

To setup Azure AD B2C SSO for the SleekFlowToDoListAPI, the Swagger UI and the Blazor Web Assembly - these 8 environment variables must be registered in the Web App configuration:

- AzureAdB2CApi\_ClientId (string)
- AzureAdB2CApi \_\_Instance (string)
- AzureAdB2CApi \_\_SignUpSignInPolicyId (string)
- AzureAdB2CApi \_\_Domain (string)
- AzureAdB2CSwagger\_ClientId (string)
- AzureAdB2CSwagger\_ClientSecret (string)
- AzureAdB2CSwagger\_Scope (string)
- AzureAdB2CSwagger\_AppName (string)

### **Setting up App registrations and UserFlows for Single Sign On.**

**Important:** Before following this document, please make sure that Azure AD B2C tenant is setup.

#### **Overview**

To implement SSO for the application we are using and Microsoft.Identity.Web package for backend API and Swagger client authentication. To utilize the packages, the authentication context needs to be created in the app and the following configurations needs to be setup in Azure Portal:

- App registration API
- App registration Swagger Client
- App registration Blazor Web Assembly (Not documented yet)
- User Flow (SignIn)

**A. App Registration API:**

1. Navigate to Azure AD B2C tenant and select the "App registrations" tab to register a new app.

Home > Azure AD B2C

## Azure AD B2C | App registrations

onmicrosoft.com

Search

2. + New registration Endpoints Troubleshooting

Overview

Manage

1. App registrations Applications (Legacy) Identity providers API connectors Company branding User attributes Users Roles and administrators Policies User flows

Welcome to the new and improved App registrations (now General availability)

Starting June 30th, 2020 we will no longer add any new features to the legacy App registrations (now General availability) interface. We recommend upgrading to Microsoft Authentication Library (MSAL) and Microsoft Identity Platform for Web.

All applications Owned applications Deleted applications

Start typing a name or Application ID to filter these results

2 applications found

Display name ↑↓

PR	preview_portal_backend
PR	preview_portal_backend_swagger

2. Register a new app:

- Give the app registration a suitable name, e.g. "preview\_portal\_backend".
- Select the option "Accounts in the organizational directory only" (Single tenant) for supported account types.

Home > Azure AD B2C | App registrations >

## Register an application

**\* Name**  
The display name for this application (this can be changed later).

preview\_portal\_backend

**Supported account types**  
Who can use this application or access this API?

☐ Accounts in this organizational directory only (Annata B2C only - Single tenant)

☐ Accounts in any organizational directory (Any Azure AD directory – Multitenant)

☒ Accounts in any identity provider or organizational directory (for authenticating users with user flows)

[Help me choose...](#)

**Redirect URI (recommended)**  
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Select a platform e.g. https://example.com/auth

**Permissions**  
Azure AD B2C requires this app to be consented for openid and offline\_access permissions. You must be an app administrator to grant admin con

☒ Grant admin consent to openid and offline\_access permissions

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applica](#)

By proceeding, you agree to the [Microsoft Platform Policies](#)

**Register**

3. Navigate to "Expose an API" tab and add a new scope to authenticate the backend API. Make sure to assign the scope name as "Authorization" – the secured backend WebApi controller will authenticate against the hardcoded scope value "Authorization". Make sure to assign the scope value "https://{domain refer to **A.5**}/{clientId refer to **A.4**}/Authorization" to **AzureAdB2CSwagger\_Scope** parameter in the web app config.

Home > Azure AD B2C | App registrations > preview\_portal\_backend

## preview\_portal\_backend | Expose an API

Search

Got feedback?

Application ID URI https://onmicrosoft.com/

**Manage**

- Overview
- Integration assistant
- Branding & properties
- Authentication
- Certificates & secrets
- API permissions
- Expose an API**
- Owners
- Manifest
- Support + Troubleshooting
- Troubleshooting

**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.

**Add a scope**

Scopes

Scopes	Who can consent	Admin consent display name	State
https://onmicrosoft.com/	Admins only	Authorization	Enabled

**Add a scope**

Scope name \* Authorization

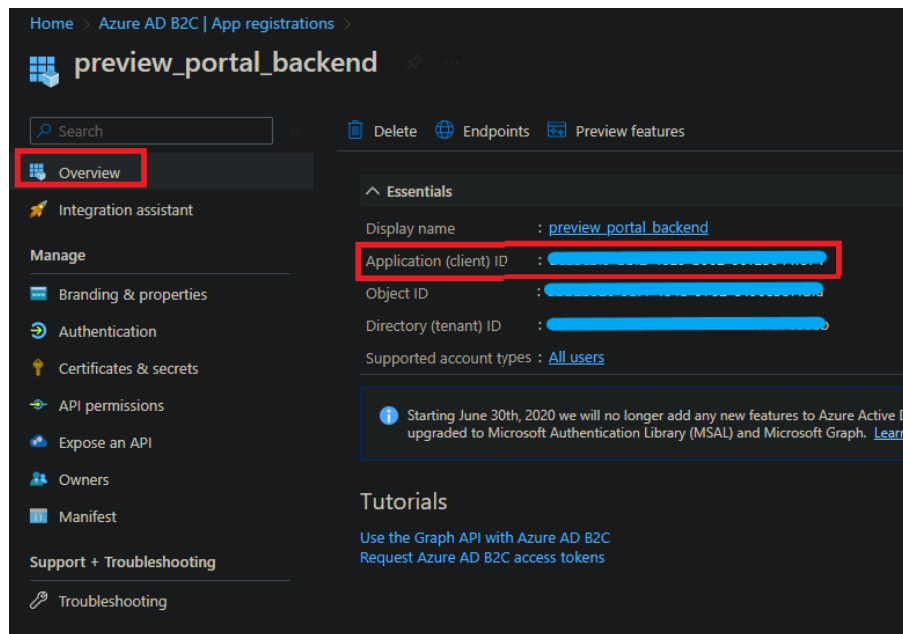
https://onmicrosoft.com/Authorization

Admin consent display name \* Authorization

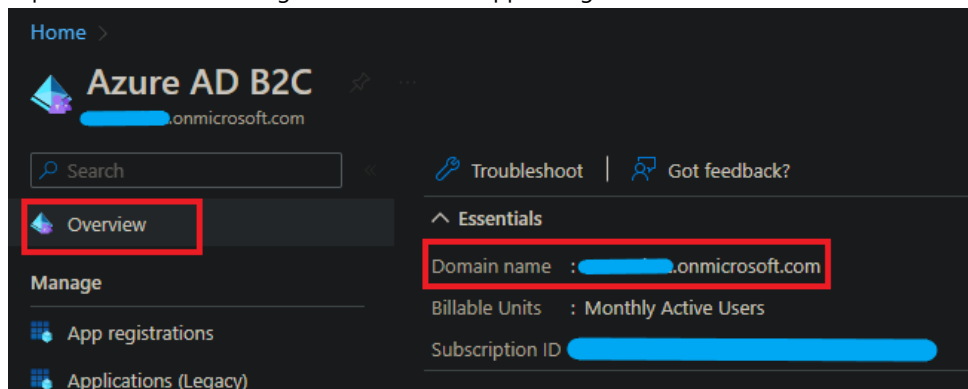
Admin consent description \* Authorization

State Enabled Disabled

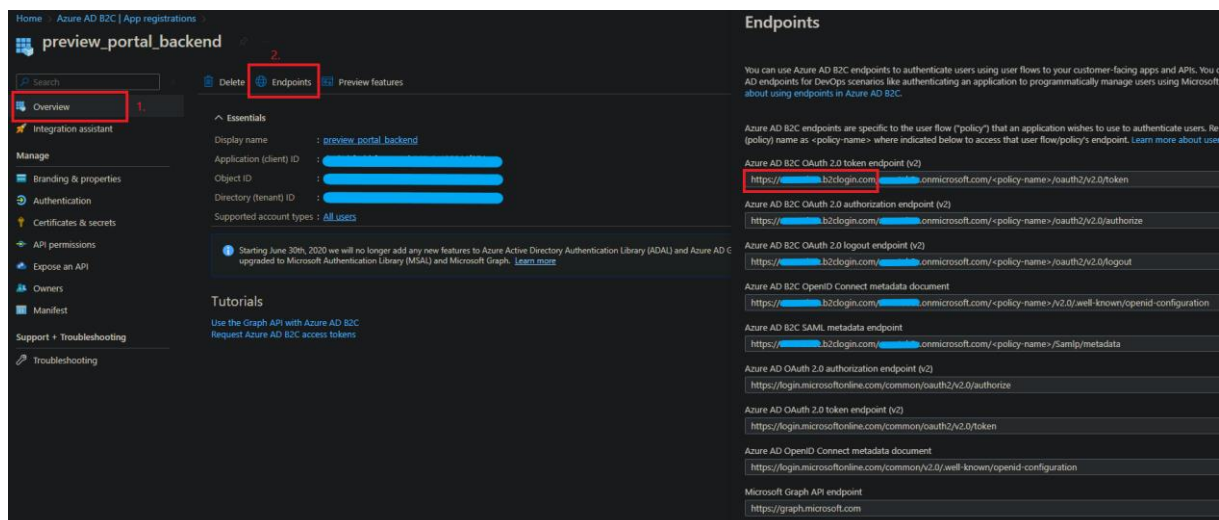
- Make sure to assign the "Application (client) ID" value to **AzureAdB2CApi\_ClientId** parameter in the Web App config.



- Navigate to the Azure Active Directory "Overview" tab to locate the **AzureAdB2CApi\_Domain** parameter value and register it in the web app config.



- For the **AzureAdB2CApi\_Instance** parameter, set its value to the app's URL in the web app config. E.g. "https://test.b2clogin.com" as shown below:

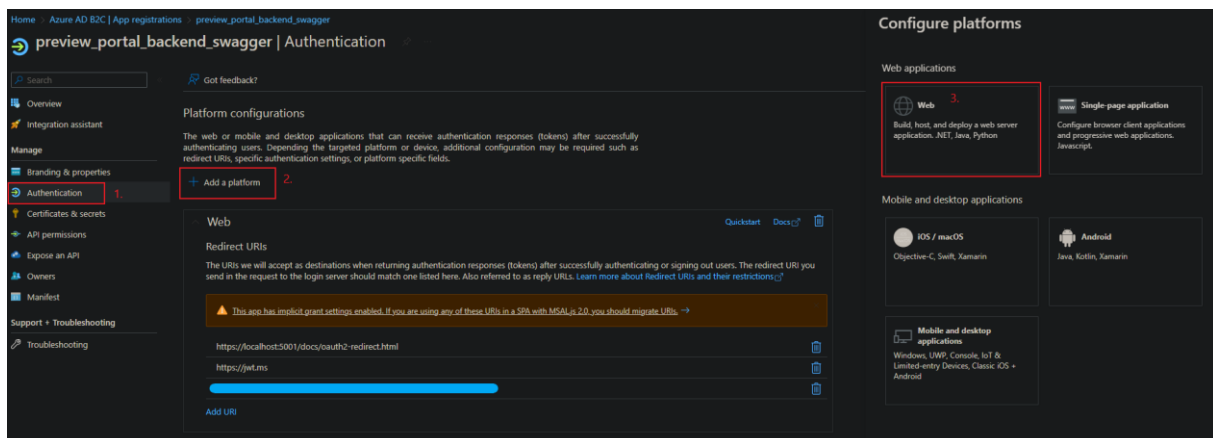


7. For the **AzureAdB2CApi\_SignUpSignInPolicyId** parameter, set its value to the user flow name created in section **D.3**, e.g. "B2C\_1\_DealerPortPreview"

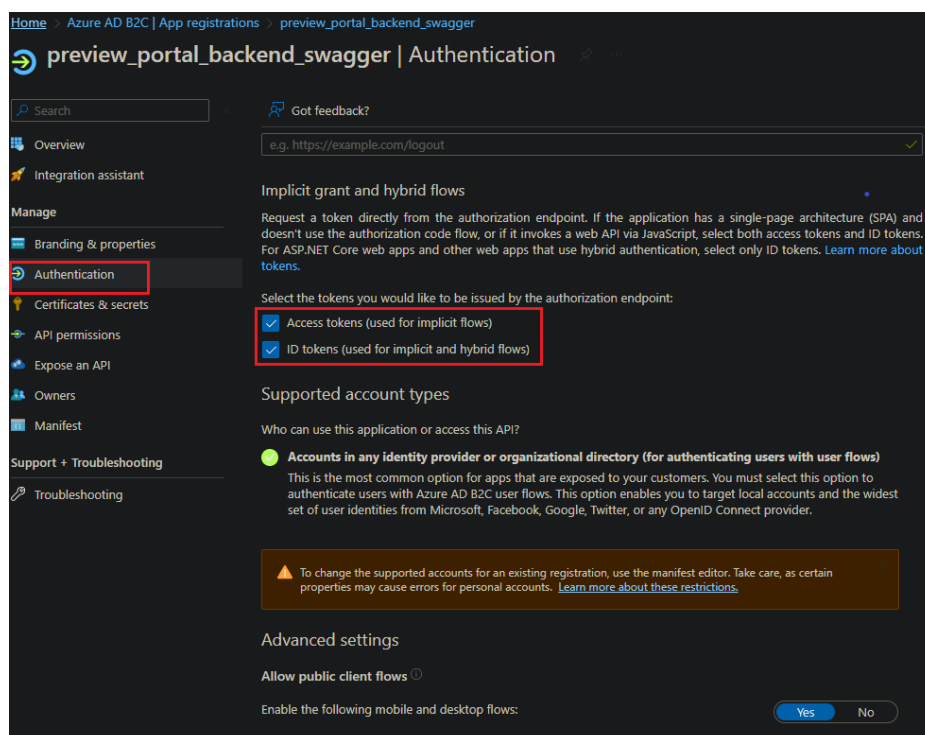
## B. App Registration Swagger Client:

App registration is required for swagger authorization to work. The following steps need to be followed

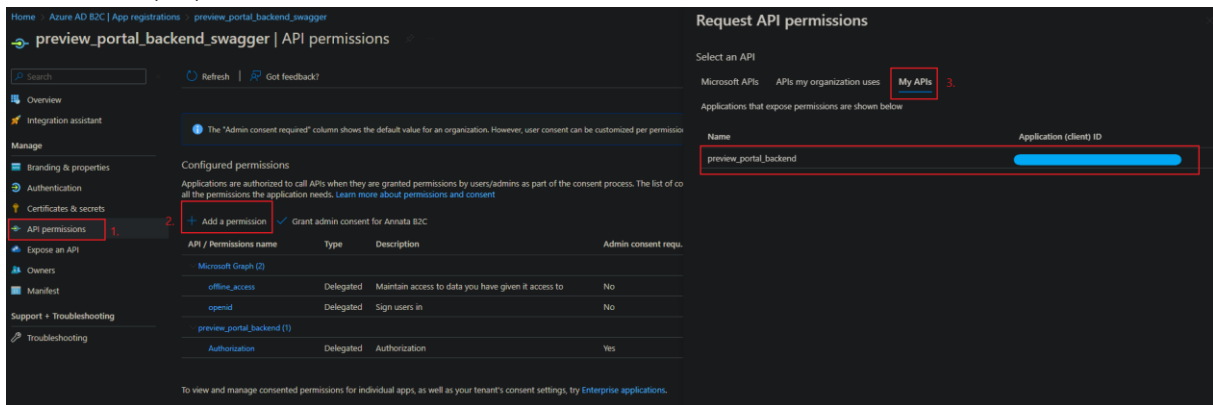
1. Navigate to Azure AD B2C tenant and select the "App registrations" tab to register a new app.
2. Register a new app:
  - Give the app registration a suitable name e.g. "preview\_portal\_backend\_swagger" (Assign the registered app name to **AzureAdB2CSwagger\_AppName** parameter in WebApp config).
  - Select the option "Accounts in the organizational directory only" (Single tenant) for supported account types.
3. Navigate to "Authentication" tab, add a new platform for "Web" and set the following redirect URLs to match the Swagger client's app:
  - <https://{appURL}/docs/oauth2-redirect.html>
  - <https://localhost:5001/docs/oauth2-redirect.html> (for developer to debug)
  - <https://jwt.ms> (for testing user flow)



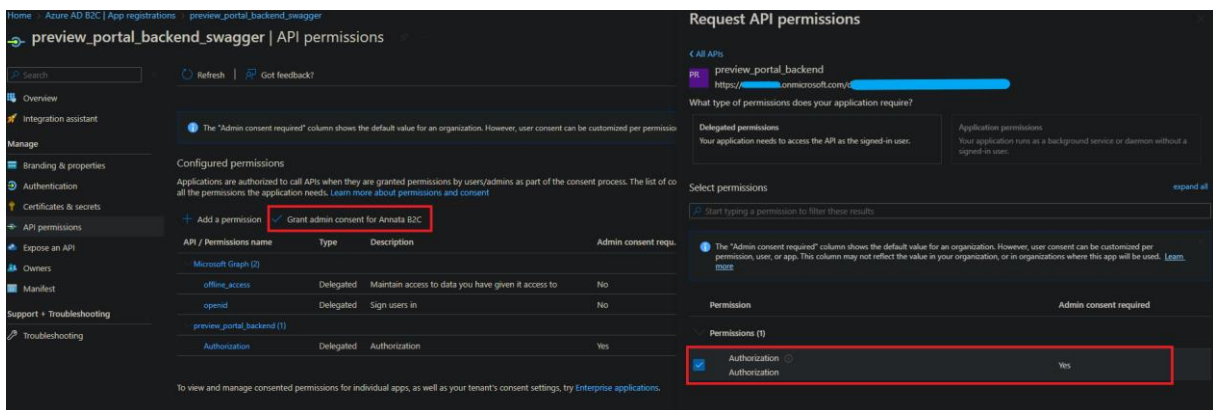
4. Navigate to "Authentication tab" and add support for ID tokens and access tokens.



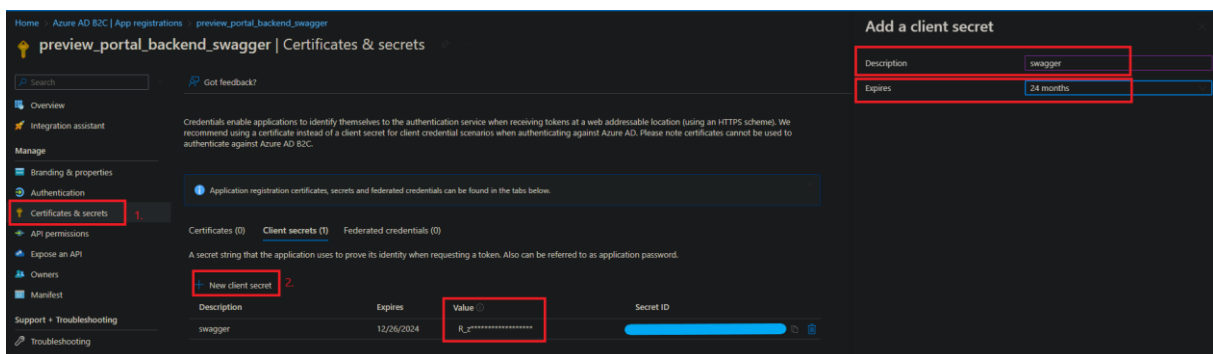
5. Navigate to “API permissions” tab, select “Add a permission” and select the API exposed in section (A.3).



6. Select the permission created in section (A.3) and add the permission. Make sure to grant admin consent for the added permission.



7. Navigate to “Certificates & secrets” tab and add a new client secret:
- Give a suitable description e.g. “swagger”.
  - Select the preferable expiration interval from the “Expires” drop-down.
  - Assign the client secret’s “Value” to **AzureAdB2CSwagger\_ClientSecret** in WebApp config.



8. Make sure to assign the "Application (client) ID" value to **AzureAdB2CSwagger\_ClientId** parameter in the Web App config.

The screenshot shows the Azure AD B2C App registrations page for an application named 'preview\_portal\_backend\_swagger'. The left sidebar contains navigation links: Overview (highlighted with a red box), Integration assistant, Manage (with sub-links: Branding & properties, Authentication, Certificates & secrets, API permissions, Expose an API, Owners, Manifest), and Support + Troubleshooting (with a link to Troubleshooting). The main content area is titled 'Essentials' and displays the following information:

- Display name : [preview\\_portal\\_backend\\_swagger](#)
- Application (client) ID : [Redacted] (This field is highlighted with a red box)
- Object ID : [Redacted]
- Directory (tenant) ID : [Redacted]
- Supported account types : [All users](#)

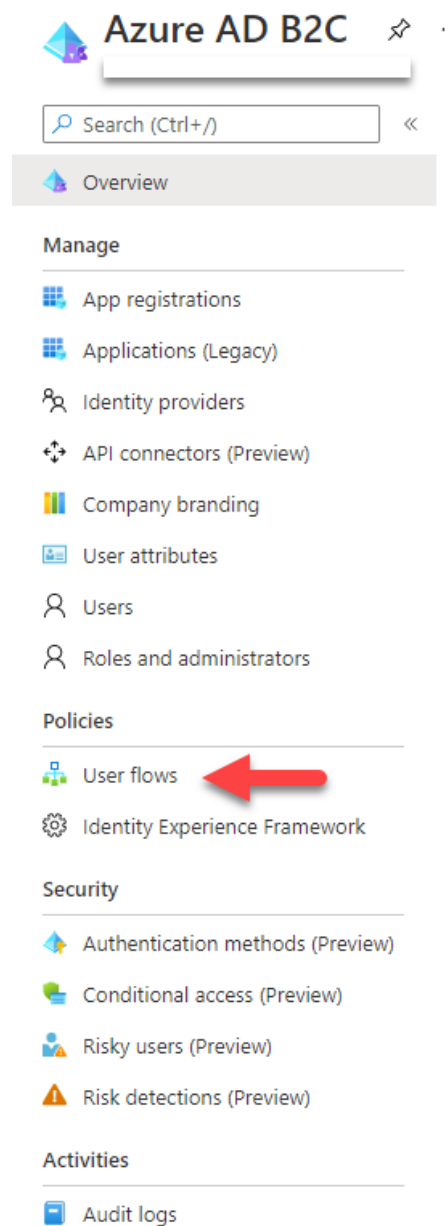
Below the Essentials section, there is a blue information box with a message: 'Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Auth upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)'. At the bottom, there is a 'Tutorials' section with links: 'Use the Graph API with Azure AD B2C' and 'Request Azure AD B2C access tokens'.



C. **App Registration Blazor Web Assembly**

#### D. User Flow (SignIn)

1. Navigate to "User flows" tab in Azure AD B2C directory main page.




## 2. Create a Sign In User flow (keep the version recommended).


Home > Azure AD B2C >


### Create a user flow


User flows are predefined, configured policies that you can use to set up authentication experiences for your end users. Select a user flow type to get started. [Learn more.](#)


**Select a user flow type**


**Sign up and sign in**  
Enables a user to create an account or sign in to their account.

**Profile editing**  
Enables a user to configure their user attributes.

**Password reset**  
Enables a user to choose a new password after verifying their email.

**Sign up**  
Enables a user to create a new account.

**Sign in**  
Enables a user to sign in to their account.

**Sign in using password credential**  
Enables a user to sign in directly to applications (in preview).

**Version**

**Recommended**  
This is the next-generation, preview user flow with the latest features.

**Standard**  
This is the generally available, production-ready user flow. No enhancements are being developed for this version. Select Recommended for next-generation features.

Create

## 3. Register a suitable name for the user flow e.g., DealerPortPreview. The final name would come out as B2C\_1\_DealerPortPreview.

### 1. Name

The unique string used to identify this user flow in requests to Azure AD B2C. This cannot be changed after a user flow has been created.

B2C\_1\_ \*

## 4. In Multifactor Authentication select the following values:

^ Multifactor authentication

Enabling multifactor authentication (MFA) requires your users to verify their identity with a second factor before allowing them into your application. [Learn more about multifactor authentication](#)

MFA method

☒ Email

☐ SMS or phone call


☐ SMS only

☐ Phone call only

MFA enforcement


☒ Conditional (Recommended)

☐ Always on

 Conditional delegates the MFA decision to conditional access policies. When conditional is selected, MFA will be OFF unless a conditional access policy requires it.

^ Conditional access (Preview)

Conditional access policies evaluate signals in real time and enforce grants like MFA only when required. [Learn more about conditional access.](#)

Enforce conditional access policies  ☐

## 5. In Identity Providers select the Email signup under Local accounts.

Identity providers are the different types of accounts your users can use to sign up or sign in to your application. You need to select at least one for a valid user flow. [Learn more about identity](#)

### Local accounts

- ☒ Email signup
- ☐ None

### Social identity providers

Identity Provider

Name

6. In “Application claims” tab, select all claims and save. Make sure “Email Addresses” and “Identity Provider Access Token” is selected, otherwise the B2C login in Dealer Portal will not work.

Home > Azure AD B2C | User flows > B2C\_1\_DealerPortPreview

## B2C\_1\_DealerPortPreview | Application claims

Sign up and sign in (Recommended)

Search

Run user flow | Save | Discard | Manage user attributes | Got feedback?

Overview

Settings

- Properties
- Identity providers
- User attributes
- Application claims**
- API connectors

Customize

- Page layouts
- Languages

User attributes are values collected on sign up. Claims are values about the user returned to the application in the token. You can create custom attributes for use in your directory. [Learn more about user attributes and claims.](#)

Name	Data Type	Description	Attribute type
<input checked="" type="checkbox"/> City	String	The city in which the user is located.	Built-in
<input checked="" type="checkbox"/> Country/Region	String	The country/region in which the user is located.	Built-in
<input checked="" type="checkbox"/> Display Name	String	Display Name of the User.	Built-in
<input checked="" type="checkbox"/> Email Addresses	StringCollection	Email addresses of the user.	Built-in
<input checked="" type="checkbox"/> Given Name	String	The user's given name (also known as first name).	Built-in
<input type="checkbox"/> Identity Provider	String	The social identity provider used by the user to access to your application.	Built-in
<input checked="" type="checkbox"/> Identity Provider Access Token	String	The access_token returned by the OAuth identity provider.	Built-in
<input checked="" type="checkbox"/> Job Title	String	The user's job title.	Built-in
<input checked="" type="checkbox"/> Legal Age Group Classification	String	The legal age group that a user falls into based on their country and date of birth	Built-in
<input checked="" type="checkbox"/> Postal Code	String	The postal code of the user's address.	Built-in
<input checked="" type="checkbox"/> State/Province	String	The state or province in user's address.	Built-in
<input checked="" type="checkbox"/> Street Address	String	The street address where the user is located.	Built-in
<input checked="" type="checkbox"/> Surname	String	The user's surname (also known as family name or last name).	Built-in
<input checked="" type="checkbox"/> User is new	Boolean	True, if the user has just signed-up for your application.	Built-in
<input checked="" type="checkbox"/> User's Object ID	String	Object identifier (ID) of the user object in Azure AD.	Built-in