

Table of Contents

- Set Up Federated Login for LastPass Using Azure Active Directory2
- Summary2
- System Requirements.....2
- Before you begin2
- Step #1: Generate a Provisioning Token3
- Step #2: Configure Azure AD with LastPass4
- Step #3: Capture the Application ID and OpenID Connect from Azure AD.....9
- Step #4: Configure Federated login settings in LastPass 10
- Step #5: Configure a Redirect URI in Azure AD 11
- Step #6: Configure API permissions in Azure AD..... 12
- Step #7: Add users to the LastPass app in Azure AD 14
- Step #8: Set up Multifactor Authentication on Azure AD (optional)..... 14
- You're all set!..... 14
- Troubleshooting & Tips..... 15
- Contact Us 15

Set Up Federated Login for LastPass Using Azure Active Directory

This guide provides setup instructions for using LastPass with Azure Active Directory (Azure AD) for your LastPass Enterprise or LastPass Identity account.

Summary

LastPass supports the following provisioning features:

- Create Users
- Update User Attributes
- Sync User Groups
- Deactivate or Disable Users

Federated login for LastPass Enterprise and LastPass Identity accounts allows users to log in to LastPass using their Azure AD account (instead of a username and separate Master Password) to access their LastPass Vault.

System Requirements

To enable federated login for LastPass using Azure AD, the following is required:

- An active Premium subscription to Microsoft Azure AD
- An active trial or paid LastPass Enterprise or LastPass Identity account
- An active LastPass Enterprise or LastPass Identity admin (required when activating your trial or paid account)

The LastPass Azure AD SCIM endpoint for federated login does not require any software installation.

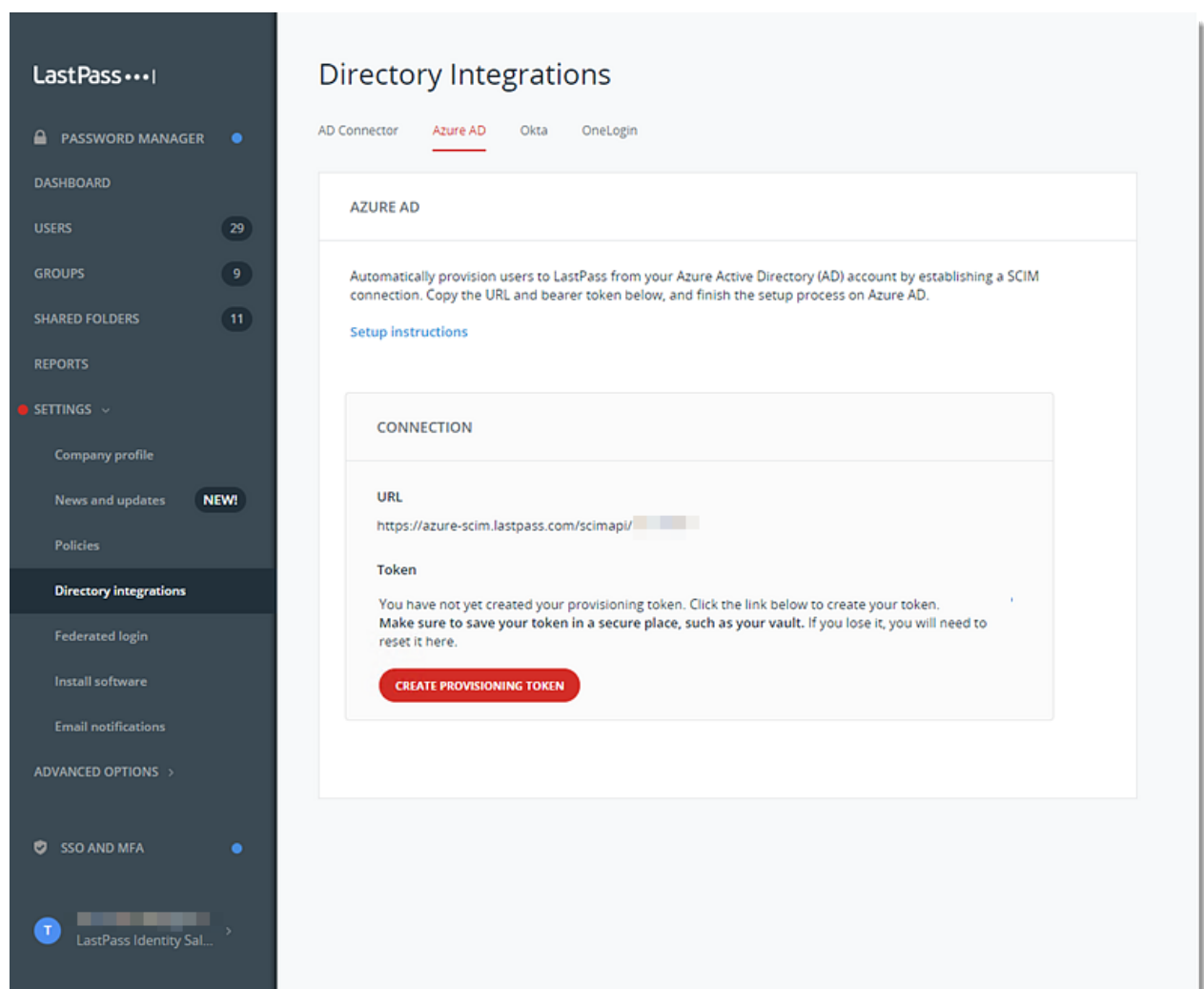
Before you begin

- It is **required** that you [enable the “Permit super admins to reset Master Passwords” policy](#) for at least 1 LastPass admin (who is also a non-federated admin) in the LastPass Admin Console. This ensures that all LastPass user accounts can still be recovered (via Master Password reset) if a critical setting is misconfigured or changed for federated login after setup is complete.
- It is helpful to open a text editor application so that you can copy and paste values that will be used between your LastPass Admin Console and the Azure AD Admin portal.

Step #1: Generate a Provisioning Token

1. Access the LastPass Admin Console by opening a web browser and going to <https://lastpass.com/company/#!/dashboard>.
2. Enter your administrator username and Master Password, then click **Log In**.
3. Select **Settings > Directory integrations** in the left navigation.
4. Click on the **Azure AD** tab.
5. Copy the URL and paste it into your text editor application.
6. Click the **Create Provisioning Token** to generate it, then copy the token and paste it into your text editor application.

Note: If you navigate away from the Azure AD tab within the Directory Integrations page, the Provisioning Token will no longer be accessible through the LastPass Admin Console. If the Token is lost, a new one can be generated, but this will invalidate the previous code. Any process that used the old Token will need to be updated with the new one. A new Provisioning Token can be generated by navigating back to the Azure AD tab and clicking **Reset Provisioning Token**.



Step #2: Configure Azure AD with LastPass

Once you have acquired the URL and Provisioning Token, you will need to enter them into the Azure AD Admin portal.

1. Log in to your Azure AD portal with your administrator account credentials at <https://portal.azure.com>.
2. Navigate to **Azure Active Directory > Enterprise Applications > New application > All > Non-gallery application**.
3. Enter a name for your application (LastPass) and click **Add** to create an app object. The application object created is intended to represent the target app (for which you would be provisioning and setting up single sign-on, not just as the SCIM endpoint).
4. Select the **Provisioning** tab in the left navigation.
5. For Provisioning Mode, use the drop-down menu and select **Automatic**.
Under Admin Credentials, enter the following:
 - a) Locate the "Tenant URL" field and paste the URL you copied from the LastPass Admin Console.
 - b) Locate the "Secret Token" field and paste the Provisioning Token you copied from the LastPass Admin Console.
6. Click Test Connection to have Azure AD attempt to connect to the SCIM endpoint. If the attempts fail, error information is displayed.
7. If the connection test succeeds, click **Save** to store the admin credentials.
8. Next, select **Mappings**.
9. First, modify user object mappings with the following:
 - a) Check the box for **Show advanced options** at the bottom of Attribute Mapping.
 - b) Click **Edit attribute list for...**

Attribute Mapping

Save

Discard

Source Object Scope

All records

Target Object (customappsso)

urn:ietf:params:scim:schemas:extension:enterprise:2.0:User

Target Object Actions

☒ Create
 ☒ Update
 ☒ Delete

Attribute Mappings

Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

Azure Active Directory Attribute	Customappsso	Matching	
objectId	externalId	1	Delete
Switch([IsSoftDeleted], , "False", "True", "True")	active		Delete
displayName	displayName		Delete
userPrincipalName	userName		Delete
Join(" ", [givenName], [surname])	name.forma...		Delete

Add New Mapping

☒ Show advanced options

Supported Attributes

View and edit the list of attributes that appear in the source and target attribute lists for this application.

The attribute list for Azure Active Directory is up to date with all supported attributes.
 [Request additional attributes you would like to see supported here.](#)

[Edit attribute list for customappsso](#)

- c) In the **Edit Attribute List**, make the following selections:
- Name = id, Type = String – Check the boxes for **Primary** and **Required**
 - Name = userName, Type = String – Check the box for **Required**
 - Name = externalID, Type = String – Check the box for **Required**
- Click **Save** and return to Attribute Mapping.
- d) Click **Save** and return to Attribute Mapping.

Edit Attribute List							
<div> <div>Save</div> <div>Discard</div> </div>							
customappsso User Attributes							
NAME	TYPE	PRIMAR...	REQUIR...	MULTI...	EXACT ...	API EXPRESSION	REFERENCED OBJE...
id	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<div>Delete</div>
active	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<div>Delete</div>
displayName	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<div>Delete</div>
userName	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<div>Delete</div>
externalId	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<div>Delete</div>

10. Set 4 Attribute Mapping rules. By default, Azure may have created mappings already, but those can be modified or deleted if needed. Only the required 4 mappings should be present after editing, and must be configured correctly:

- ExternalID** – Use the objectID attribute from Azure AD and set this as a matching attribute with Precedence set as **1**.
 - TIP!** This should be the only mapping with any Precedence set. For any existing mapping you have set with a Precedence, set that mapping's Precedence to greater than 1, then create the ExternalID mapping outlined above and delete all unneeded mappings.
- Active** – The default Azure AD mapping can be used, or a custom one which will be used to set the user as enabled/disabled in LastPass.
- DisplayName** – Use any property from Azure AD. This should be a string which will be the synchronized user's name in LastPass.
- UserName** – Map the user's email address from Azure AD. Please note that the userPrincipalName might not be equal to the email address. In this case, use an attribute from Azure AD which contains the email address the user will utilize and can read (e.g., Mail or in most cases, userPrincipalName should be fine).

WARNING! If you already have users in LastPass, their email address **MUST** match the Azure AD attribute mapped to the userName value. If this is not mapped correctly, a duplicate user will be created for every existing user in LastPass.

Attribute Mapping

Save

Discard

Source Object (Azure Active Directory)

User

Source Object Scope

All records

Target Object (customappsso)

urn:ietf:params:scim:schemas:extension:enterprise:2.0:User

Target Object Actions

☒ Create
☒ Update
☒ Delete

Attribute Mappings

Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

AZURE ACTIVE DIRECTORY ATTRIBUTE	CUSTOMAPPS...	MATCHING ...	
objectId	externalid	1	Delete
Switch([IsSoftDeleted], , "False", "True", "Tr	active		Delete
displayName	displayName		Delete
userPrincipalName	userName		Delete

Add New Mapping

×

Edit Attribute

×

A mapping lets you define how the attributes in o...

Mapping type ⓘ

Direct

* Source attribute ⓘ

objectId

Default value if null (optional) ⓘ

* Target attribute ⓘ

externalid

Match objects using this attribute

Yes

Matching precedence ⓘ

1

Apply this mapping ⓘ

Always

11. Click **Save**, then return to the Provisioning settings and select **Mappings** (from **Step #9** above).
12. Next, modify group object mappings as follows:
 - a) Check the box for **Show advanced options** at the bottom of Attribute Mapping.
 - b) Click **Edit attribute list for...**
 - c) In the Edit Attribute List, make the following selections:
 - Name = id, Type = String – Check the boxes for **Primary** and **Required**
 - Name = externalID, Type = String – Check the box for **Required**
 - Name = displayName, Type = String – Check the box for **Required**
 - Name = members, Type = Reference – Check the box for **Multi-Valued**, then set referenced objects for:
 - urn:ietf:params:scim:schemas:core:2.0:Group
 - urn:ietf:params:scim:schemas:extension:enterprise:2.0:User
 - d) Click **Save** and return to Attribute Mapping.

Attribute Mapping

Save X Discard

Source Object (Azure Active Directory)
Group

Source Object Scope
All records

Target Object (customappsso)
urn:ietf:params:scim:schemas:core:2.0:Group

Target Object Actions
☒ Create
☒ Update
☒ Delete

Attribute Mappings
Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

AZURE ACTIVE DIRECTORY ATTRIBUTE	CUSTOMAPPS...	MATCHING ...
objectId	externalid	1

Edit Attribute List

Save X Discard

customappsso Group Attributes

NAME	TYPE	PRIMARY...	REQUIR...	MULTI...	EXACT ...	API EXPRESSION	REFERENCED OBJE...
id	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Delete
externalid	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Delete
displayName	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Delete
members	Reference	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		2 selected Delete

members

2 selected

the schema of

13. Set 3 Attribute Mapping rules, as follows:

- ExternalID** – Use the objectId attribute from Azure AD and set this as a matching attribute with Precedence set as **1**. This should be the only mapping with any Precedence set.
- DisplayName** – Use any attribute for group name.
- Members** – User members from Azure AD.

Attribute Mapping

Save X Discard

Name
Synchronize Azure Active Directory Groups to customappsso

Enabled
Yes No

Source Object (Azure Active Directory)
Group

Source Object Scope
All records

Target Object (customappsso)
urn:ietf:params:scim:schemas:core:2.0:Group

Target Object Actions
☒ Create
☒ Update
☒ Delete

Attribute Mappings
Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

AZURE ACTIVE DIRECTORY ATTRIBUTE	CUSTOMAPPS...	MATCHING ...
objectId	externalid	1 Delete
displayName	displayName	Delete
members	members	Delete

Add New Mapping

- ### Step #3: Capture the Application ID and OpenID Connect from Azure AD

- Home > **logmeinaccountspidevlogmet (Default Directory) - App registrations (Preview)** > **LastPass**

LastPass

PREVIEW

Overview

Quickstart

Manage

Branding

Authentication

Certificates & secrets

API permissions

Expose an API

Owners

Manifest

Support + Troubleshooting

Troubleshooting

New support request

Delete

Endpoints

Display name

: LastPass

Application (client) ID

: a97fb9d2-ad5f-47b7-9d56-a1c9-111111111111

Directory (tenant) ID

: 3da690d3-8bd8-4056-b645-111111111111

Object ID

: 1f560493-6360-4bc2-8b57-111111111111

Call APIs

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

View API Permissions

Sign in users in 5 minutes

Use our SDKs to sign in users and call APIs in a few steps

View all quickstart guides

Endpoints

PREVIEW

OAuth 2.0 authorization endpoint (v2)

https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/oauth2/authorize

OAuth 2.0 token endpoint (v2)

https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/oauth2/token

OAuth 2.0 authorization endpoint (v1)

https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/oauth2/authorize

OAuth 2.0 token endpoint (v1)

https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/oauth2/token

OpenID Connect metadata document

https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/.well-known/openid-configuration

Microsoft Graph API endpoint

https://graph.microsoft.com

Federation metadata document

https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/federationmetadata

WS-Federation sign-on endpoint

https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/wsfed/signon

SAML-P sign-on endpoint

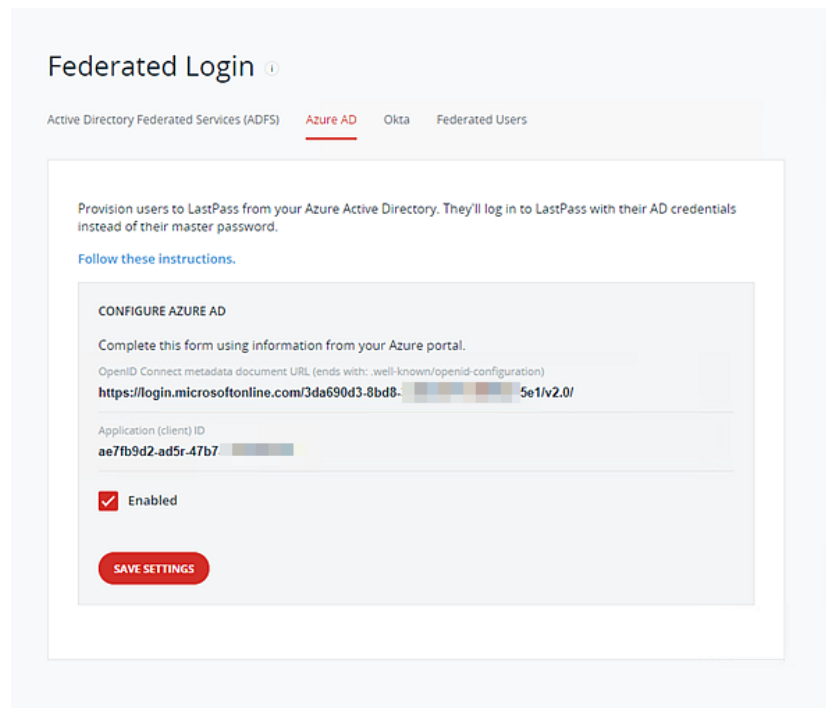
https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/samlp/signon

SAML-P sign-out endpoint

https://login.microsoftonline.com/3da690d3-8bd8-4056-b645-111111111111/samlp/signout

Step #4: Configure Federated login settings in LastPass

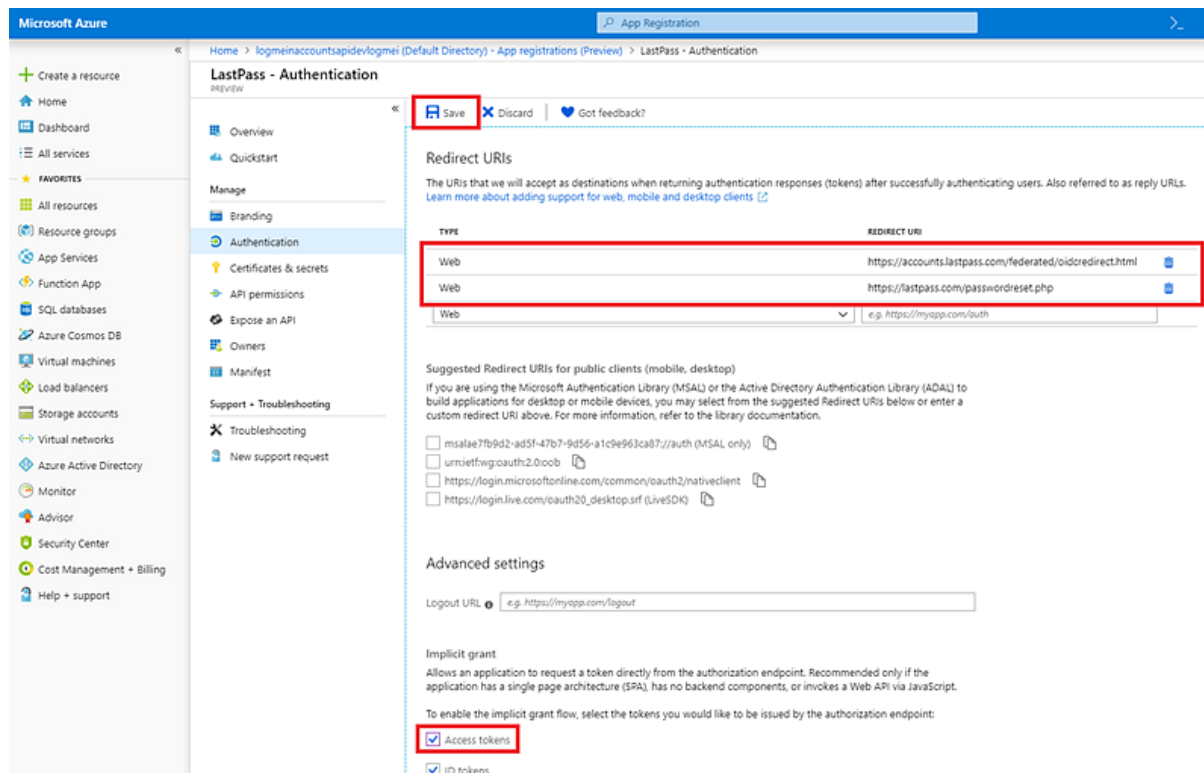
1. Go back to the LastPass Admin Console, then select **Settings > Federated login** in the left navigation.
2. Select the **Azure AD** tab, then enter the following:
 - In the "Directory (tenant) ID" field, paste the **OpenID Connect metadata document** from **Step #3** (in the previous section).
 - In the "Application (client) ID" field, paste the **Application (client) ID** from **Step #3** (in the previous section).
3. Check the box for **Enabled**.
4. Click **Save Settings** when finished.



The screenshot shows the 'Federated Login' configuration page in the LastPass Admin Console. The page has a header 'Federated Login' with a help icon. Below the header are four tabs: 'Active Directory Federated Services (ADFS)', 'Azure AD' (which is selected and underlined in red), 'Okta', and 'Federated Users'. The main content area contains instructions: 'Provision users to LastPass from your Azure Active Directory. They'll log in to LastPass with their AD credentials instead of their master password.' and a link 'Follow these instructions.' Below this is a 'CONFIGURE AZURE AD' section with the instruction 'Complete this form using information from your Azure portal.' It contains two input fields: 'OpenID Connect metadata document URL (ends with: .well-known/openid-configuration)' with the value 'https://login.microsoftonline.com/3da690d3-8bd8-5e1/v2.0/' and 'Application (client) ID' with the value 'ae7fb9d2-ad5r-47b7'. There is a checkbox labeled 'Enabled' which is checked. At the bottom of the form is a red 'SAVE SETTINGS' button.

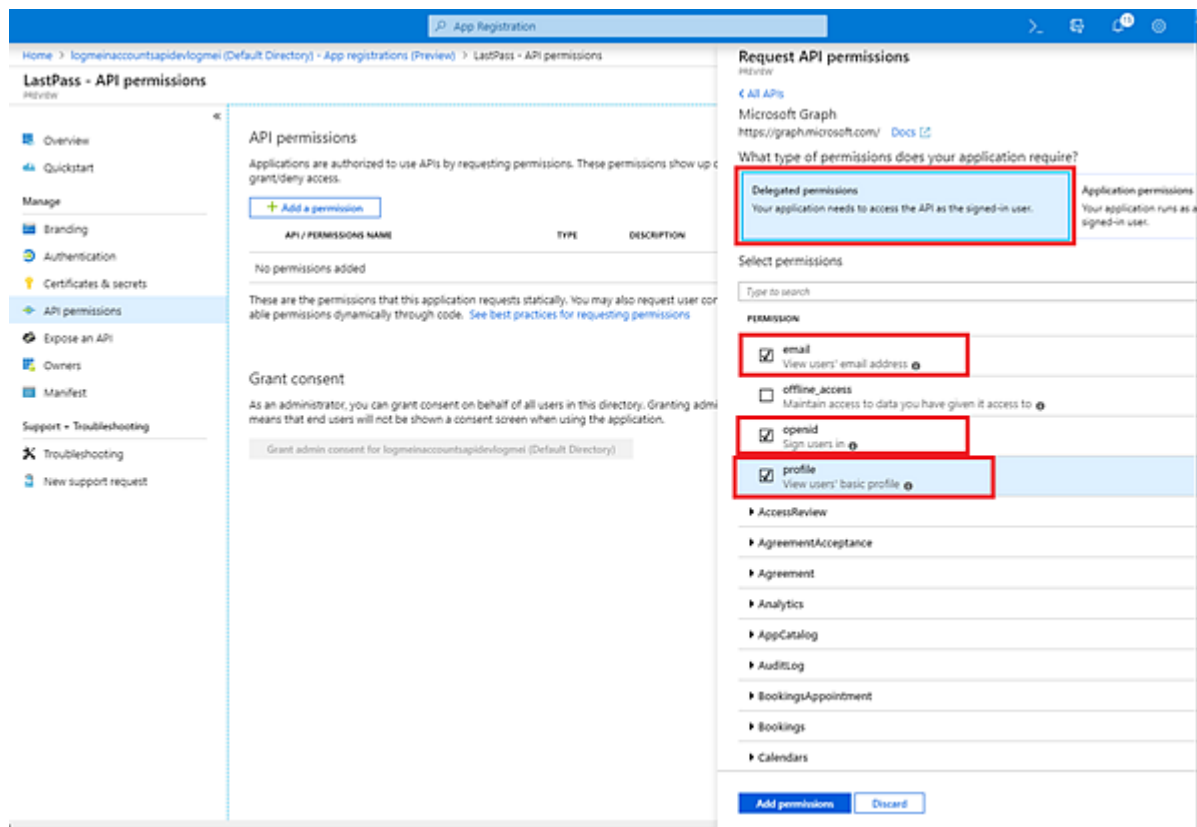
Step #5: Configure a Redirect URI in Azure AD

1. In the Azure AD portal, with your LastPass application selected, select **Overview** in the left navigation.
2. Under Redirect URI in the upper-right navigation, click **Add a Redirect URI**.
3. Add the first Redirect URI, as follows:
 - For the Type column, use the drop-down menu and select **Web**
 - For the Redirect URI column, enter <https://lastpass.com/passwordreset.php>
4. Add the second Redirect URI, as follows:
 - For the Type column, use the drop-down menu and select **Web**
 - For the Redirect URI column, enter <https://accounts.lastpass.com/federated/oidcredirect.html>
5. Under the Advanced settings, check the boxes to enable the following settings:
 - Access tokens
 - ID tokens
6. Click **Save** when finished.

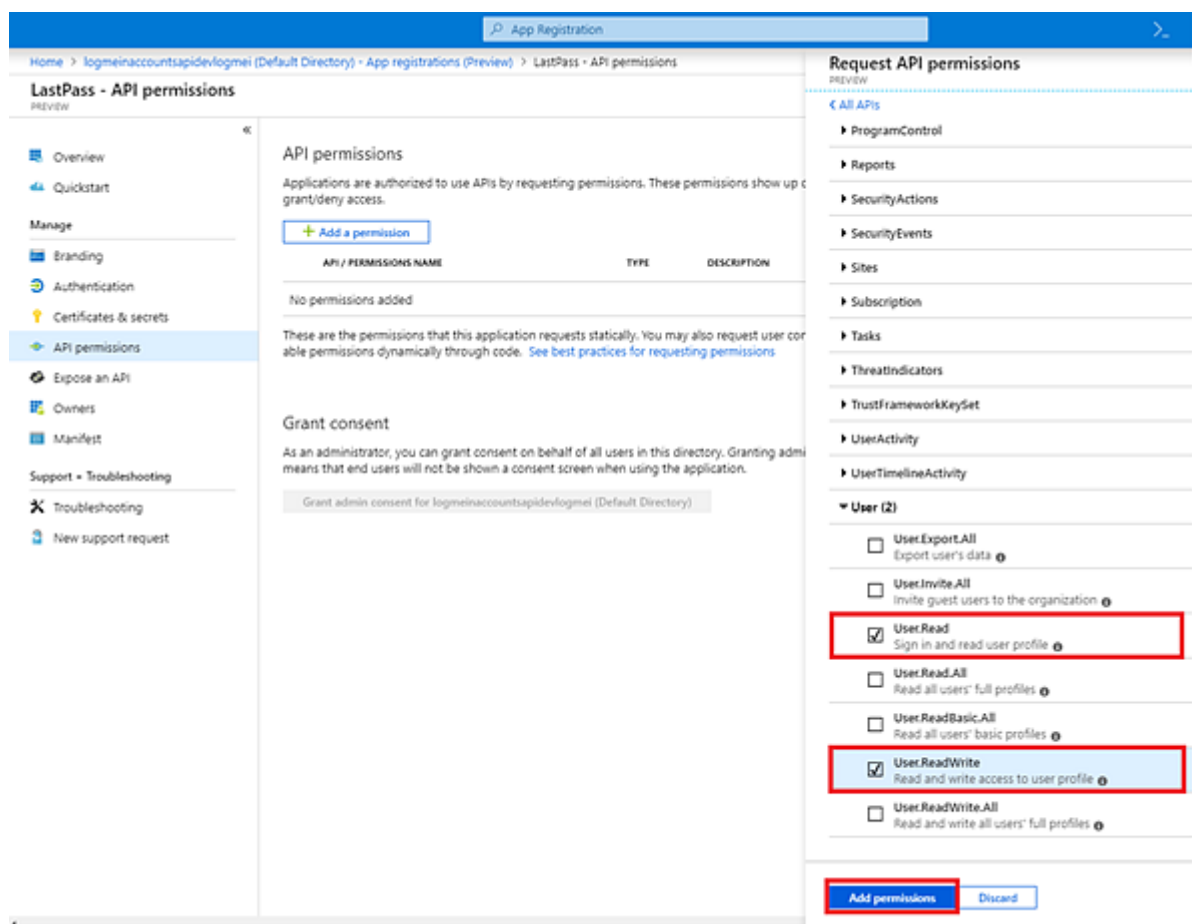


Step #6: Configure API permissions in Azure AD

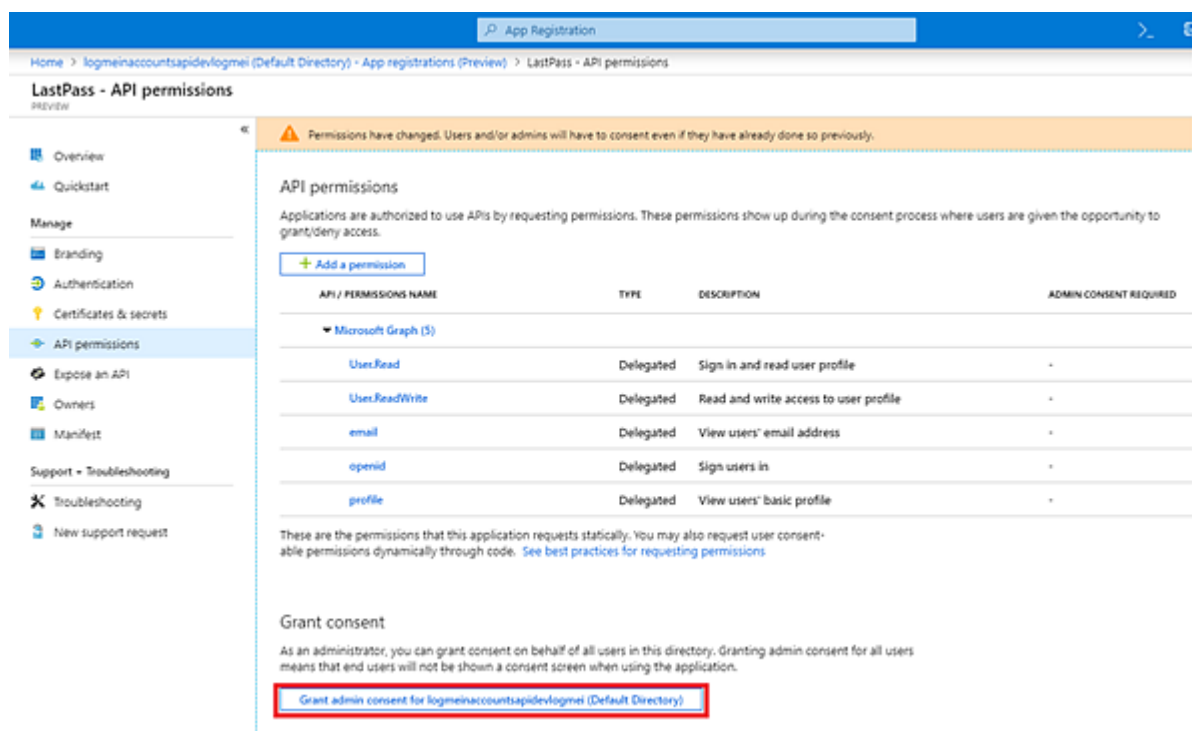
1. In the Azure AD portal, select **API permissions** in the left navigation.
2. Click the **Add a permission** button, then select **Microsoft Graph**.
3. In the right navigation, select **Delegated permissions**.
4. Under the Permission menu, check the boxes to enable the following permission settings:
 - email
 - openid
 - profile



5. Under the User menu, check the boxes to enable the following user settings:
 - User.Read
 - User.ReadWrite
6. When finished, click **Add permissions**.

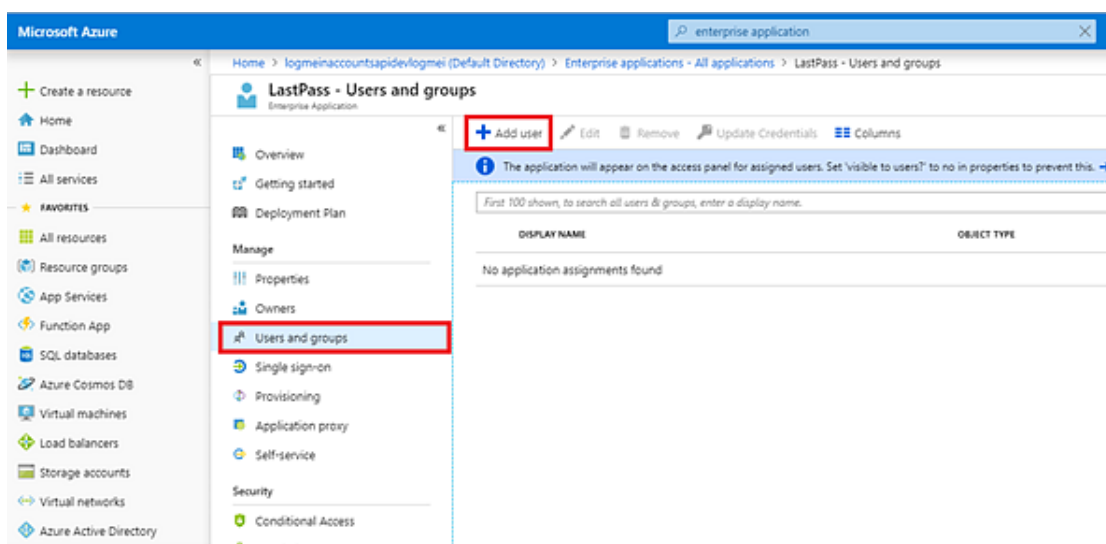


7. Under Grant consent, click the **Grant <your LastPass application name>** hyperlink to finish configuring API permissions for your LastPass app.



Step #7: Add users to the LastPass app in Azure AD

1. In the Azure AD portal, with your LastPass application selected, go to **Overview > Enterprise applications** in the left navigation.
2. Select your newly created LastPass application.
3. Select **Users and groups** in the left navigation.
4. Click **Add user**.
5. Locate each of the users and/or groups in the list, then click **Select** to grant access to the LastPass app.



Step #8: Set up Multifactor Authentication on Azure AD (optional)

If desired, you can [set up Multifactor Authentication at the Azure AD \(Identity Provider\) level](#).

You're all set!

You have successfully set up your LastPass Enterprise or LastPass Identity account to use federated login with your Azure Active Directory. All of your newly populated federated users will receive a Welcome email informing them that they can now log in to use LastPass. Please note that your LastPass users must log in using the LastPass web browser extension in order to use federated login for their Azure AD account with LastPass.

- To learn more about deploying the LastPass web browser extension to your organization, please see [Install LastPass Software Using the Admin Console](#).
- To see your end users' experience, please see [Federated Login Experience for LastPass Users](#).
- If your end users have linked personal accounts associated with their federated login account, please see [How do I verify my linked personal account?](#)
- To convert a non-federated user to a federated user, please see [How do I convert an existing LastPass user to a federated \(Azure AD\) user?](#)

Troubleshooting & Tips

- It is **strongly recommended** that you have at least 1 LastPass admin that is [enabled with the “Permit super admins to reset Master Passwords” policy](#).

Contact Us

If you have not started a LastPass Enterprise or LastPass Identity trial, please contact our Sales team at lastpass.com/contact-sales for more information.

For additional help, please see [Set Up Federated Login for LastPass Using Azure Active Directory](#), and if desired, select a contact option at the bottom of the article.