Office 365 Proposal Manager

Getting Started Guide

Version 3.0

Contents

[1 Overview 1](#_Toc528079718)

[2 Setup Azure App Service 1](#_Toc528079719)

[2.1 Azure App Registration 1](#_Toc528079720)

[2.2 Application Insights 3](#_Toc528079721)

[2.3 Confirm Azure web app deployment status 3](#_Toc528079722)

[3 Register Proposal Manager App 4](#_Toc528079723)

[3.1 Register App 4](#_Toc528079724)

[3.2 Configure Application Registration 5](#_Toc528079725)

[4 Setup Office 365 Groups 7](#_Toc528079726)

[5 SharePoint Configuration 8](#_Toc528079727)

[6 Deploy Solution 9](#_Toc528079728)

[6.1 Update App Settings 10](#_Toc528079729)

[6.2 Update Client App Settings 11](#_Toc528079730)

[6.3 Publish to Azure 12](#_Toc528079731)

[6.4 Accept Application Consent as Admin 12](#_Toc528079732)

[7 Teams add-in for Proposal Manager 13](#_Toc528079733)

[7.1 Enable side-loading of add-ins 13](#_Toc528079734)

[7.2 Create custom add-in for Teams 14](#_Toc528079735)

[8 Guided Setup 15](#_Toc528079736)

[9 Load Proposal Manager add-in 18](#_Toc528079737)

[10 Configure Notifications Bot 20](#_Toc528079738)

[11 Configure Analytics Dashboard 21](#_Toc528079739)

[12 Office add-in for Proposal Creation 24](#_Toc528079740)

[12.1 Deploying Office add-in 24](#_Toc528079741)

[12.2 Configure SharePoint Document ID Service 24](#_Toc528079742)

[13 API Guidance 25](#_Toc528079743)

[14 User Experience Overview 25](#_Toc528079744)

[14.1 User Personas 25](#_Toc528079745)

[14.2 User Interfaces 26](#_Toc528079746)

[14.3 Key Entities 26](#_Toc528079747)

[14.4 User Permissions 27](#_Toc528079748)

[15 Proposal Management Process 28](#_Toc528079749)

[15.1 High Level Workflow 28](#_Toc528079750)

[15.2 Data Flow 28](#_Toc528079751)

[15.3 User Interaction Flow 29](#_Toc528079752)

[16 Extensibility 31](#_Toc528079753)

[17 Maintenance 31](#_Toc528079754)

[18 Security Considerations 33](#_Toc528079755)

[19 Troubleshooting 33](#_Toc528079756)

[20 Known Issues 34](#_Toc528079757)

[Appendix: Permissions List 34](#_Toc528079758)

# Overview

The purpose of this document is to provide guidance on setup and deployment to an ISV partner who is familiar with Microsoft 365, Microsoft Azure and web application implementation. This document is part of the October 2018 release of the solution - Office 365 Proposal Manager.

**Prerequisites**

To setup the solution, make sure that you have:

* Microsoft 365 Tenant (Business or Enterprise SKU)
* Microsoft Azure subscription
* Visual Studio 2017 (Enterprise, recommended), version 15.7.4 and above
* ASP.NET Core 2.1
* Node.js

Note that the following access permissions are required to be able to deploy the solution:

* Ensure that you have global admin access on the Office 365 tenant where the solution is being deployed
* You need to have Contributor or Owner access on the Microsoft Azure subscription where the web application is planned to be deployed and made available to the users.

# Setup Azure App Service

In this section, we set up an app service on Microsoft Azure to deploy the solution. Make sure that you have contributor access on the Microsoft Azure subscription where the web application is planned to be deployed and made available to the users.

## Azure App Registration

First step is to create the web application where users can access the Proposal Manager application. Make sure that you have Owner or Contributor access on the Azure subscription before proceeding.

Sign into [Microsoft Azure portal](https://portal.azure.com/).

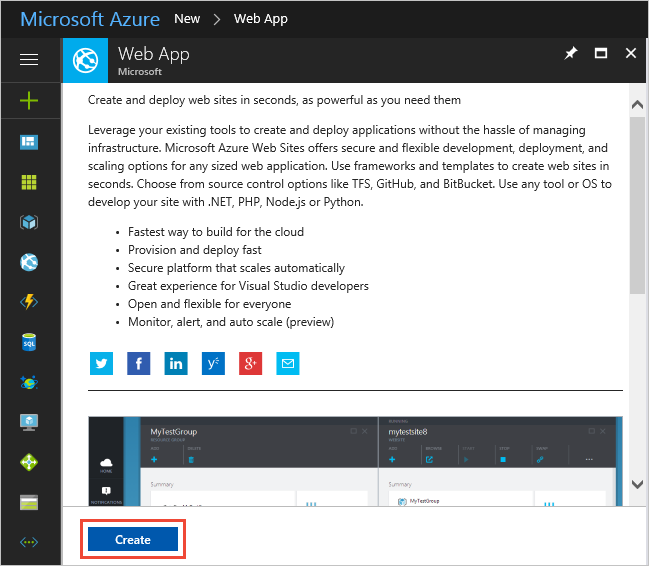


Choose the **+** icon in the left navigation bar, then choose **Web App**.

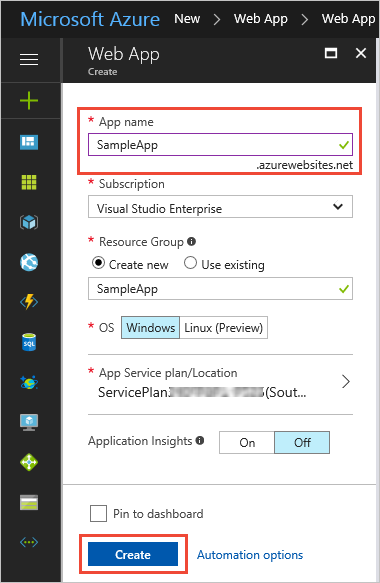


If you don't see **Web App** in the list, use the search box to find it.

At the bottom of the introduction page, choose **Create**.

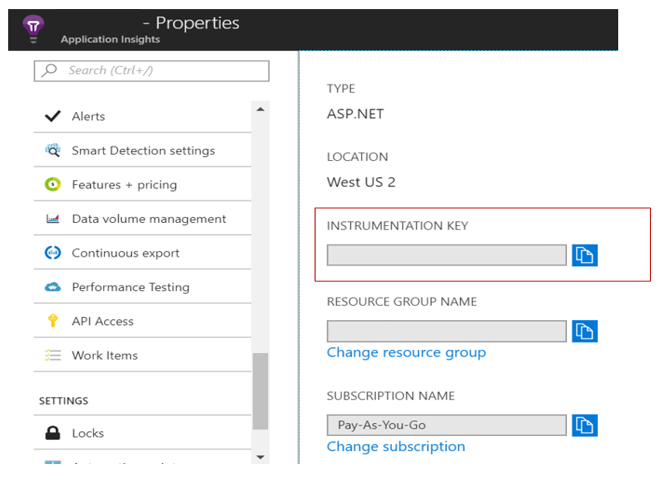


Enter a name for the new web app. You'll see a green checkmark when the name is unique. Then choose **Create**.



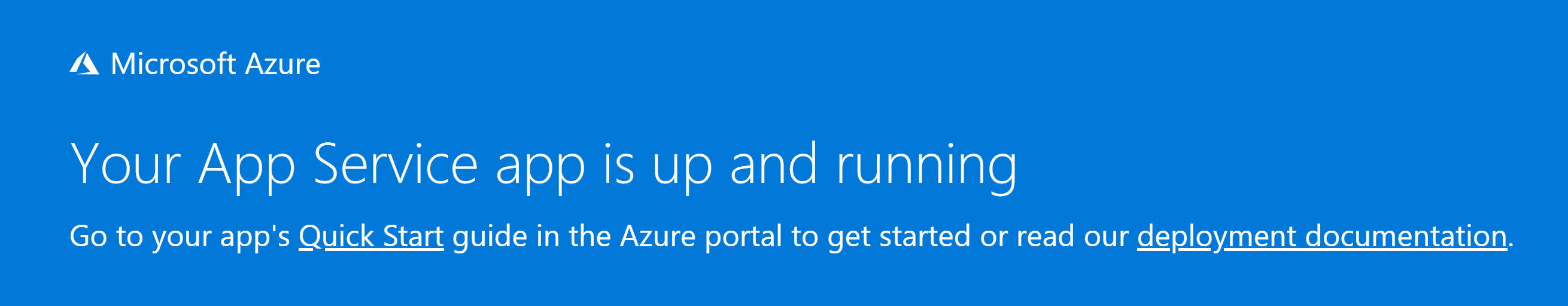
## Application Insights

In case you would like to monitor the live web application, at least initially, it is a good idea to leverage Azure Application Insights. This can be configured from the Azure Portal, Application Insights tab. Take care to note the Instrumentation ID.



## Confirm Azure web app deployment status

Go to the newly created app service and click on the app URL. A default landing page will be displayed if the app service has been successfully setup.



# Register Proposal Manager App

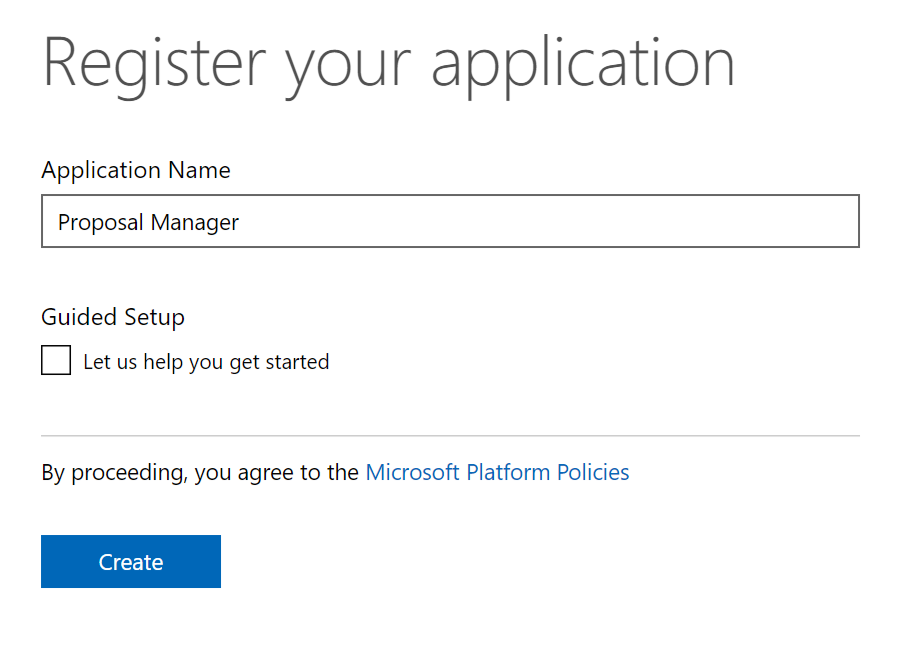
## 3.1 Register App

First, we need to register the “Proposal Manager” app to enable users in the organization to access the solution and to facilitate communication between the dashboard and the solution implementation.

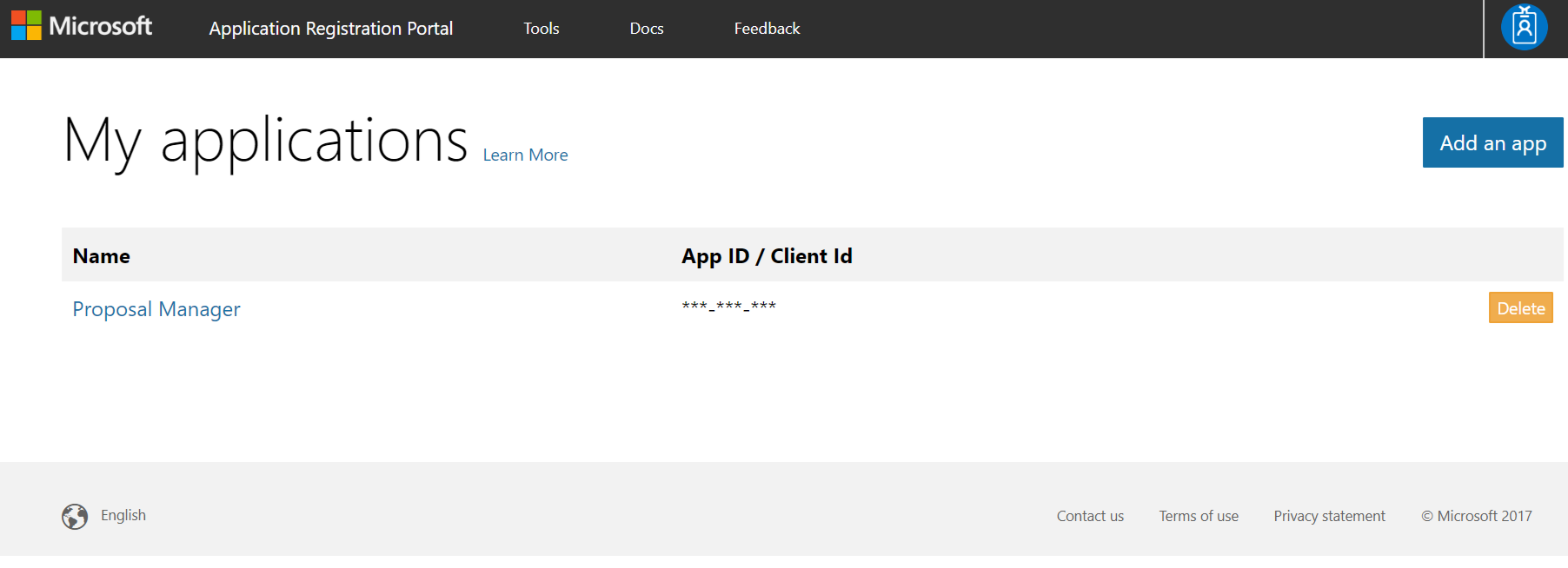
1. To access the Application Registration Portal, go to <https://apps.dev.microsoft.com> and login using your **Office 365 tenant admin account**
2. Click on **Add an app**



1. Give an appropriate name, say **‘Proposal Manager’** and Click **Create**. **Note the Application Id**



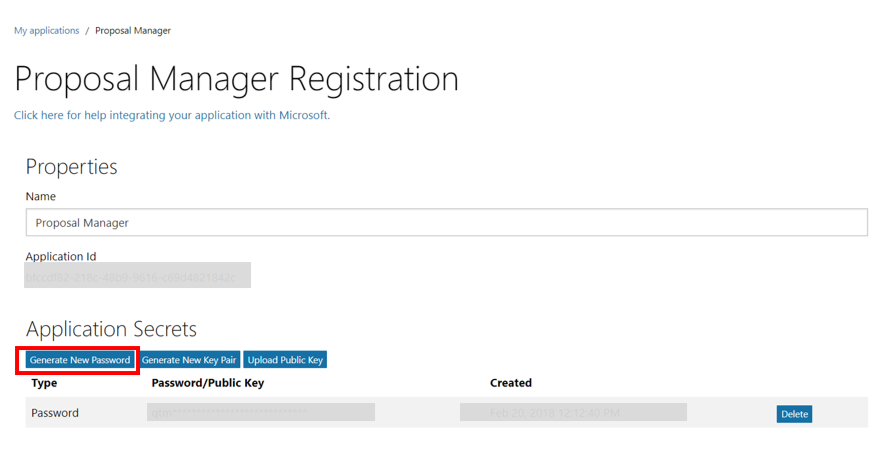
1. Once it is registered you will see it listed as shown below.



## 3.2 Configure Application Registration

Now let’s configure the Proposal Manager Azure Registration using the following details below:

1. Property Name: Enter the name of your application if needed
2. Application ID: **Note down the Application Id** for future use, this will need to be updated as Client Id in appsettings.json and Appsettings.js
3. Application Secret: Click on ‘Generate New Password’ and **note down the Application Secret** that will need to be updated as Client Secret in appsettings.json and Appsettings.js



Ensure you note the Password\Public Key which will be needed. If this is lost, another key will need to be generated to configure the application.

1. Platform: In the Platform section select Web Platform option and continue with its configuration
2. Identify the web URLS that will be needed as shown below example:

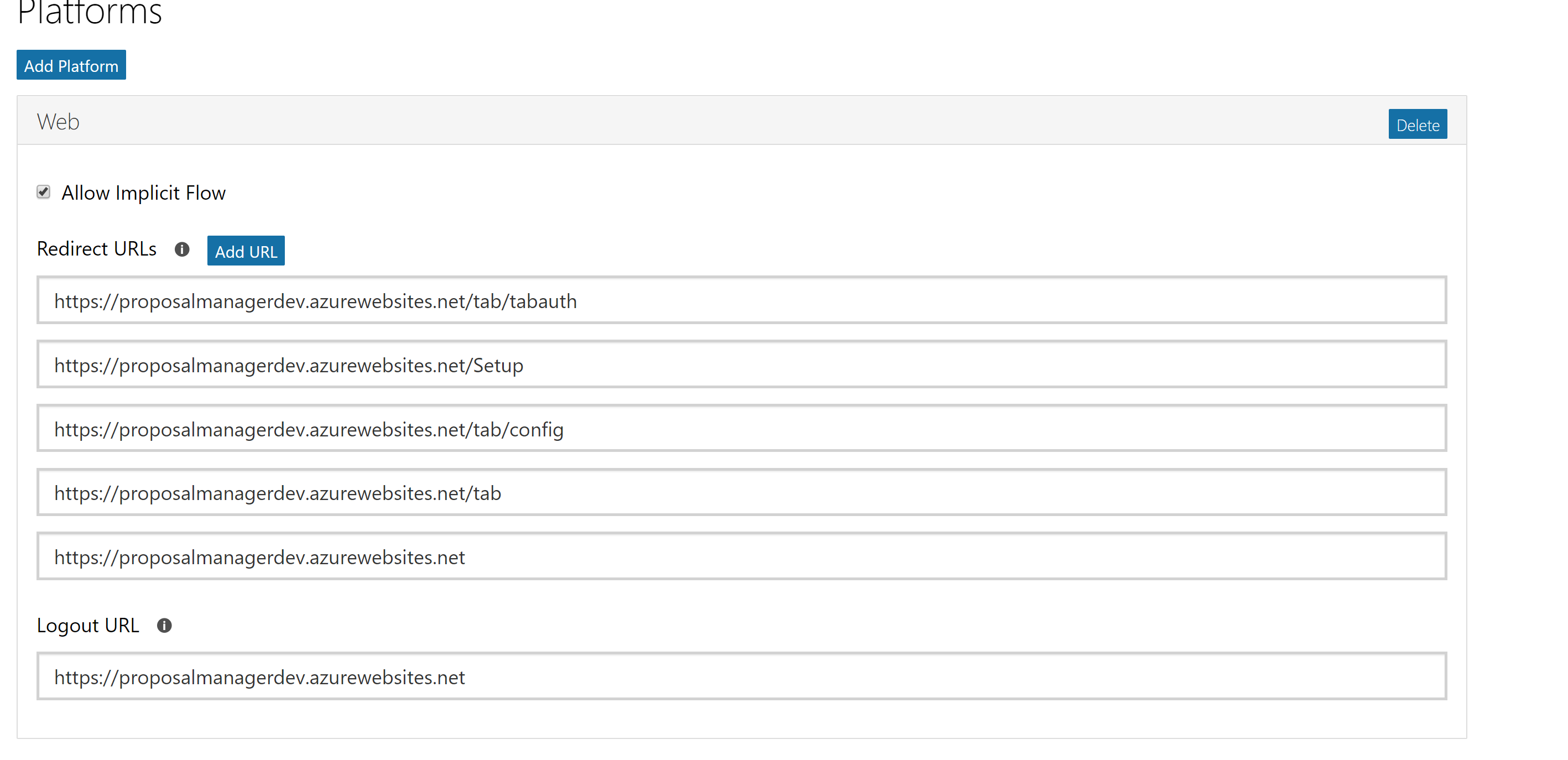
Check 'Allow Implicit Flow'. In the Redirect URLs section, add the URLs that are indicated as allowed for the Proposal Manager web application, once deployed

For example, if the Azure web site is created as https://proposalmanagerdev.azurewebsites.net, take care to add the following:

* https://proposalmanagerdev.azurewebsites.net
* https://proposalmanager.azurewebsites.net/tab
* https:// proposalmanagerdev.azurewebsites.net/tab/tabauth
* https:// proposalmanagerdev.azurewebsites.net/tab/config
* https:// proposalmanagerdev.azurewebsites.net/Setup

Note that the Redirect URLs are **case-sensitive**.

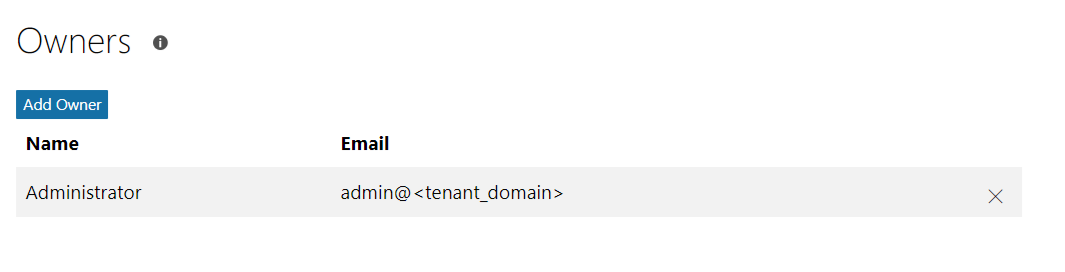
Screen will look something like this:



1. Recommended logout URL is https://proposalmanagerdev.azurewebsites.net/
2. Now we need to add another section, click on Web API under platforms and add scope as follows:

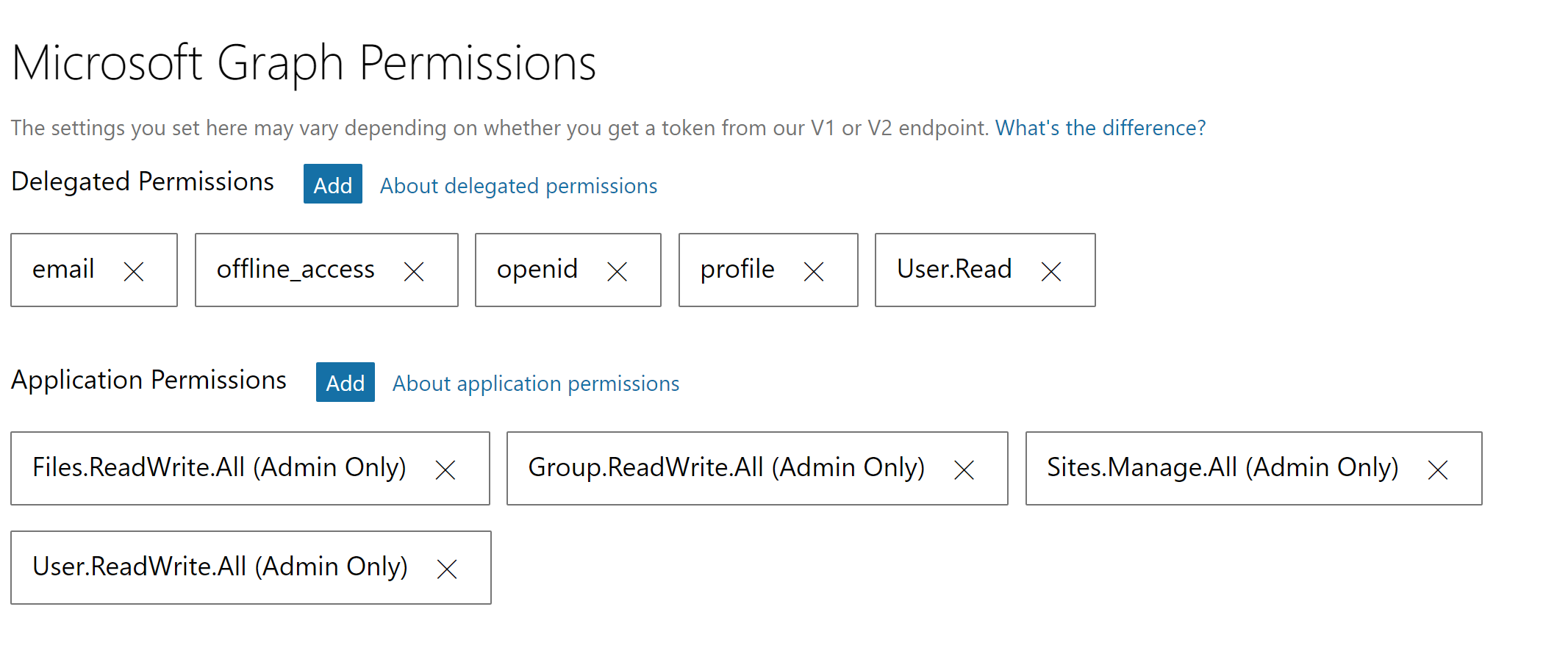
**api://<application id>/access\_as\_user**

1. Identify your Owners: This section defines who all have access to update the application configuration from the Application Registration Portal.



1. Identify Microsoft Graph Permissions as shown below:

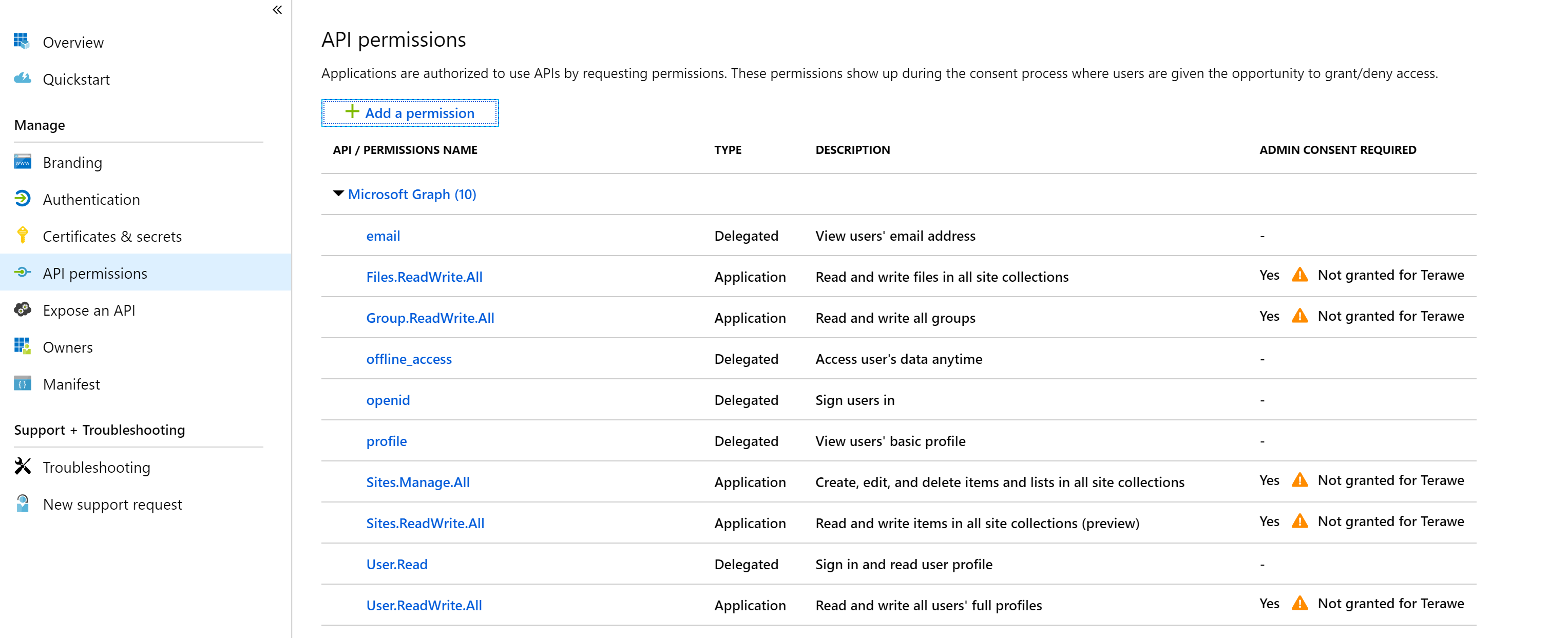
Set appropriate **Delegated Permissions** and **Application Permissions** by Clicking **Add.** See below for permissions to set for Proposal Manager.



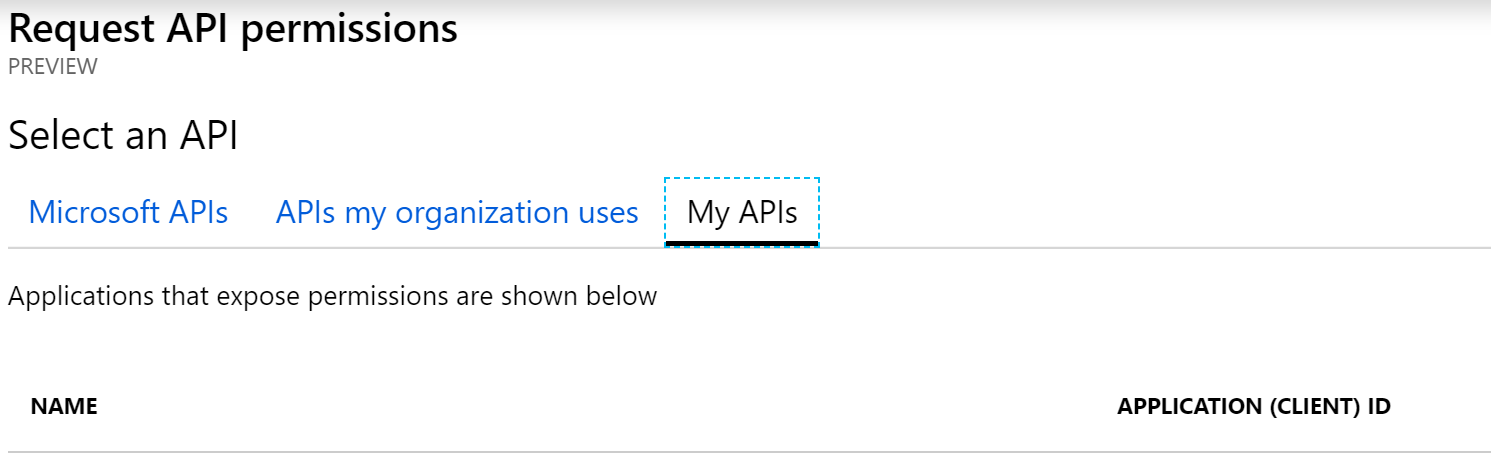
1. Identify your Home Page URL, Terms of Service URL and Privacy URL as needed
2. In the Profile selection, choose the logo that you would like to use and specify the home page URL as the web application URL that you plan to deploy the proposal manager solution to



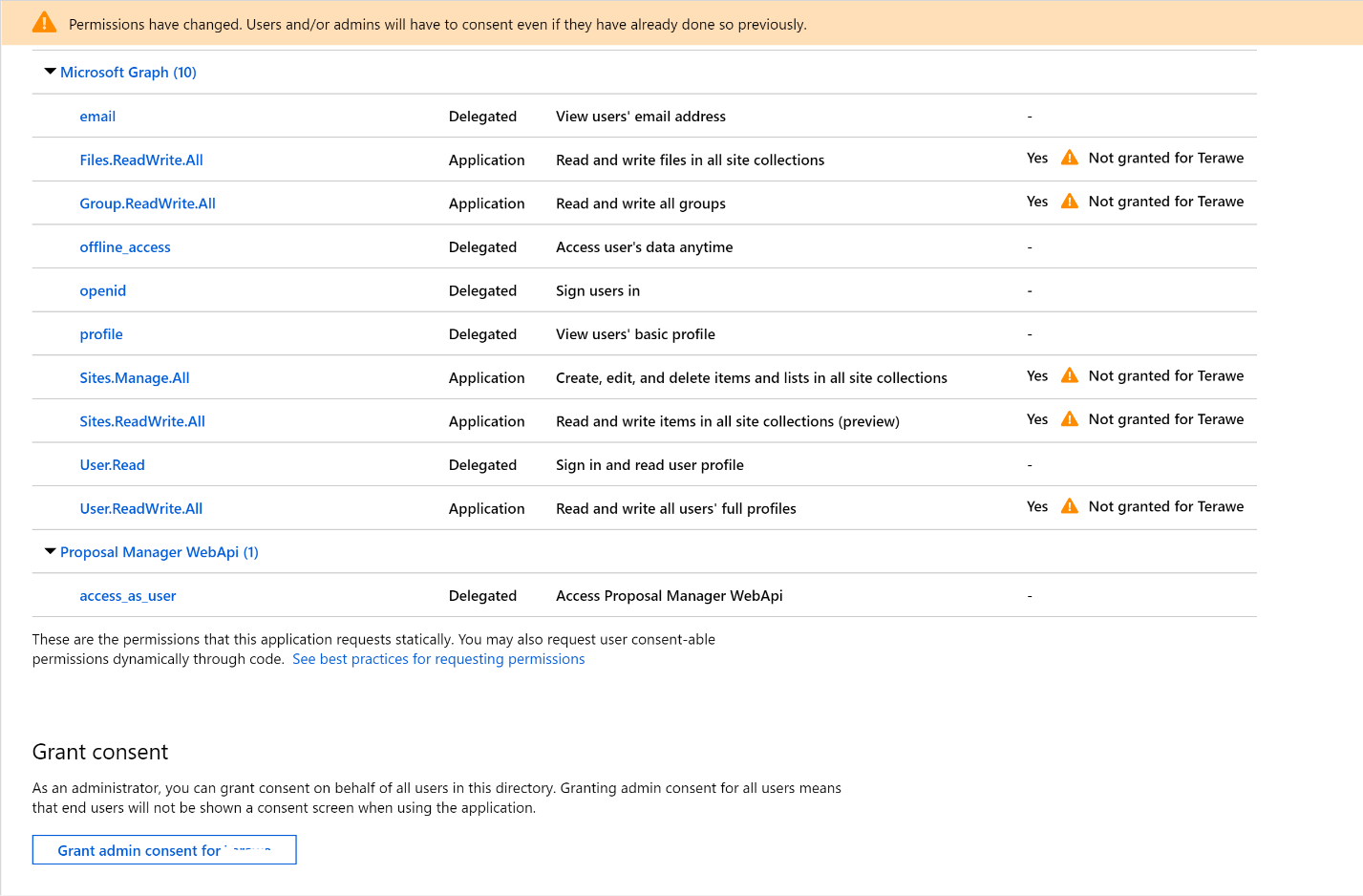
1. Save Settings to finish the process
2. Go to <https://portal.azure.com/>
3. Select Azure Active Directory and then select **App registrations (Preview)**
4. Select the app service which we created in <https://apps.dev.microsoft.com>
5. Go to **Api Permissions**



**Click on Add a permission**



1. Click on the App Registration which we created in <https://apps.dev.microsoft.com>
2. Check the access\_as\_user permission and then click add permission
3. After this click ***Grant admin consent for***



# Setup Office 365 Groups

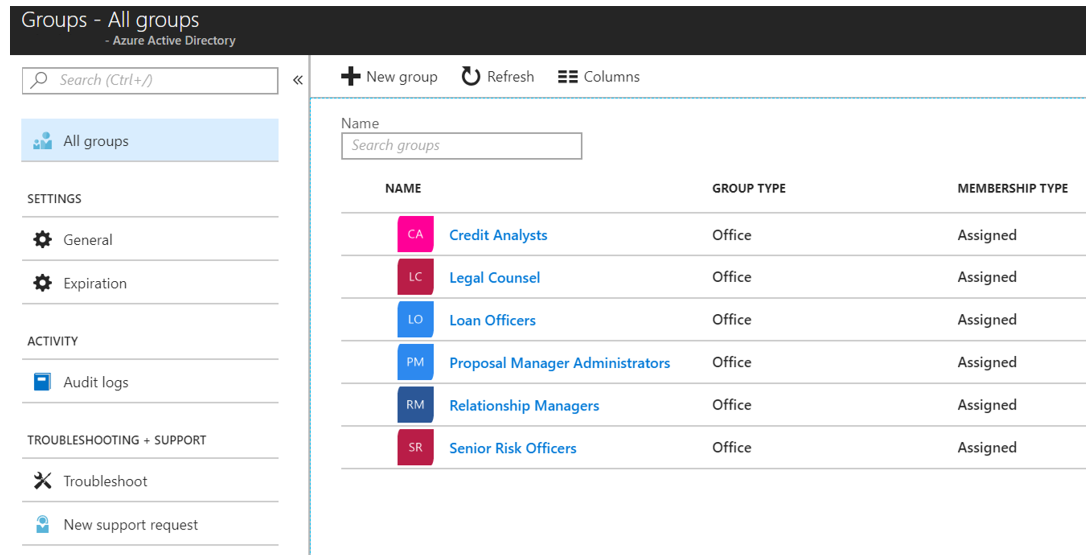
Proposal Manager requires the users to be added to role-specific groups to facilitate access with required permissions.

Note that the group names given below are indicative examples. Any group name can be used based or organizational mapping. This can be mapped to roles defined in the application at the time of deployment.

|  |  |  |
| --- | --- | --- |
| Example Group Name | Purpose | Type |
| Relationship Managers | Relationship Managers, who are authorized to create opportunities | Office 365 |
| Loan Officers | Loan Officers, who manage assigned opportunities, including selecting proposal document template | Office 365 |
| Legal Counsel | Legal Counsel for an opportunity who take care of compliance review | Office 365 |
|  | Risk Officers for an opportunity who take care of risk assessment | Office 365 |
| Credit Analysts | Credit Analysts for an opportunity who take care of credit check | Office 365 |
| Proposal Management Administrators | Administrators who are authorized to create Microsoft Teams and associated channels for an opportunity. Note that members of this group should have ‘Global Administrator’ role in the Office 365 tenant | Security |

To create Office 365 groups:

* Go to <https://portal.azure.com> (login as Office 365 tenant admin)
* Select Azure Active Directory tab from the left navigation menu (or choose from the 'All services' list)
* Choose Groups and click on 'New group'



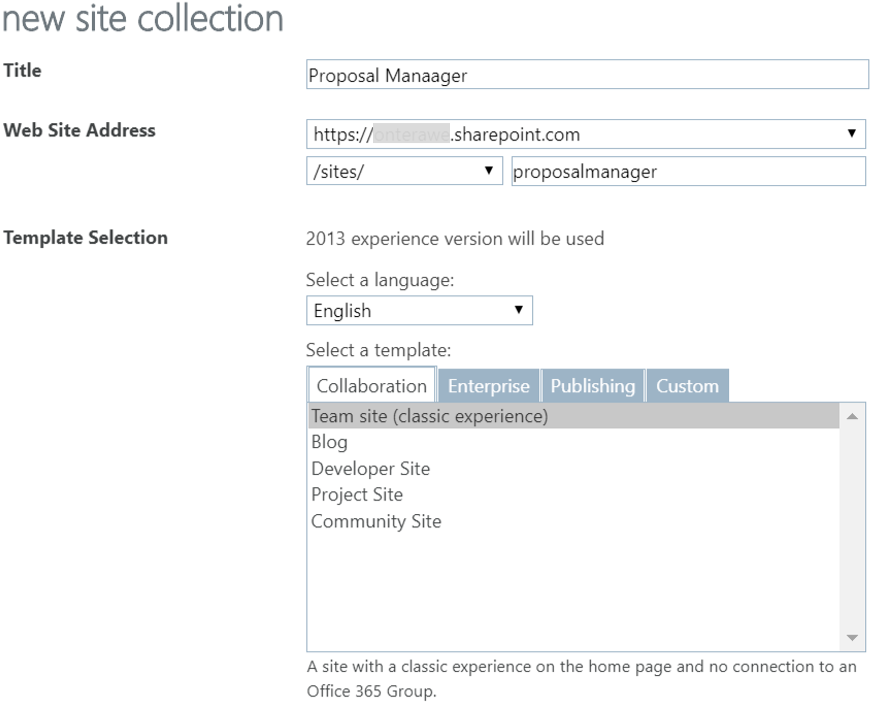
Ensure that users who require access to the solution are added to the relevant groups.

# SharePoint Configuration

In this section we will go over how to setup and configure SharePoint to be used as a repository to store information on opportunities. First step is to create a SharePoint site collection.

Access SharePoint Management portal at **https://<tenant>-admin.sharepoint.com**

Create a new site, say, Proposal Manager



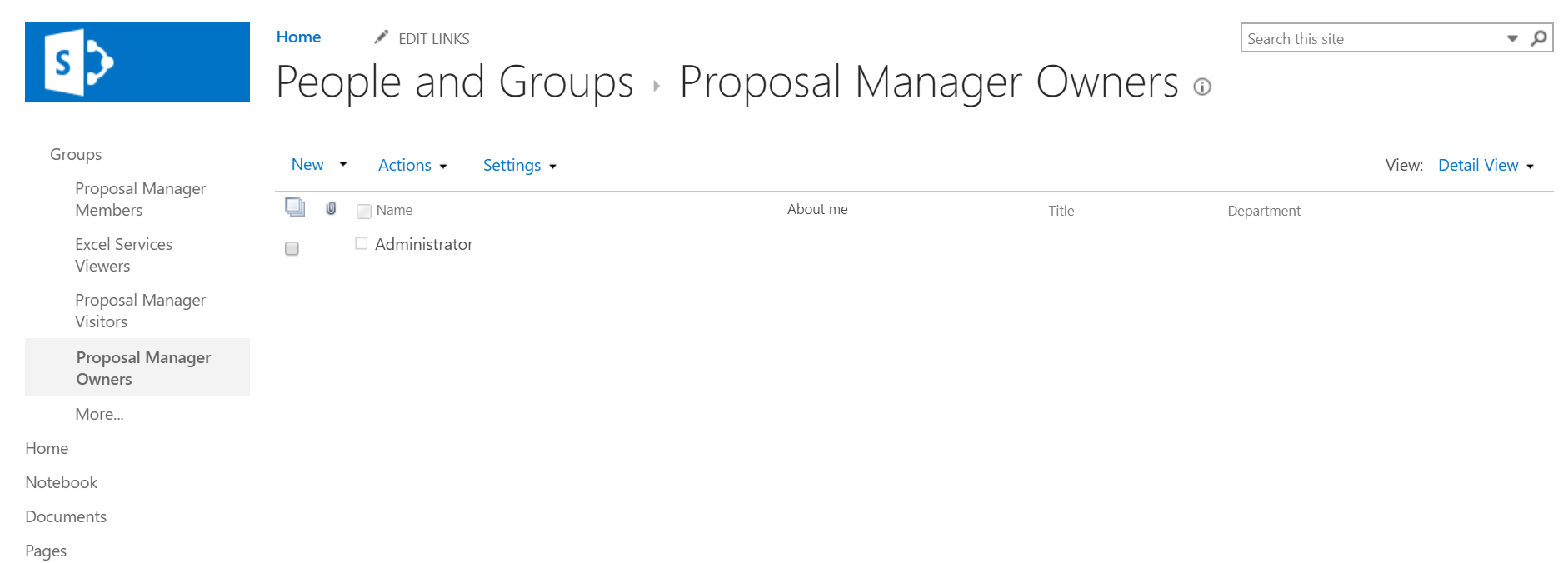
**Note:** If you are using the new version of SharePoint, which is currently in Preview (as of October 2018), you will need to select “New Team Site”.

Note down the site root host name and the relative name, which in this case is <tenant>.sharepoint.com and ‘ProposalManager’ respectively. This information will be needed in the solution setup step.

Add the Admin user as an owner on the site:

https://<tenant>.sharepoint.com/sites/ProposalManager/\_layouts/15/people.aspx?MembershipGroupId=3

Example:



The user who created the site will automatically be added to this group.

# Deploy Solution

Download the latest source code from <https://github.com/OfficeDev/ProposalManager> and proceed as detailed below.

Proposal Manager follows a two-step deployment process:

Step 1: Update the core app settings

* appsettings.json (relative path : WebReact\appsettings.json)
* Appsettings.js (WebReact\ClientApp\src\helpers\AppSettings.js)

Step 3: Publish to Azure, which enables one to access the Setup page

Step 2: Use the Setup page for a guided setup experience, which will take care of completing the required configuration to get the solution fully functional

To start, get a local copy of the latest version of the source code and open the solution in Visual Studio.

## Update App Settings

Open appsettings.json located at (.\WebReact\) and update as follows. Note that the property values not mentioned in the below table can be left as-is and will be updated automatically during the setup process.

|  |  |  |
| --- | --- | --- |
| Property | Description | Example Value |
| **AzureAd** |
| ClientId | This is the unique ID of the service principal object associated with the application, which is the application id in application portal. | "<client\_id>" |
| ClientSecret | Client secret for the app registered in tenant | <Noted from the Application Registration portal - apps.dev.microsoft.com> |
| Instance | Public Instance name for AAD | "https://login.microsoftonline.com/" |
|  |  |  |
| Domain | Domain name of the tenant | “<tenant\_domain>” |
| TenantId | Login into Azure portal as tenant admin account (Office 365 admin account) and go to properties and get Directory Id | "<tenant\_id>" |
| CallbackPath | Landing page for the app after authentication | "/signin-oidc" |
| Authority |  | "https://login.microsoftonline.com/<tenantid>" |
| BaseUrl | Return URL for the app after authentication, this should match one of the reply-to URLs specified in the app manifest | "https://<app\_url>.azurewebsites.net" |
| Scopes | Application scope | “https://graph.microsoft.com/.default“ |
| GraphResourceId | Public end-point for Graph API | "https://graph.microsoft.com/" |
| GraphScopes | Scope required by the application for accessing Microsoft Graph | “email User.Read.All” |

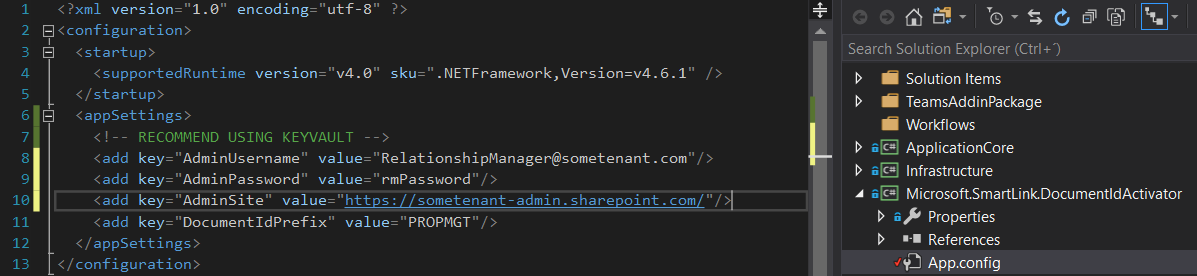
## Update Client App Settings

Open appsettings.js located at .\WebReact\ClientApp\src\helpers and update the fields indicated below. Take care to retain the scopes as indicated:

|  |
| --- |
| /\*  \* Copyright (c) Microsoft. All rights reserved. Licensed under the MIT license.  \* See LICENSE in the source repository root for complete license information.  \*/  // General settings  export const appUri = ''; //Base Url generated by Azure for the web service.  // This Section is Required to be updated before the initial publish to Azure.  export const clientId = ''; //Registered Application Id from apps.dev.microsoft.com.  export const redirectUri = appUri + "/"; //Redircet Url used at authentication.  export const instanceId = 'https://login.microsoftonline.com/';  export const graphScopes = ["offline\_access", "profile", "User.ReadBasic.All", "mail.send"]; //User scopes defined at app registration.  export const graphScopesAdmin = ["offline\_access", "profile", "User.Read.All", "mail.send", "Sites.ReadWrite.All", "Files.ReadWrite.All", "Group.ReadWrite.All"]; //Application scopes defined at app registration.  export const webApiScopes = ["api://<clientId>/access\_as\_user"];// web Api scope generated at app registration from apps.dev.microsoft.com.  export const authority = "https://login.microsoftonline.com/<tenantId>"; // Null for login as common (multi-tenant also) eg. https://login.microsoftonline.com/common/oauth2/v2.0/authorize  //No need to update anything bellow these are placeholders  const generalProposalManagementTeam = ""; //The Proposal Manager general team name that contains all the administration functionality.  const teamsAppInstanceId = ""; //Id of the Proposal Manager application instaled in teams.  const localStorePrefix = "env1\_"; //Local Store Prefix.  const teamsAppName = ""; //The short Name specified in the appllication manifest file.  const reportId = ""; //PowerBI Report Id.  const workspaceId = ""; //PowerBI WorkSpace Id.  export const appSettingsObject = {  generalProposalManagementTeam,  teamsAppInstanceId,  localStorePrefix,  teamsAppName,  reportId,  workspaceId  };  export default appSettingsObject; |

## Configure Document ID Activator

The Proposal Creation add-in relies on a unique Document ID to map the Proposal Document to an opportunity. For this, it is required enable the Document ID feature for each new opportunity. You will find a WebJob that does just that, included in the solution; the project is called Microsoft.SmartLink.DocumentIdActivator. To set it up, open App.config, and set up the values like displayed in the following figure:



The credentials stored here need to be of a user who has access to the opportunities and has the SharePoint administrator role. We recommend using the Relationship Manager for this, and storing the credentials in a more secure place, like Azure KeyVault.

## Publish to Azure

Choose Build 🡪 Publish Solution to build and deploy the solution to Azure from the Publish UI.

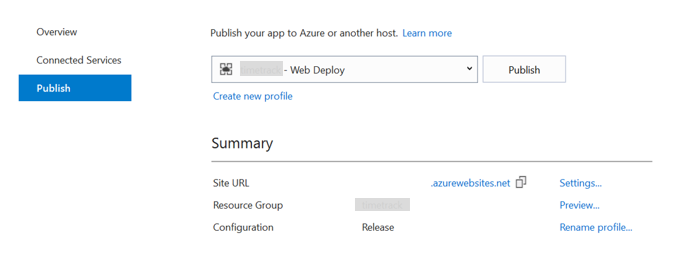
* Select the New Publish profile option and choose Azure App



* Logon using the Azure account that you used to setup the application
* Select the Resource Group where the application is setup and expand, then choose the application



* Click on Publish to deploy the application to the Azure app service

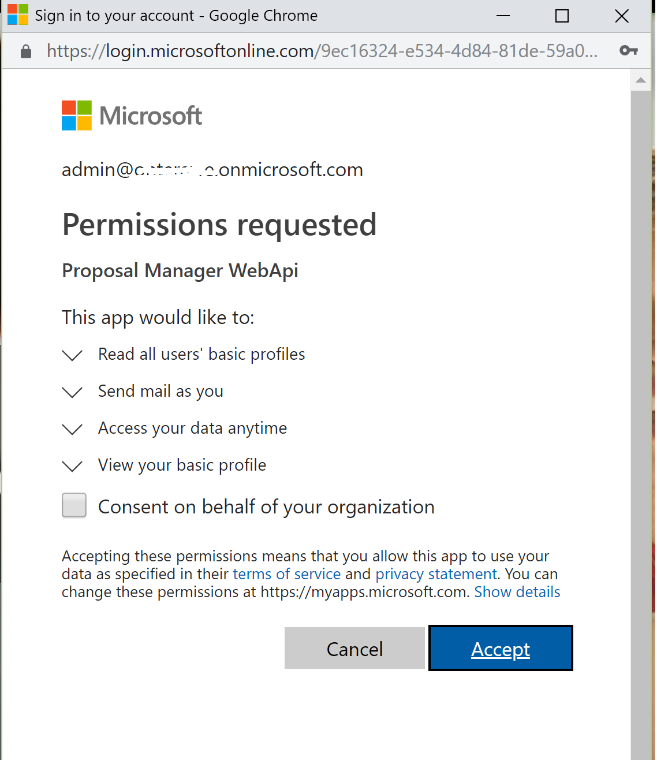


Once publish is completed, the web application is automatically launched on the default browser.

## Accept Application Consent as Admin

Once the application is deployed, logon as the admin user at:

https://<app>.azurewebsites.net



Click Accept on the list of permissions that show up. This enables the administrator to set consent on the application context. Once this step is completed, sign in.

You will get another admin consent page for Setup, which will also need to be accepted to proceed.

This sets application permissions across all users in the tenant, which will enable any user to login and use the app after the setup process has been completed.

# Teams add-in for Proposal Manager

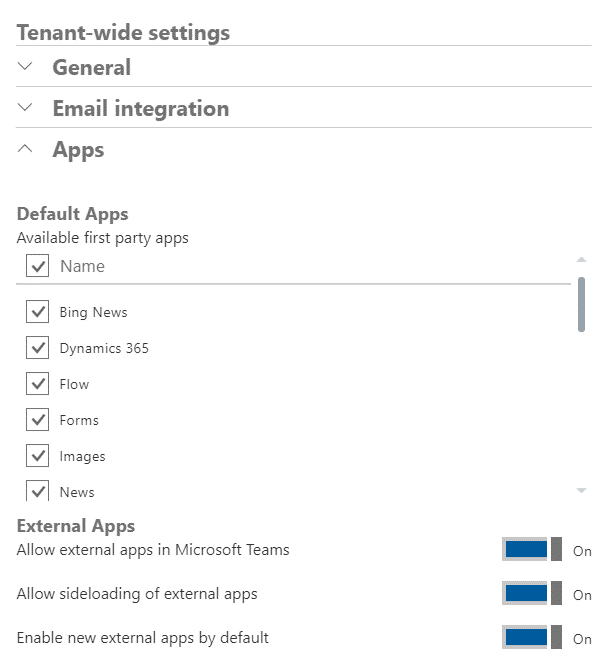
To facilitate managing the end-to-end process of handling opportunities involving multiple team members using the rich collaboration capabilities of Microsoft Teams, we have extended Microsoft Teams by adding a team for each opportunity with channels and corresponding content to address different parts of the process.

This section details how to create an add-in package which will need to be side-loaded to the root Proposal Manager team created during the Setup process.

## 7.1 Enable side-loading of add-ins

To manage admin settings for apps in Teams, go to the Office 365 admin center and open Settings > Services & add-ins, then choose Microsoft Teams. As an admin, you can directly get to this section with the following link: <https://portal.office.com/adminportal/home#/Settings/ServicesAndAddIns>

Sideloading is how you add an app to Teams by uploading a zip file directly to a team. Only team owners, or members who are granted permissions, can sideload apps into Teams.



Ensure that sideloading of external apps is enabled at the tenant level.

## 7.2 Create custom add-in for Teams

To share with Relationship Managers and Loan Officers who need to add this to the channels in the teams to access the capabilities of Proposal Manager, the app needs to be packaged as detailed [here](https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/apps/apps-package).

This is directly get from code repository

A Teams app package is a .zip file containing the following:

* **A manifest file named "manifest.json", which specifies attributes of your app and points to required resources for your experience, such the location of its tab configuration page or the Microsoft app ID for its bot.**
* A transparent "outline" icon and a full "color" icon.

Define the manifest schema for the custom add-in, as detailed [here](https://docs.microsoft.com/en-us/microsoftteams/platform/resources/schema/manifest-schema).

An example manifest for Proposal Manager solution is shared below (take care to replace <app\_name> appropriately based on the URL for accessing the solution.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| {    "$schema": "https://statics.teams.microsoft.com/sdk/v1.2/manifest/MicrosoftTeams.schema.json",    "manifestVersion": "1.2",    "version": "1.0.0",    "id": "<id>",    "packageName": "com.microsoft.proposalmanager",    "developer": {      "name": "GoLocal Solutions",      "websiteUrl": "https://<app\_name>.azurewebsites.net/",      "privacyUrl": "https://<app\_name>.azurewebsites.net/tab/privacy",      "termsOfUseUrl": "https://<app\_name>.azurewebsites.net/tab/termsofuse"    },    "icons": {      "color": "color.png",      "outline": "outline.png"    },  "name": {  "short": "Proposal Manager",  "full": "Proposal Manager Add-in for Microsoft Teams"  },  "description": {  "short": "Proposal Manager GoLocal Solution",  "full": "Proposal Manager GoLocal Solution"  },    "accentColor": "#FFFFFF",    "configurableTabs": [      {  "configurationUrl":"https://<app\_name>.azurewebsites.net/tab/config?channelName={channelName}&teamName={teamName}&groupId={groupId}&channelId={channelId}&upn={upn}",        "canUpdateConfiguration": true,        "scopes": [          "team"        ]      }    ],      "permissions": [      "identity",      "messageTeamMembers"    ],    "validDomains": [      "<app name>.azurewebsites.net",      "login.microsoftonline.com"    ]  } Guided Setup Once all the steps till Section 7 have been completed, launch the Guided Setup experience by navigating to the setup page at https://<app\_url>.azurewebsites.net/Setup and proceed as follows   |  |  | | --- | --- | | Step | Description | | 1 | First, we need to create the root Proposal Manager team to facilitate centralized administration and configuration of the solution | | 2 | Sideload the Proposal Manager add-in to the Proposal Manager team as detailed [here](#_Load_Proposal_Manager_1), and configure it to associate with the solution | | 3 | Specify the SharePoint site information where Proposal Management content is managed | | 4 | Configure user profile cache expiration and local storage prefix | | 5 | Specify the AD Group for managing Proposal Manager administrators and add default data.    This will add an entry in Role Mappings for the Administrator with the specified AD Group Name and add default data to the lists specified in Step 3.  With this, SharePoint site has been setup and the solution is ready to be used with the core features. Next two steps can be completed later based on when you wish to enable them in the organization | | 6 | Integrate the Notifications bot feature | | 7 | Configure Analytics Dashboard | |  | 8    You can obtain these values by following these steps:   1. Go to the azure portal and visualize the Proposal Manager WebApp. 2. Go to WebJobs (under “Settings”) 3. Select DocumentIdActivator from the list and click “Properties”. You should see the values on the right-hand side, similar to those displayed on the figure. | | 9 | It is recommended to maintain a local copy of the application settings file, by clicking on the Export button, to facilitate updates later.    Admin can come back to the Setup page at any time and make updates until the Finalize button is clicked. This disables the Setup page and prevents further updates. |  Load Proposal Manager add-in  |  | | --- | | **NOTE:** This step needs to be executed during the Guided Setup process after the root Proposal Manager team has been created. |   After the root Proposal Manager team has been created, Proposal Manager Administrator can now open Teams app in Web or Desktop to add the Proposal Manager to the newly created Proposal Manager team, which will be used for administration and configuration of the solution. This step is not supported in the mobile version of the Microsoft Teams app.  **Note:** Take care to use the most recent Teams add-in package for Proposal Manager shared by the administrator.  Open Teams application in Web or Desktop and navigate to the Team, whose name corresponds to the name of the root Proposal Manager team specified during Setup.  Select more options (Click on '...') next to the Team Name and choose 'Manage team'    Choose the Apps tab and click on ‘Upload a custom app’    Upload the Proposal Manager add-in package shared by the administrator – Proposal Manager will now show up as a new app in the list    On each of the channels, add the Proposal Manager add-in by clicking on the + sign next to the Files tab and choosing Proposal Manager    Proposal Manager settings page will load, click on Save to add the new tab.  The contents of the tab will be different based on the channel where you have added it.  Once the add-in has been side-loaded in this step, it is now automatically available for every Team that is created for any opportunity that is created using the solution. The same process can be followed by a Relationship Manager or Loan Officer to enable the Proposal Manager solution on the channels in the respective teams.  Each channel in the Proposal Manager team or the team corresponding to each opportunity have specific purpose facilitated by the add-in experience.  After Adding the custom app, we need to add Proposal Manager Administrator into the app as a member. |

# Configure Notifications Bot

|  |
| --- |
| **NOTE:** You can choose to complete this step later based on when you wish to enable this feature. This can be updated later from the Setup page, until Setup has been marked as Final. |

A bot is used to relay event notifications in the form of cards that get displayed in the General channel of the team created for an opportunity. Team name and Opportunity Name have a one to one mapping.

The following events are currently notified via a card in the general channel:

* Opportunity creation – This is applicable only in the rare scenario of a team (with the same name as the opportunity) already being present.
* Opportunity status changes – Any action in the application that changes the state of the opportunity.
* Update of status in Checklist related channels
* Assigning of a proposal document section to an owner

The registered bot has to be mentioned in the manifest file in the below section along with the given scopes.

|  |
| --- |
| "bots": [  {  "botId": {Bot id},  "scopes": [  "personal",  "team"  ]  }  ] |

The bot details should also be added in appsettings.json as below in the section ProposalManagement:

"BotServiceUrl": "https://smba.trafficmanager.net/amer-client-ss.msg/",

"BotName": {Bot Name},

"BotId": "{Bot Id}"

Refer [here](https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/bots/bots-create) to learn more about implementing and integrating bots.

# Configure Analytics Dashboard

|  |
| --- |
| **NOTE:**   * You can choose to complete this step later based on when you wish to enable this feature. This can be updated later from the Setup page, until Setup has been marked as Final. * This step can be completed only after Step 5 of Guided Setup, as the Dashboard list created during site setup is required for configuration |

**Prerequisites**

Make sure that you have downloaded and installed [Power BI Desktop](https://powerbi.microsoft.com/en-us/desktop/) to proceed with the setup

Identify a user who will need to be assigned a Power BI Pro license to be able to access the dashboard. This is required to facilitate the “apps own data” model for embedded Power BI workspaces

|  |  |
| --- | --- |
| Step | Description |
| 1 | Register an Azure AD application as detailed [here](https://docs.microsoft.com/en-us/power-bi/developer/register-app). Take care to ensure that the app is configured as **Application type: Native** |
| 2 | Navigate to the Power BI [portal](https://powerbi.microsoft.com/en-us/landing/signin/) and create an application workspace as detailed [here](https://docs.microsoft.com/en-us/power-bi/service-create-distribute-apps) |
| 3 | Load the Dashboard template file, dashboard-template.pbit, located in the solution folder at ./SolutionItems |
| 4 | An empty dashboard is now displayed in the workspace: |
| 5 | Note the URL for the Dashboard list in the Proposal Manager SharePoint site |
| 6 | Click on ‘Get Data’ from the top ribbon and set the Dashboard list URL as the data source |
| 7 | Make sure that all measures, set as detailed below, are of type ‘whole number’. For this, select each measure in Power BI Desktop, click on ‘Modeling’ tab on top and set the data type to ‘Whole Number’.   |  |  | | --- | --- | | Measure Name | Formula | | Accepted | *IF(ISBLANK( CALCULATE( COUNTA('DashBoard'[Status]), 'DashBoard'[Status] IN { "Accepted", "Completed" }*  *)),0,CALCULATE( COUNTA('DashBoard'[Status]), 'DashBoard'[Status] IN { "Accepted", "Completed" } ))* | | Archived | *IF(ISBLANK(*  *CALCULATE(*  *COUNTA('DashBoard'[Status]),*  *'DashBoard'[Status] IN { "Archived" }*  *)),0,CALCULATE(*  *COUNTA('DashBoard'[Status]),*  *'DashBoard'[Status] IN { "Archived" }*  *))* | | Completed | *CALCULATE(*  *COUNTA('DashBoard'[Status]),*  *'DashBoard'[Status] IN { "Completed" }*  *)* | | Open | *IF(ISBLANK(CALCULATE(*  *COUNTA('DashBoard'[Status]),*  *'DashBoard'[Status] IN { "Creating","InProgress" })),0,CALCULATE(COUNTA('DashBoard'[Status]),*  *'DashBoard'[Status] IN { "Creating","InProgress" }))* | | FormalProposalCompletedCount | *NOT(CALCULATE(*  *COUNTA('DashBoard'[FormalProposalEndDate]),*  *(ISBLANK('DashBoard'[FormalProposalEndDate]))*  *))* | | ComplianceReviewCompletedCount | *NOT(CALCULATE(*  *COUNTA('DashBoard'[ComplianceRewiewCompletionDate]),*  *(ISBLANK('DashBoard'[ComplianceRewiewCompletionDate]))*  *))* | | CreditCheckCompletedCount | *NOT(CALCULATE(*  *COUNTA('DashBoard'[CreditCheckCompletionDate]),*  *(ISBLANK('DashBoard'[CreditCheckCompletionDate]))*  *))* | | RiskAssessmentCompletedCount | *NOT(CALCULATE(*  *COUNTA('DashBoard'[RiskAssesmentCompletionDate]),*  *(ISBLANK('DashBoard'[RiskAssesmentCompletionDate]))*  *))* | |  | ***Note for the measures* FormalProposalCompletedCount *,* ComplianceReviewCompletedCount, CreditCheckCompletedCount, RiskAssessmentCompletedCount, once it got added then**   1. **Click on each measure** 2. **Go to Modeling tab** 3. **Select “Data Type” as whole number** | | ValueOfTotalNoOfDays | *VALUE(DashBoard[TotalNoOfDays])*  *This is created as a column and not a measure* | | Average Time (in Days) to close an opportunity | *IF(DashBoard[Completed]=0,0,SUM(DashBoard[ValueOfTotalNoOfDays])/DashBoard[Completed])*  *First create as ‘AverageTime’, later select the first visualization and rename this as the above in Fields section in Power BI Desktop* | | AverageCompReviewTime | *SUM(DashBoard[ComplianceReviewNoOfDays])/DashBoard[ComplianceReviewCompletedCount]* | | AverageCreditAnalyisTime | *SUM(DashBoard[CreditCheckNoOfDays])/DashBoard[CreditCheckCompletedCount]* | | AverageFormalProposalTime | *SUM(DashBoard[FormalProposalNoOfDays])/DashBoard[FormalProposalCompletedCount]* | | AverageRiskAssessmentTime | *SUM(DashBoard[RiskAssessmentNoOfDays])/DashBoard[RiskAssessmentCompletedCount]* | |
| 8 | Once the measures have been configured, confirm that the reports are loading fine. Note that until the first opportunity has been created, the lists will be empty. |
| 9 | Publish the report to the workspace created in Step 2 |
| 10 | Login to <https://app.powerbi.com> with the credentials of the licensed user and go to the specific workspace |
| 11 | Ensure that the licensed user has access to the published report |
| 12 | Note the Report ID and Workspace ID that will be required to integrate the Power BI report with Proposal Manager |

# Office add-in for Proposal Creation

The Word add-in for Proposal Creation facilitates collaborative finalization of the Proposal document, with support for advanced features such as assignment of owners for sections and creation of tasks associated with specific sections. This is integrated with the Proposal Manager solution and activities are published as cards in the General channel in the Team associated with the Opportunity.

## 12.1 Deploying Office add-in

Please refer to the Proposal Creation add-in documentation for details on deploying the add-in on a new tenant.

# API Guidance

The guide for using the API is present at ./APIGuide. To view the API documentation, right click on the solution, select ‘Open Folder in File Explorer’, and navigate to the root directory. The folder named APIGuide contains the API documentation.

# User Experience Overview

This section details at a high level, how different personas interact with the solution.

This section details the different user personas interacting with the Proposal Management solution, their experience across different interfaces and end-to-end Proposal Management process for Corporate Lending, facilitated by the solution.

## 14.1 User Personas

Solution considers 3 key personas as part of the team handling the opportunity, along with addition team members as defined in the user role mapping at the tenant level:

|  |  |
| --- | --- |
| Persona | Description |
| Relationship Manager | Owner of the opportunity, who identifies the opportunity, enters it into the system and drives it to a conclusion. Takes care of providing relevant documents to the participants involved in the proposal process, and acts as the point of contact with the customer |
| Loan Officer | Owns the proposal preparation process and forms the team handling the opportunity, based on decision taken on loan type and specific process to be followed for arriving at a proposal for the same |
| Credit Analyst | Example roles - Part of the opportunity team, selected by the loan officer. There could be one or more people associated with each role based on the process type, taking into consideration factors such as industry, geographic region and opportunity size. The solution is flexible to adapt to more personas to be part of the team. |
| Legal Counsel |
| Risk Officer |
| Proposal Manager Administrator | Dedicated administrator(s) in the tenant authorized to handle the administration and configuration of Proposal Manager. For each opportunity that is created, the Admin has to add the add-in for the team to be able to use the solution. |

## 14.2 User Interfaces

Different members of the opportunity team can interact with the solution using two different interfaces:

|  |  |
| --- | --- |
| Team Type | Description |
| Proposal Manager Team | Root team for Proposal Manager where the administrator can perform the administration and configuration of the solution and add the add-in for each opportunity. Relationship Manager has the ability to create an opportunity and Loan Officer can add or remove team members, choose deal type and upload the Proposal Document template used for the opportunity from this team |
| Opportunity Team | A team is formed in MS Teams for each opportunity, and all members of the opportunity team are automatically added to the same. Each Team consists of different channels to facilitate collaboration, and the Proposal Manager add-in enables specific process steps across all personas on each of the channels |

## 14.3 Key Entities

The Proposal Management process is built around three main entities – the Opportunity that gets processed by an internal team, the specific process/workflow identified to validate and move the opportunity forward, and the associated Proposal that is prepared as an outcome of the process.

**Opportunity**

An Opportunity is a potential deal identified by the Relationship Manager for one of his/her own clients based on knowledge gathered from customer discussions and market analysis. He then uses the Proposal Manager application to convert the opportunity, by means of well-defined corporate lending processes and a hand-picked team of experts brought together to execute the process, to a proposal document that can then be presented to the customer, offering a loan or line of credit that they can use in line with the terms and conditions.

Opportunity consists of the following key details:

* Client Information
* Relevant notes and documents
* Specific info for lending process

Opportunity is considered as Complete when a decision has been taken on the proposal by the customer.

**Workflow**

Specific process to be followed for an opportunity is based on the type of loan determined to be the optimal fit as per the loan officer, who takes the decision based on different considerations such as deal size, past history with the client and other details. The solution supports different types of work flows that can be created from the Deal Types section in the Configuration channel in the Proposal Manager team.

**Proposal Document**

Proposal document, to be presented to the customer by the Relationship Manager, for the customer’s review and decision is the final deliverable from the Proposal Management process for each opportunity. Each loan type and process can be associated with a template chosen by the loan officer at the beginning of the process.

A proposal document is composed of clearly defined sections, each with a specific purpose and expected set of details to be updated based on inputs gathered from documents or information obtained from different steps of the process. Owners are identified for each section based on which step of the process is associated with the same. Loan Officer owns the end-to-end process for preparation of a formal document, that is then handed off to the Relationship Manager.

## 14.4 User Permissions

Different types of users with associated permissions are as follows:

* Regular user
  + profile, User.ReadBasic.All, mail.send
* App Admin user (user that adds the add-in for each opportunity)
  + profile, User.Read.All, mail.send, Sites.ReadWrite.All, Files.ReadWrite.All, Group.ReadWrite.All
    - The above permissions needs to be granted by a tenant admin (via admin consent)
* App context
  + Email, profile, User.Read.All, Mail.Send, Sites.ReadWrite.All, Files.ReadWrite.All, Group.ReadWrite.All, Directory.ReadWrite.All
    - The above permissions needs to be granted by a tenant admin (via admin consent)
* Tenant admin
  + Grants the permissions as noted above and config/deploy the system
  + Authorize the app in Azure and initial SharePoint site (details in deployment guide)

# Proposal Management Process

This section details the end-to-end process and defines how each team member interacts with the solution as part of the process.

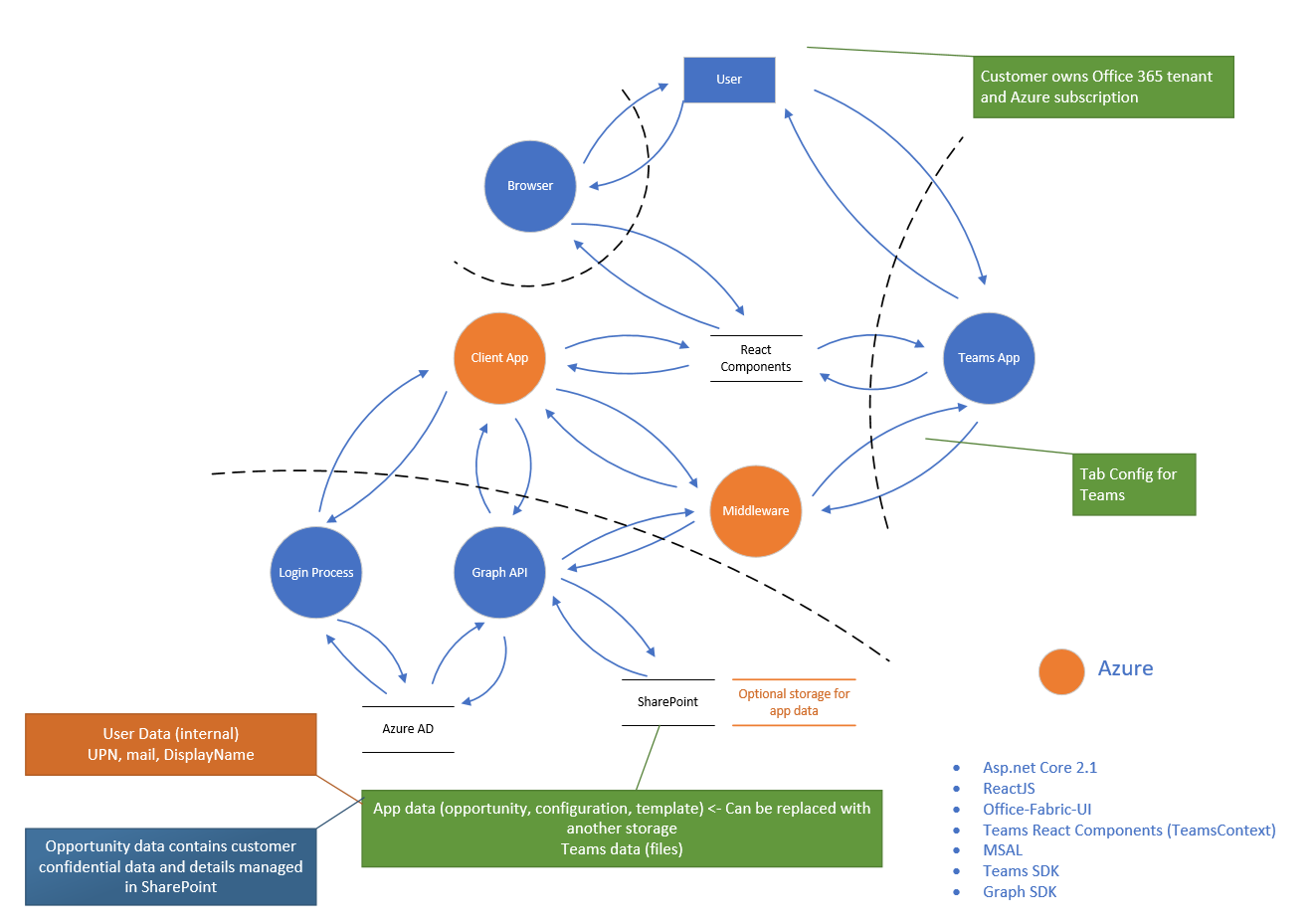
## 15.1 High Level Workflow

At a high level, the process starts with identification of an opportunity and ends with the customer taking a decision on a proposal document prepared during the course of the process.

1. Relationship Manager identifies a corporate lending opportunity for one of his/her clients and creates a new opportunity using the Proposal Manager application
2. Opportunity is assigned to a loan officer who identifies the process to be followed and forms a team
3. Team members collaborate and perform their assigned responsibilities at different stages of the process to prepare a Proposal document
4. Relationship Manager presents the proposal document to the client for review and decision
5. Client conveys his decision on if the proposal has been Accepted or Rejected

## 15.2 Data Flow

Shared below is a high level data flow showcasing the different components and how they interact with each other:



## 15.3 User Interaction Flow

This section details the interaction experience for each persona.

### 15.3.1 Proposal Manager Team Experience

**Relationship Manager**

Relationship Manager owns the opportunity, from creation to closure.

1. Relationship Manager identifies an opportunity and creates it in the system
   1. Documents can be uploaded as part of the opportunity creation process, which gets copied to the General channel in corresponding Team once created
   2. Metadata defined during the opportunity creation process cannot be updated later
2. Sees a list of all opportunities associated with the user in the Opportunities list
3. Select an opportunity to see a summary view of the opportunity, with all relevant information and quick access to team members
4. For each opportunity:
   1. Review process workflow at a high level
   2. Add/update notes about the opportunity, visible only to the Relationship Manager
   3. Assign loan officer if not assigned, or Change loan officer

**Loan Officer**

Loan Officer owns the Proposal Document, from identification to formalization.

1. Sees a list of all opportunities associated with the user in the Opportunities list
2. Select an opportunity to see a summary view of the opportunity, with all relevant information and quick access to team members
3. For each opportunity:
   1. Choose the deal type, which triggers the following steps:
      1. Create a Team with same name as opportunity
      2. Create pre-defined channels in each team, based on the process steps in the deal type, each with a specific purpose
      3. Copy documents uploaded at the time of opportunity creation to the Files tab in the General channel
   2. Select team members for each role and finalize the team working on the opportunity
   3. Review process workflow at a high level
   4. Edit team by adding or removing specific team members

**Proposal Manager Administrator**

Administrator is a designated admin on the tenant who is authorized to manage the administration of the solution and add the add-in for teams associated with new opportunities

1. Access Administration page from the Administration channel in the Proposal Manager
2. For opportunities listed in ‘Requires Action’ tab, review opportunities with status ‘Creating’, and click on the Action button, which automatically adds the Proposal Manager add-in to the Team corresponding to the selected opportunity

**Other Personas**

All other personas are not currently supported by default in the Proposal Manager Team.

### 15.3.2 Opportunity Team Experience

A Team is formed in MS Teams for each opportunity, with everyone working on the opportunity added as members.

A custom add-in, Proposal Manager, facilitates the process within Teams. The add-in can be loaded automatically for each Team associated for an opportunity by following the configuration steps [here](#_Load_Proposal_Manager). This add-in needs to be shared with the team members by the administrator and should be added to all the channels manually by Relationship Manager or Loan Officer. Every channel in Teams have Conversation and Files tabs by default, which facilitates the collaborative experience.

The Team for an opportunity, consists of the following channels by default, each with a tab for Proposal Manager add-in, which in turn consists of one or more sections.

|  |  |  |
| --- | --- | --- |
| Channel | Section | Description |
| General | Workflow | * Overview of the process workflow with the owner and status for each step |
| Team Updates | * Team overview, with quick access links, and status |
| Channel for each process step in the deal type | Checklist | Example channels in line with the three personas listed as examples in [User Personas](#_15.1_User_Personas)   * Checklist for each process step * Ability to upload documents or download uploaded documents for each checklist item * Mark status as Completed, In Progress or Blocked at the process level |
| Formal Proposal | Proposal Status | * Overview of the status of different sections of the proposal document, updated by the loan officer |
| Customer Decision | Customer Decision | * Decision by the customer on the final proposal and the details of loan disbursement, if applicable |

# Extensibility

Proposal Manager is designed to be extensible to support potential new requirements for partners and customers across multiple industry verticals and regions.

# Maintenance

This section details some of the common maintenance scenarios.

### Update AD users and groups

Managing users and their roles are a key administrator responsibility.

The activities in this regard can all be performed from the Office 365 Admin portal at https://portal.office.com/adminportal

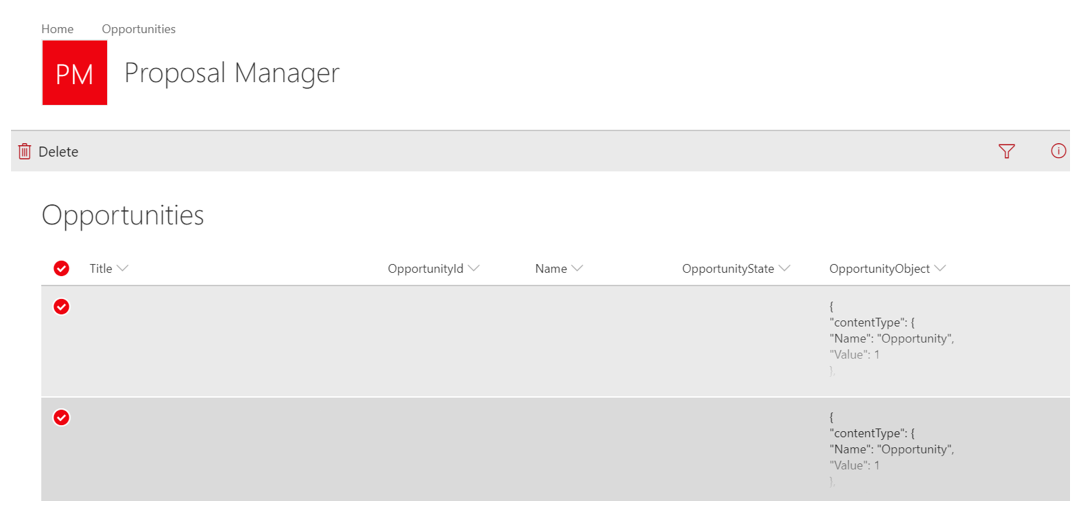
|  |  |
| --- | --- |
| Add a new user | On the admin portal, click on Add a user under the Active Users section, provide required details and click on Add    Navigate to the Groups list at https://portal.office.com/adminportal#/groups and add the newly added user to the group corresponding to their role |
| Update user role | Navigate to the Groups list at https://portal.office.com/adminportal#/groups and add or remove users from the groups corresponding to their role |
| Remove user | On the admin portal, click on Delete a user under the Active Users section |

### Clear SharePoint site

It may sometimes be necessary for the administrator to clear the list of opportunities from the system, such as for archiving reasons or to reset a test or prototype environment.

To delete all opportunities from the application, so as to have a clear dashboard for all users, proceed as follows:

* Navigate to the SharePoint site setup for Proposal Manager, and select the list for Opportunities
* Choose Select All from the top left of the list, and then click Delete.



Note that when cleaning up the Opportunities list, it is recommended to also delete the corresponding Teams manually by logging into Microsoft Teams.

### Update master data

An admin can maintain master data stored in SharePoint lists via the Settings pages accessible from the Configuration channel in the Proposal Manager team.

# Security Considerations

After app has been deployed and tested ad before go live, the following steps are recommended for the production environment:

* Lockdown SharePoint site: Remove all users from members list
* Replace app secret with certificates as detailed in Azure AD [guidelines](https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-protocols-oauth-service-to-service)

# Troubleshooting

|  |  |
| --- | --- |
| **Error** | **Recommended Solution** |
| “Reply-to address does not match one defined for the application” | Confirm that the reply-to URLs specified for the application in the Application Registration Portal exactly matches the reply-to address specified in the error message |
| Setup page shows 500 error after first time successful load on Azure | Scale up the web app to add more memory to see if it resolves the issue |
| Landing page gets stuck at “Loading your experience” | Make sure that pop-ups are enabled on the site to facilitate opening the sign-in window |
| Clicking on the Proposal Manager add-in link from a channel in Teams Mobile app opens a page stuck in “Loading” | Logon to the account in another window on the default browser in the phone on some other site as http://portal.microsoftonline.com before clicking on the add-in link in Teams |

Local Test and Debug

To test the solution locally against the application setup in the tenant, update the appsettings.json appropriately as indicated above, and then set the BaseUrl to a local value such as https://localhost:44385/ Make sure that this is added as a valid reply-to URL in the application as detailed [here](#_Registering_Time_Tracker)

Before building the solution for the first time, take care to navigate to .\WebReact\ClientApp from the command line, and run npm install to install all dependencies

Now run the application in Debug mode from Visual Studio by using the debug controls.

[Debugging in Visual Studio](https://docs.microsoft.com/en-us/visualstudio/debugger/index)

# Known Issues

This section lists some of the key known issues:

* File selection and upload does not work on mobile devices
* Proposal Manager has been validated only on Microsoft Edge, Google Chrome and Mozilla on Desktop and on the default browser on iOS and Android phones
* Opportunities created from Dynamics 365 using the Web Hook will not show up in the dashboard
* In the mobile, after the authentication, proposal manager auth pop-up won’t close by itself, from a dev standpoint this mobile auth issue can be addressed by adding 7 URLs explicitly to app settings.

# Appendix: Permissions List

This section gives an overview of the different permissions that can be set for an AD Group from the Role Mapping list page to setup granular access control.

|  |  |
| --- | --- |
| Permission | Permitted Actions |
| Opportunity\_Create | Create an opportunity |
| Opportunity\_Read\_All | Read everything related to an opportunity |
| Opportunity\_ReadWrite\_All | Read and write every aspect related to an opportunity  (A super set of all permissions so if it exists then the user gets complete access to an opportunity) |
| Opportunity\_Read\_Partial | Read a portion of an opportunity, example credit analyst will only able to read , general area and credit check area of an opportunity |
| Opportunity\_ReadWrite\_Partial | Read and write a portion of an opportunity, example credit analyst will only able to read , general area and credit check area of an opportunity |
| Opportunities\_Read\_All | Same as Opportunity\_Read\_All, but can read every opportunity irrespective of whether the person is a team member of that opportunity |
| Opportunities\_ReadWrite\_All | Same as Opportunity\_ReadWrite\_All but can read and write every opportunity, irrespective of whether the person is a team member of that opportunity |
| Opportunity\_ReadWrite\_Team | The user can add or remove any team member disable button if this permission doesn’t exist. |
| Opportunity\_ReadWrite\_Dealtype | The deal type dropdown/Start Process will be disabled if the user doesn’t have this permission. Enable the deal type tab if this permission exists. |
| Administrator | Access to administration channel in Proposal Manager Team, and with this permission he can create dealtypes, permissions, regions, industries and categories |
| CustomerDecision\_Read(part of the superset) | Read customer decision channel |
| CustomerDecision\_ReadWrite(part of the super set) | Read and write customer decision channel |
| CreditCheck\_Read(part of the superset) | Read customer credit check channel |
| CreditCheck\_ReadWrite(part of the superset) | Read and write credit check channel |
| Compliance\_Read(part of the superset) | Read customer compliance channel |
| Compliance\_ReadWrite(part of the superset) | Read and write compliance channel |
| RiskAssessement\_Read(part of the superset) | Read customer risk assessment channel |
| RiskAssessement\_ReadWrite(part of the superset) | Read and write risk assessment channel |
| ProposalDocument\_Read(part of the superset) | Read customer proposal document channel |
| ProposalDocument\_ReadWrite(part of the superset) | Read and write proposal document channel |