

✓ OVERVIEW OF THE SOLUTION

You will build:

1. A **SharePoint Events List** to hold the synced events
 2. An **Azure AD app registration** (for Graph API permissions)
 3. A **Power Automate Flow** that:
 - Calls **Microsoft Graph** → gets events from the M365 Group Calendar
 - Writes/updates them into the SharePoint list
 4. Display the synced events in **SharePoint Events web part** or **List web part**
-

✦ PART 1 — Create the SharePoint Events List

Step 1 — Create a custom list

1. Go to your SharePoint site
2. Click **New** → **List**
3. Choose **Blank list**
4. Name it: **Synced Group Events**
5. Click **Create**

Step 2 — Add event fields

Go to **List settings** → **Columns** → **Add column**

Create the following columns:

| Column Name | Type | Notes |
|--------------------|------------------|--|
| Title | Single line text | Event subject (already exists) |
| StartTime | Date & time | Event start time |
| EndTime | Date & time | Event end time |
| Location | Single line text | Optional |
| BodyPreview | Multiple lines | Optional |
| EventID | Single line text | Stores Graph event ID (used for updates) |

Why EventID?

→ It prevents creating duplicates every time the flow runs.

✦ PART 2 — Register an Azure AD App (for Microsoft Graph access)

You need an app so Power Automate can call Graph to read the Group Calendar.

Step 1 — Go to Azure App Registrations

1. Visit **Azure Portal**
2. Search → **Azure Active Directory**
3. In the left menu → **App registrations**
4. Click **New registration**

Step 2 — Configure the app

- Name: **GroupCalendarSyncApp**
- Supported account types: **Single tenant**
- Redirect URI: **Leave blank**

Click **Register**

Step 3 — Add API Permissions

1. Go to **API permissions**
2. Click **Add permission**

3. Select **Microsoft Graph**
4. Select **Application permissions**
5. Add:
 - Calendars.Read
6. Click **Grant admin consent**

Step 4 — Generate a client secret

1. Go to **Certificates & secrets**
2. Click **New client secret**
3. Copy the **Value** (you won't see it again)

Save these three values (you will need them in Power Automate):

- Tenant ID
- Client ID
- Client Secret

✦ PART 3 — Build the Power Automate Flow

You will create a scheduled cloud flow.

Step 1 — Create the flow

1. Go to **Power Automate**
2. Select **Create → Scheduled cloud flow**
3. Run every:
 - **1 hour OR 1 day** depending on your needs
4. Click **Create**

Step 2 — Add HTTP Action (to call Microsoft Graph)

Click **New step → HTTP**

Configure it:

METHOD

GET

URI

<https://graph.microsoft.com/v1.0/groups/{GroupID}/events>

Find the Group ID:

- Go to Outlook on the web → Right-click the Group → “Settings” → see its ID
OR
- In Azure AD → Groups → find the Microsoft 365 group

HEADERS

Content-Type: application/json

AUTHENTICATION

Choose: **Active Directory OAuth**

Fill in:

- **Tenant:** *Your Tenant ID*
- **Audience:** <https://graph.microsoft.com>
- **Client ID:** *Your App's Client ID*
- **Secret:** *Your Client Secret*

Step 3 — Add “Parse JSON” Action

This extracts event fields from the Graph response.

Click **Add step** → **Parse JSON**

Schema (paste this):

```
{
  "type": "object",
  "properties": {
    "value": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": { "type": "string" },
          "subject": { "type": "string" },
          "start": {
            "type": "object",
            "properties": {
              "dateTime": { "type": "string" }
            }
          },
          "end": {
            "type": "object",
            "properties": {
              "dateTime": { "type": "string" }
            }
          },
          "location": {
            "type": "object",
            "properties": {
              "displayName": { "type": "string" }
            }
          },
          "bodyPreview": { "type": "string" }
        }
      }
    }
  }
}
```

Set **Content** to:

Body from the HTTP step

Step 4 — “Apply to each” loop (for each event)

Click:

Add step → **Apply to each**

Select the array:

value (from Parse JSON)

Inside the loop, add:

Step 4a — “Get items” (to check if the event already exists)

Action: Get items → SharePoint

Site: *Your Site*

List: *Synced Group Events*

Filter Query:

EventID eq '@{items('Apply_to_each')?['id']}'

Step 4b — Condition: Event exists?

Add Condition:

length(body('Get_items')?['value']) is greater than 0

If event exists → Update item

Inside "If Yes":

Add Update item:

- Title → subject
- StartTime → start/dateTime
- EndTime → end/dateTime
- Location → location/displayName
- BodyPreview → bodyPreview
- EventID → id

Make sure to use the **item ID** from Get items (the existing list item).

If event does NOT exist → Create item

Inside "If No":

Add Create item:

- Title → subject
 - StartTime → start/dateTime
 - EndTime → end/dateTime
 - Location → location/displayName
 - BodyPreview → bodyPreview
 - EventID → id
-

PART 4 — Display the synced events on the SharePoint page

You have two choices:

Option A — Calendar-style display

1. Edit the SharePoint page
2. Add **Events** web part
3. Point it to your list (Synced Group Events)

Option B — Full list view

1. Add **List** web part
 2. Select **Synced Group Events**
 3. Configure the view (calendar, list, compact, filters, etc.)
-

YOU'RE DONE

Your SharePoint page now shows your Group Calendar events, synced automatically.