**Prerequisites**

* [.NET Core SDK](https://dotnet.microsoft.com/download) version 6.0

determine dotnet version

dotnet --version

* [Ngrok](https://ngrok.com/download) (For local environment testing) Latest (any other tunneling software can also be used)
* [Teams](https://teams.microsoft.com/) Microsoft Teams is installed and you have an account

**Setup**

**Register your Teams Auth SSO with Azure AD**

1. Register a new application in the [Azure Active Directory – App Registrations](https://go.microsoft.com/fwlink/?linkid=2083908) portal.

* Select **New Registration** and on the *register an application page*, set following values:
  + Set **name** to your app name.
  + Choose the **supported account types** (any account type will work)
  + Leave **Redirect URI** empty.
  + Choose **Register**.
* On the overview page, copy and save the **Application (client) ID, Directory (tenant) ID**. You’ll need those later when updating your Teams application manifest and in the appsettings.json.
* Under **Manage**, select **Expose an API**.
* Select the **Set** link to generate the Application ID URI in the form of api://{AppID}. Insert your fully qualified domain name (with a forward slash "/" appended to the end) between the double forward slashes and the GUID. The entire ID should have the form of: api://fully-qualified-domain-name/{AppID}
  + ex: api://%ngrokDomain%.ngrok-free.app/00000000-0000-0000-0000-000000000000.
* Select the **Add a scope** button. In the panel that opens, enter access\_as\_user as the **Scope name**.
* Set **Who can consent?** to Admins and users
* Fill in the fields for configuring the admin and user consent prompts with values that are appropriate for the access\_as\_user scope:
  + **Admin consent title:** Teams can access the user’s profile.
  + **Admin consent description**: Allows Teams to call the app’s web APIs as the current user.
  + **User consent title**: Teams can access the user profile and make requests on the user's behalf.
  + **User consent description:** Enable Teams to call this app’s APIs with the same rights as the user.
* Ensure that **State** is set to **Enabled**
* Select **Add scope**
  + The domain part of the **Scope name** displayed just below the text field should automatically match the **Application ID** URI set in the previous step, with /access\_as\_user appended to the end:
    - `api://[ngrokDomain].ngrok-free.app/00000000-0000-0000-0000-000000000000/access\_as\_user.
* In the **Authorized client applications** section, identify the applications that you want to authorize for your app’s web application. Each of the following IDs needs to be entered:
  + 1fec8e78-bce4-4aaf-ab1b-5451cc387264 (Teams mobile/desktop application)
  + 5e3ce6c0-2b1f-4285-8d4b-75ee78787346 (Teams web application) **Note** If you want to test or extend your Teams apps across Office and Outlook, kindly add below client application identifiers while doing Azure AD app registration in your tenant:
  + 4765445b-32c6-49b0-83e6-1d93765276ca (Office web)
  + 0ec893e0-5785-4de6-99da-4ed124e5296c (Office desktop)
  + bc59ab01-8403-45c6-8796-ac3ef710b3e3 (Outlook web)
  + d3590ed6-52b3-4102-aeff-aad2292ab01c (Outlook desktop)
* Navigate to **API Permissions**, and make sure to add the follow permissions:
  + Select Add a permission
  + Select Microsoft Graph -> Delegated permissions.
    - **User.Read (enabled by default)**
    - **Calendars.ReadWrite**
    - **Mail.Send**
    - **MailboxSettings.ReadWrite**
  + Click on Add permissions. Please make sure to grant the admin consent for the required permissions.
  + Navigate to **Authentication** If an app hasn't been granted IT admin consent, users will have to provide consent the first time they use an app. Set a redirect URI:
  + Select **Add a platform**.
  + Select **Single Page Application**.
  + Enter the **redirect URI** for the app in the following format: https://%ngrokDomain%.ngrok-free.app/Auth/End.
* Navigate to the **Certificates & secrets**. In the Client secrets section, click on "+ New client secret". Add a description (Name of the secret) for the secret and select “Never” for Expires. Click "Add". Once the client secret is created, copy its value, it need to be placed in the appsettings.json.
* Ensure that you've [enabled the Teams Channel](https://learn.microsoft.com/en-us/azure/bot-service/channel-connect-teams?view=azure-bot-service-4.0)

1. **Setup NGROK**

* Run ngrok - point to port 3978

# ngrok http 3978 --host-header="localhost:3978"

1. **Setup for code**

* Clone the repository

git clone https://github.com/fsdp15/CoorporateHelper.git

* Modify the /appsettings.json and fill in the following details:
  + {{YOUR-TENANT-ID}} - Generated from Step 1 while doing AAd app registration in Azure portal Directory (tenant) ID.
  + {{YOUR-MICROSOFT-APP-ID}} - Generated from Step 1 while doing AAd app registration in Azure portal.
  + {{ YOUR-CLIENT-SECRET}} - Generated from Step 1, also referred to as Client secret
  + {{ ApplicationIdURI }} - Your application's ApplicationIdURI. E.g. api://%ngrokDomain%.ngrok-free.app/00000000-0000-0000-0000-000000000000.,
  + Run app in visual studio.
  + If you are using Visual Studio
  + Launch Visual Studio
  + File -> Open -> Project/Solution
  + Select CoorporateHelper.csproj file

1. **Setup Manifest for Teams**

* ***This step is specific to Teams.***
* **Edit** the manifest.json contained in the ./Manifest folder to replace your Microsoft App Id (that was created when you registered your app registration earlier) *everywhere* you see the place holder string <<YOUR-MICROSOFT-APP-ID>> (depending on the scenario the Microsoft App Id may occur multiple times in the manifest.json)
* **Edit** the manifest.json for validDomains and replace {{domain-name}} with base Url of your domain. E.g. if you are using ngrok it would be https://1234.ngrok-free.app then your domain-name will be 1234.ngrok-free.app.
* **Edit** the manifest.json for webApplicationInfo resource "api://{{domain-name}}/<<YOUR-MICROSOFT-APP-ID>>" with MicrosoftAppId. E.g. "api://1234.ngrok-free.app/00000000-0000-0000-0000-000000000000".
* **Zip** up the contents of the Manifest folder to create a Manifest.zip file (Make sure that zip file does not contains any subfolder otherwise you will get error while uploading your .zip package)
* Upload the manifest.zip to Teams (in the Apps view click "Upload a custom app")
  + Go to Microsoft Teams. From the lower left corner, select Apps
  + From the lower left corner, choose Upload a custom App
  + Go to your project directory, the ./Manifest folder, select the zip folder, and choose Open.
  + Select Add in the pop-up dialog box. Your app is uploaded to Teams.

1. Run your app, either from Visual Studio with F5 or using dotnet run in the appropriate folder.