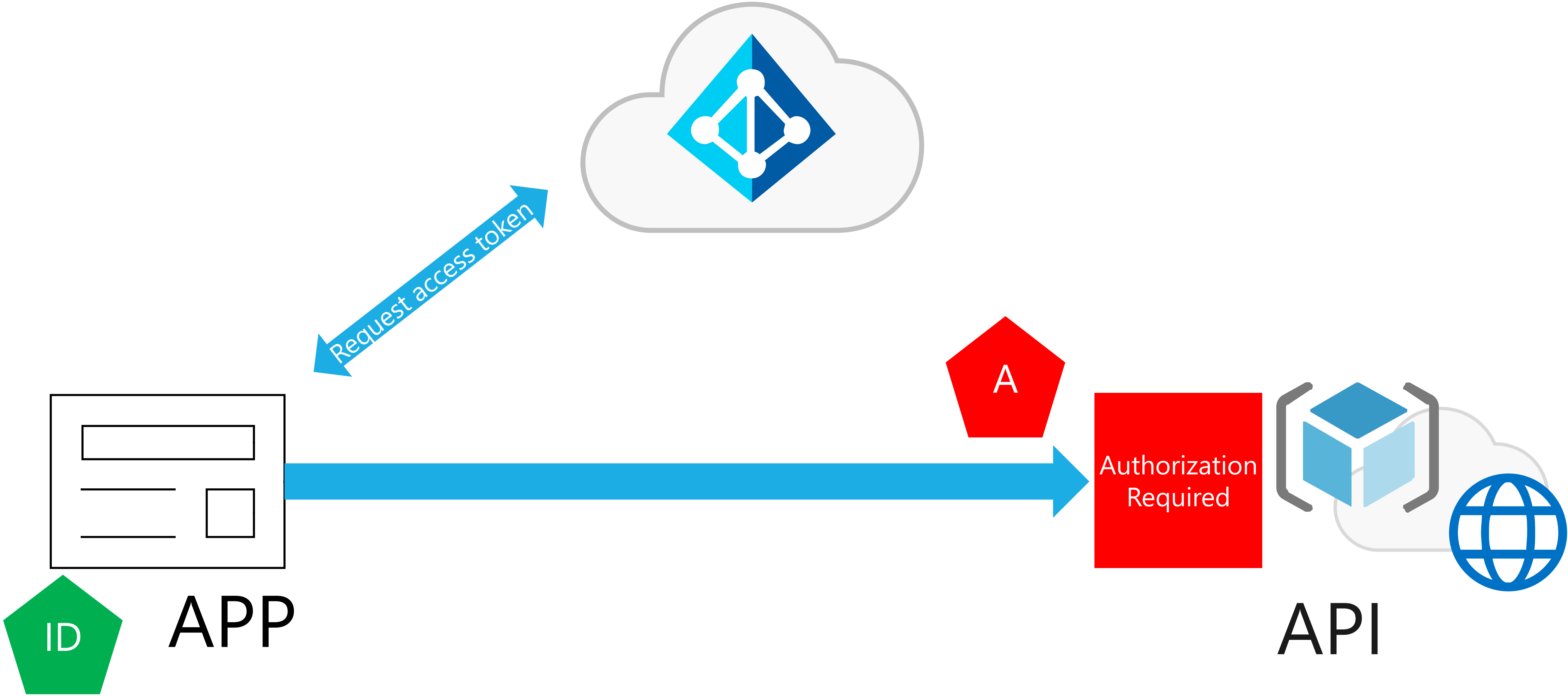
Microsoft

Microsoft Identity platform
Developer training

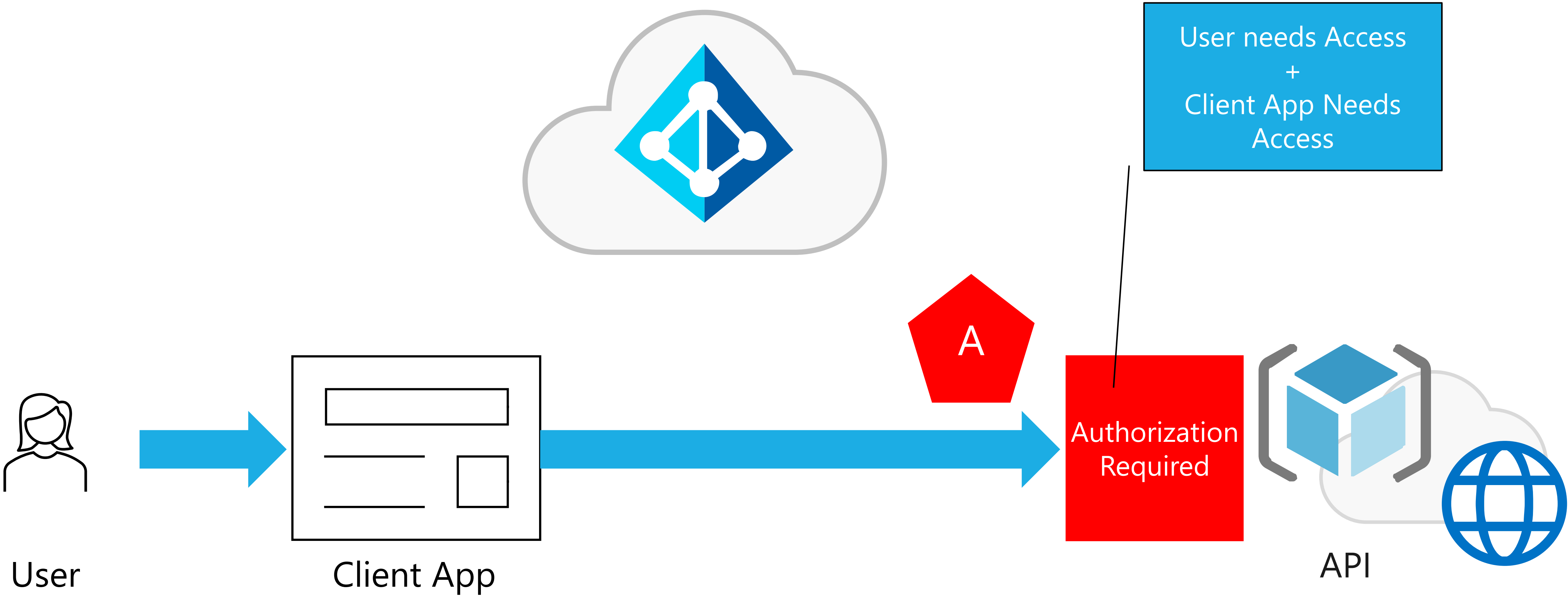
Permissions and Consent

An Authorization Recap

Call an API with an Access_Token

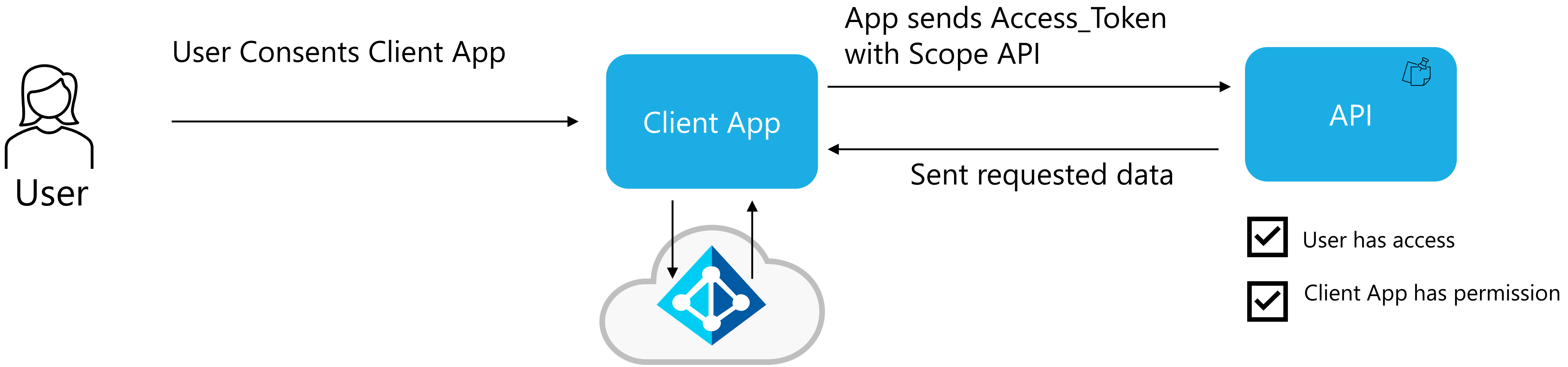


What does your application authorize on?



What “does Client App needs access” means

- The API delivers a number of Services
- The Client App want to use these services on behalf of the user
- The User needs to **Consent** the Client App to have permission to perform the services on behalf of the user
- The Client App requests access_token that contains the permissions as scopes



User Consents Client App

App sends Access_Token with Scope API

Sent requested data

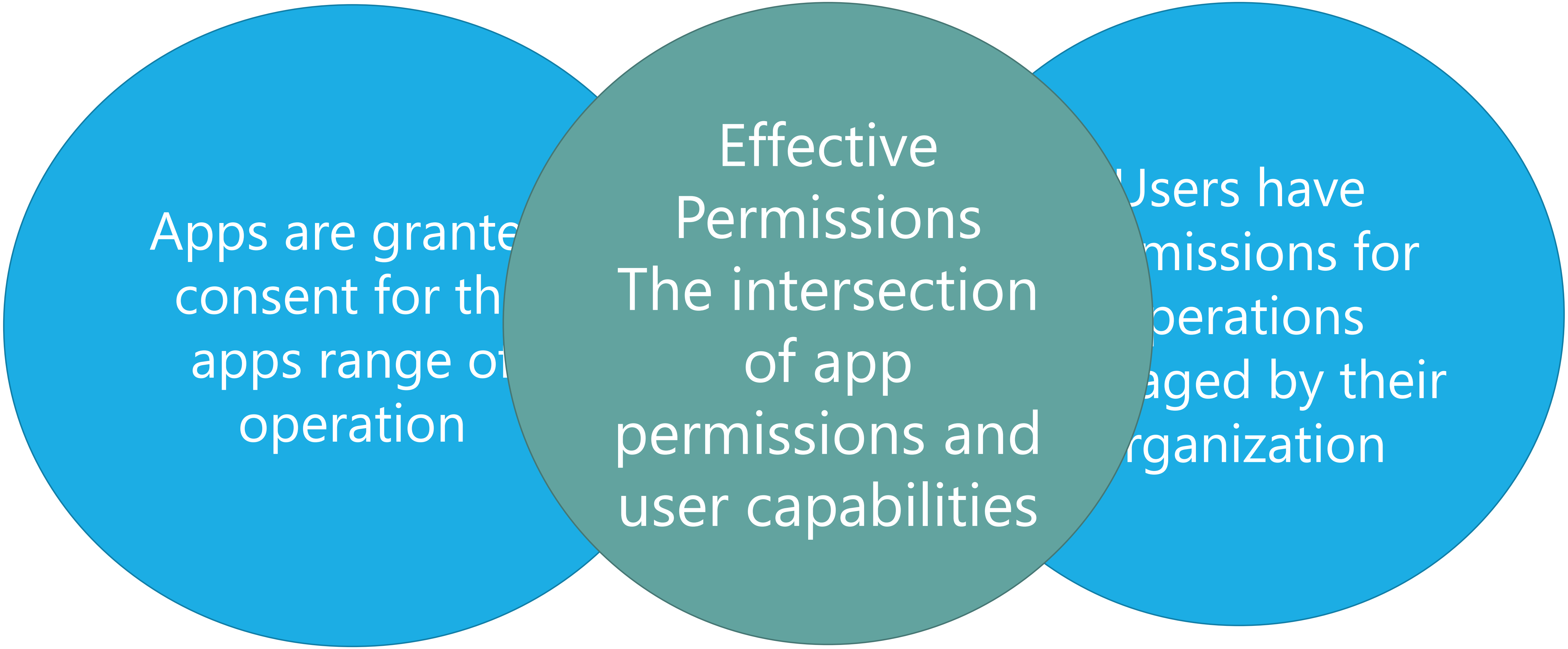
☑ User has access

☑ Client App has permission

Apps are granted consent for the apps range of operation

Apps are granted consent for the apps range of operation

Users have permissions for operations managed by their organization



Apps are grante
consent for th
apps range o
operation

Effective
Permissions
The intersection
of app
permissions and
user capabilities

Users have
missions for
perations
aged by their
rganization

How does Microsoft use Permissions and Consent?

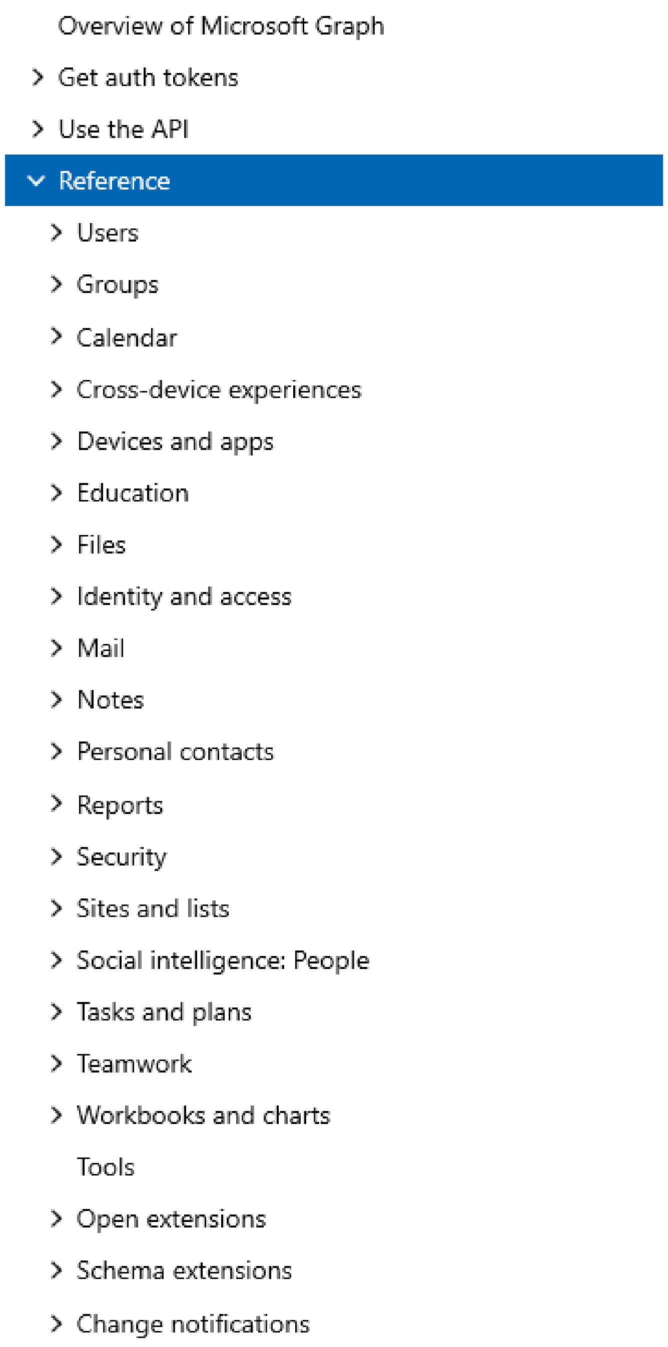
https://graph.microsoft.com

Permission	Display String	Description	Admin Consent Required
User.Read	Sign-in and read user profile	Allows users to sign-in to the app, and allows the app to read the profile of signed-in users. It also allows the app to read basic company information of signed-in users.	No
User.ReadWrite	Read and write access to user profile	Allows the app to read the signed-in user's full profile. It also allows the app to update the signed-in user's profile information on their behalf.	No
User.ReadBasic.All	Read all users' basic profiles	Allows the app to read a basic set of profile properties of other users in your organization on behalf of the signed-in user. This includes display name, first and last name, email address, open extensions and photo. Also allows the app to read the full profile of the signed-in user.	No
User.Read.All	Read all users' full profiles	Allows the app to read the full set of profile properties, reports, and managers of other users in your organization, on behalf of the signed-in user.	Yes
User.ReadWrite.All	Read and write all users' full profiles	Allows the app to read and write the full set of profile properties, reports, and managers of other users in your organization, on behalf of the signed-in user. Also allows the app to create and delete users as well as reset user passwords on behalf of the signed-in user.	Yes

https://graph.microsoft.com

Applications specify the range of operation they REQUIRE by requesting a scope

If it is a REQUEST, the request must be granted or denied



Microsoft Graph permissions names

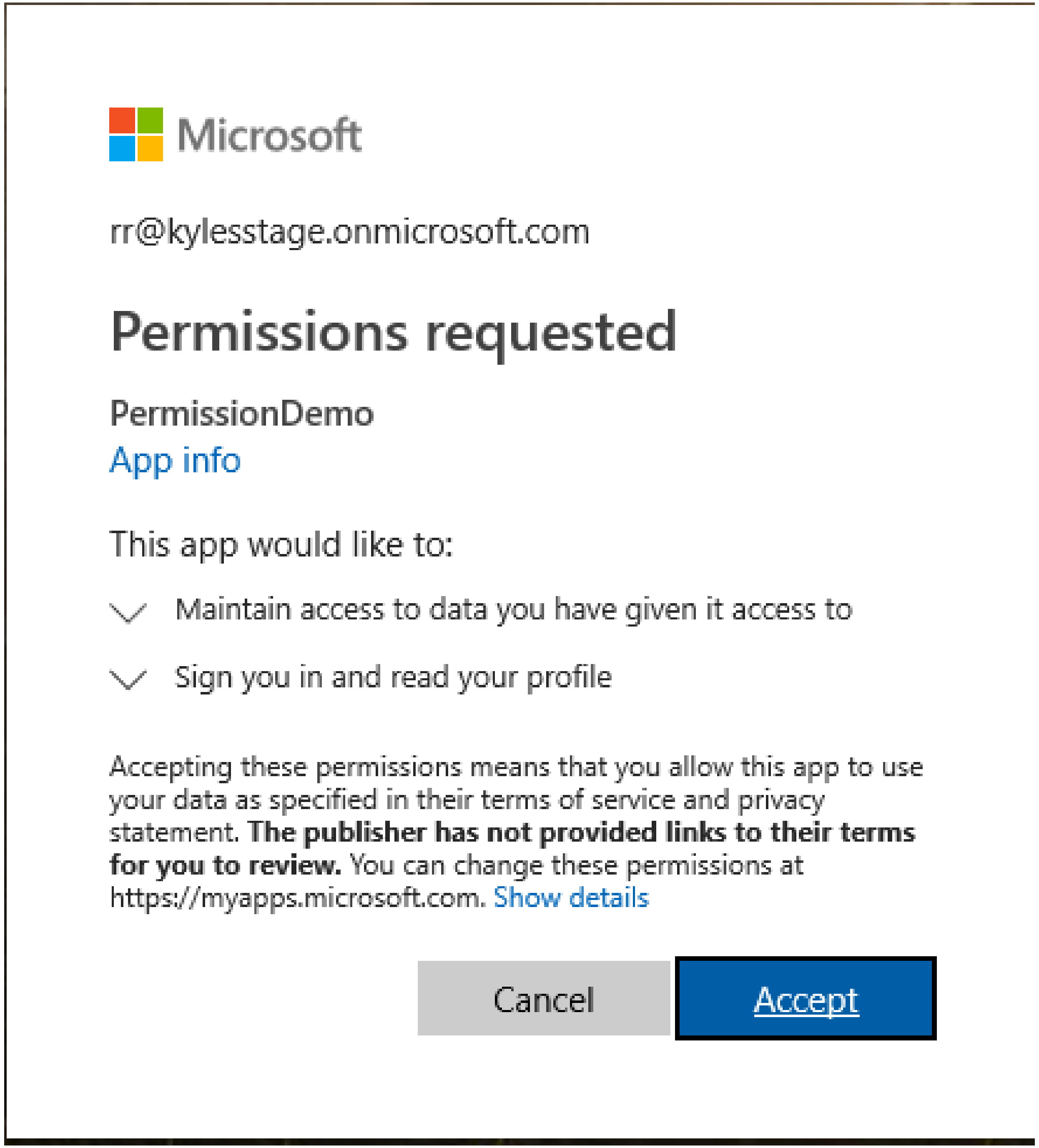
```
Resource.Operation.Constraint
```

Constraint	What it does
All	Grants permission for the app to perform the operations on all of the resources of the specified type in a directory.
Shared	Grants permission for the app to perform the operations on resources that other users have shared with the signed-in user. This constraint is mainly used with Outlook resources like mail, calendars, and contacts.
AppFolder	Grants permission for the app to read and write files in a dedicated folder in OneDrive. This constraint is only exposed on Files permissions and is only valid for Microsoft accounts.
none	If no constraint is specified the app is limited to performing the operations on the resources owned by the signed-in user.

App requests a scope

```
myMSALObj.acquireTokenSilent(“User.Read”)

if (requiresInteraction(error.errorCode)) {
                    myMSALObj.acquireTokenPopup (“User.Read”)
```



Microsoft

rr@kylesstage.onmicrosoft.com

Permissions requested

PermissionDemo
App info

This app would like to:

- Maintain access to data you have given it access to
- Sign you in and read your profile

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://myapps.microsoft.com. Show details

App requests a scope

User is presented a Permission dialog listing the scopes

As result we often refer to *scopes* as *permissions*

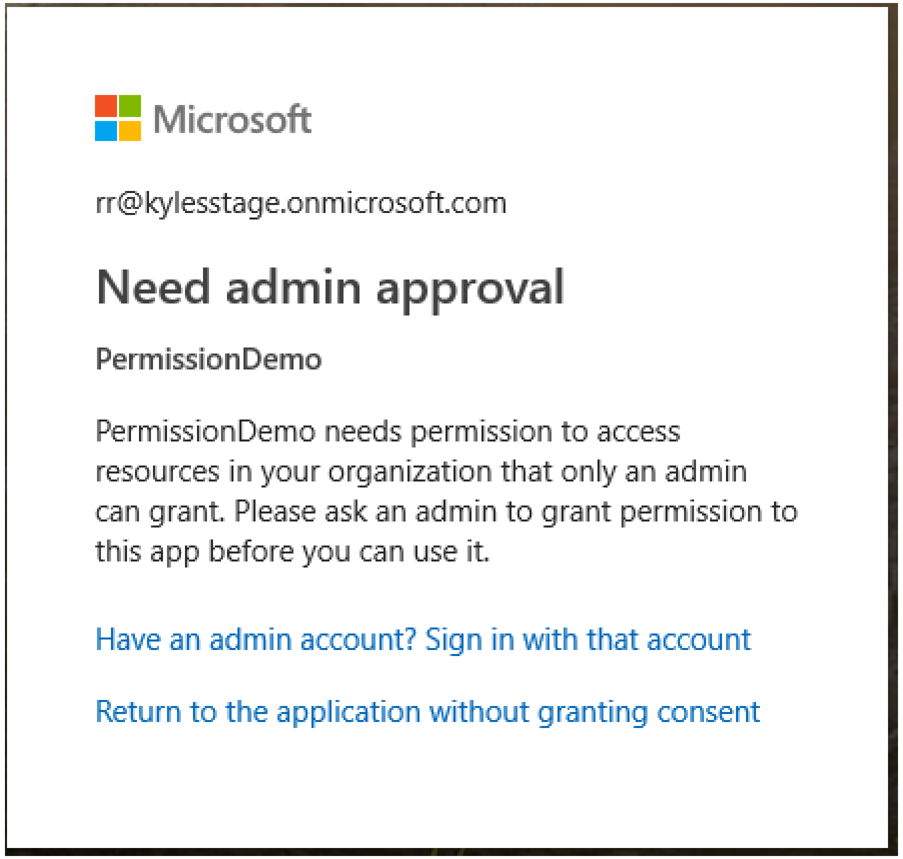
eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUFDRWZleFh4amFtUWIzT2VHUT
RHdWd2STdUZ085c0NEWVc5SlVQMzRaUG1tak5ZRTd5XzJvMTRRbTYtRzBwcXFld
E9JaENadjNFTmtLdGkyZTJTUVJNTjZteWdIbmt2MW1UcFJNQXotbDhRcGlBQSIsIm
FsZyI6IlJTMjU2IiwieDV0IjoiLXN4TUpNTENJRFdNVFB2WnlKNnR4LUNEeHcwIiwia2l
kIjoiLXN4TUpNTENJRFdNVFB2WnlKNnR4LUNEeHcwIn0.eyJhdWQiOiJodHRwczov
L2dyYXBoLm1pY3Jvc29mdC5jb20iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzL
m5ldC85NmM5YzY2Ny1lMGQ2LTQzNzMtYmRlNi1iOTcwOTg5ODQ5NTAvIiwiaWF
0IjoxNTUwNzI1NTQyLCJuYmYiOjE1NTA3MjU1NDIsImV4cCI6MTU1MDcyOTQ0Mi
wiYWNjdCI6MCwiYWNyIjoiMSIsImFpbyI6IkFTUUEyLzhLQUFBQWplbUJ5aC9yN29v
ZzJPZG5LcVc4NGViNU5TSm1adEV6dWcrajBwamhEY2M9IiwiYW1yIjpbInB3ZCJdLC
JhcHBfZGlzcGxheW5hbWUiOiJQZXJtaXNzaW9uRGVtbyIsImFwcGlkIjoiZWUxZjcyN
GYtMmM3Ni00NjRjLTgyNTgtZDBhZDM1M2YzOGMzIiwiYXBwaWRhY3IiOiIwIiwiZG
V2aWNlaWQiOiI5YjI4ZTY1Zi05Yzc5LTQ4Y2QtYTA1OC01YmY3YWIyZWM1YzkiLCJ
mYW1pbHlfbmFtZSI6IlJvYmVydHNvbiIsImdpdmVuX25hbWUiOiJSb2JiaWUiLCJpc
GFkZHIiOiI1MC4zNS42NC4xODQiLCJuYW1lIjoiUm9iYmllIFJvYmVydHNvbiIsIm9pZ
CI6ImZkZDBkMGYwLTJiMTQtNDYxNC1hZTc2LWRjOGZlNzZlYzI3MiIsInBsYXRmIjoi
MyIsInB1aWQiOiIxMDAzM0ZGRkFFQThEMzM2Iiwic2NwIjoicHJvZmlsZSBvcGVua
WQgZW1haWwgVXNlci5SZWFkIiwic3ViIjoiYXh3c2JWTmVSc2V0bDFMYnp1bmtGe
WxtTlFfaUpPXy1sbTUyd25wcnBqYyIsInRpZCI6Ijk2YzljNjY3LWUwZDYtNDM3My1iZ
GU2LWI5NzA5ODk4NDk1MCIsInVuaXF1ZV9uYW1lIjoicnJAa3lsZXNzdGFnZS5vbm
1pY3Jvc29mdC5jb20iLCJ1cG4iOiJyckBreWxlc3N0YWdlLm9ubWljcm9zb2Z0LmNvb
SIsInV0aSI6Imt5Zk11Y18zMjBxZ3FKOEFscXhBQUEiLCJ2ZXIiOiIxLjAiLCJ4bXNfc3Qi
Onsic3ViIjoiam9rS0xYcG1CVXo5cDh0YWVzYnVuZmdzRFhHYjFKZ0Q2NTJtMFV0bj
NYcyJ9LCJ4bXNfdGNkdCI6MTUzNzMzNDc1NH0.LmNqF8hbd7EpvtKhiAJtbMPanC
h6q0SAjgTgFUqP23Sn_m4A2hNfJwZYURpb040jun6JXK7zwFyHvIE9vMU_veWqn-
adrDcS7ATT44jAb-
chrCLxeT0kirc81xSSWh1vY3JEL0esR5zmCK_RiA0xZpOkf7fAASGlyxpUeIrGoFj66PNI
YG3GMuTQP7vye74X13m2z9txcKrGAYQHPvbaz_E2tFCJyc7rQtQNXqURoeTATTIVU
c4ZDxQYOhvaZ0ebGhsYewv5V4SwJjPWNboE2_PNiNNWLxAFqzm47oYQe34VS7JL
d8zjYbr375ojjNQfBDfQ2EeuuoA362BD9c1_NQ

App receives, or not, an Access Token to use as a Bearer Token when calling the API

We say the app has be granted or denied *consent*

Using the Access Token in JS

```
function callMSGraph(theUrl, accessToken, callback) {
    var xmlHttp = new XMLHttpRequest();
    xmlHttp.onreadystatechange = function () {
      if (this.readyState == 4 && this.status == 200)
        callback(JSON.parse(this.responseText));
    }
    xmlHttp.open("GET", theUrl, true); // true for asynchronous
    xmlHttp.setRequestHeader('Authorization', 'Bearer ' + accessToken);
    xmlHttp.send();
  }
```



Microsoft

rr@kylesstage.onmicrosoft.com

Need admin approval

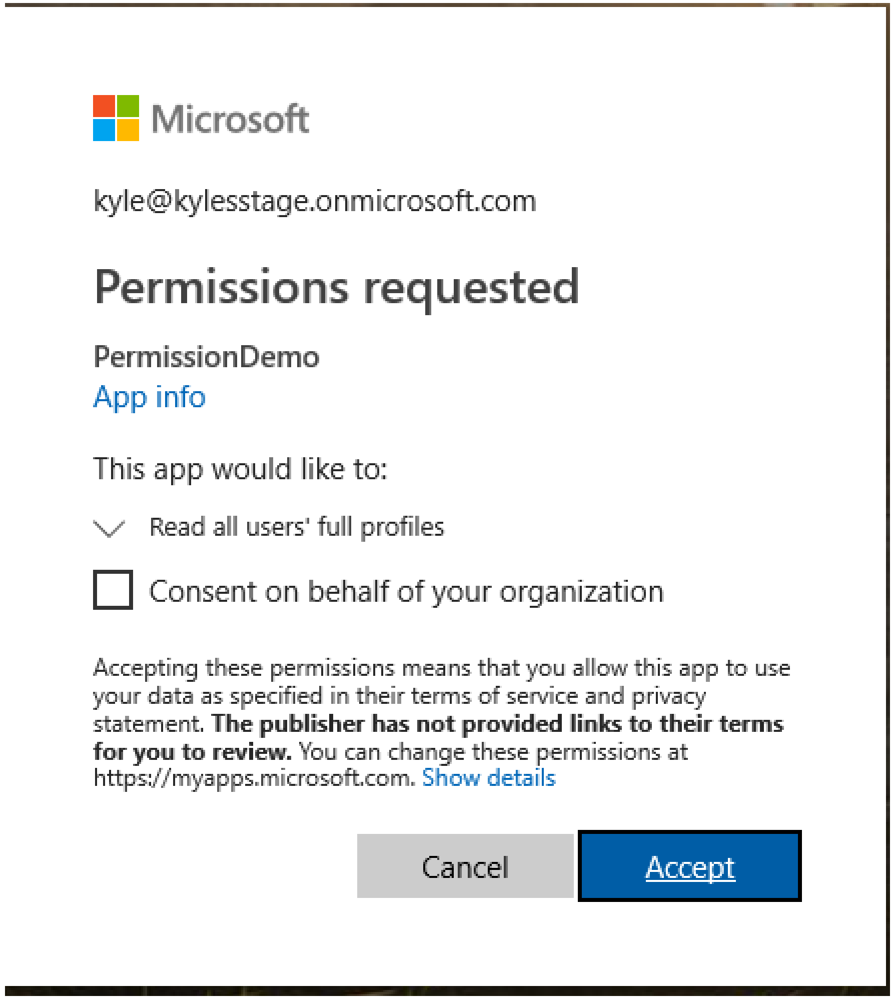
PermissionDemo

PermissionDemo needs permission to access resources in your organization that only an admin can grant. Please ask an admin to grant permission to this app before you can use it.

Have an admin account? Sign in with that account

Return to the application without granting consent

Users cannot grant consent for a permission that requires Admin consent



Microsoft

kyle@kylesstage.onmicrosoft.com

Permissions requested

PermissionDemo
App info

This app would like to:

Read all users' full profiles

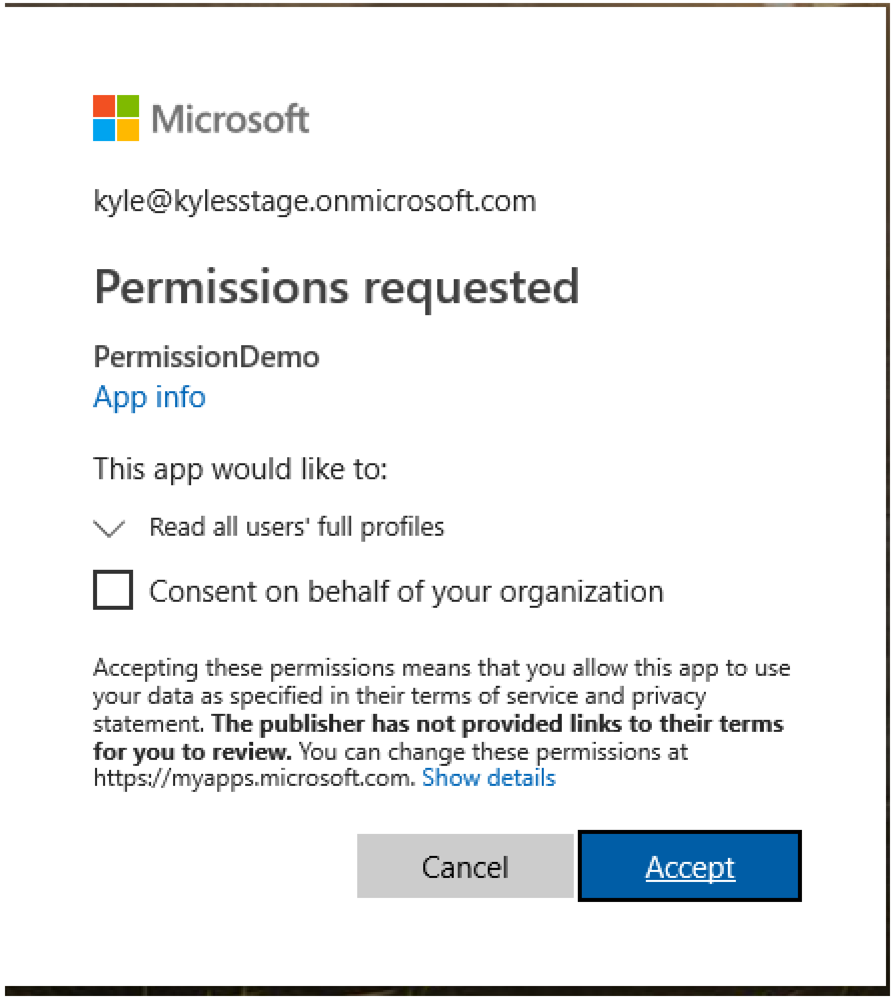
□ Consent on behalf of your organization

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://myapps.microsoft.com. Show details

Users can not grant consent for a permission that requires Admin consent

Admins can grant this consent

When running the app



Microsoft

kyle@kylesstage.onmicrosoft.com

Permissions requested

PermissionDemo
App info

This app would like to:

Read all users' full profiles

□ Consent on behalf of your organization

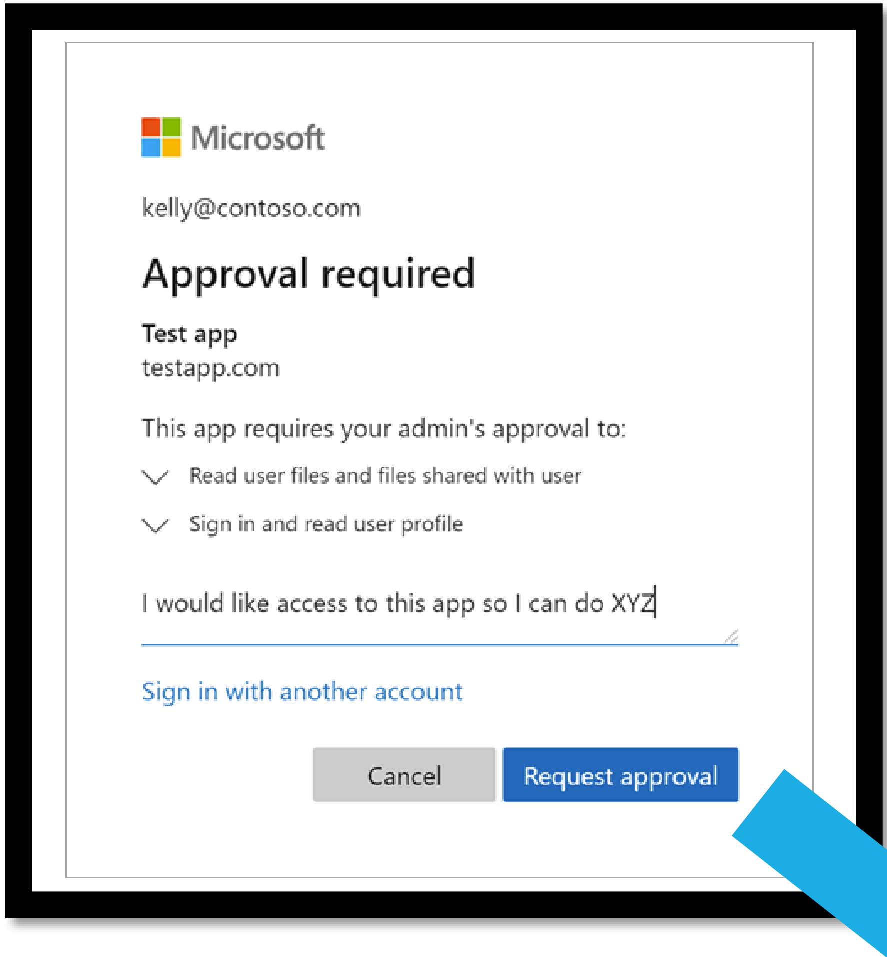
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://myapps.microsoft.com. Show details

Admins can also consent for all users in the organization

Applies to both consent User and Admin permissions

When running the app

In the Azure Portal



Microsoft

kelly@contoso.com

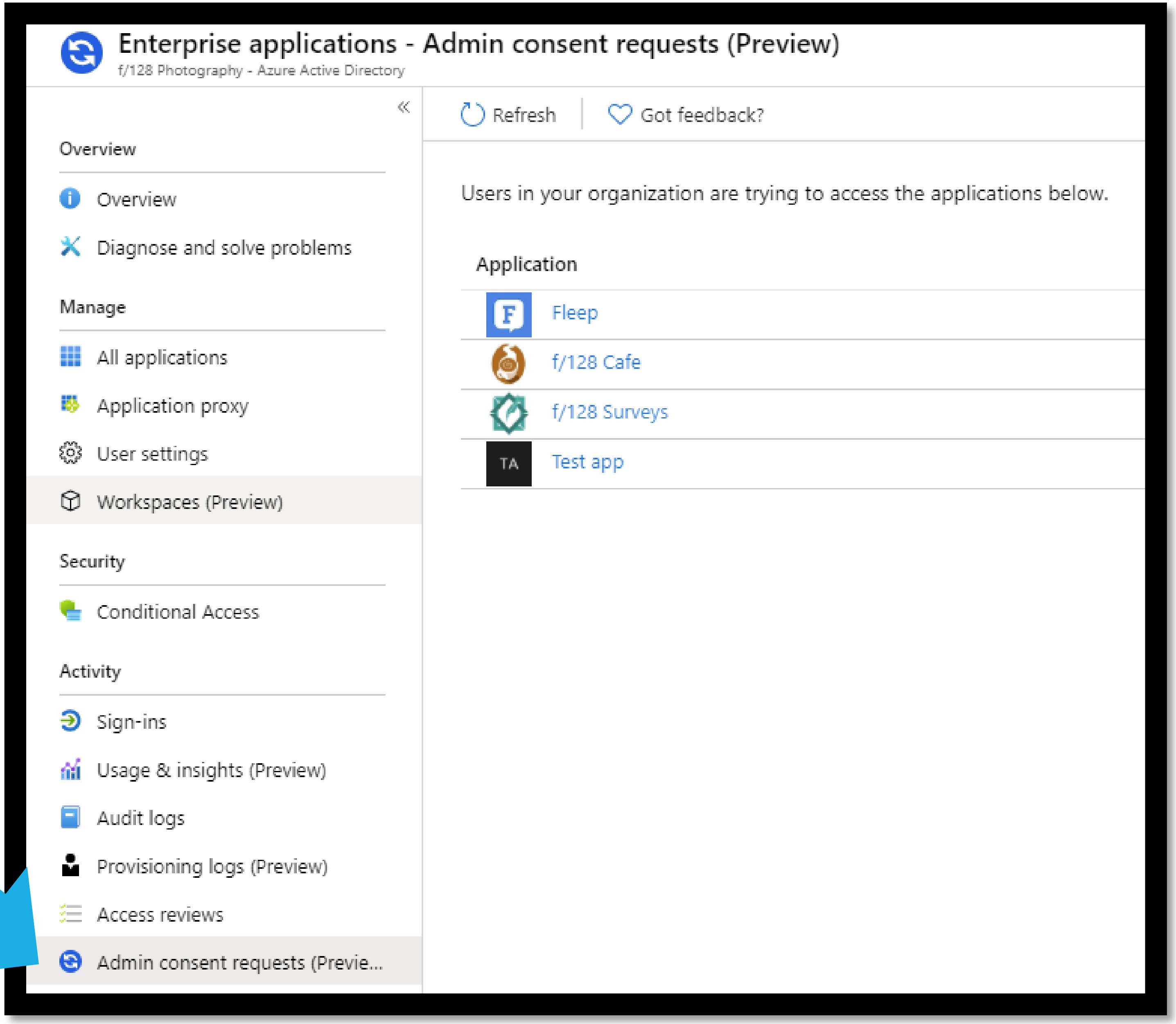
Approval required

Test app
testapp.com

This app requires your admin's approval to:

- Read user files and files shared with user
- Sign in and read user profile

I would like access to this app so I can do XYZ

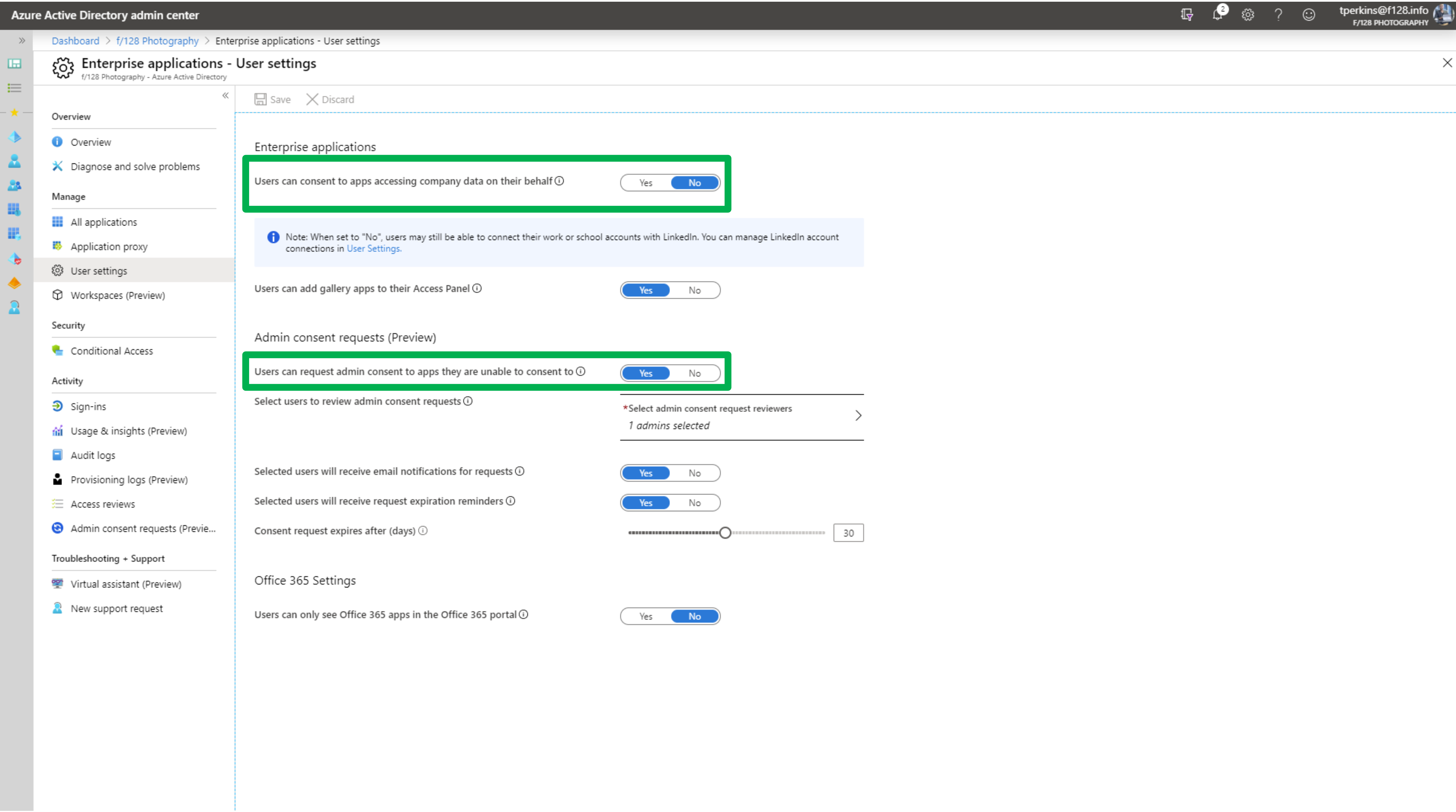


Enterprise applications - Admin consent requests (Preview)

f/128 Photography - Azure Active Directory

Users in your organization are trying to access the applications below.

Application
Fleep
f/128 Cafe
f/128 Surveys
Test app



Azure Active Directory admin center
tperkins@f128.info
F/128 PHOTOGRAPHY
Dashboard > f/128 Photography > Enterprise applications - User settings

Enterprise applications - User settings

f/128 Photography - Azure Active Directory

Save
Discard
Overview
Overview
Diagnose and solve problems
Manage
All applications
Application proxy
User settings
Workspaces (Preview)
Security
Conditional Access
Activity
Sign-ins
Usage & insights (Preview)
Audit logs
Provisioning logs (Preview)
Access reviews
Admin consent requests (Previe...
Troubleshooting + Support
Virtual assistant (Preview)
New support request

Enterprise applications

Users can consent to apps accessing company data on their behalf ⓘ Yes [✓ No]

Note: When set to "No", users may still be able to connect their work or school accounts with LinkedIn. You can manage LinkedIn account connections in User Settings.

Users can add gallery apps to their Access Panel ⓘ [✓ Yes] No

Admin consent requests (Preview)

Users can request admin consent to apps they are unable to consent to ⓘ [✓ Yes] No

Select users to review admin consent requests ⓘ
*Select admin consent request reviewers
1 admins selected

Selected users will receive email notifications for requests ⓘ [✓ Yes] No

Selected users will receive request expiration reminders ⓘ [✓ Yes] No

Consent request expires after (days) ⓘ 30

Office 365 Settings

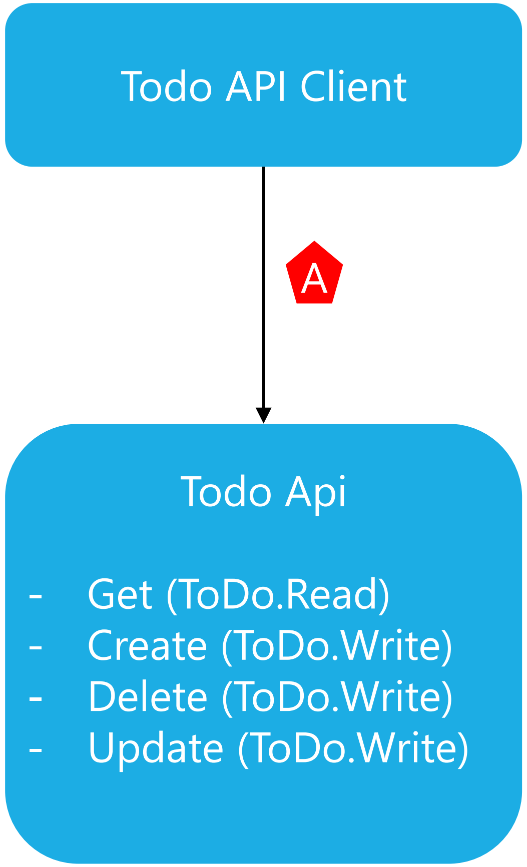
Users can only see Office 365 apps in the Office 365 portal ⓘ Yes [✓ No]

When to ask for permission

- **Dynamic user consent** – At login or first access token request
- **Incremental user consent** – As needed
- **The /.default scope**

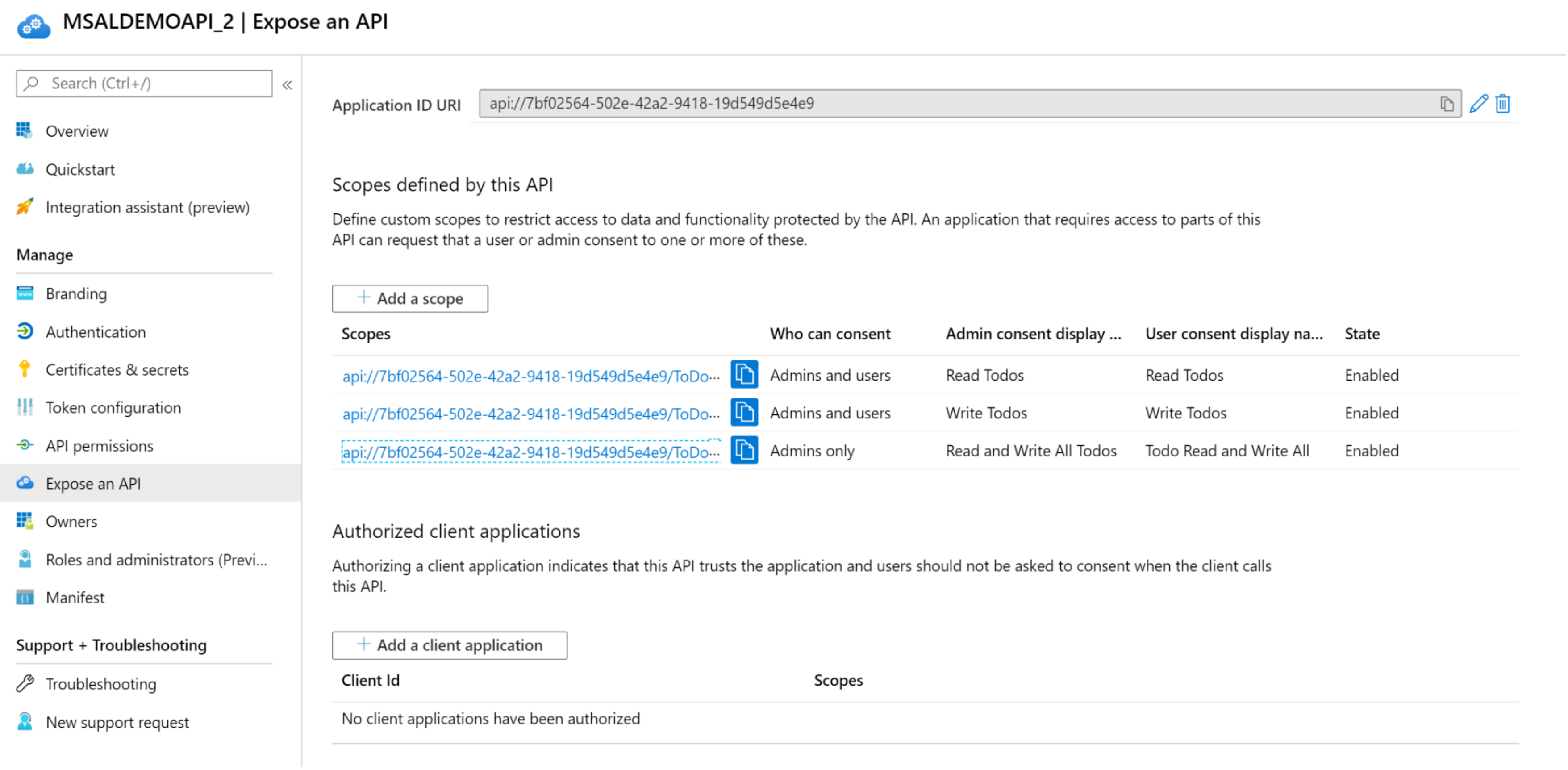
What if you want to build your own API’s

The ToDo App (Client and an API)



- Get (ToDo.Read)
- Create (ToDo.Write)
- Delete (ToDo.Write)
- Update (ToDo.Write)

Add Scopes in API App registration



Application ID URI: api://7bf02564-502e-42a2-9418-19d549d5e4e9

Scopes defined by this API

Define custom scopes to restrict access to data and functionality protected by the API. An application that requires access to parts of this API can request that a user or admin consent to one or more of these.

Scopes	Who can consent	Admin consent display ...	User consent display na...	State
api://7bf02564-502e-42a2-9418-19d549d5e4e9/ToDo...	Admins and users	Read Todos	Read Todos	Enabled
api://7bf02564-502e-42a2-9418-19d549d5e4e9/ToDo...	Admins and users	Write Todos	Write Todos	Enabled
api://7bf02564-502e-42a2-9418-19d549d5e4e9/ToDo...	Admins only	Read and Write All Todos	Todo Read and Write All	Enabled

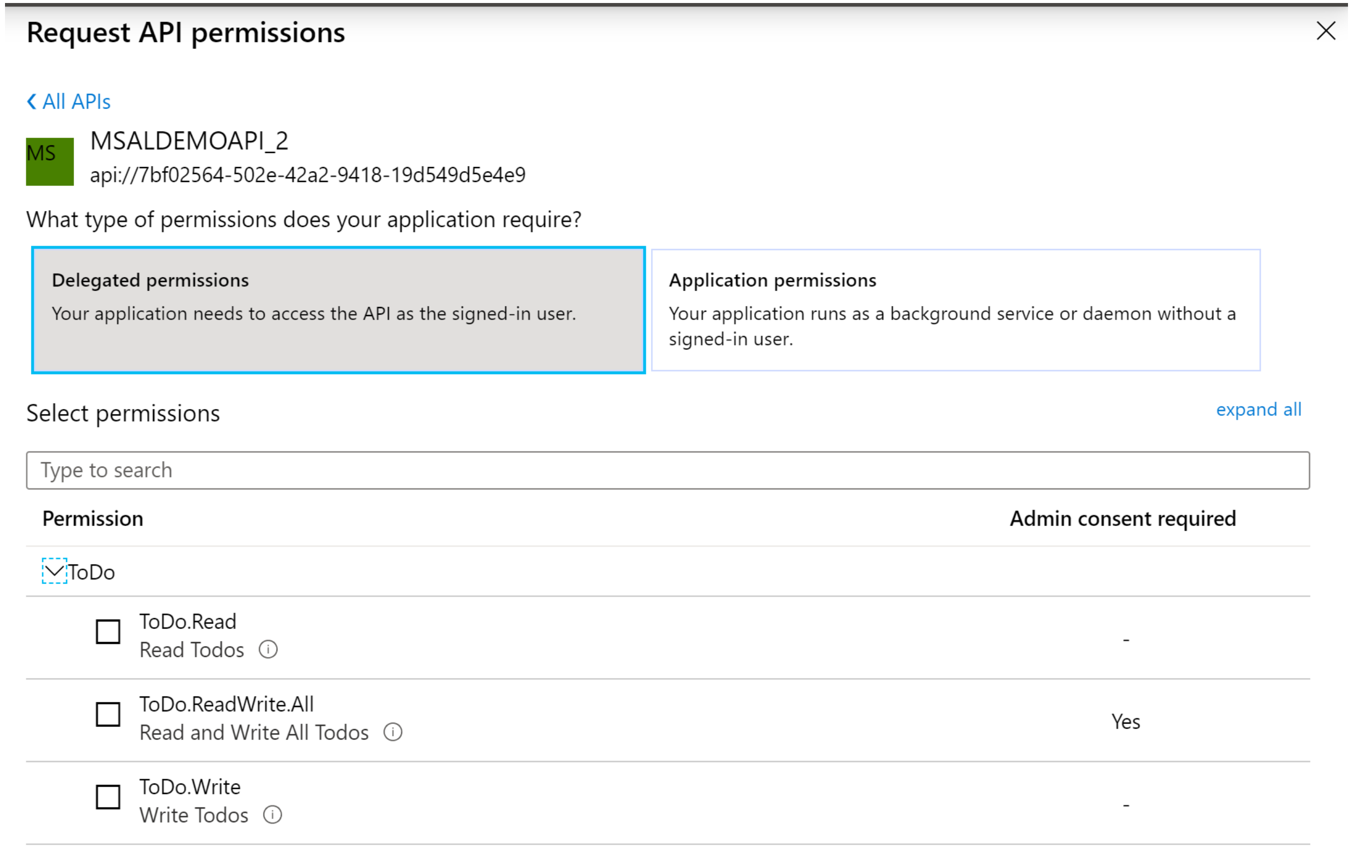
Authorized client applications

Authorizing a client application indicates that this API trusts the application and users should not be asked to consent when the client calls this API.

Client Id	Scopes
No client applications have been authorized	

Add Permission to API to Client App Registration

- Optional is Azure AD V2 Endpoints are used



Request API permissions

‹ All APIs

MS MSALDEMOAPI_2
api://7bf02564-502e-42a2-9418-19d549d5e4e9

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select permissions

expand all

Type to search

Permission	Admin consent required
ToDo	
□ ToDo.Read Read Todos ⓘ	-
□ ToDo.ReadWrite.All Read and Write All Todos ⓘ	Yes
□ ToDo.Write Write Todos ⓘ	-

Authorize against client app and get access_token

Authorize and get token

https://login.microsoftonline.com/common/oauth2/v2.0/authorize
?client_id=ac8d6bc2-a0ff-40db-9102-8a21dff251c3
&response_type=token
&redirect_uri=https%3A%2F%2Fjwt.ms
&response_mode=fragment
&scope=profile openid api://7bf02564-502e-42a2-9418-19d549d5e4e9/ToDo.Write
&state=123456
&nonce=123456789
&lc=9

Authorize and get admin consent token:

https://login.microsoftonline.com/common/oauth2/v2.0/authorize
?client_id=ac8d6bc2-a0ff-40db-9102-8a21dff251c3
&response_type=token
&redirect_uri=https%3A%2F%2Fjwt.ms
&response_mode=fragment
&scope=profile openid api://7bf02564-502e-42a2-9418-19d549d5e4e9/ToDo.ReadWrite.All
&state=123456
&nonce=123456789
&lc=9

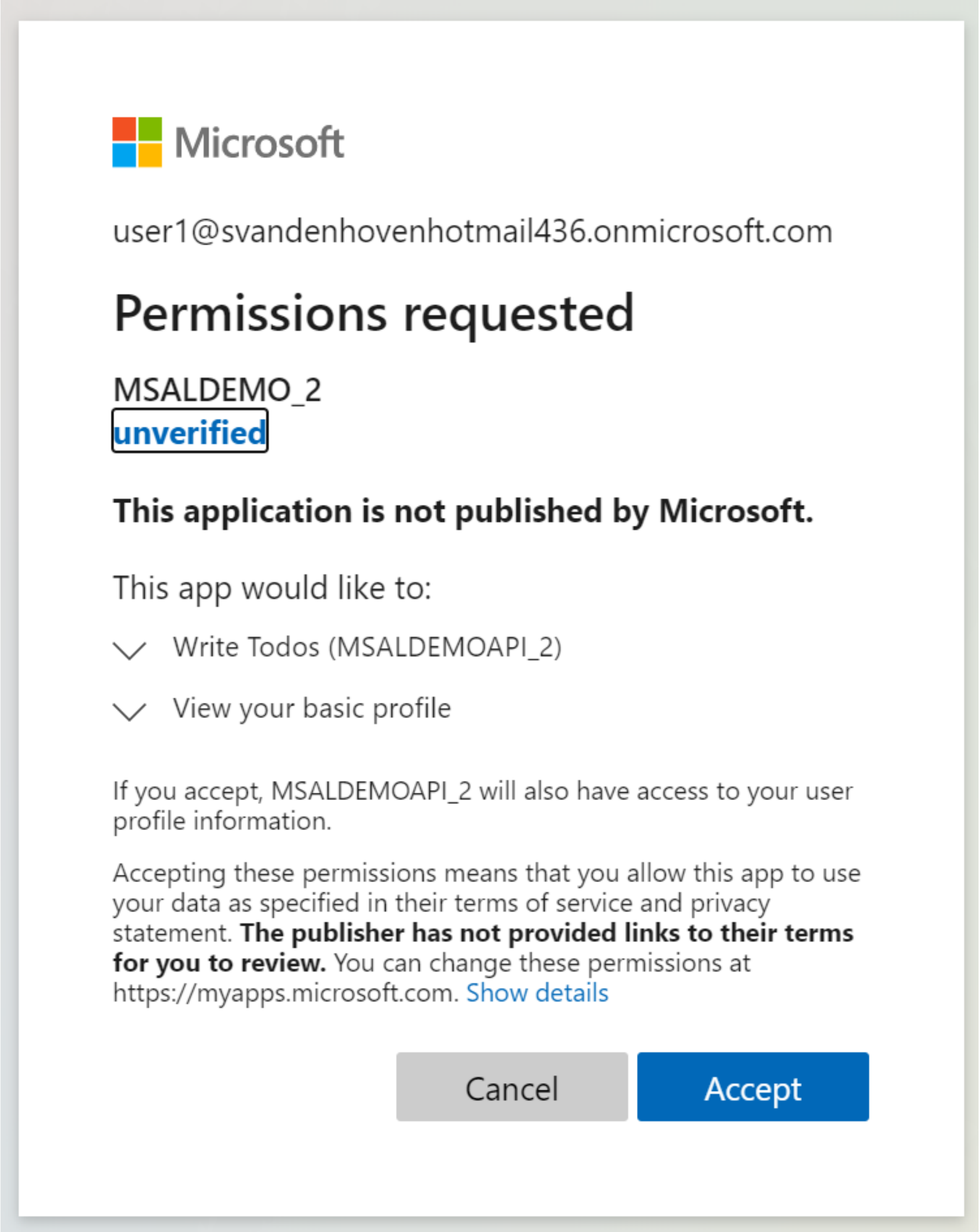
A Scope need to be Accepted

User needs to decide to approve or not

If user accepts, an Access Token can be returned

If user does not accept, Access Token will not be returned

Consent can be removed via myapps.microsoft.com



user1@svandenhovenhotmail436.onmicrosoft.com

Permissions requested

MSALDEMO_2
unverified

This application is not published by Microsoft.

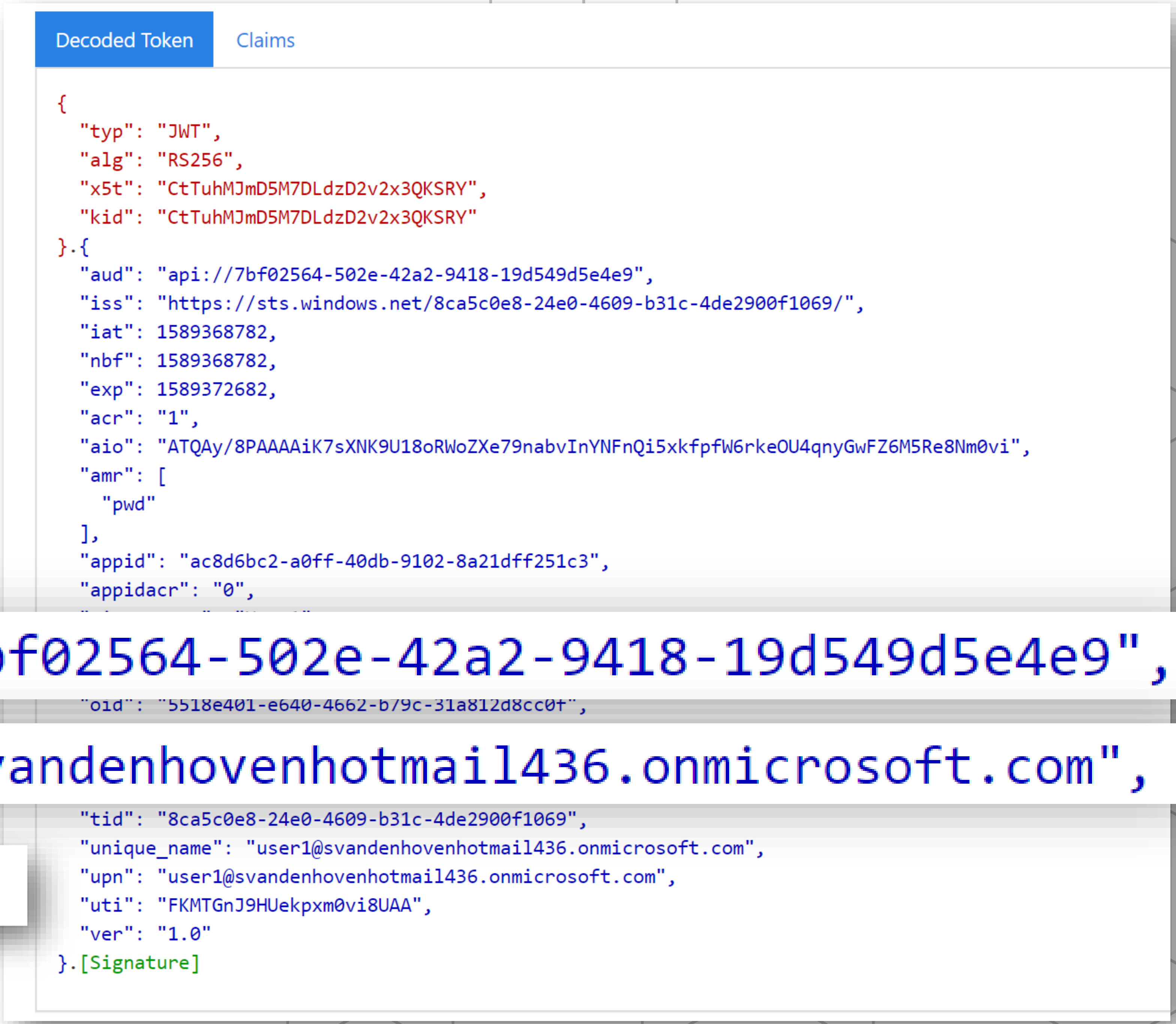
This app would like to:

- Write Todos (MSALDEMOAPI_2)
- View your basic profile

If you accept, MSALDEMOAPI_2 will also have access to your user profile information.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://myapps.microsoft.com. Show details

The requested Access_Token



```
{
  "typ": "JWT",
  "alg": "RS256",
  "x5t": "CtTuhMJmD5M7DLdzD2v2x3QKSRY",
  "kid": "CtTuhMJmD5M7DLdzD2v2x3QKSRY"
}.{
  "aud": "api://7bf02564-502e-42a2-9418-19d549d5e4e9",
  "iss": "https://sts.windows.net/8ca5c0e8-24e0-4609-b31c-4de2900f1069/",
  "iat": 1589368782,
  "nbf": 1589368782,
  "exp": 1589372682,
  "acr": "1",
  "aio": "ATQAy/8PAAAAiK7sXNK9U18oRWoZXe79nabvInYNFnQi5xkfpfW6rkeOU4qnyGwFZ6M5Re8Nm0vi",
  "amr": [
    "pwd"
  ],
  "appid": "ac8d6bc2-a0ff-40db-9102-8a21dff251c3",
  "appidacr": "0",
  "oid": "5518e401-e640-4662-b79c-31a812d8cc0f",
  "tid": "8ca5c0e8-24e0-4609-b31c-4de2900f1069",
  "unique_name": "user1@svandenhovenhotmail436.onmicrosoft.com",
  "upn": "user1@svandenhovenhotmail436.onmicrosoft.com",
  "uti": "FKMTGnJ9HUekpxm0vi8UAA",
  "ver": "1.0"
}.[Signature]
```

```
"aud": "api://7bf02564-502e-42a2-9418-19d549d5e4e9",
```

```
"upn": "sander@svandenhovenhotmail436.onmicrosoft.com",
```

```
"scp": "ToDo.Write",
```

Call the Api With the AccessToken C# & MSAL

Create HTTPClient with AccessToken as Bearer Header

```
private ITokenAcquisition _tokenAcquisition;
var scope = "User.Read api://7bf02564-502e-42a2-9418-19d549d5e4e9/ToDo.Write",
var accessToken = await _tokenAcquisition
        .GetAccessTokenOnBehalfOfUserAsync(scope.Split(" "));
_httpClient.DefaultRequestHeaders.Authorization =
        new AuthenticationHeaderValue("Bearer", accessToken);
```

Call the API (post to add a todo item)

```
var response = await _httpClient
        .PostAsync($"{_TodoListBaseAddress}/api/todolist", jsoncontent);
```

How to protect the api with the accesstoken will be discussed in next webinar

Demo to register API with Permissions and get Access_Token in a client with Consent of User

Authorization of APIs

Service, Daemon, Machine to Machine

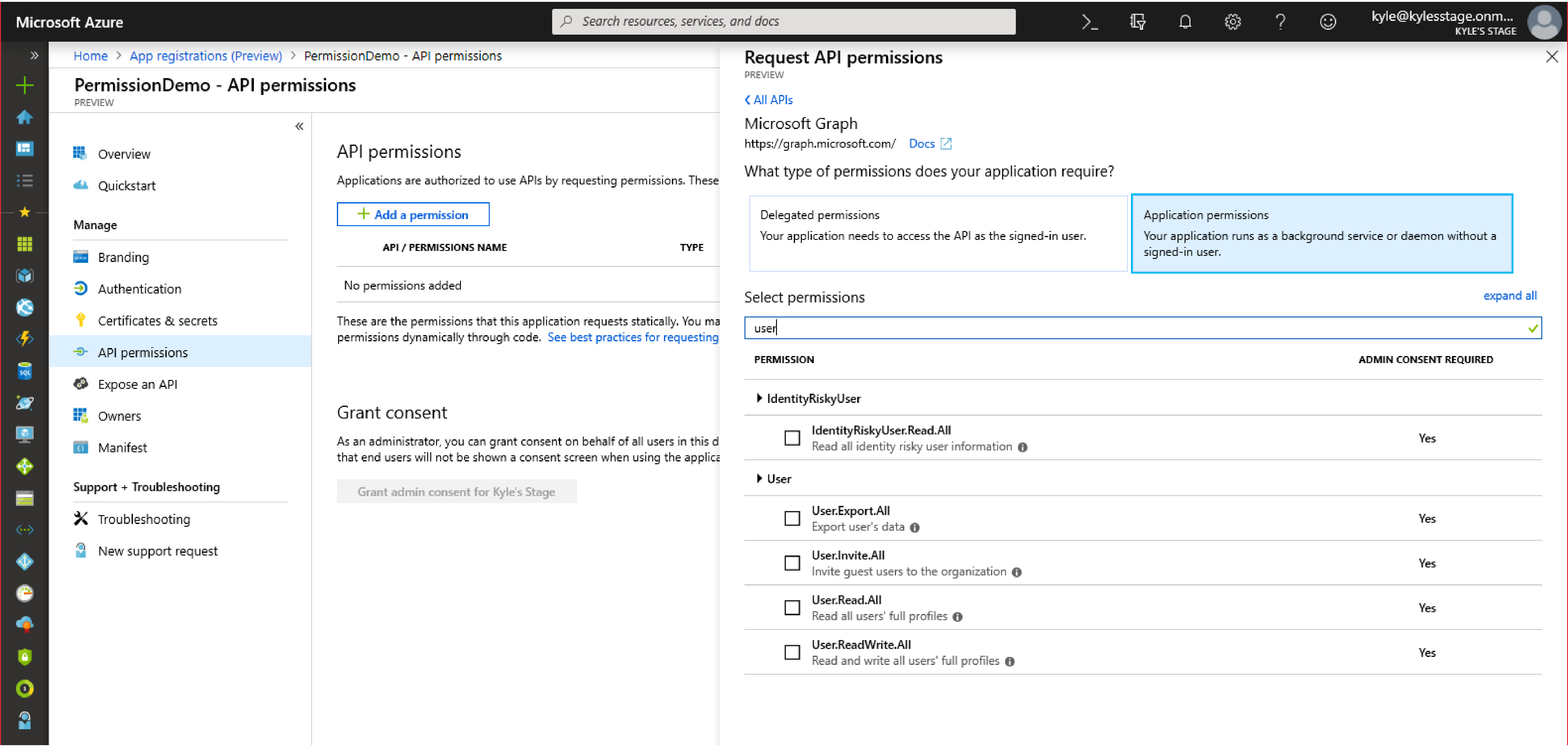
Delegated versus Application Permissions

- **Delegation Permissions**: Your application needs to access the web API as the signed-in user, but with access limited by the selected permission. This type of permission can be granted by a user unless the permission is configured as requiring administrator consent.

- **Application Permissions**: Your application needs to access the web API directly as itself (no user context). This type of permission requires administrator consent and is also not available for native client applications.

Application Permissions

- App proves it is the app requesting permission
  - Secret, Cert, Managed Identity for Azure Services
- Admin consent – Always
- Cross user functionality
  - User.ReadWrite.All
  - Files.Read.All
- Permissions granted the app are always the permissions used



Microsoft Azure

Search resources, services, and docs

kyle@kylesstage.onm...
KYLE'S STAGE

Home > App registrations (Preview) > PermissionDemo - API permissions

PermissionDemo - API permissions
PREVIEW

- Overview
- Quickstart

Manage

- Branding
- Authentication
- Certificates & secrets
- API permissions
- Expose an API
- Owners
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

API permissions

Applications are authorized to use APIs by requesting permissions. These

+ Add a permission

API / PERMISSIONS NAME	TYPE
No permissions added	

These are the permissions that this application requests statically. You ma
permissions dynamically through code. See best practices for requesting

Grant consent

As an administrator, you can grant consent on behalf of all users in this d
that end users will not be shown a consent screen when using the applica

Grant admin consent for Kyle's Stage

Request API permissions
PREVIEW

< All APIs

Microsoft Graph
https://graph.microsoft.com/ Docs

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select permissions expand all

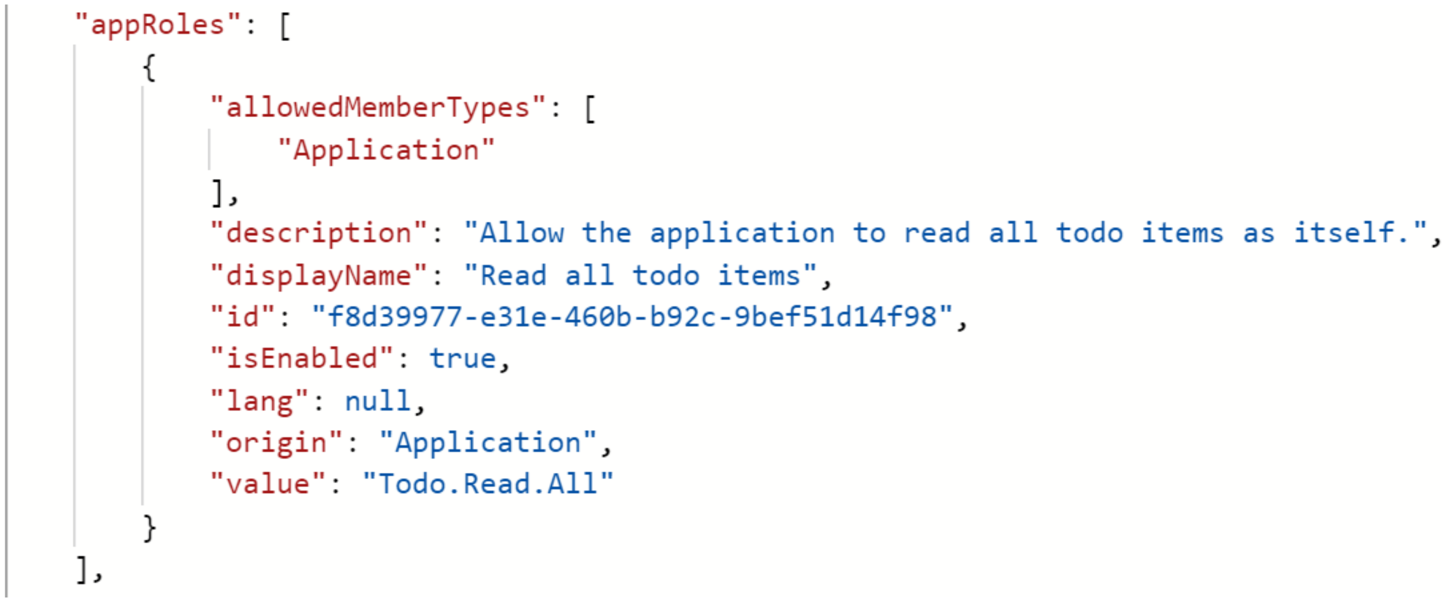
user

PERMISSION	ADMIN CONSENT REQUIRED
▸ IdentityRiskyUser	
□ IdentityRiskyUser.Read.All Read all identity risky user information	Yes
▸ User	
□ User.Export.All Export user's data	Yes
□ User.Invite.All Invite guest users to the organization	Yes
□ User.Read.All Read all users' full profiles	Yes
□ User.ReadWrite.All Read and write all users' full profiles	Yes

How to create Application Permissions in your API?

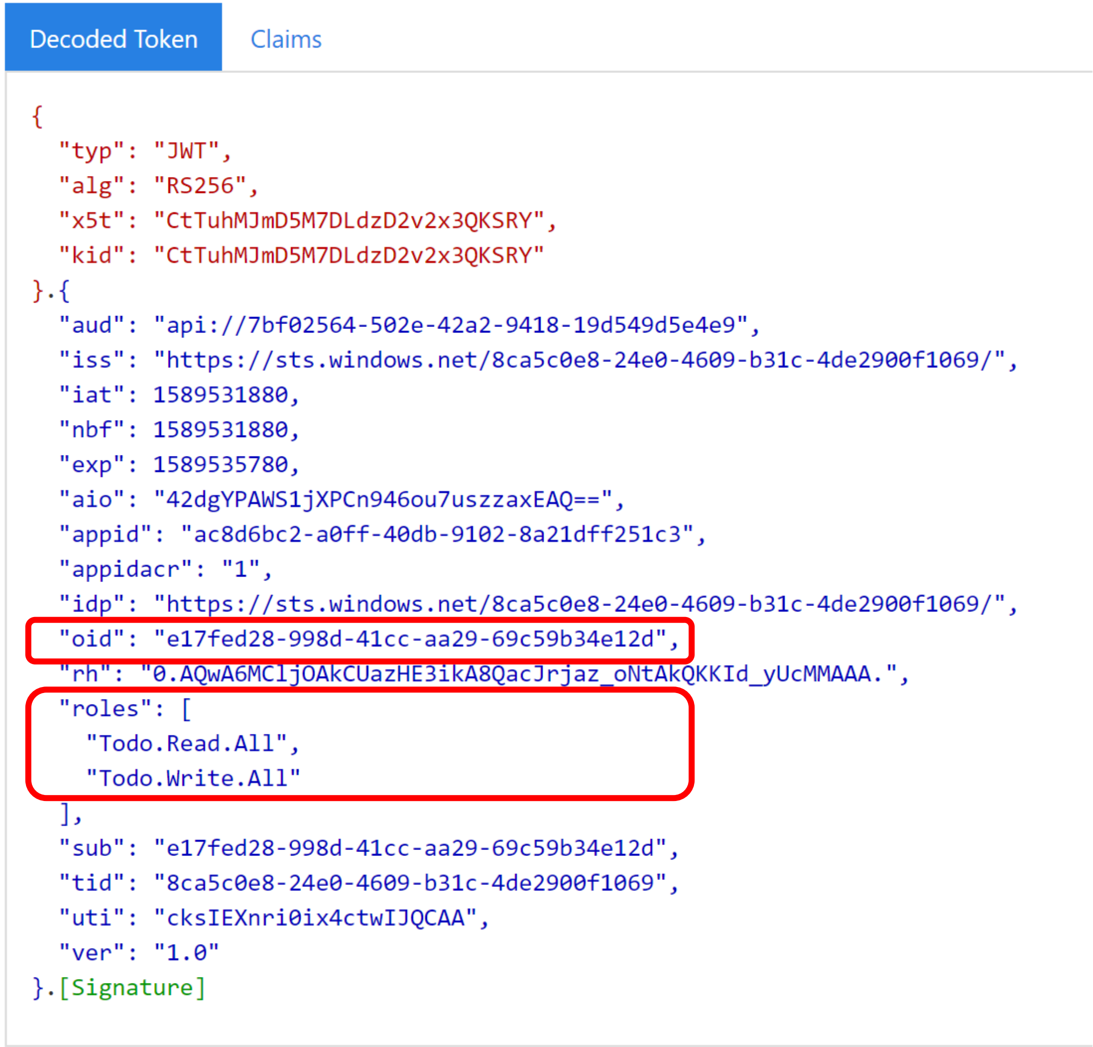
Application Permission are not scope but roles

Application permissions are registering in the app registration as roles



```
"appRoles": [
    {
        "allowedMemberTypes": [
            "Application"
        ],
        "description": "Allow the application to read all todo items as itself.",
        "displayName": "Read all todo items",
        "id": "f8d39977-e31e-460b-b92c-9bef51d14f98",
        "isEnabled": true,
        "lang": null,
        "origin": "Application",
        "value": "Todo.Read.All"
    }
],
```

Application Permission will be added as Roles



```
{
  "typ": "JWT",
  "alg": "RS256",
  "x5t": "CtTuhMJmD5M7DLdzD2v2x3QKSRY",
  "kid": "CtTuhMJmD5M7DLdzD2v2x3QKSRY"
}.{
  "aud": "api://7bf02564-502e-42a2-9418-19d549d5e4e9",
  "iss": "https://sts.windows.net/8ca5c0e8-24e0-4609-b31c-4de2900f1069/",
  "iat": 1589531880,
  "nbf": 1589531880,
  "exp": 1589535780,
  "aio": "42dgYPAWS1jXPCn946ou7uszzaxEAQ==",
  "appid": "ac8d6bc2-a0ff-40db-9102-8a21dff251c3",
  "appidacr": "1",
  "idp": "https://sts.windows.net/8ca5c0e8-24e0-4609-b31c-4de2900f1069/",
  "oid": "e17fed28-998d-41cc-aa29-69c59b34e12d",
  "rh": "0.AQwA6MCljOAkCUazHE3ikA8QacJrjaz_oNtAkQKKId_yUcMMAAA.",
  "roles": [
    "Todo.Read.All",
    "Todo.Write.All"
  ],
  "sub": "e17fed28-998d-41cc-aa29-69c59b34e12d",
  "tid": "8ca5c0e8-24e0-4609-b31c-4de2900f1069",
  "uti": "cksIEXnri0ix4ctwIJQCAA",
  "ver": "1.0"
}.[Signature]
```

Demo application permissions

Call the To Api from a Daemon application

Is this going to be on the test?

Example: Cactus API

A *very* simple API

- `GET /items` – list all items
- `GET /items/{id}` – get one item

Exposes two delegated permissions

- *Catalog.View.All*: List all items
- *Catalog.View.Published*: List only published items

Has two types of users

- *Admin*: Can view all items
- *User*: Can view only published items

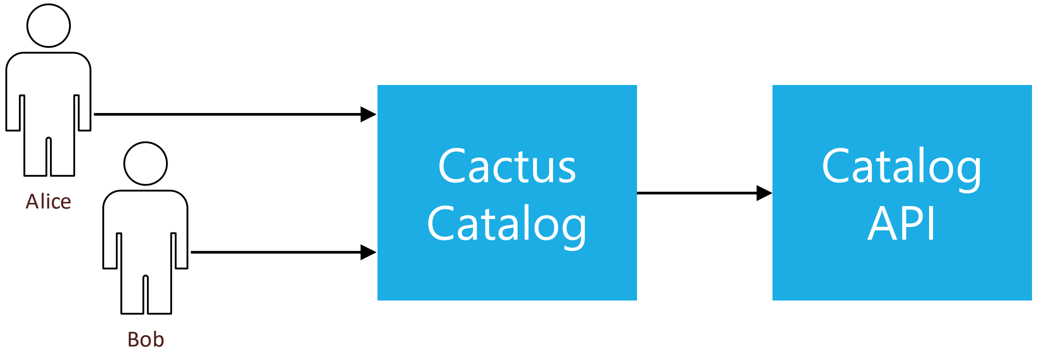
Catalog contents

```
[
  {
    "id": 1,
    "name": "Cephalocereus senilis",
    "status": "published"
  },
  {
    "id": 2,
    "name": "Neobuxbaumia polylopha",
    "status": "unpublished"
  },
  {
    "id": 3,
    "name": "Myrtillocactus geometrizans",
    "status": "published"
  }
]
```

Example: Cactus Catalog

A client of Cactus API

- A single-page JavaScript app requesting the *Catalog.View.Published* permission, used to list catalog items
  - Q: Is this a delegated permission (scope) or application permission (app role)?
  - A: It *must* be delegated, because the client is a public client.



Example: Cactus Catalog

A client of Cactus API

ID	Name	Status
1	Cephalocereus senilis	published
2	Neobuxbaumia polylopha	unpublished
3	Myrtillocactus geometrizans	published

App has consent for Catalog.View.Published scope

Alice, who is an *Admin*, signs in to *Cactus Catalog* and calls `GET /items`

- Q: What’s the result?
- A: Only published items are listed. Although Alice is considered an *Admin* by the API and is allowed to view all items, the client app has limited delegated permissions.

```
GET /items
```

```
[
  {
    "id": 1,
    "name": "Cephalocereus senilis",
    "status": "published"
  },
  {
    "id": 3,
    "name": "Myrtillocactus geometrizans",
    "status": "published"
  }
]
```

Example: Cactus Catalog

A client of Cactus API

ID	Name	Status
1	Cephalocereus senilis	published
2	Neobuxbaumia polylopha	unpublished
3	Myrtillocactus geometrizans	published

App has consent for Catalog.View.Published scope

Bob, who is a *User*, signs in to *Cactus Catalog* and calls `GET /items/1`

- Q: What’s the result?
- A: The requested item is returned. Because it is a published item, both Bob and the client app (on Bob’s behalf) have permission.

```
GET /items/1

{
  "id": 1,
  "name": "Cephalocereus senilis",
  "status": "published"
}
```

Example: Cactus Catalog

A client of Cactus API

ID	Name	Status
1	Cephalocereus senilis	published
2	Neobuxbaumia polylopha	unpublished
3	Myrtillocactus geometrizans	published

App has consent for Catalog.View.Published scope

Alice, who is an *Admin*, signs in to *Cactus Catalog* and calls `GET /items/2`

Q: What’s the result?

A: An authorization error is returned. Although Alice is considered an *Admin* by the API and is allowed to view all items, the client app has limited delegated permissions and is not allowed to read unpublished items.

```
GET /items/2


403 Forbidden
```

Example: Cactus Catalog

A client of Cactus API

ID	Name	Status
1	Cephalocereus senilis	published
2	Neobuxbaumia polylopha	unpublished
3	Myrtillocactus geometrizans	published

App has consent for Catalog.View.All scope

Now suppose the *Cactus Catalog* app is granted the delegated permission *Catalog.View.All*

Bob, a *User*, signs in to *Cactus Catalog* and calls `GET /items/2`

Q: What should be the result?

A: An authorization error. Even though the app has delegated permissions to read all items, these permissions are restricted by what the user is actually allowed to do, and Bob isn’t allowed to read unpublished items.

```
GET /items/2

403 Forbidden
```

Best Practices

- Adopt consent and authorization best practices
  - The most widely adopted apps in Azure AD generally follow these already
  - http://aka.ms/GraphBestPractices
- Show that you have been thoughtful with your permission requests:
  - Only ask for what is absolutely necessary
  - Choose permission type based on your scenario
  - Use incremental consent to request granular permissions just in time
- Provide Terms of Service and Privacy Statement

Next Session:
Accessing and protecting API’s
WEWC586 Thursday May 28th 14:00-15:00

Documentation on the webinar:
http://aka.ms/modernIdentityForDevelopers