Deployment Guide: Microsoft Graph Calendar API Automation

Friday, July 25, 2025 10:23 AM

🚀 Deployment Guide: Microsoft Graph Calendar API Automation

📁 Prerequisites

1. Azure Subscription with permissions to:

- Register applications in Azure AD
- Manage Azure Key Vault
- Assign Graph API permissions

2. PowerShell Environment with the following modules installed:

- Az.Accounts
- Az.Resources
- Az.KeyVault
- Az.AD
- Microsoft.Graph.Calendar
- Microsoft.Graph.Applications

Install modules (if needed):

3. Azure Key Vault:

- Name: wolffofficekvkv2
- Resource Group: Adminrg
- Must allow network access from your IP or automation IP

🛠️ Step-by-Step Setup

1. 🔐 Authenticate and Set Context

```
Connect-AzAccount
$subscription = Get-AzSubscription | Out-GridView -Title "Select a subscription:" -PassThru
Set-AzContext -Subscription $subscription.Id
```

2. 🧬 Register Azure AD Application

```
$app = New-AzADApplication -DisplayName "WOLFFCalendarManagerAppspn" `
  -IdentifierUris "https://wpi-corp.com/CalendarManagerApp" `
  -HomePage "https://wpi-corp.com/CalendarManagerApp"
```

3. 👤 Create Service Principal

```
$sp = New-AzADServicePrincipal -ApplicationId $app.AppId
```

4. 🔑 Assign Microsoft Graph Permissions

```
Connect-MgGraph -TenantId $subscription.TenantId -
Scopes "Application.ReadWrite.All", "AppRoleAssignment.ReadWrite.All"
$graphSp = Get-MgServicePrincipal -Filter "displayName eq 'Microsoft Graph'"
$permissions = @("Calendars.Read", "Calendars.ReadWrite", "MailboxSettings.Read", "User.Read.All")
foreach ($perm in $permissions) {
  $appRole = $graphSp.AppRoles | Where-Object { $_.Value -eq $perm -and $_.AllowedMemberTypes -
contains "Application" }
  if ($appRole) {
    New-MgServicePrincipalAppRoleAssignment -ServicePrincipalId $sp.Id `
     -PrincipalId $sp.Id -ResourceId $graphSp.Id -AppRoleId $appRole.Id
  }
}
```

5. 🔐 Configure Azure Key Vault

```
Update-AzKeyVault -ResourceGroupName "Adminrg" -VaultName "wolffofficekvkv2" -
PublicNetworkAccess Enabled
Set-AzKeyVaultAccessPolicy -VaultName "wolffofficekvkv2" -ObjectId $sp.Id -
PermissionsToSecrets get,list,set
```

6. 🔐 Generate and Store Client Secret

```
$clientSecret = New-AzADAppCredential -ApplicationId $app.AppId
$secureSecret = ConvertTo-SecureString -String $clientSecret.SecretText -AsPlainText -Force
Set-AzKeyVaultSecret -VaultName "wolffofficekvkv2" -Name $app.DisplayName -
SecretValue $secureSecret
```

📅 Running the Calendar Event Script

Once the app and secret are configured:

1. Retrieve the secret from Key Vault
2. Acquire a token from Microsoft Identity Platform
3. Use Microsoft Graph API to create a calendar event

You can use the reusable script template I provided earlier to automate this process.

🔗 Admin Consent (Optional)

To grant admin consent for the app:

https://login.microsoftonline.com/<tenantId>/adminconsent?client_id=<appId>

Replace <tenantId> and <appId> with your actual values.