Required Constants:

1. **Tenant ID**
2. **Client ID**
3. **Client Secret**
4. **Team ID**

Step-by-Step Instructions to Obtain Each Constant:

1. Tenant ID

Where to Find: This identifier represents your organization's instance within Microsoft Azure.

- **Steps**:
  - Log into the **Azure Portal** (portal.azure.com).
  - Navigate to **Azure Active Directory**.
  - Your **Directory (tenant) ID** listed in the overview section is your Tenant ID.

2. Client ID and Client Secret

Purpose: These credentials authenticate your application to Azure AD, allowing it to interact with Microsoft services like Microsoft Graph API.

- **Steps**:
  - In the **Azure Portal**, navigate to **Azure Active Directory** > **App registrations** > **New registration**.
  - Provide a name for your application, specify account types, and optionally set a redirect URI (e.g., `http://localhost` for local development).
  - After registration, the **Application (client) ID** displayed on the overview page is your Client ID.
  - Under **Certificates & secrets**, generate a **client secret**. Note it down immediately, as it will not be visible again once you navigate away.

3. Team ID

Purpose: This is the unique identifier for a Microsoft Teams team that you want to manage.

- **Steps**:
  - If you have administrative access, you might find the Team ID directly in Microsoft Teams admin center.
  - Alternatively, use the Microsoft Graph API to list all teams:
    - Utilize a GET request to `https://graph.microsoft.com/v1.0/me/joinedTeams` with a tool like Postman or the Microsoft Graph Explorer.

Required API Permissions:

- **Application Permissions** (for service-to-service interactions):
  - `TeamMember.ReadWrite.All`: Allows the app to manage teams and their memberships.
  - `User.Read.All`: Allows the app to read all users' profiles.
- **Delegated Permissions** (for user-based interactions):
  - `TeamMember.ReadWrite.All`: Enables the app to manage teams on behalf of the logged-in user.
  - `User.Read`: Enables the app to read the logged-in user's profile.

Configuring API Permissions and Granting Admin Consent:

1. Navigate back to the **Azure Portal**.
2. Select your registered application under **Azure Active Directory** > **App registrations**.
3. Go to **API permissions**:
   - Click **Add a permission** > **Microsoft Graph**.
   - Choose **Application permissions** or **Delegated permissions** depending on your application's needs.
   - Add the necessary permissions such as `TeamMember.ReadWrite.All` and `User.Read.All`.
4. Click on **Grant admin consent for [Your Organization]** to activate these permissions.