Governance_Office365_App

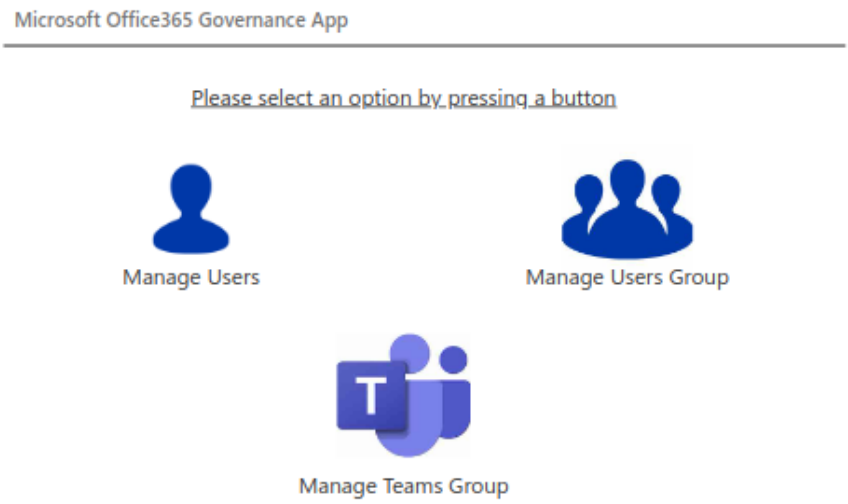
PowerApp Documentation

21/08/2022

Author: Manuel Portero Leiva

Introduction

This document has the purpose to explain the different parts of the Governement_Office365_App, its code and functionalities, for understanding and replication purposes. The different parts of the architecture solution are show below.



Picture 1: Governement_Office365_App Layout

Architecture

The composition of the architecture starts in the PowerApp, The Users or different groups are represented in their respective galleries. After any groups or user request, the respective flow is activated and the user or group is created.

Main Screen

The main screen or navigation screen is composed by three buttons , the manage users button will navigate the user to the Users Screen, the manage groups button will navigate the user to the Groups Screen and the Manage Teams Group will manage the user to the Teams Main Screen.

Picture 2: Governement_Office365_App main Screen

The Code On start of the Main Screen will load the teams users and groupusers collection,a part of that the login user and the admin user is setted onStart. A sample of the onStart code is shown below :

```
//Collect all the teams of logged in user
ClearCollect(
    colMyTeams;
    MicrosoftTeams.GetAllTeams().value
);;
//Collect all the users of the tenant
ClearCollect(
    colMyUsers;
    Office365Users.SearchUser()
);;
//Collect all the groups of the tenant
ClearCollect(
```

```
    colMyGroups;
    GruposdeOffice365.ListGroups().value
);;
//Set logged in user to a variable
Set(
    LoggedInUser;
    User()
);;
//Set admins here to show login as admin buttton on home screen only to admin users
Set(
    AdminUsers;
    [
        "manuelportero@t8656.onmicrosoft.com"//,
        //"adminEmailId@yourOrg.com"
    ]
)
```

Users Screen

The Users Screen has a Gallery displaying a users collection, a reload and a Request a new User button.

Picture 5: Users Screen

The code of the reload button is shown below :

```
ClearCollect(
    colMyUsers;
    Office365Users.SearchUser()
);;
Notify("Users refreshed successfully";Success)
```

Groups Screen

The Groups Screen has a Gallery displaying a groups collection, a reload and a Request a new group button.

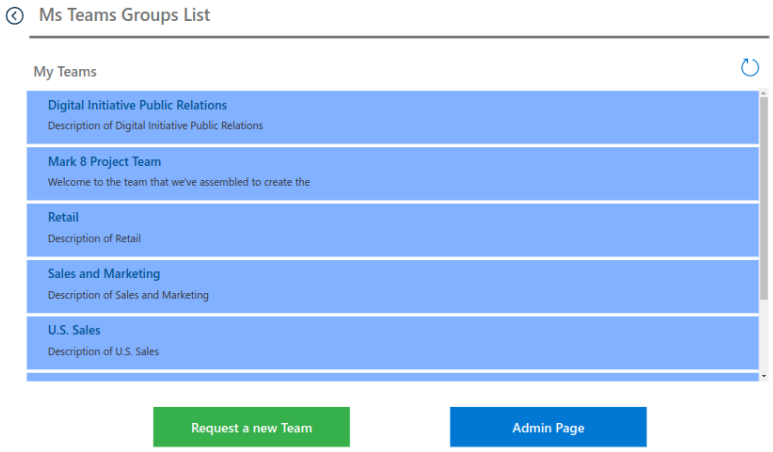
Picture 5: Groups Screen

The code of the reload button is shown below :

```
ClearCollect(
    colMyGroups;
    GruposdeOffice365.ListGroups().value
);;
Notify("Groups refreshed successfully";Success)
```

Teams Screen

The Teams Screen has a Gallery displaying a Teams groups collection, a reload and a Request a new teams group button and an Admin Page button.



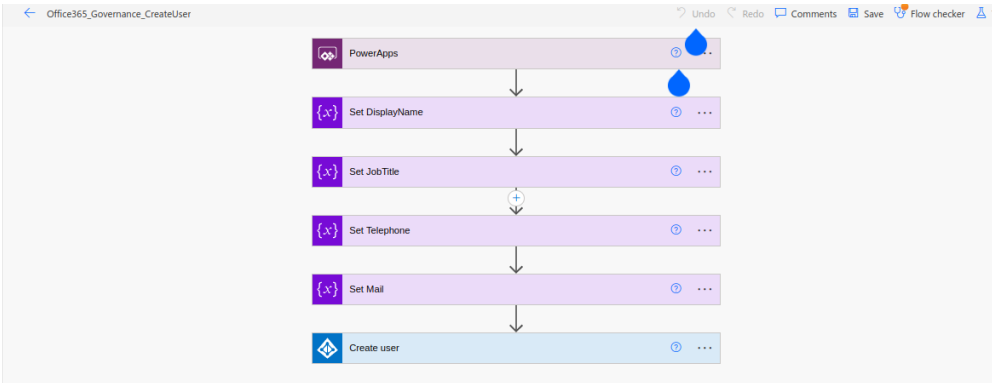
Picture 5: Teams Groups Screen

The code of the reload button is shown below :

```
ClearCollect(
    colMyTeams;
    MicrosoftTeams.GetAllTeams().value
);;
Notify("Teams Groups refreshed successfully";Success)
```

Office365_Governance_CreateUser Flow

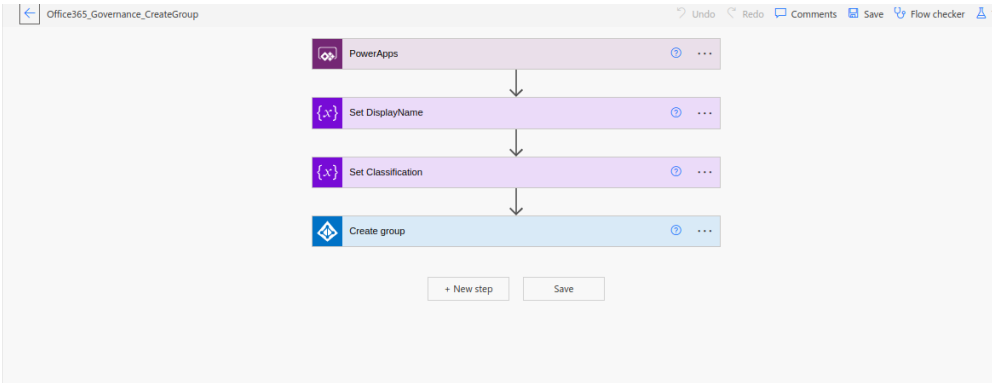
The Office365_Governance_CreateUser Flow will receive the variables from the powerapp once pressed the powerapp “create user” button and will store them in the DisplayName, JobTitle, Telephone and Mail variables. After that a create user action in Azure AD will create the user



Picture 6: Office365_Governance_CreateUser Flow

Office365_Governance_CreateGroup Flow

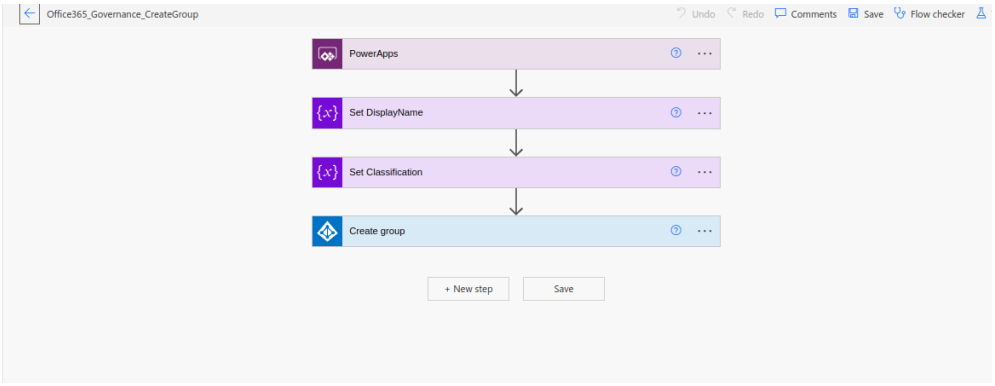
The Office365_Governance_CreateGroup Flow will receive the variables from the powerapp once pressed the powerapp “create group” button and will store them in the DisplayName and Classification variables. After that a create group action in Azure AD will create the group



Picture 6: Office365_Governance_CreateGroup Flow

Office365_Governance_CreateGroup Flow

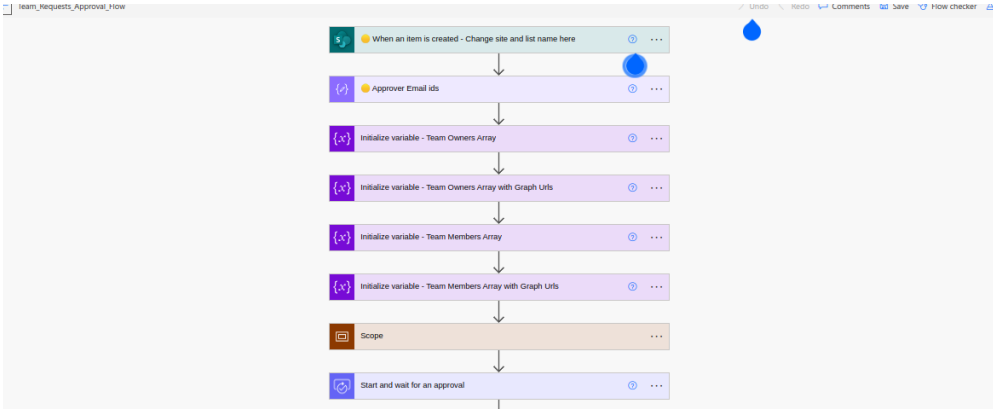
The Office365_Governance_CreateGroup Flow will receive the variables from the powerapp once pressed the powerapp “create group” button and will store them in the DisplayName and Classification variables. After that a create group action in Azure AD will create the group



Picture 7: Office365_Governance_CreateGroup Flow

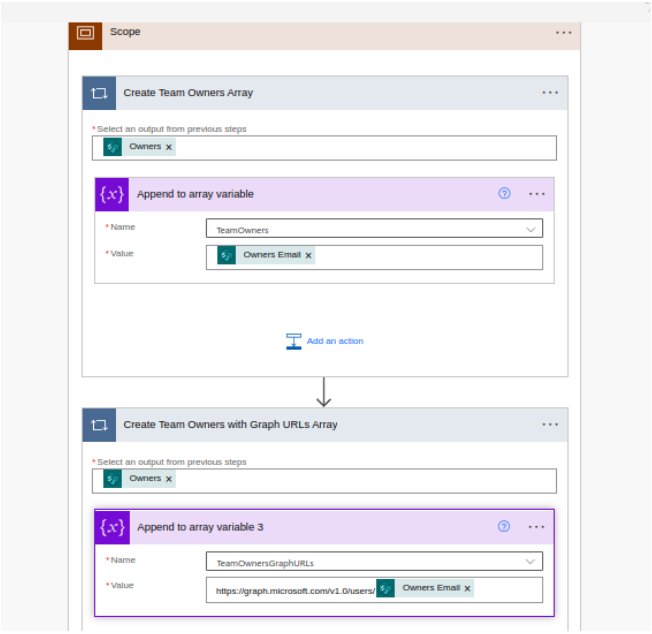
Team_Request_Approval Flow

The Team_Request_Approval Flow will be actionated from the powerapp when we press the create teams gropup button. When an item in the Sharepoint list ist added or modified the Teams_Request_Approval flow will be triggerd and the variables from the item will be stored in the Team Owners and Team Members array. After that an approvals with all the information to the new group will be sent to the Approval manager. If the manager approves an http Ad Premium will post a create Office 365 group action, and after a delay of 30 seconds (important because otherwise the next action would not be able to see the created Office365 group) another HTTP AD post action will create the Teams group. Finally the details of the request will be added to the approvals sharepoint list. If the manager reject the approval the details of the request will be added to the approvals sharepoint list.

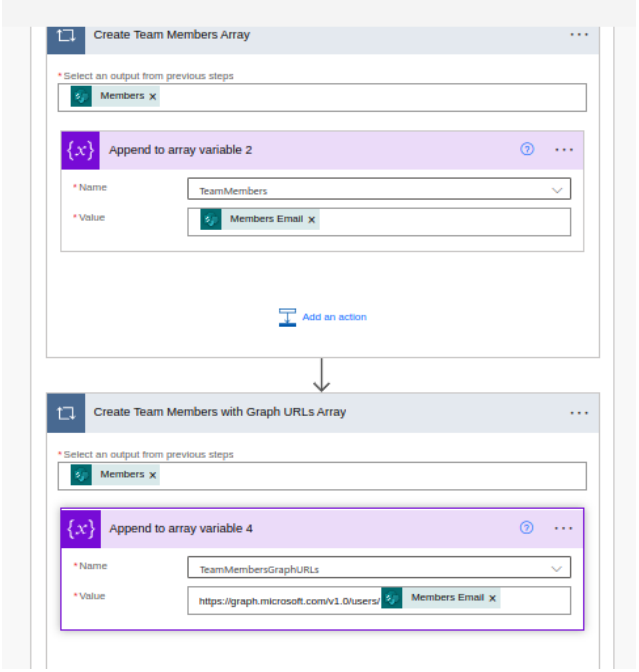


Picture 8: Office365_Governance_CreateGroup Flow

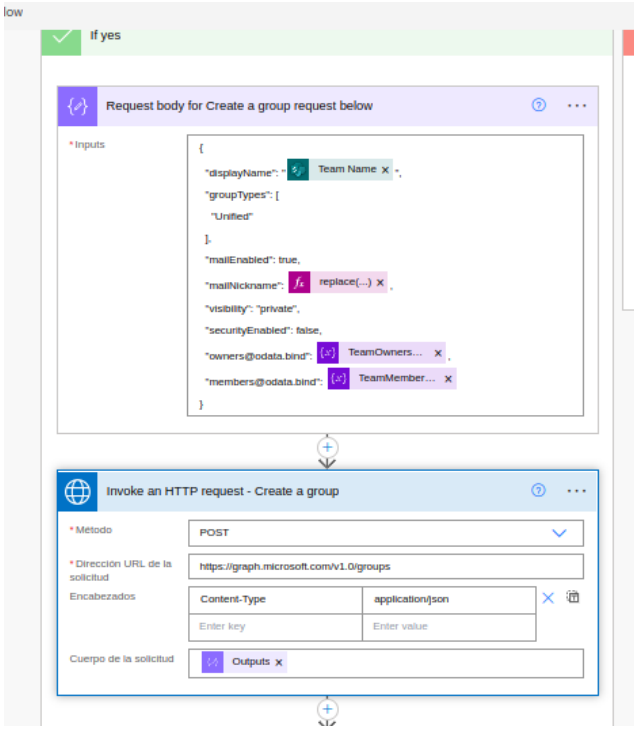
The details of the different post actions are shown below:



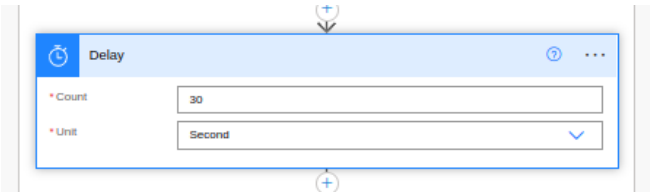
Picture 9: Arrays appending operation



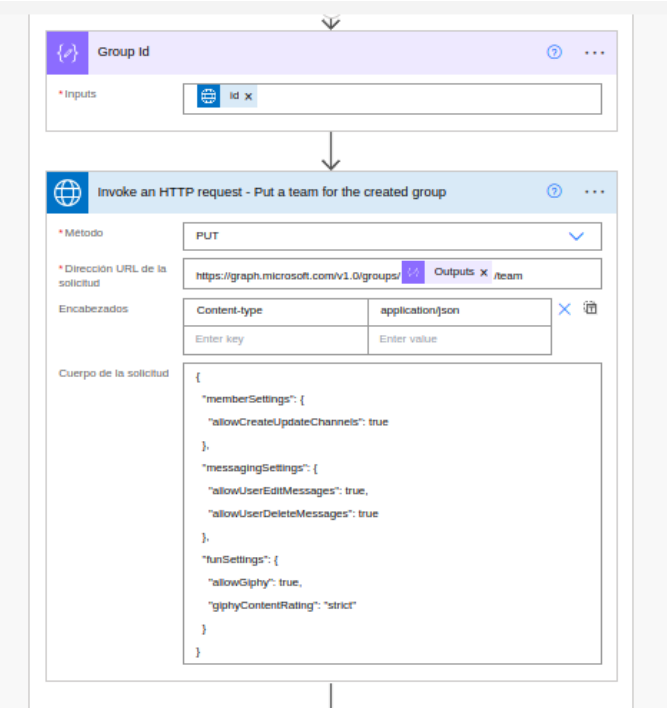
Picture 10: Arrays appending operation



Picture 11: Office365 Http Post Group



Picture 12: Delay in between Http Posts



Picture 13: Http Teams group Post