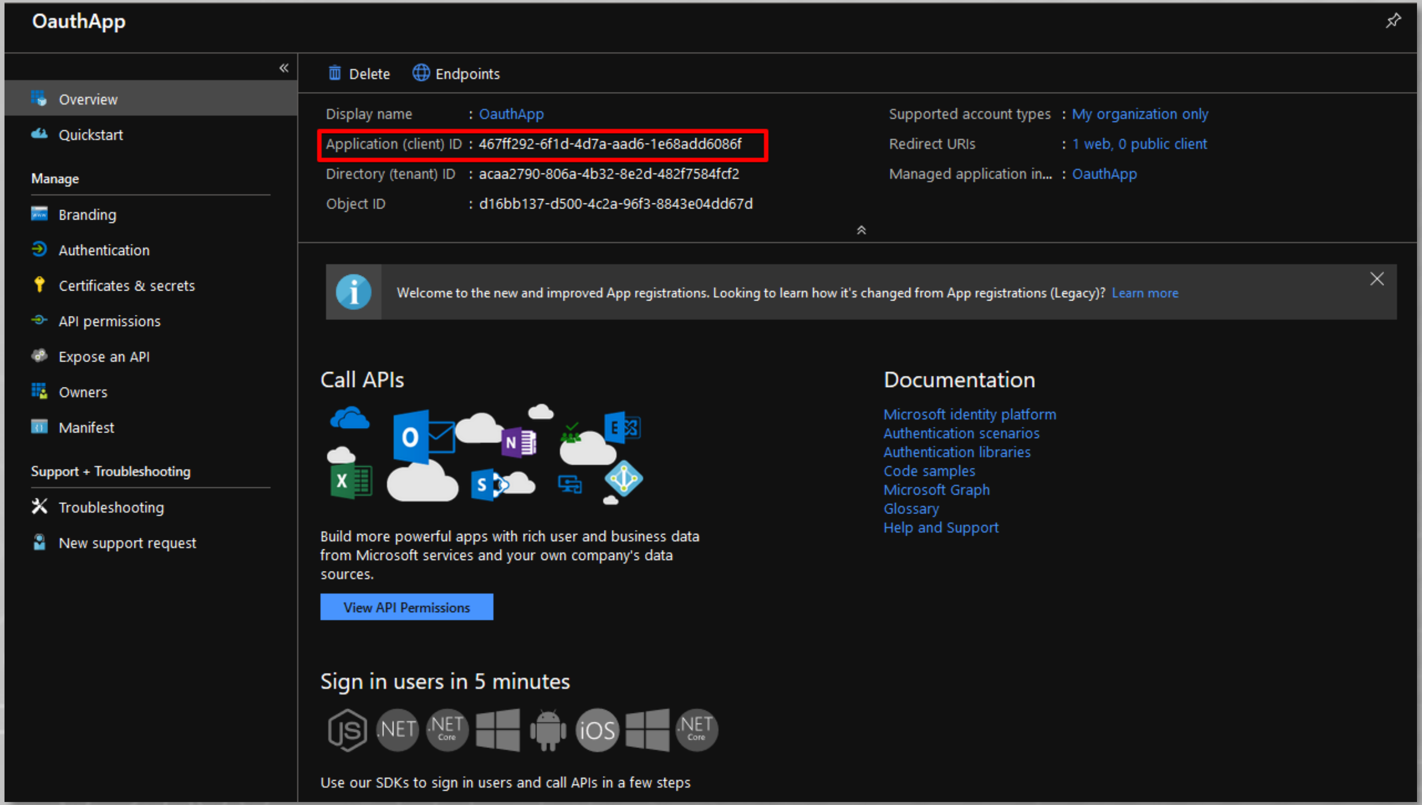

Authorization Server
Microsoft identity platform endpoint
REST
Bearer token
OAuth client (native or web app)
Resource server (REST API)
Resource owner (End-user)

AUTHENTICATE TO GRAPH API

Service Principal

Azure AD Application

- **Application (client) ID**
- Application (client) Secret
- Directory (tenant) ID
- Permissions
- User or Admin Consent
- Azure AD User Account Credentials


OauthApp

Delete Endpoints

Display name	: OauthApp
Application (client) ID	: 467ff292-6f1d-4d7a-aad6-1e68add6086f
Directory (tenant) ID	: acaa2790-806a-4b32-8e2d-482f7584fcf2
Object ID	: d16bb137-d500-4c2a-96f3-8843e04dd67d
Supported account types	: My organization only
Redirect URIs	: 1 web, 0 public client
Managed application in...	: OauthApp

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Call APIs

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

View API Permissions

Sign in users in 5 minutes

Use our SDKs to sign in users and call APIs in a few steps

Documentation

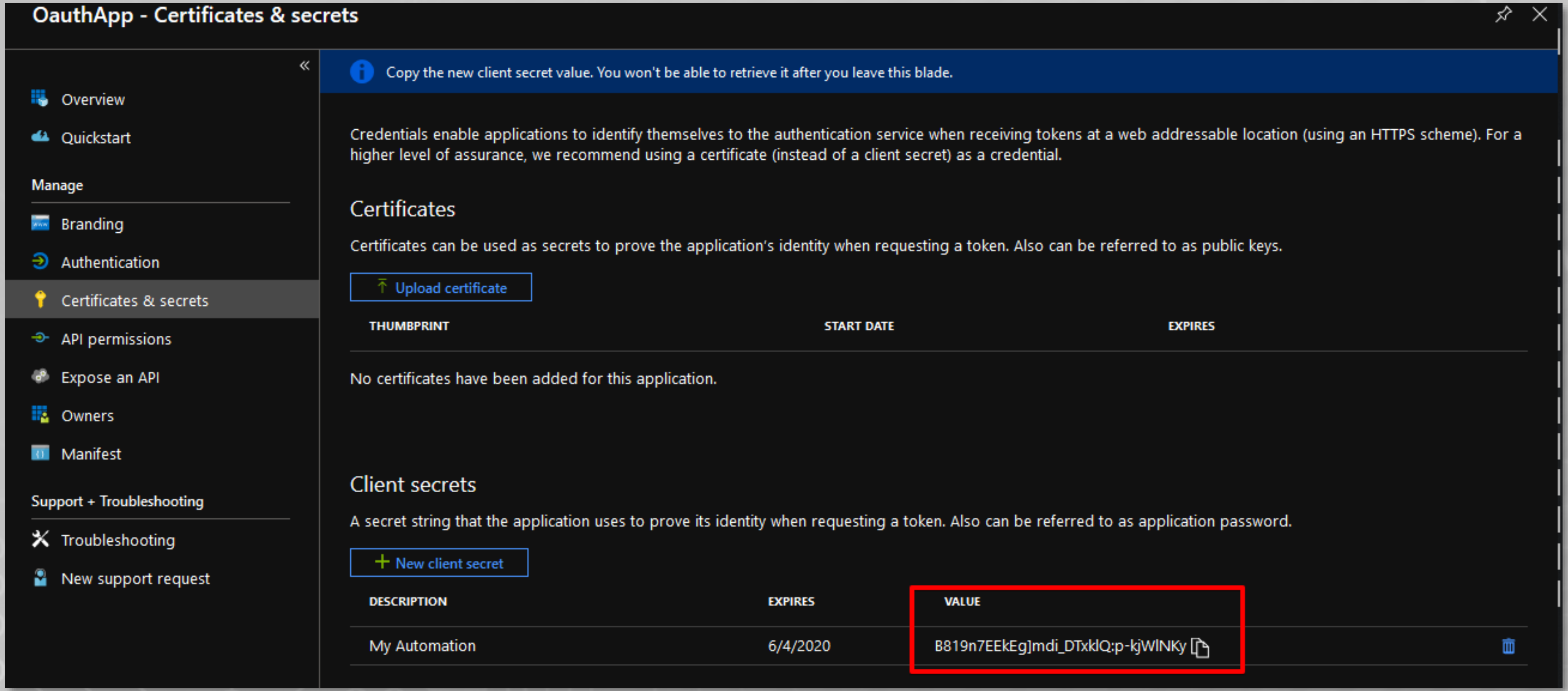
Microsoft identity platform
Authentication scenarios
Authentication libraries
Code samples
Microsoft Graph
Glossary
Help and Support

AUTHENTICATE TO GRAPH API

Service Principal

Azure AD Application

- Application (client) ID
- **Application (client) Secret**
- Directory (tenant) ID
- Permissions
- User or Admin Consent
- AAD User Account Credentials



OauthApp - Certificates & secrets

Copy the new client secret value. You won't be able to retrieve it after you leave this blade.

Credentials enable applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

THUMBPRINT	START DATE	EXPIRES

No certificates have been added for this application.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

New client secret

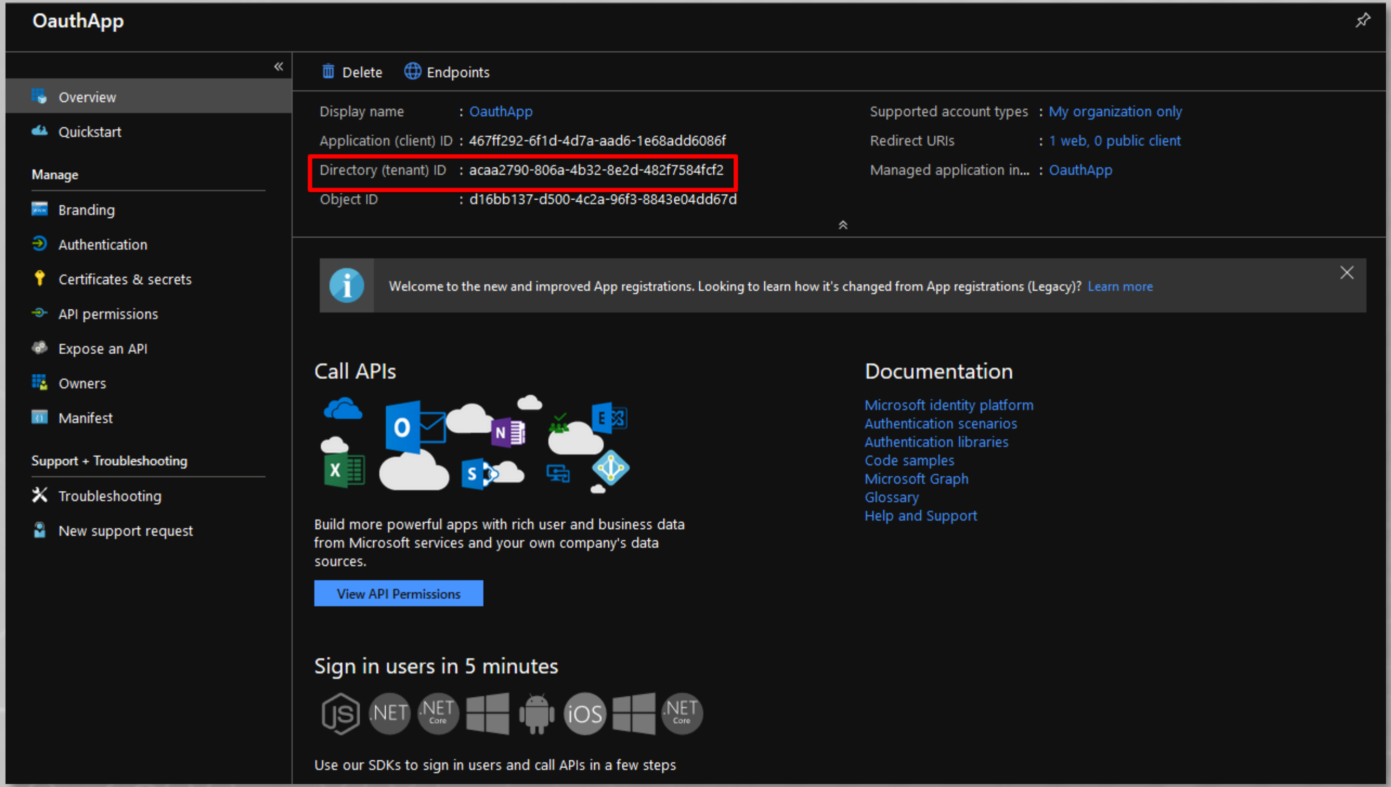
DESCRIPTION	EXPIRES	VALUE
My Automation	6/4/2020	B819n7EEkEg]mdi_DTxklQ:p-kjWlNKy

AUTHENTICATE TO GRAPH API

Service Principal

Azure AD Application

- Application (client) ID
- Application (client) Secret
- **Directory (tenant) ID**
- Permissions
- User or Admin Consent
- Azure AD User Account Credentials



Display name : OauthApp
Application (client) ID : 467ff292-6f1d-4d7a-aad6-1e68add6086f
Directory (tenant) ID : acaa2790-806a-4b32-8e2d-482f7584fcf2
Object ID : d16bb137-d500-4c2a-96f3-8843e04dd67d
Supported account types : My organization only
Redirect URIs : 1 web, 0 public client
Managed application in... : OauthApp

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

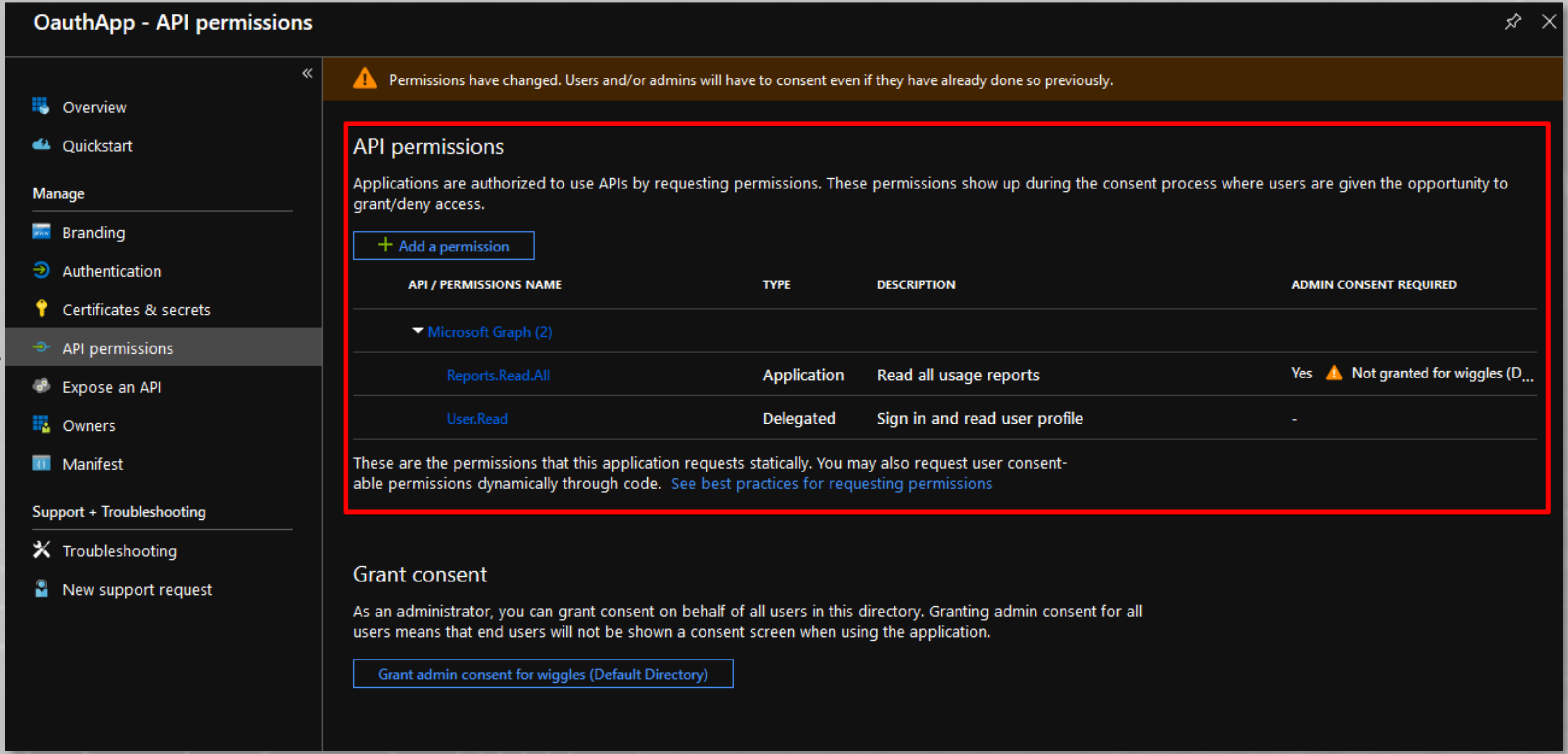
Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

Use our SDKs to sign in users and call APIs in a few steps

Service Principal

Azure AD Application

- Application (client) ID
- Application (client) Secret
- Directory (tenant) ID
- **Permissions**
- User or Admin Consent
- Azure AD User Account Credentials



Permissions have changed. Users and/or admins will have to consent even if they have already done so previously.

API permissions

Applications are authorized to use APIs by requesting permissions. These permissions show up during the consent process where users are given the opportunity to grant/deny access.

API / PERMISSIONS NAME	TYPE	DESCRIPTION	ADMIN CONSENT REQUIRED
Microsoft Graph (2)			
Reports.Read.All	Application	Read all usage reports	Yes ⚠ Not granted for wiggles (D...
User.Read	Delegated	Sign in and read user profile	-

These are the permissions that this application requests statically. You may also request user consent-able permissions dynamically through code. See best practices for requesting permissions

Grant consent

As an administrator, you can grant consent on behalf of all users in this directory. Granting admin consent for all users means that end users will not be shown a consent screen when using the application.

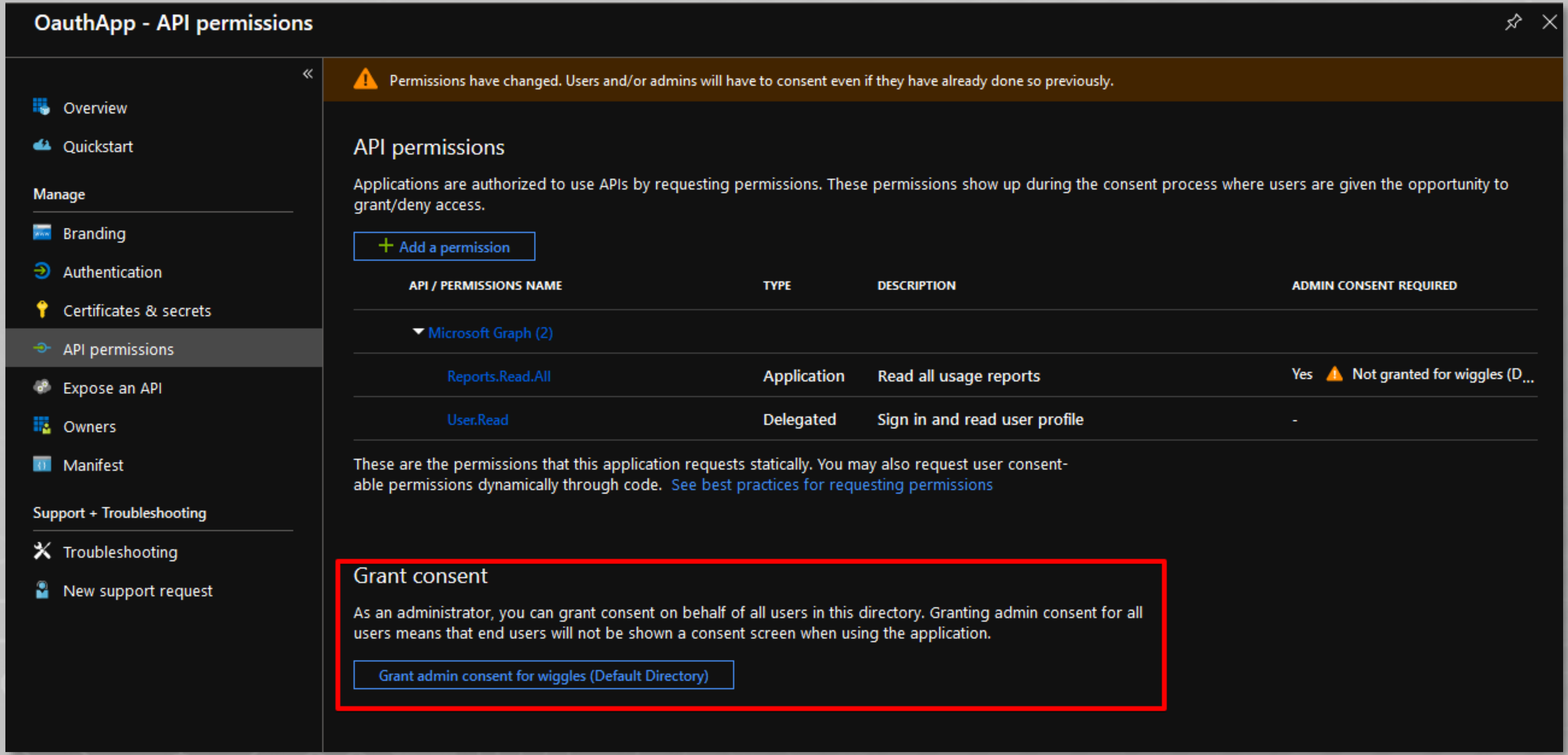
Grant admin consent for wiggles (Default Directory)

AUTHENTICATE TO GRAPH API

Service Principal

Azure AD Application

- Application (client) ID
- Application (client) Secret
- Directory (tenant) ID
- Permissions
- **User or Admin Consent**
- Azure AD User Account Credentials



Permissions have changed. Users and/or admins will have to consent even if they have already done so previously.

API permissions

Applications are authorized to use APIs by requesting permissions. These permissions show up during the consent process where users are given the opportunity to grant/deny access.

API / PERMISSIONS NAME	TYPE	DESCRIPTION	ADMIN CONSENT REQUIRED
Microsoft Graph (2)			
Reports.Read.All	Application	Read all usage reports	Yes ⚠ Not granted for wiggles (D...
User.Read	Delegated	Sign in and read user profile	-

These are the permissions that this application requests statically. You may also request user consent-able permissions dynamically through code. See best practices for requesting permissions

Grant consent

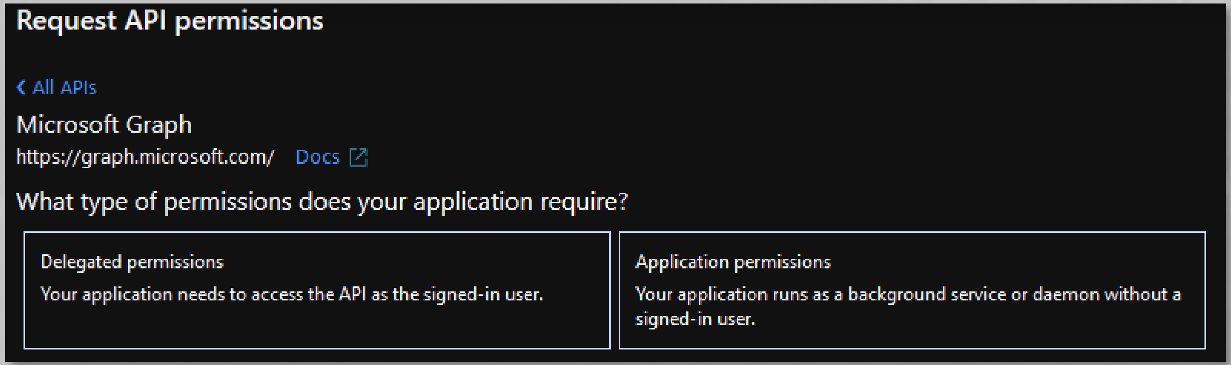
As an administrator, you can grant consent on behalf of all users in this directory. Granting admin consent for all users means that end users will not be shown a consent screen when using the application.

AUTHENTICATE TO GRAPH API

APP PERMISSIONS

Delegated
- Resources that need user context
    - E.g. Sending email or message to teams
- Needs user or admin consent
- Access the API as the signed-in user.

Application
- App-only – without a user
- Best for daemon apps
- Only admin can consent



Request API permissions

< All APIs

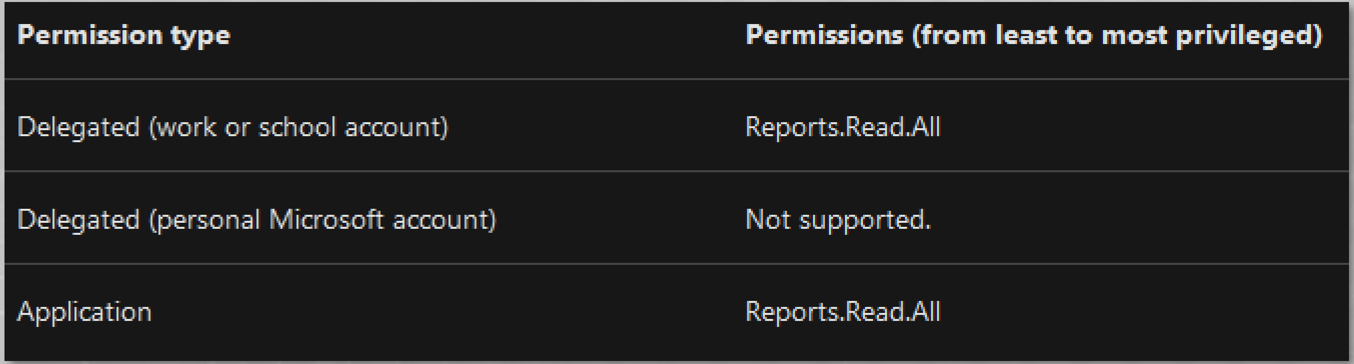
Microsoft Graph

https://graph.microsoft.com/ Docs

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.



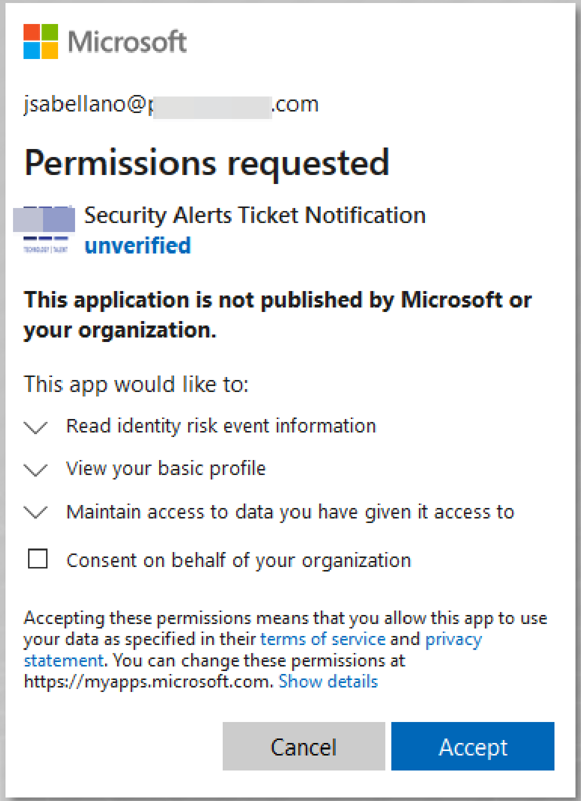
Permission type	Permissions (from least to most privileged)
Delegated (work or school account)	Reports.Read.All
Delegated (personal Microsoft account)	Not supported.
Application	Reports.Read.All

AUTHENTICATE TO GRAPH API

USER AUTH | ADMIN CONSENT

USER

- https://login.microsoftonline.com
  /{tenant}
  /oauth2
  /v2.0
  /authorize
  ?client_id={client ID}
  &response_type=code
  &redirect_uri={Redirect URL in the Azure app}
  &scope={permission.code} offline_access
  &prompt=consent



jsabellano@p[redacted].com

Permissions requested

Security Alerts Ticket Notification
unverified

This application is not published by Microsoft or your organization.

This app would like to:

- Read identity risk event information
- View your basic profile
- Maintain access to data you have given it access to

☐ Consent on behalf of your organization

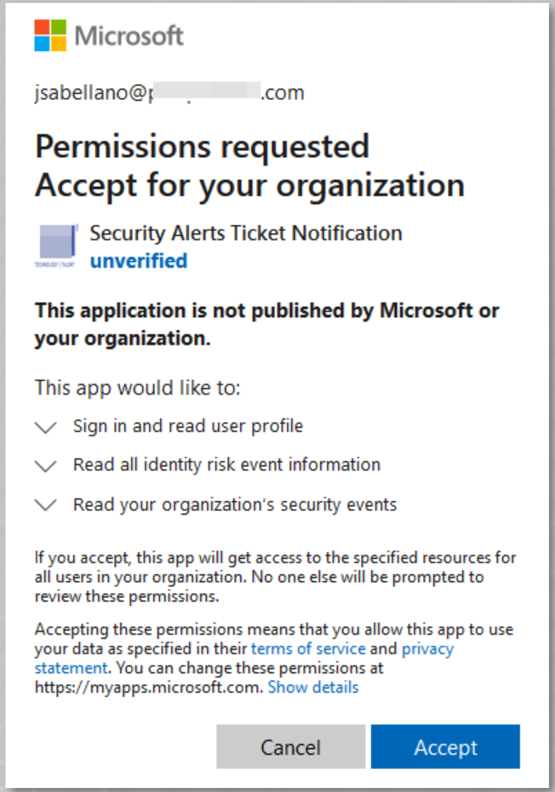
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://myapps.microsoft.com. Show details

AUTHENTICATE TO GRAPH API

USER AUTH | ADMIN CONSENT

ADMIN

- https://login.microsoftonline.com
  /{tenant}
  /adminconsent
  ?client_id={client ID}
  &redirect_uri={Redirect URL in the Azure app}



Microsoft

jsabellano@[redacted].com

Permissions requested
Accept for your organization

Security Alerts Ticket Notification
unverified

This application is not published by Microsoft or your organization.

This app would like to:

- Sign in and read user profile
- Read all identity risk event information
- Read your organization's security events

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

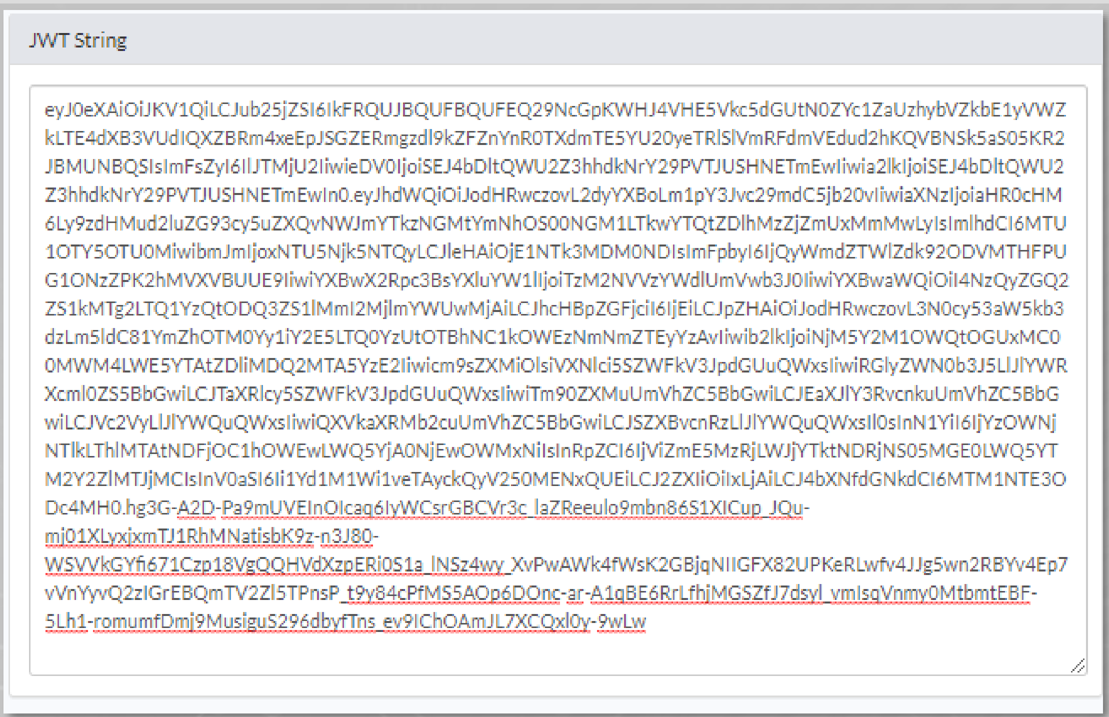
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://myapps.microsoft.com. Show details

AUTHENTICATE TO GRAPH API

Access Token

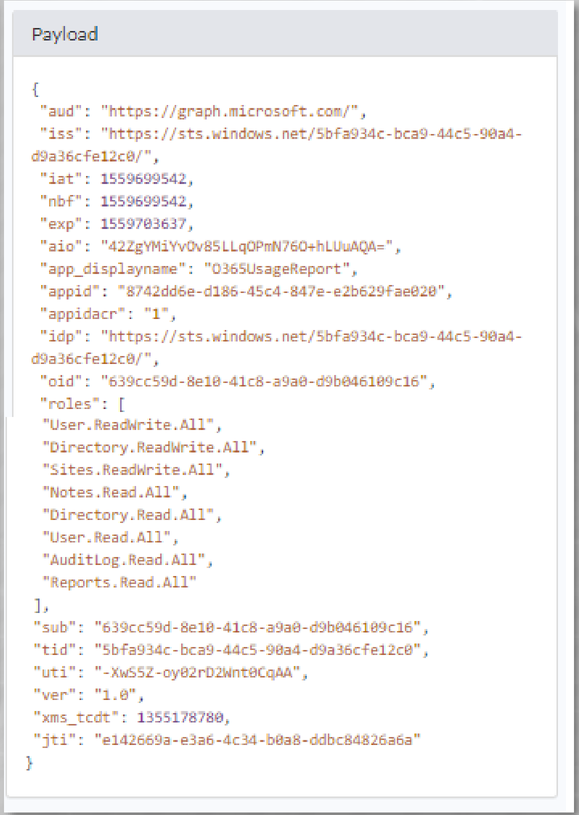
Access Token

- Needed to Access Graph API
- JSON Web Token (JWT)
- https://login.microsoftonline.com
  /{Tenant ID | “common”}
  /oauth2
  /v2.0
  /token



JWT String

```
eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUFEQ29NcGpKWHJ4VHE5Vkc5dGUtN0ZYc1ZaUzhybVZkbE1yVWZ
kLTE4dXB3VUdIQXZBRm4xeEpJSGZERmgzdl9kZFZnYnR0TXdmTE5YU20yeTRlSlVmRFdmVEdud2hKQVBNSk5aS05KR2
JBMUNBQSIsImFsZyI6IlJTMjU2IiwieDV0IjoiSEJ4bDltQWU2Z3hhdkNrY29PVTJUSHNETmEwIiwia2lkIjoiSEJ4bDltQWU2
Z3hhdkNrY29PVTJUSHNETmEwIn0.eyJhdWQiOiJodHRwczovL2dyYXBoLm1pY3Jvc29mdC5jb20vIiwiaXNzIjoiaHR0cHM
6Ly9zdHMud2luZG93cy5uZXQvNWJmYTkzNGMtYmNhOS00NGM1LTkwYTQtZDlhMzZjZmUxMmMwLyIsImlhdCI6MTU
1OTY5OTU0MiwibmJmIjoxNTU5Njk5NTQyLCJleHAiOjE1NTk3MDM0NDIsImFpbyI6IjQyWmdZTWlZdk92ODVMTHFPU
G1ONzZPK2hMVXVBUUE9IiwiYXBwX2Rpc3BsYXluYW1lIjoiTzM2NVVzYWdlUmVwb3J0IiwiYXBwaWQiOiI4NzQyZGQ2
ZS1kMTg2LTQ1YzQtODQ3ZS1lMmI2MjlmYWUwMjAiLCJhcHBpZGFjciI6IjEiLCJpZHAiOiJodHRwczovL3N0cy53aW5kb3
dzLm5ldC81YmZhOTM0Yy1iY2E5LTQ0YzUtOTBhNC1kOWEzNmNmZTEyYzAvIiwib2lkIjoiNjM5Y2M1OWQtOGUxMC0
0MWM4LWE5YTAtZDliMDQ2MTA5YzE2Iiwicm9sZXMiOlsiVXNlci5SZWFkV3JpdGUuQWxsIiwiRGlyZWN0b3J5LlJlYWR
Xcml0ZS5BbGwiLCJTaXRlcy5SZWFkV3JpdGUuQWxsIiwiTm90ZXMuUmVhZC5BbGwiLCJEaXJlY3RvcnkuUmVhZC5BbG
wiLCJVc2VyLlJlYWQuQWxsIiwiQXVkaXRMb2cuUmVhZC5BbGwiLCJSZXBvcnRzLlJlYWQuQWxsIl0sInN1YiI6IjYzOWNj
NTlkLThlMTAtNDFjOC1hOWEwLWQ5YjA0NjEwOWMxNiIsInRpZCI6IjViZmE5MzRjLWJjYTktNDRjNS05MGE0LWQ5YT
M2Y2ZlMTJjMCIsInV0aSI6Ii1Yd1M1Wi1veTAyckQyV250MENxQUEiLCJ2ZXIiOiIxLjAiLCJ4bXNfdGNkdCI6MTM1NTE3O
Dc4MH0.hg3G-A2D-Pa9mUVEInOIcaq6IyWCsrGBCVr3c_laZReeulo9mbn86S1XICup_JQu-
mj01XLyxjxmTJ1RhMNatisbK9z-n3J80-
WSVVkGYfi671Czp18VgQQHVdXzpERi0S1a_lNSz4wy_XvPwAWk4fWsK2GBjqNIIGFX82UPKeRLwfv4JJg5wn2RBYv4Ep7
vVnYyvQ2zIGrEBQmTV2Zl5TPnsP_t9y84cPfMS5AOp6DOnc-ar-A1qBE6RrLfhjMGSZfJ7dsyl_vmlsqVnmy0MtbmtEBF-
5Lh1-romumfDmj9MusiguS296dbyfTns_ev9IChOAmJL7XCQxl0y-9wLw
```



Payload

```
{
 "aud": "https://graph.microsoft.com/",
 "iss": "https://sts.windows.net/5bfa934c-bca9-44c5-90a4-
d9a36cfe12c0/",
 "iat": 1559699542,
 "nbf": 1559699542,
 "exp": 1559703637,
 "aio": "42ZgYMiYvOv85LLqOPmN76O+hLUuAQA=",
 "app_displayname": "O365UsageReport",
 "appid": "8742dd6e-d186-45c4-847e-e2b629fae020",
 "appidacr": "1",
 "idp": "https://sts.windows.net/5bfa934c-bca9-44c5-90a4-
d9a36cfe12c0/",
 "oid": "639cc59d-8e10-41c8-a9a0-d9b046109c16",
 "roles": [
  "User.ReadWrite.All",
  "Directory.ReadWrite.All",
  "Sites.ReadWrite.All",
  "Notes.Read.All",
  "Directory.Read.All",
  "User.Read.All",
  "AuditLog.Read.All",
  "Reports.Read.All"
 ],
 "sub": "639cc59d-8e10-41c8-a9a0-d9b046109c16",
 "tid": "5bfa934c-bca9-44c5-90a4-d9a36cfe12c0",
 "uti": "-XwS5Z-oy02rD2Wnt0CqAA",
 "ver": "1.0",
 "xms_tcdt": 1355178780,
 "jti": "e142669a-e3a6-4c34-b0a8-ddbc84826a6a"
}
```

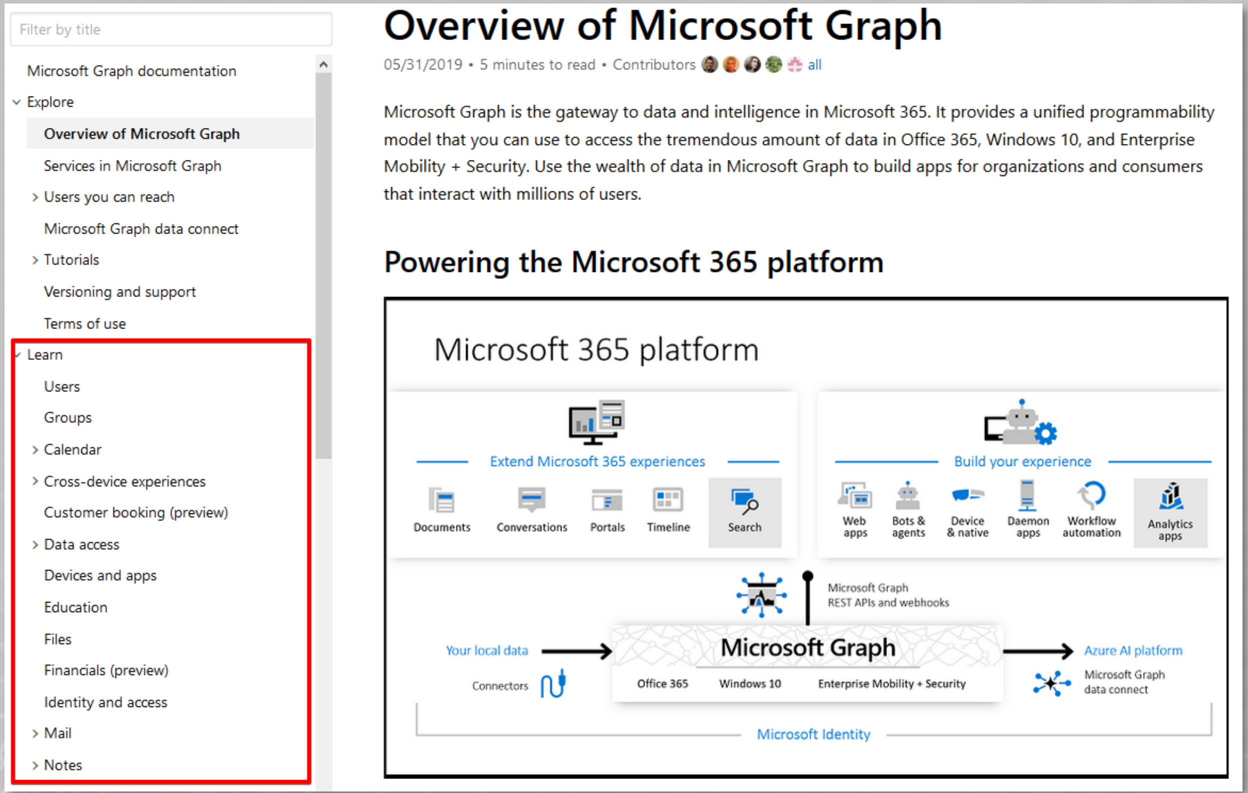
API Endpoint

How to Know the Endpoint?

- https://docs.microsoft.com
  /graph
  /overview

Examples:
https://graph.microsoft.com/v1.0

- Reports - /reports/
- Audit Logs - /auditLogs/
- Onedrive - /me/drive/
- Mail Drafts - /me/mailfolders('Drafts')/
- Sharepoint - /sites/



Microsoft Graph documentation
- Explore
  - Overview of Microsoft Graph
  - Services in Microsoft Graph
  - Users you can reach
  - Microsoft Graph data connect
  - Tutorials
  - Versioning and support
  - Terms of use
- Learn
  - Users
  - Groups
  - Calendar
  - Cross-device experiences
  - Customer booking (preview)
  - Data access
  - Devices and apps
  - Education
  - Files
  - Financials (preview)
  - Identity and access
  - Mail
  - Notes

Overview of Microsoft Graph

05/31/2019 • 5 minutes to read • Contributors all

Microsoft Graph is the gateway to data and intelligence in Microsoft 365. It provides a unified programmability model that you can use to access the tremendous amount of data in Office 365, Windows 10, and Enterprise Mobility + Security. Use the wealth of data in Microsoft Graph to build apps for organizations and consumers that interact with millions of users.

Powering the Microsoft 365 platform

AUTHENTICATE TO GRAPH API

GRANT TYPES

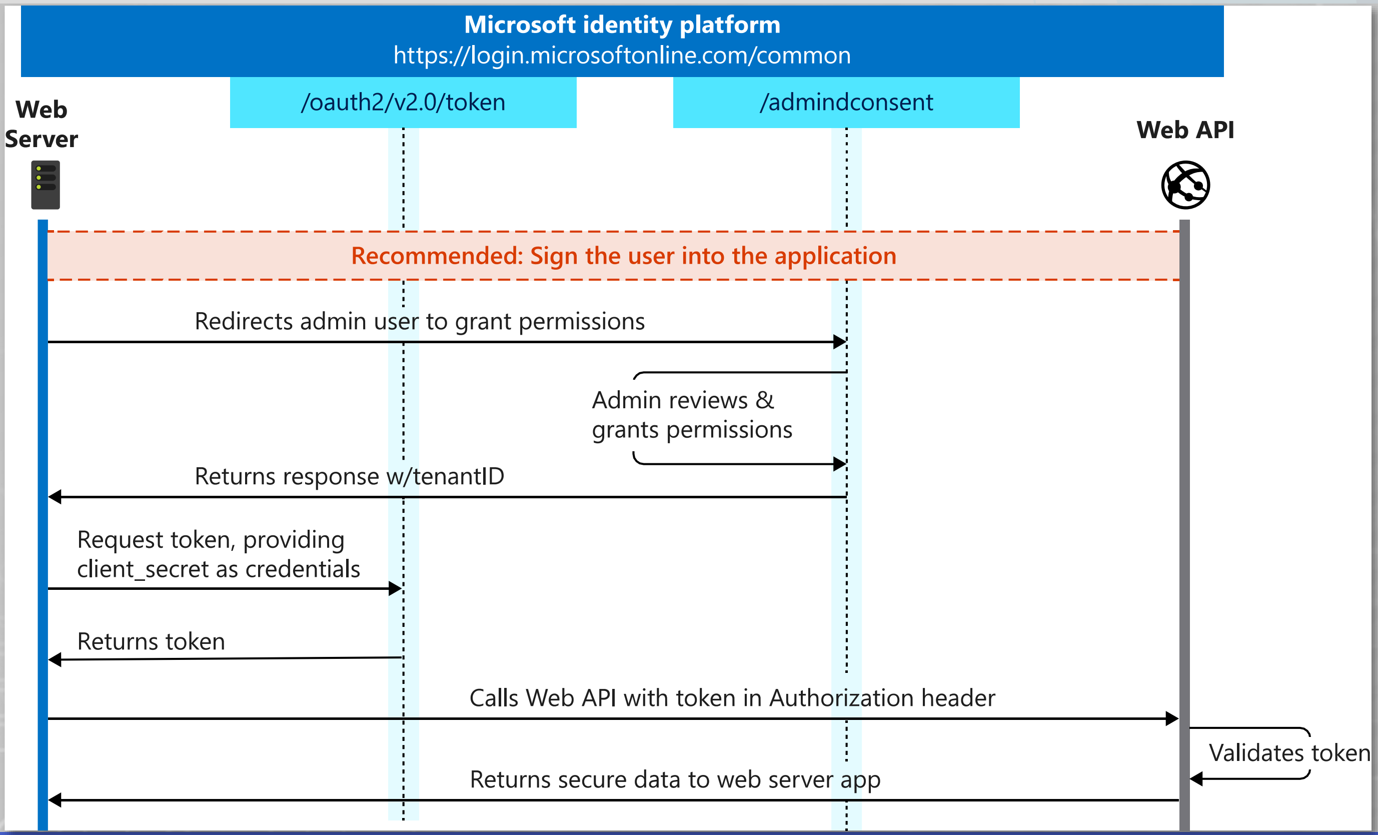
- **Client_Credentials**
- **Password**
- **Authorization_Code**
- **Device_Code**
- **Refresh_Token**

AUTHENTICATE TO GRAPH API

GRANT TYPES

Client_Credentials
- Best used for daemon apps
- Uses Azure AD App only for authentication
- Application permissions are used
- Token contains no user context
- Needs admin consent for permissions

Required POST Body:
- Grant_Type = Client_Credentials
- Client_ID = “{Azure App Client ID}”
- Client_Secret = “{Azure App Client Secret}”
- Scope = https://graph.microsoft.com/.default


Microsoft identity platform
https://login.microsoftonline.com/common
/oauth2/v2.0/token
/admindconsent
Web Server
Web API
Recommended: Sign the user into the application
Redirects admin user to grant permissions
Admin reviews & grants permissions
Returns response w/tenantID
Request token, providing client_secret as credentials
Returns token
Calls Web API with token in Authorization header
Validates token
Returns secure data to web server app

AUTHENTICATE TO GRAPH API

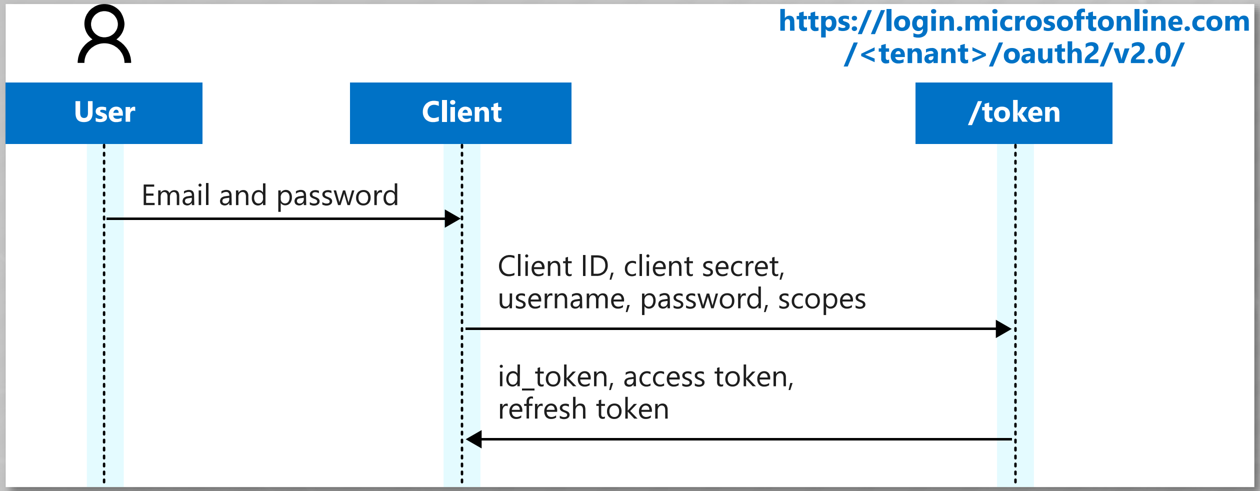
GRANT TYPES

Password

- Token contains user context
- Needs AAD user credentials
- Can create a refresh token (Scope: Offline_Access)
- Does not support web-based authentication (MFA)
- Not recommended by Microsoft

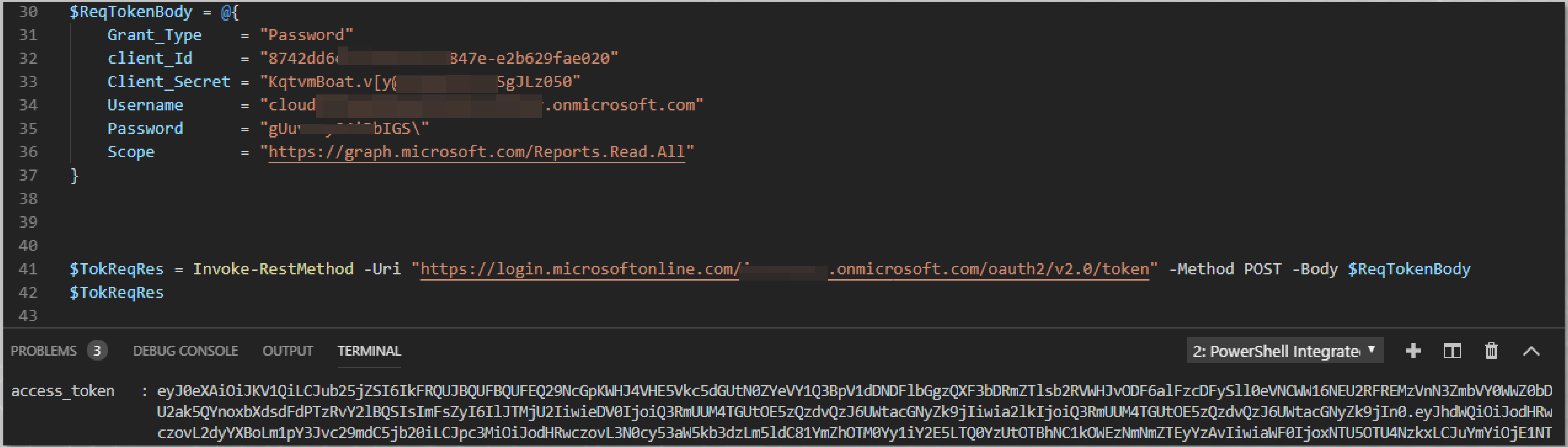
Required POST Body:

- Grant_Type = Password
- Client_ID = “{Azure App Client ID}”
- Client_Secret = “{Azure App Client Secret}”
- Username = “{Username}”
- Password = “{Password}”
- Scope = https://graph.microsoft.com/{permission.code}


https://login.microsoftonline.com
/<tenant>/oauth2/v2.0/
User
Client
/token
Email and password
Client ID, client secret, username, password, scopes
id_token, access token, refresh token

PASSWORD - STITCHING THE COMPONENTS

REQUESTING TOKEN



```
$ReqTokenBody = @{
    Grant_Type    = "Password"
    client_Id     = "8742dd6[redacted]847e-e2b629fae020"
    Client_Secret = "KqtvmBoat.v[y([redacted]5gJLz050"
    Username      = "cloud[redacted].onmicrosoft.com"
    Password      = "gUu[redacted]bIGS\"
    Scope         = "https://graph.microsoft.com/Reports.Read.All"
}



$TokReqRes = Invoke-RestMethod -Uri "https://login.microsoftonline.com/[redacted].onmicrosoft.com/oauth2/v2.0/token" -Method POST -Body $ReqTokenBody
$TokReqRes
```

```
access_token   : eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUFEQ29NcGpKWHJ4VHE5Vkc5dGUtN0ZYeVY1Q3BpV1dDNDFlbGgzQXF3bDRmZTlsb2RVWHJvODF6alFzcDFySll0eVNCWW16NEU2RFREMzVnN3ZmbVY0WWZ0bD
                 U2ak5QYnoxbXdsdFdPTzRvY2lBQSIsImFsZyI6IlJTMjU2IiwieDV0IjoiQ3RmUUM4TGUtOE5zQzdvQzJ6UWtacGNyZk9jIiwia2lkIjoiQ3RmUUM4TGUtOE5zQzdvQzJ6UWtacGNyZk9jIn0.eyJhdWQiOiJodHRw
                 czovL2dyYXBoLm1pY3Jvc29mdC5jb20iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC81YmZhOTM0Yy1iY2E5LTQ0YzUtOTBhNC1kOWEzNmNmZTEyYzAvIiwiaWF0IjoxNTU5OTU4NzkxLCJuYmYiOjE1NT
```

AUTHENTICATE TO GRAPH API

GRANT TYPES

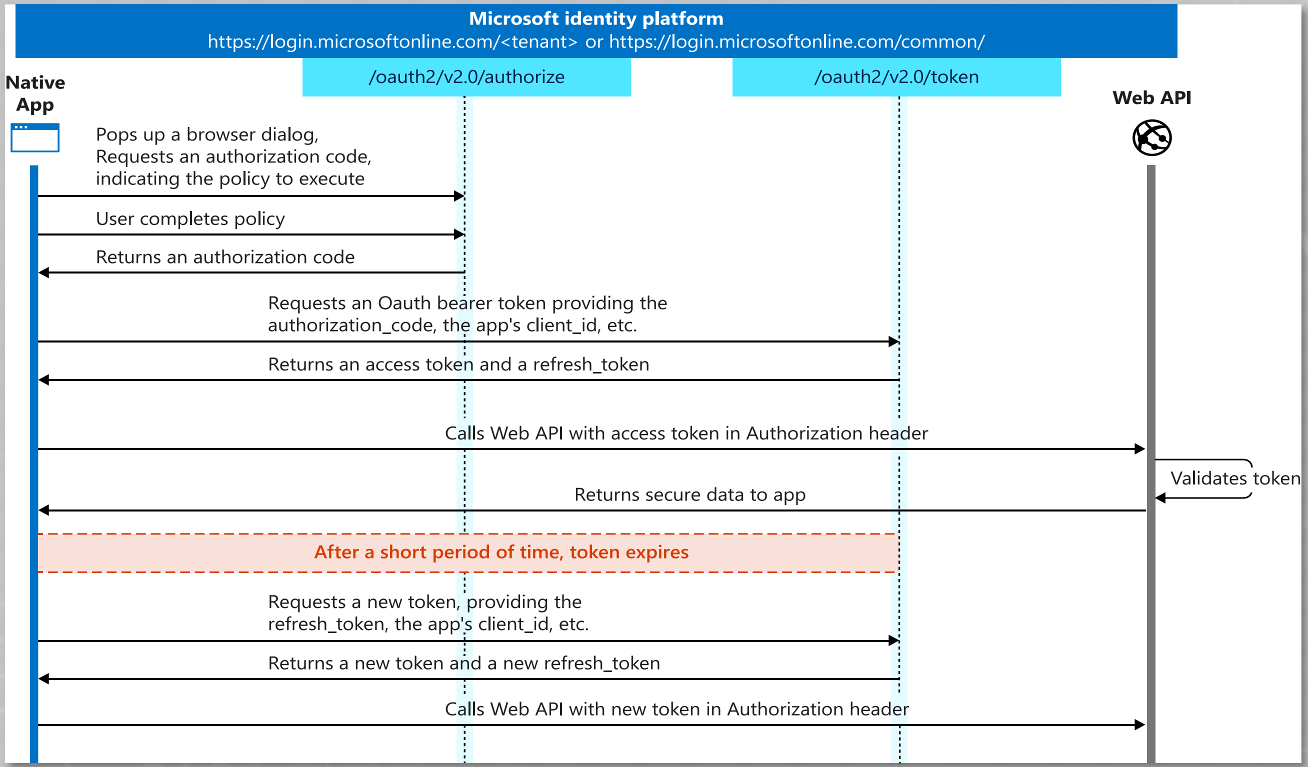
Authorization_Code

Auth Request URL:

- https://login.microsoftonline.com/{tenantID}/oauth2
  /v2.0/authorize
  ?client_id={clientID}
  &redirect_uri={RedirectURI}
  &scope={Scope}
  &response_type=code

Token Request POST Body:

- Grant_Type = Authorization_Code
- Client_ID = “{ClientID}”
- Client_Secret = “{Client Secret}”
- Scope = https://graph.microsoft.com/{permission.code}
- Code = “{Authorization Code}”
- Redirect_Uri = {Redirect URI in the Azure App}


Microsoft identity platform
https://login.microsoftonline.com/<tenant> or https://login.microsoftonline.com/common/
/oauth2/v2.0/authorize
/oauth2/v2.0/token
Native App
Web API
Pops up a browser dialog, Requests an authorization code, indicating the policy to execute
User completes policy
Returns an authorization code
Requests an Oauth bearer token providing the authorization_code, the app's client_id, etc.
Returns an access token and a refresh_token
Calls Web API with access token in Authorization header
Validates token
Returns secure data to app
After a short period of time, token expires
Requests a new token, providing the refresh_token, the app's client_id, etc.
Returns a new token and a new refresh_token
Calls Web API with new token in Authorization header

AUTHENTICATE TO GRAPH API

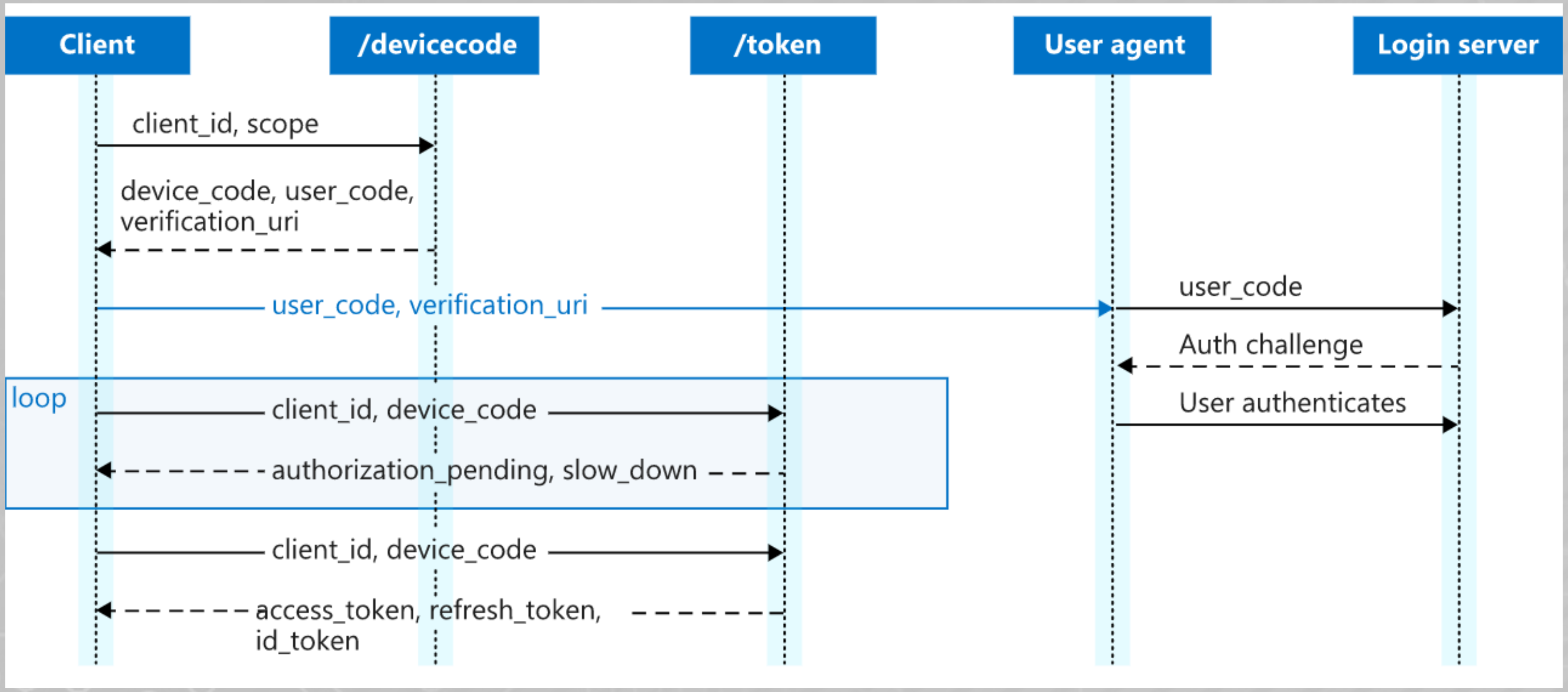
GRANT TYPES

Device_Code

Auth Request POST Body:
- Client_ID = “{Azure App Client ID}”
- Code = “{Device Code}”

Send to:
- https://login.microsoftonline.com/{tenant}
/oauth2/v2.0/devicecode

Token Request POST Body:
- Grant_Type = “Device_Code”
- Client_ID = “{Azure App Client ID}”
- Code = “{Device Code}”



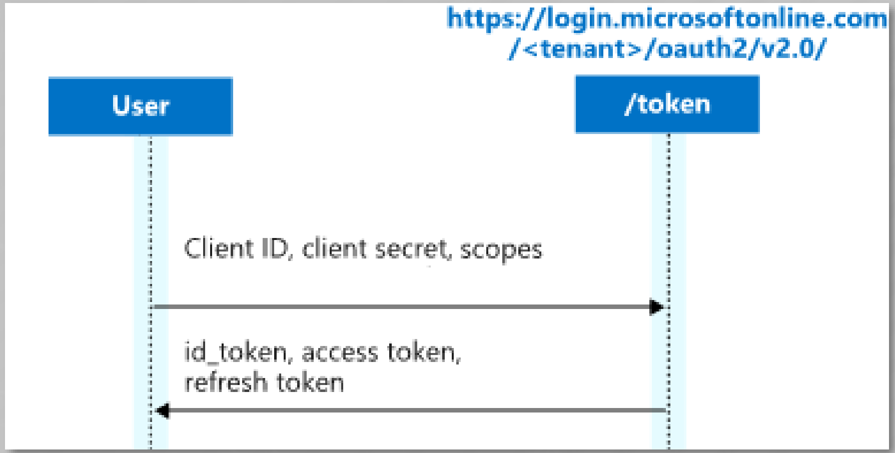
AUTHENTICATE TO GRAPH API

GRANT TYPES

Refresh_Token

Required POST Body:

- Grant_Type = Refresh_Token
- Client_ID = “{Azure App Client ID}”
- Client_Secret = “{Azure App Client Secret}”
- Refresh_Token = “{Refresh Token}”
- Redirect_URI = “{Redirect URI}”
- Scope = https://graph.microsoft.com/{permission.code}



GRAPH API INTERACTIONS

SENDING REQUESTS

- **Query Parameters**
  - Filter results
- **Paging Results**
  - Dealing with huge datasets
- **Batch Processing**
  - Send multiple requests at ones

Graph API Endpoint

- Where you will send your request
- Root URL - https://graph.microsoft.com/v1.0/
- Root URL - https://graph.microsoft.com/beta/
- https://graph.Microsoft.com
  /{version}
  /{resource}
  /{id}
  /{property}
  /{query-parameters}

GRAPH API INTERACTIONS

FILTERING QUERIES

Name	Description	Example
$count	Retrieves the total count of matching resources.	/me/messages?$top=2&$count=true
$expand	Retrieves related resources.	/groups?$expand=members
$filter	Filters results (rows).	/users?$filter=startswith(givenName,'J')
$format	Returns the results in the specified media format.	/users?$format=json
$orderby	Orders results.	/users?$orderby=displayName desc
$search	Returns results based on search criteria. Currently supported on messages and person collections.	/me/messages?$search=pizza
$select	Filters properties (columns).	/users?$select=givenName,surname
$skip	Indexes into a result set. Also used by some APIs to implement paging and can be used together with $top to manually page results.	/me/messages?$skip=11
$top	Sets the page size of results.	/users?$top=2

Microsoft Graph SDK

PowerShell Graph SDK

Key Benefits

- Available from the PowerShell Gallery
  - *Install-Module Microsoft.Graph*
- 3219 different commands currently
- Supports PowerShell 5.1+ (Windows and Core)
- No requirement to create an Azure AD Application, the SDK will use it's own baked-in application
- Supports device code auth, so not reliant on Windows Forms or WPF to display sign-in/MFA prompts
- Has built in authentication methods such as refreshing OAuth tokens automatically etc.
- Prefix of mg (ex: get-mguser, get-mgusermessage)

Questions / Open Discussion