ADMINISTRATION OFFICE 365

API / POWERSHELL

TANET Margaux

BARILLY Dylan

Introduction :

Microsoft 365 utilise des rôles d'administrateur pour attribuer des fonctions d'administration spécifiques aux utilisateurs. Les administrateurs gèrent les rôles d'administrateur Microsoft 365 à l'aide du Centre d'administration Microsoft 365 ou du Windows PowerShell.

Une API (application programming interface ou interface de programmation d'application) est une interface logicielle qui permet de connecter un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

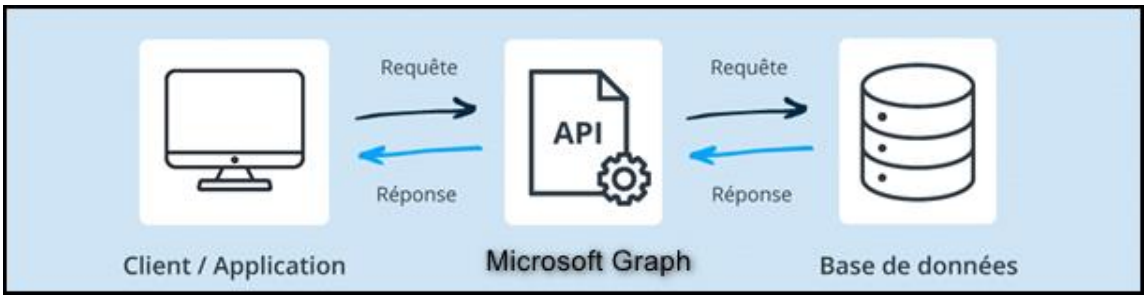


Schéma du fonctionnement d'une API

Microsoft Graph est une API web RESTful qui permet d'accéder aux ressources de service Cloud Microsoft. Il est possible d'effectuer des requêtes dans l'API Microsoft Graph.

Outils d'interaction avec Microsoft Graph

L'afficheur Graph est un outil web que l'on peut utiliser pour créer et tester des requêtes à l'aide des API Microsoft Graph.

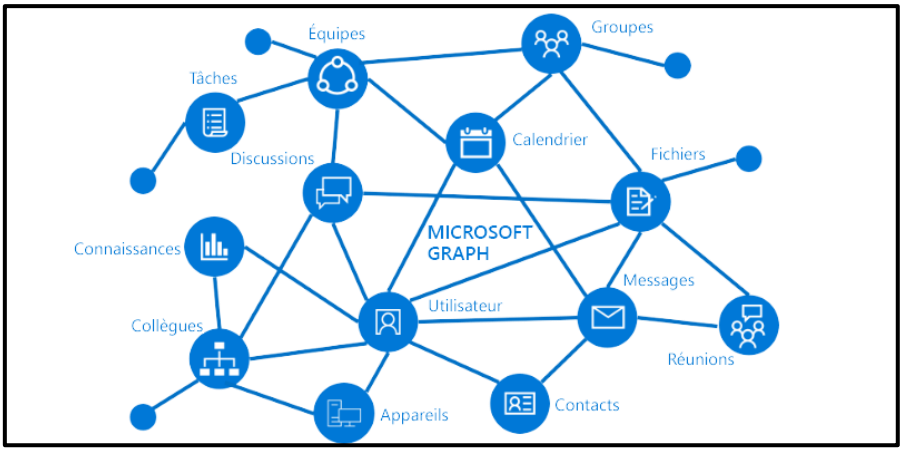


Schéma représentant le type d'informations accessibles via Microsoft Graph

Objectif :

Ce document présente l'attribution de rôles d'administrateurs à un utilisateur avec PowerShell. Par la suite, nous travaillerons avec l'API office 365 pour réaliser des tentatives d'obtentions d'informations (des requêtes) grâce au compte crée ultérieurement.

Pré requis :

- Commencer par la création d'un compte utilisateur ayant les droits administrateurs en prénom.nom pour être relié au compte Mewo qui a la même nomenclature au début.

Exemple : compte ayant les droits admin : margaux.tanet
Compte Mewo sur pc : prenom.nom@mewo-campus.fr

- Pour le bon fonctionnement lors des commandes, l'installation de **PowerShell version 7.0** est nécessaire. Lancer PowerShell en tant qu'administrateur.

Table des matières

I- Attribuer des rôles d'administrateur sous Windows PowerShell

Le module PowerShell Microsoft Graph permet d'attribuer des rôles d'administrateur. Les commandes suivantes permettent d'interagir avec Microsoft Graph qui est une API et permet d'accéder aux données et aux fonctionnalités de Microsoft 365.

➢ Cette commande installe le module Microsoft Graph sous l'utilisateur actuel :

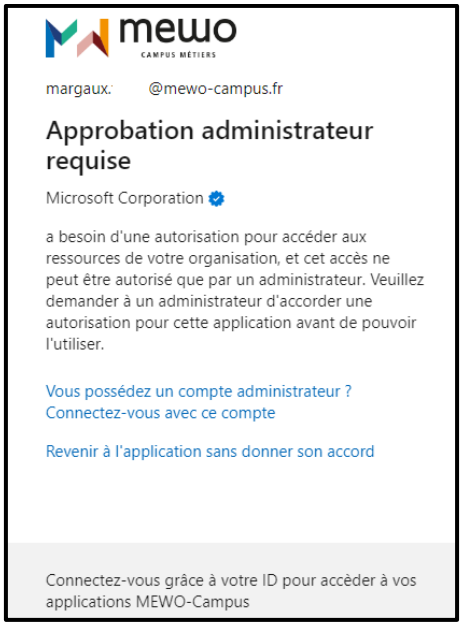
- *Install-Module Microsoft.Graph -Scope CurrentUser*

➢ Continuer avec ces deux commandes :

- *Import-Module Microsoft.Graph.Identity.DirectoryManagement*
- *Connect-MgGraph -Scopes 'User.Read.All', 'RoleManagement.ReadWrite.Directory'*

La troisième commande est utilisée pour établir la connexion qui demande des autorisations spécifiques.

Une redirection est faite vers une autre page. Elle nous informe que nous ne disposons pas des droits nécessaires pour se connecter. En effet, l'utilisateur crée n'a aucun droit d'accès aux ressources d'Office 365.



➢ Voici la quatrième commande :

- *Get-MgUser -All | Format-List ID, DisplayName, Mail, UserPrincipalName*

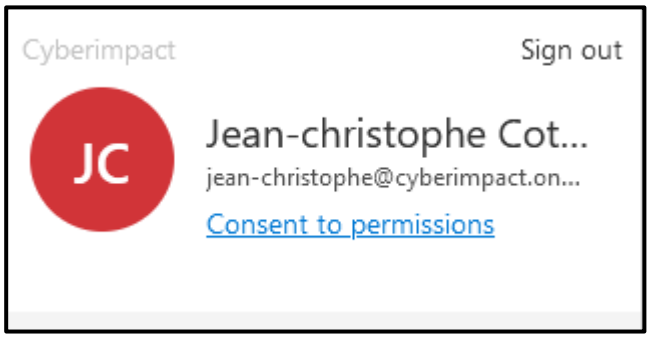
Get-MgUser permet d'obtenir l'ID d'objet de l'utilisateur pour avoir la liste de tous les utilisateurs. L'ID d'objet d'un utilisateur est l'identifiant unique qui est attribué à un utilisateur dans l'annuaire Microsoft Azure.

```
PS C:\Program Files\PowerShell\7> Get-MgUser -All | Format-List ID, DisplayName, Mail, UserPrincipalName
Get-MgUser_List: Authentication needed. Please call Connect-MgGraph.
```

Cette commande aurait marché si l'utilisateur avait les droits administrateurs. C'est ce qui est décrit lors de l'erreur « Authentification needed. ».

II- Requêtes API pour obtenir des informations d'administration avec Microsoft Graph

Il est possible d'effectuer des requêtes dans l'API Microsoft Graph comme notre utilsateur a été créer :



Informations de commandes pour les prochaines requêtes : POST modifie et GET récupère les informations.

1) Lister tous les utilisateurs

La requête : GET *https://graph.microsoft.com/v1.0/users*

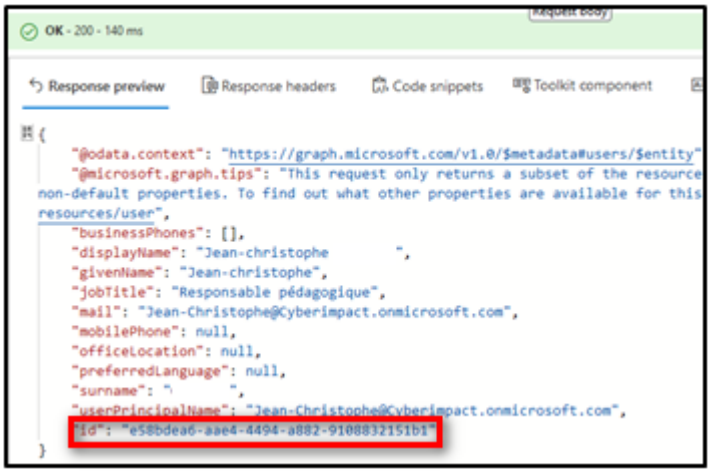
On obtient les informations de base de tous les utilisateurs dans le tenant :
Le nom, le prénom, l'ID, le mail, la profession, ...

2) Obtenir les détails d'un utilisateur spécifique

Cela permet de consulter les informations détaillées d'un utilisateur en particulier. Il faut remplacer ***id-user*** par un des id-user trouver dans la requête juste au-dessus. (Exemple : e58bdea6-aae4-4494-a882-9108832151b1)

La requête : GET *https://graph.microsoft.com/v1.0/users/{user-id}*

A modifier *https://graph.microsoft.com/v1.0/users/{e58bdea6-aae4-4494-a882-9108832151b1}*



3) Lister les groupes de sécurité

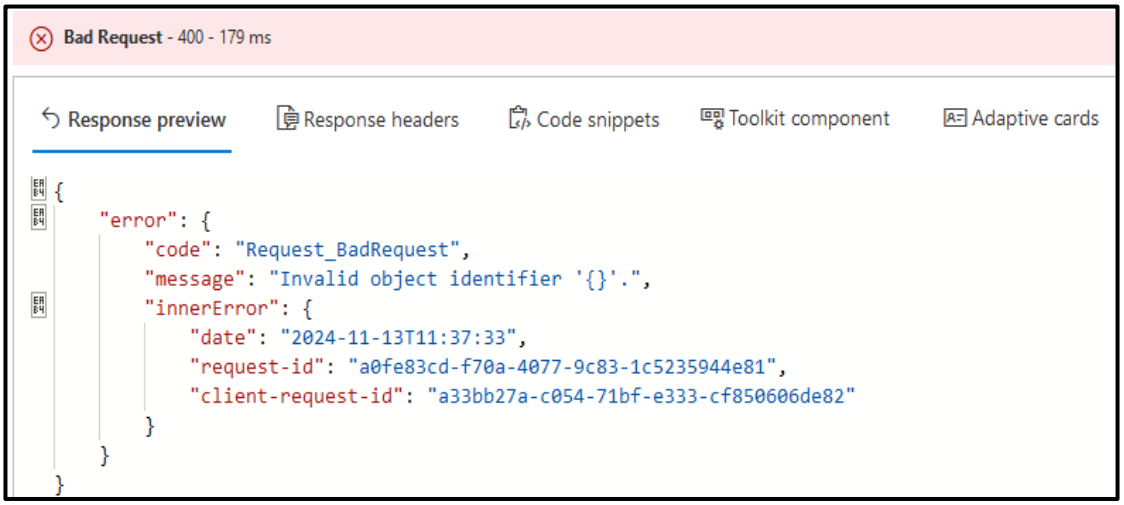
Cela permet d'obtenir la liste des groupes de sécurité et d'audit des permissions assignées à chaque groupe.

La requête : GET *https://graph.microsoft.com/v1.0/groups?$filter=securityEnabled* eq true

4) Lister les membres d'un groupe

La requête : GET *https://graph.microsoft.com/v1.0/groups/{group-id} /members*

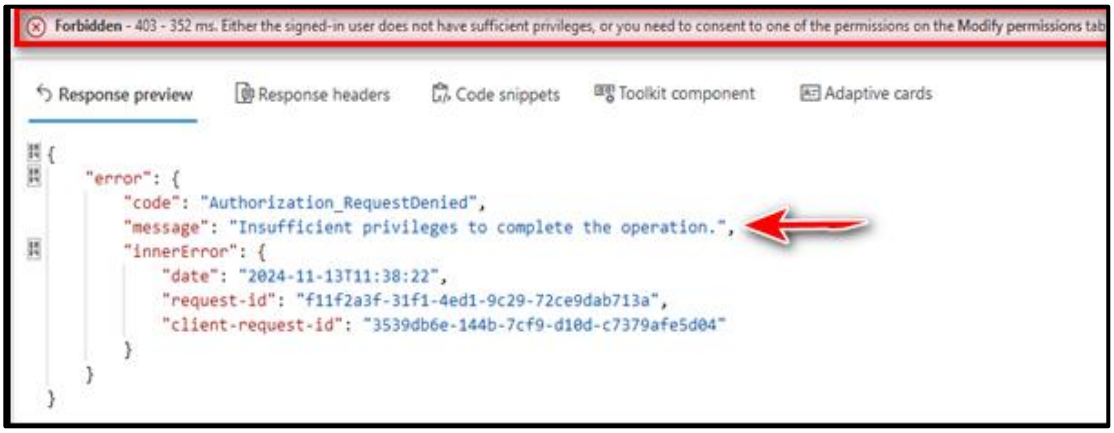
La requête ressort une erreur car les droits sont insuffisants pour créer un groupe comme l'utilisateur a juste les droits en tant que lecteur :



5) Lister les rôles d'administrations assignés

La requête retourne les rôles d'administration assignés dans Azure AD, permettant de vérifier les privilèges des utilisateurs.

Une erreur ressort, car c'est un rôle de lecteur dont fait partie Jean Christophe et manque de droits suffisants donc il est impossible de voir les rôles d'administrations.



6) Lister les utilisateurs assignés à un rôle spécifique

Cela permet d'identifier les utilisateurs jouant un rôle d'administration précis.

La requête : *GET https://graph.microsoft.com/v1.0/directoryRoles/{role-id} /members*

Cependant, une erreur survient, car il faut remplacer rôle-id par l'ID du rôle que l'on doit avoir eu dans la requête juste avant or, aucun rôle n'a été trouvé comme l'utilisateur ne disposait pas de droits suffisants et était en tant qu'utilisateur lecteur.

7) Obtenir les licences assignées à un utilisateur

Cette requête donne la possibilité de voir les licences Office 365 affectés à chaque utilisateur.

La requête : GET *https://graph.microsoft.com/v1.0/users/{user-id} /licenseDetails*

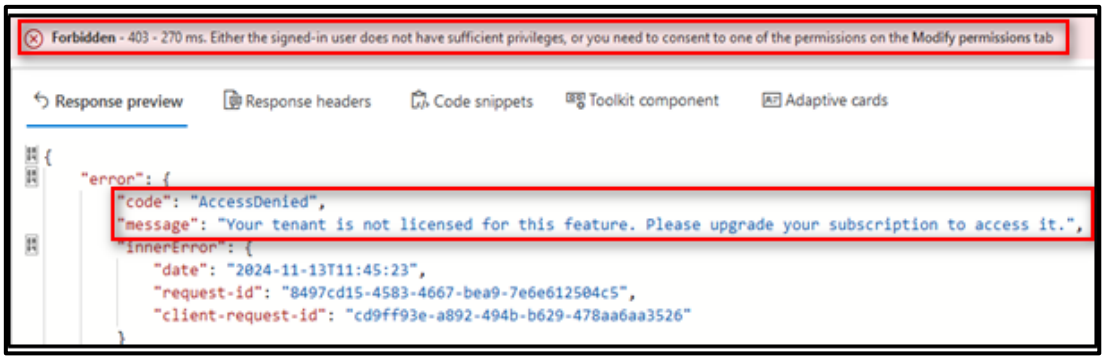
Il faut changer et remplacer **l'user-*id*** par l'ID de l'utilisateur que l'on a obtenu au départ (requête 2) : GET *https://graph.microsoft.com/v1.0/users/{e58bdea6-aae4-4494-a882-9108832151b1} /licenseDetails*

Chaque licence est identifiée grâce au « skuId » qui correspond à chaque fois à un produit bien précis.

8) Lister les connexions suspectes

La requête : GET *https://graph.microsoft.com/v1.0/identityProtection/riskyUsers*

Une erreur indique que l'utilisateur actuel a un accès refusé et ne possède pas les autorisations suffisantes pour accéder à cette liste. Il demande à l'utilisateur de mettre à niveau les permissions pour pouvoir continuer sa requête.

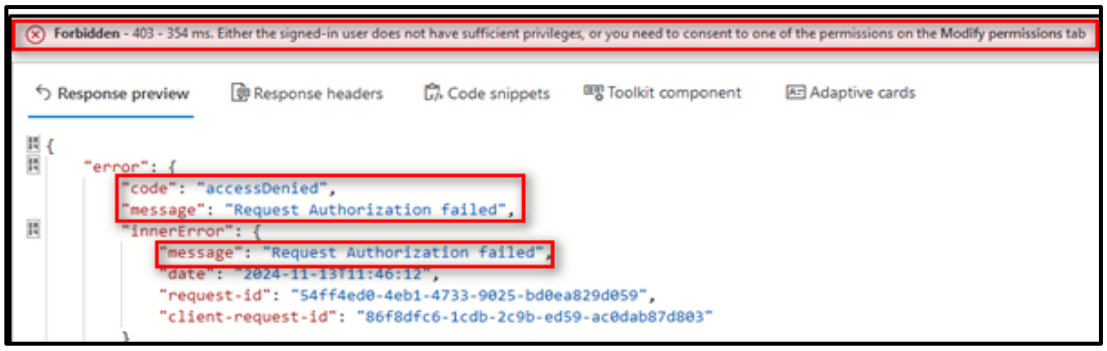


9) Vérifier les paramètres de sécurité MFA d'un utilisateur

La requête : GET *https://graph.microsoft.com/beta/users/{user-id} /authentication/methods*

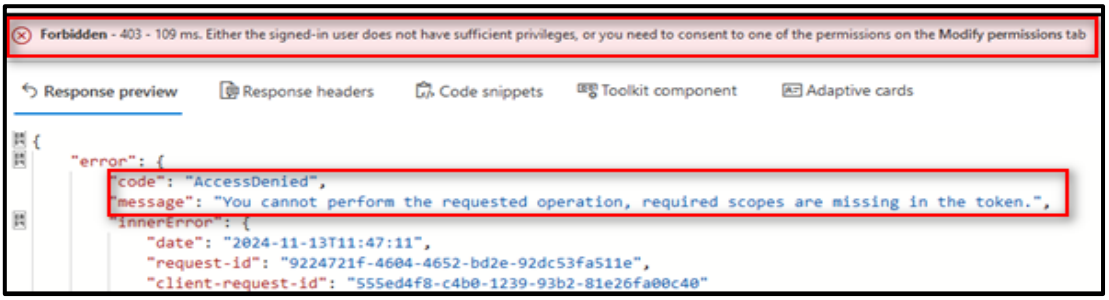
Il faut penser à changer user-id par l'ID de l'utilisateur de la première requête (requête 2) : *GET https://graph.microsoft.com/beta/users/{e58bdea6-aae4-4494-a882-9108832151b1} /authentication/methods*

Cela retourne des erreurs : accès refusé et la demande d'autorisation a échoué. L'utilisateur actuel ne possède pas les autorisations suffisantes pour accéder à cette liste. Il demande à l'utilisateur de mettre à niveau les permissions pour pouvoir continuer sa requête.



10) Obtenir les paramètres de stratégie de Conditional Access

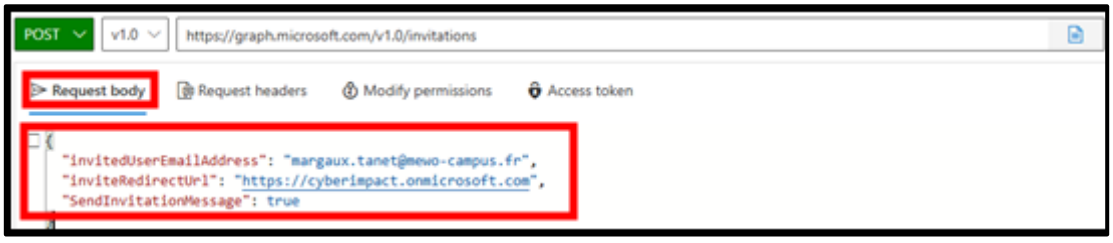
Une liste de stratégie d'accès conditionnel est une instruction *if then* d'affectations et de contrôle d'accès. Elles contrôlent les équipements qui sont enregistrés et conformes à la solution d'administration qui fonctionne avec la suite Office 365.
La requête : GET *https://graph.microsoft.com/v1.0/identity/conditionalAccess/policies*
Mais la même erreur se reproduit, car l'utilisateur n'a pas les permissions requises et l'accès est refusé.



III- Création d'un invité sur le tenant Microsoft

La création d'un utilisateur est requise pour pouvoir créer une réunion. En effet, un « token » est requis et l'utilisateur doit être déclarer sous Office 365. Pour cela, nous allons utiliser l'API Microsoft Graph pour créer une invitation.

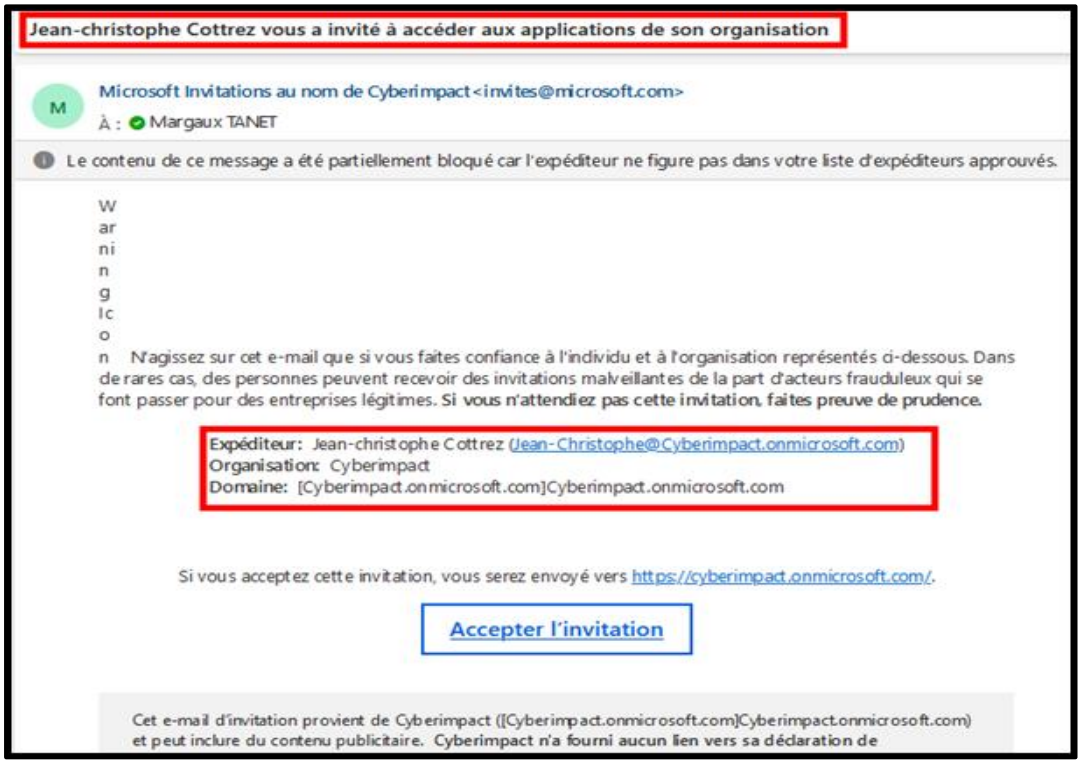
L'exemple suivant montre une demande d'ajout et d'invitation d'un utilisateur invité :



L'exemple suivant illustre la réponse :

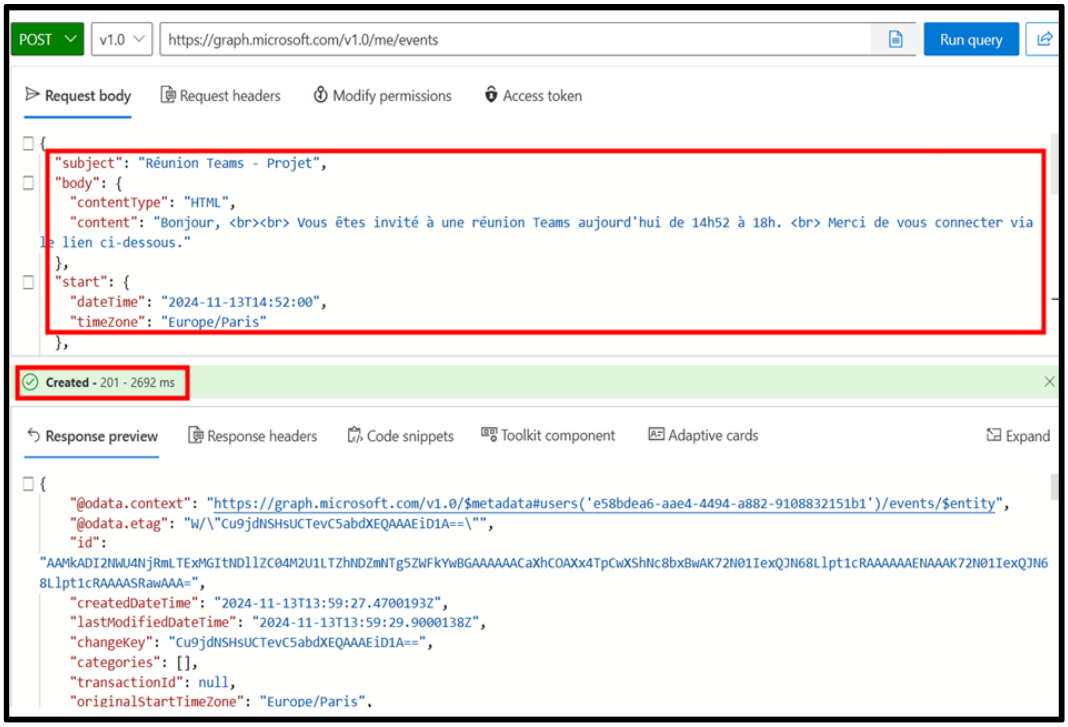

```
{
    "businessPhones": [],
    "displayName": "Dylan BARILLY",
    "givenName": null,
    "jobTitle": null,
    "mail": "dylan.barilly@mewo-campus.fr",
    "mobilePhone": null,
    "officeLocation": null,
    "preferredLanguage": null,
    "surname": null,
    "userPrincipalName": "dylan.barilly_mewo-campus.fr#EXT#@Cyberimpact.onmicrosoft.com",
    "id": "6cdf7e81-6893-49eb-b746-9f95c54293f2"
},
```

Lorsque l'invitation est créée, Microsoft Graph envoie automatiquement un mail d'invitation à l'utilisateur invité.

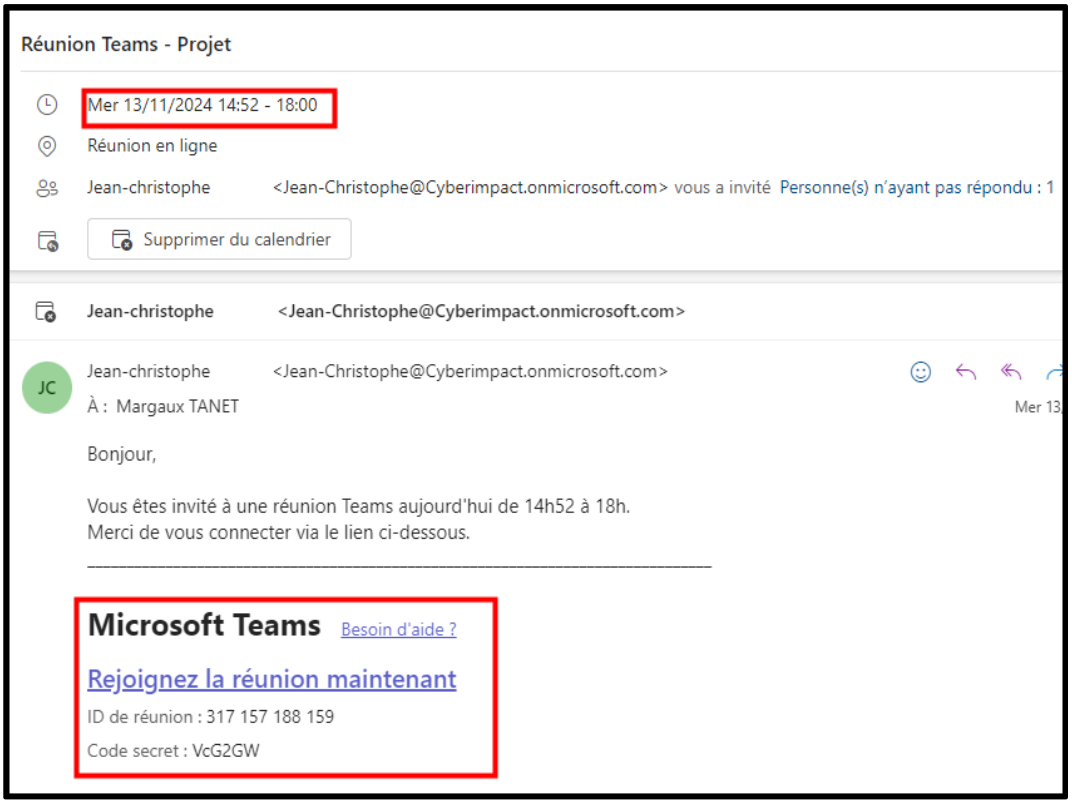


IV- Création d'une réunion

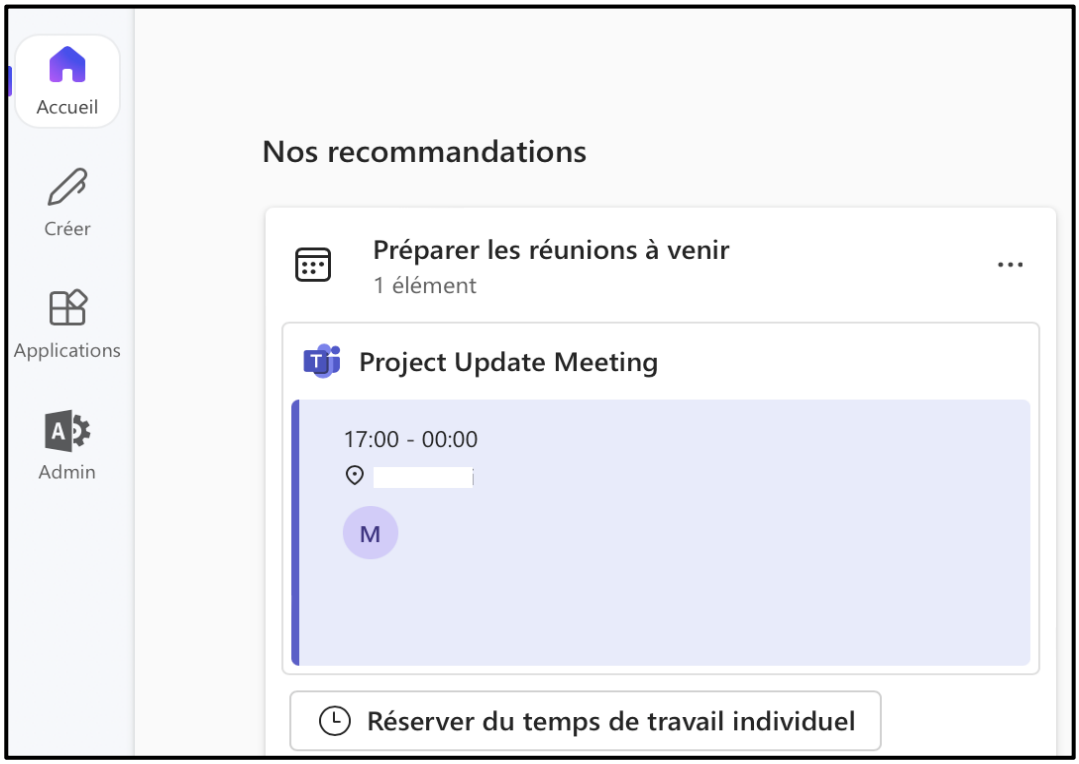
Grâce à l'ajout de requête dans l'API, il est possible de créer un évènement dans le calendrier de l'organisateur. Voici la requête pour la création d'une réunion :
Création de la réunion à 14h52 le 13-11 avec comme titre « Réunion Teams – Projet »



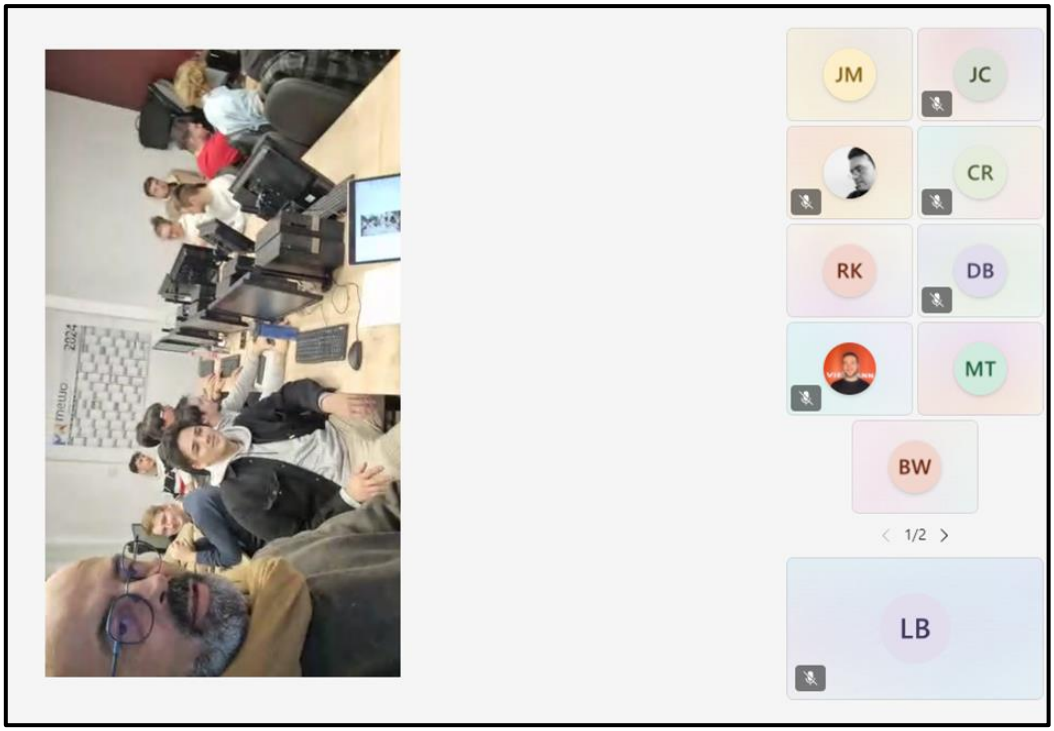
➢ Voici le mail d'invitation pour la réunion reçue à l'attention des personnes destinées :



➢ La réunion s'affiche bien dans le calendrier :



Et voici le résultat de la réunion :



V- Conclusion

Dans ce rapport, il est présenté comment Microsoft Graph fournit une API qui permet d'avoir accès aux données et aux informations stockées dans l'organisation. Microsoft donne la possibilité de pouvoir créer une réunion, d'effectuer différentes requêtes pour en extraire les différentes informations d'après le rôle qui a été appliqué.

Sources :

Microsoft Learn

Forum

Cours – Cyberimpact