

Single Sign-On

Last Modified on 04/03/2024 3:38 pm EDT

Many IT departments have configured some kind of Single Sign-On (SSO) for users to have a single set of credentials to access common applications. If you're looking to integrate DevResults more fully with your existing IT environment, you now have the option to integrate your SSO provider with DevResults to handle user credential authentication. We support SSO integration with Azure Active Directory, with Active Directory via OAuth2, and with Okta. This page will walk you through how DevResults-SSO integration works and provide instructions on how to set it up.

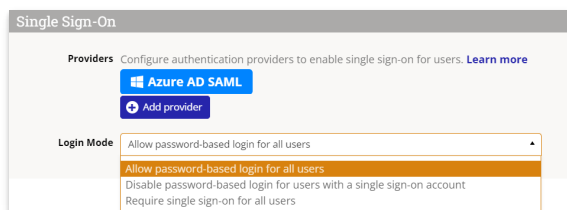
Jump to:

- [User Administration](#)
- [SSO Integration Setup](#)
 - [Azure Active Directory Setup](#)
 - [Active Directory via OAuth2 Setup](#)
 - [Okta Setup](#)
- [Frequently Asked Questions](#)

User Administration

User accounts in DevResults are created from the SSO provider once the user tries to log in to DevResults using those credentials and accepts the permissions.

The two accounts — SSO and DevResults — are linked but are independent. The **Login Mode** configuration determines whether or not password-based logins will still be allowed for some or any users once a SSO integration has been setup (see the following sections for instructions).



- **Allow password-based login for all users:** any user can establish and use a password to log into DevResults.
- **Disable password-based login for users with a single sign-on account** any user with an active SSO account will be prevented from using a password-based login, but all other users (e.g. external partners, local staff, etc.) will still be able to log on using passwords.
- **Require single sign-on for all users:** password-based logins are fully disabled for all users; the SSO platform manages all access to DevResults for all users.

Please note that deleting or deactivating a user from Active Directory **does not** automatically delete or deactivate their account in DevResults. If they've only ever logged in using SSO, it will prevent them from logging in. However, if they ever established a DevResults password outside of SSO, they would still be able to access the site using that password. Therefore, we recommend still having a designated DevResults site administrator oversee permissions and account deactivation within DevResults.

[back to top](#)

SSO Integration Setup

We support SSO integration with Azure Active Directory and with Active Directory via OAuth2. For both setups, the overall process is:

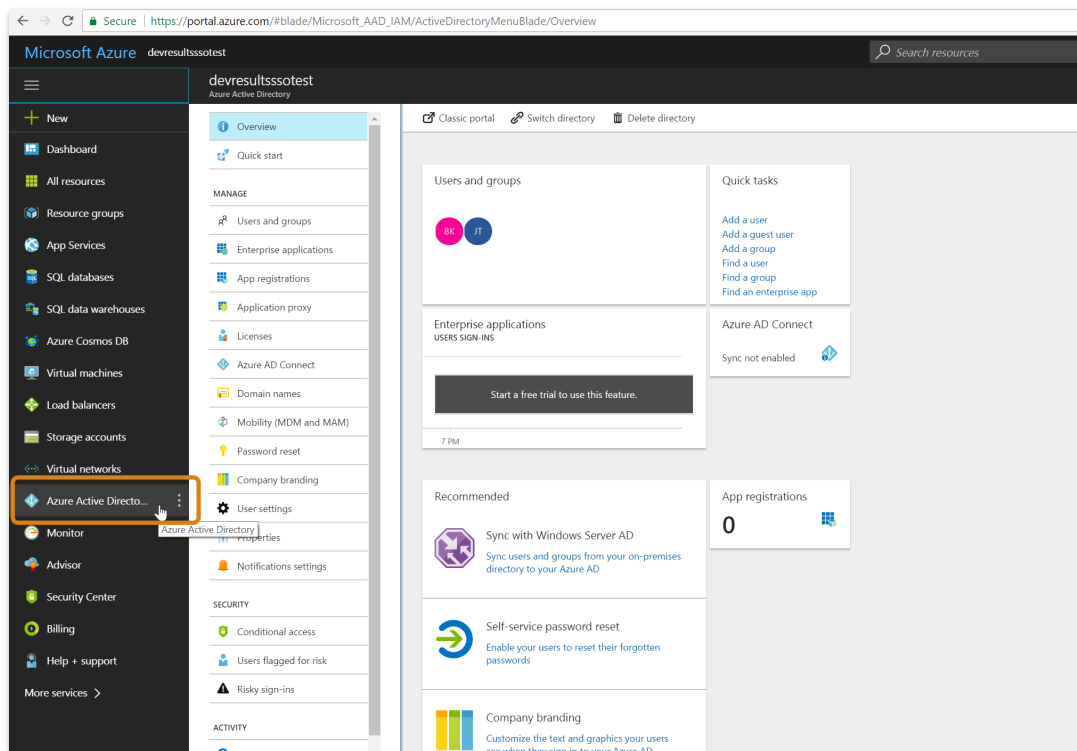
- Do some configuration and setup on your SSO provider side to gather the information you need.
- Enter some of that information in DevResults so it recognizes your SSO provider.

[back to top](#)

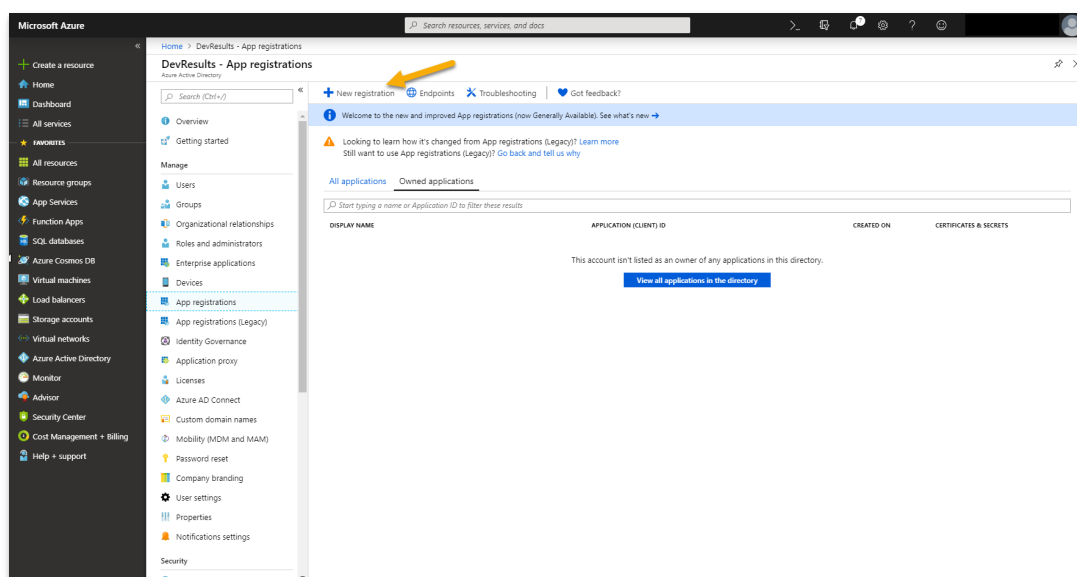
Azure Active Directory

For Azure Active Directory, you will need to have a registered application within Azure Active Directory, and you'll need to know the Application ID and have a Key generated. We'll provide a quick outline of those steps here as a sample and then walk you through the DevResults side of the setup.

In the Azure Portal, click "Azure Active Directory" in the sidebar to the left.



Select **App registrations**. Click on **New application registration** in the right-hand pane:



Enter a **Name** that will help you to identify this as your DevResults SSO config later. Next, select the accounts types you'd like to have access to DevResults. Click Register. For your **Redirect URI**, choose "Web" and enter your DevResults home page URL. Then click **Register**.

The screenshot shows the 'Register an application' form in the Microsoft Azure portal. The form is titled 'Register an application' and has a close button (X) in the top right corner. The form contains the following sections:

- Name:** A text input field with the value 'DevResults' and a checkmark icon on the right.
- Supported account types:** A section titled 'Who can use this application or access this API?' with three radio button options:
 - ☒ Accounts in this organizational directory only (DevResults)
 - ☐ Accounts in any organizational directory
 - ☐ Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)A link 'Help me choose...' is below the options.
- Redirect URI (optional):** A section titled '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.' It contains a dropdown menu set to 'Web' and a text input field with the value 'https://testaccount@devresults.com' and a checkmark icon on the right.

At the bottom of the form, there is a link 'By proceeding, you agree to the Microsoft Platform Policies' and a blue 'Register' button.

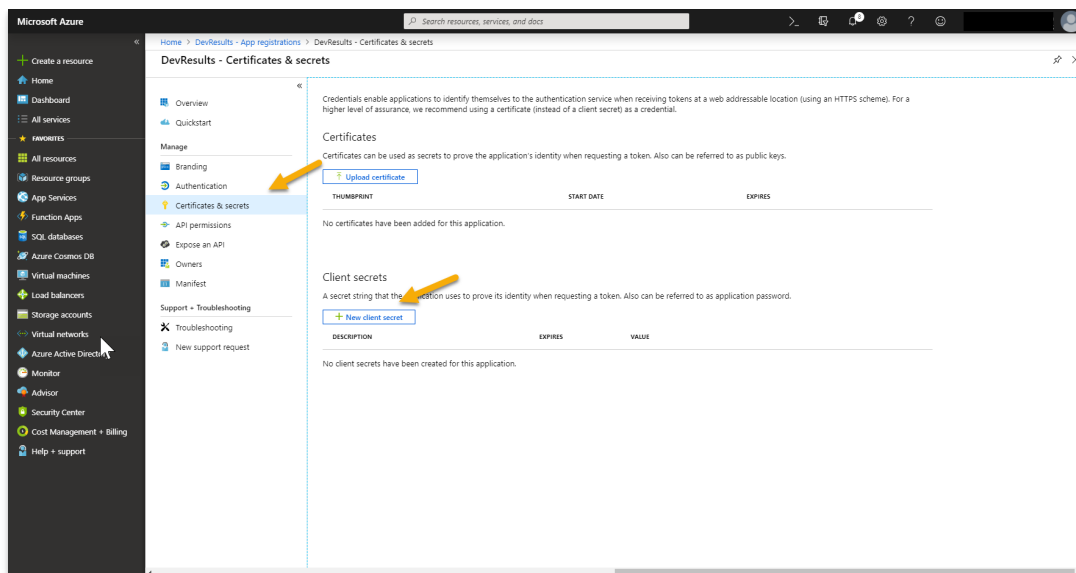
Once the application is created, click on it to open details. Copy the **Application ID**--you'll need this for the DevResults side of configuration.

The screenshot shows the 'DevResults' application details page in the Microsoft Azure portal. The page has a left sidebar with navigation options: Overview, Quickstart, Manage, Branding, Authentication, Certificates & secrets, API permissions, Expose an API, Owners, Manifest, Support + Troubleshooting, and New support request. The main content area displays the following information:

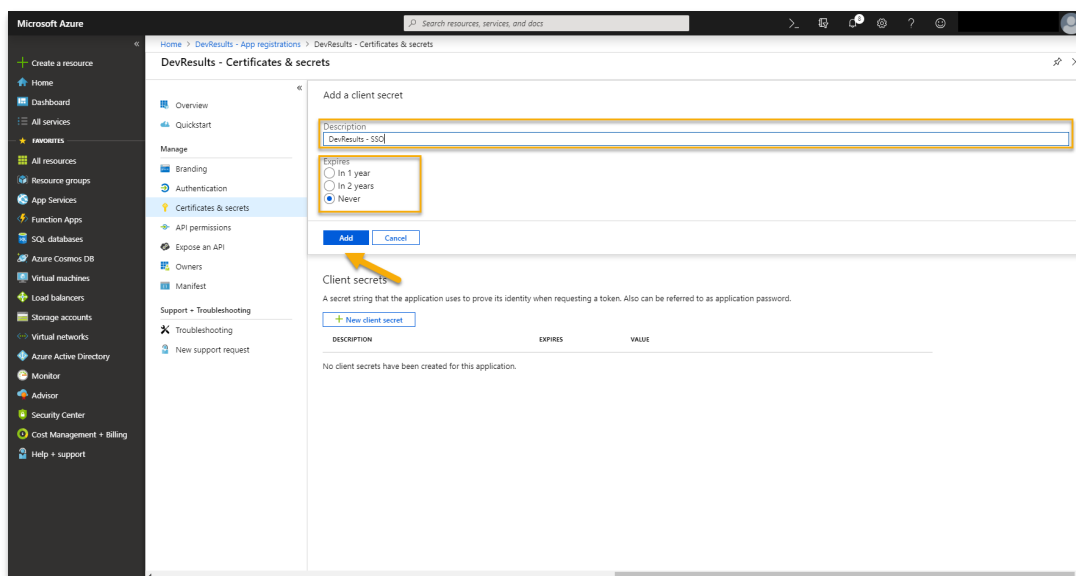
- Display name:** DevResults
- Application (client) ID:** 52fcd321-551f-4b65-b83f-513b0991cc38 (highlighted with a yellow box)
- Directory (tenant) ID:** 3408008c-3aee-4367-85c2-fc6a2a0632
- Object ID:** 062dc744-40ab-4b13-8bd3-3bb0424da4ab
- Supported account types:** My organization only
- Redirect URIs:** 1 web, 0 public client
- Managed application in ...:** DevResults

Below the application details, there is a 'Call APIs' section with a 'View API Permissions' button. To the right of the 'Call APIs' section is a 'Documentation' section with links to 'Microsoft identity platform', 'Authentication scenarios', 'Authentication libraries', 'Code samples', 'Microsoft Graph', 'Glossary', and 'Help and Support'. At the bottom of the page, there is a 'Sign in users in 5 minutes' section with a 'View all quickstart guides' button.

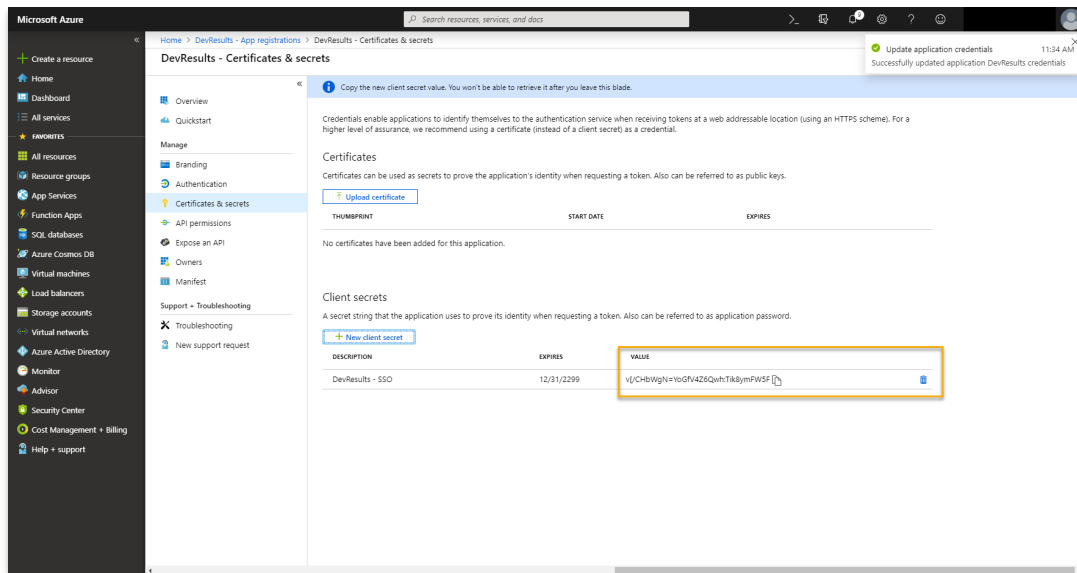
Click **Certificates & Secrets** and then **New Client Secret**.



Enter DevResults-SSO (or something else useful) in the **Description**. Select a **Duration** consistent with your organization's security policies. Then click **Add**.



When you click **Save**, Azure will generate a Key in the Value. Copy that **Value**. (If you don't copy this, you'll need to regenerate it later to add it to DevResults.)



We'll need to return to Azure Portal in a moment, but our next steps happen in the DevResults app. Now that there's an application in Azure Active Directory, we can set up the DevResults portion. In your DevResults site, go to **Administration > Settings**.



In the **Single Sign-On** section, click the **Add provider** button.

The screenshot shows the 'Program Settings' page in the DevResults application. The 'Single Sign-On' section is highlighted with an orange box, indicating the next step in the configuration process. The page includes various settings sections like Login Message, Branding, Document Storage, Fiscal Year, and IATI.

This will open the Authentication Provider pop-up. You'll need to complete these sections to configure the integration:

- **Name:** Provide a name for this SSO. We used Azure AD in our example. This label will appear to end-users on their login screen ("Use my {Name} account") so be sure it's something your users will understand!
- **Authentication Provider:** Currently DevResults supports Azure Active Directory and Active Directory via OAuth2. For this example, select Azure Active Directory.
- **Default Group:** When new users are created in DevResults from Active Directory, what group should they be added to by default? For our example, we used our standard **Users** group, but you can choose any existing group in your DevResults site.
- **Notification Email:** If you want anyone to be notified when a new DevResults user is created from Active Directory, enter their email address here. (Optional)
- **Is Active?:** You can uncheck this box if you don't want this provider to be currently used; otherwise, check the box to make sure it's going to be used.
- **Redirect Url:** This will autopopulate once you've completed the rest of the form. You'll need to copy this value - we'll be adding to to Azure Active Directory in a moment.
- **Provider Settings: Application ID:** Paste in the Application ID we copied from Azure Portal earlier
- **Provider Settings: Secret:** Paste in the Key's Value from the last set of steps in the Azure Portal directions

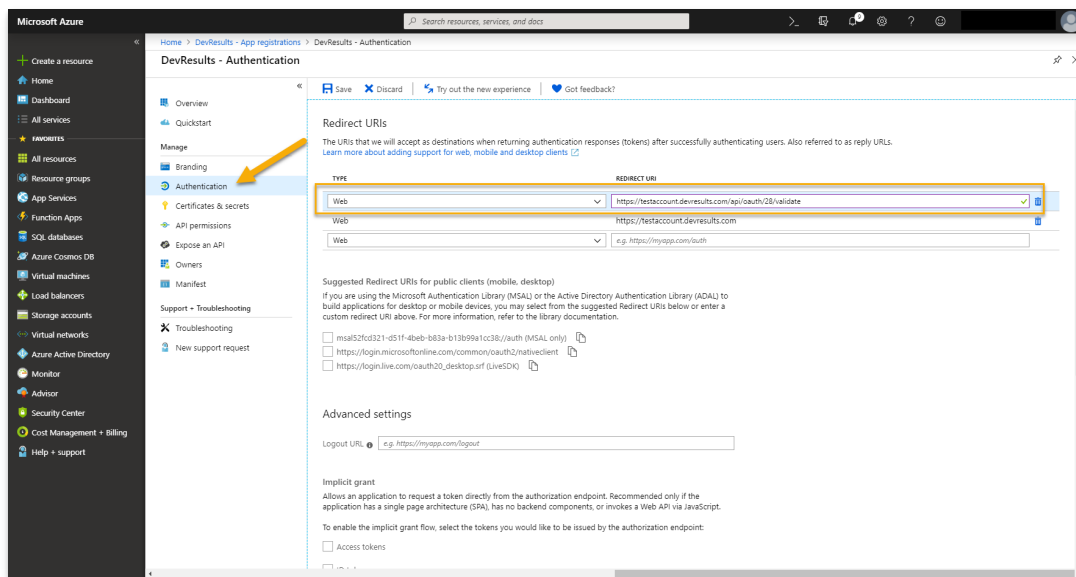
Once these fields are filled out, click the **Save** button to save your configuration settings.

The 'Authentication Provider' pop-up form is shown with the following fields and values:

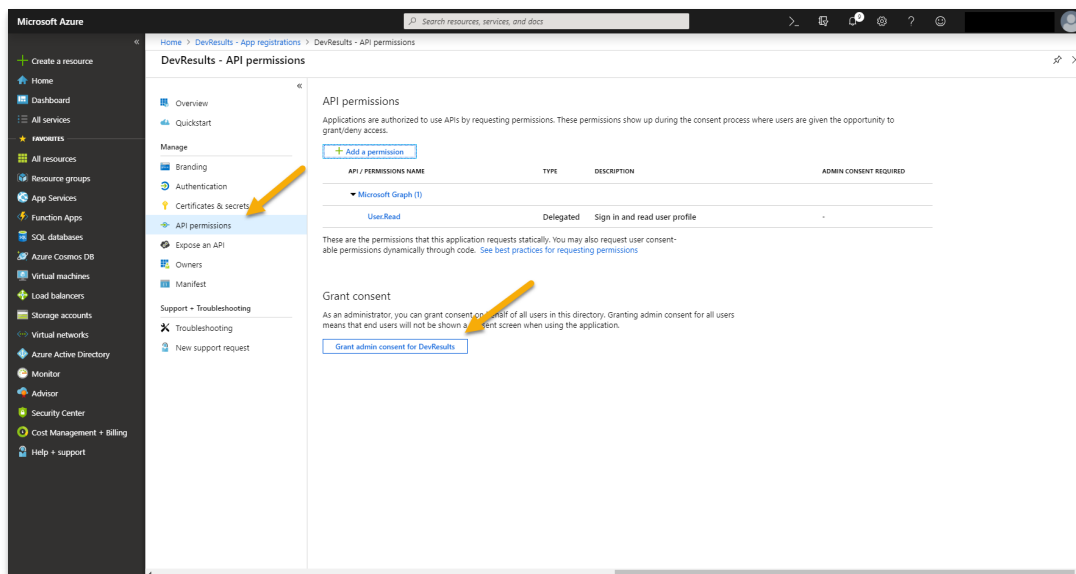
- Name:** Azure AD
- Authentication Provider:** Azure Active Directory
- Default Group:** Viewers
- Notification Email:** Who should be notified when a new user is created from this provider? leslie@devresults.com, ritika@devresults.com
- Is Active?:** ☒ Allow users to sign in with this provider
- Redirect Uri:** https://kate.devresults.com/api/oauth/4/validate
- Application ID:** 9db54d90-7b5c-4457-b66e-683919d2b6ee
- Secret:** Z5sts/zi+Ydi0Zz2Ki2DmG8KZg0p7nP8sp193/HikbY=

At the bottom, there are buttons for 'Cancel', 'Delete', and 'Save'.

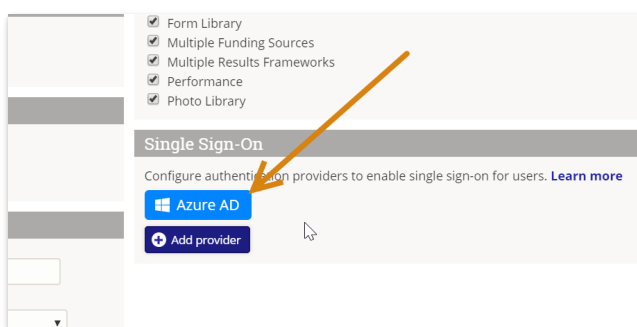
Now, we'll need to return to Azure Active Directory. Click **Authentication**. Then add the value you copied from the Redirect URI field above. You can ignore the remainder of the Authentication page.



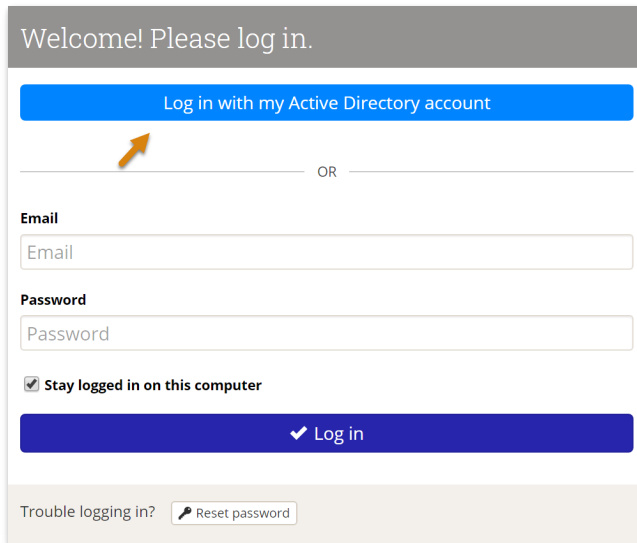
Now, click **API Permissions**. Then, click **Grant admin consent for DevResults**. (The text of this button may be different if you named the app something other than 'DevResults' in the initial configuration.)



That's it! You've done everything you need to do in Azure. Back in DevResults, you'll see the provider appear on the DevResults Administration > Settings page. You can click to edit or delete it.



With the SSO set up and enabled, your DevResults Login page will look slightly different. It will now show a "Log in with my ___ account" option as well as the regular login. Users can either use their DevResults username and password (if they already have one) or their SSO.



>Welcome! Please log in.

Log in with my Active Directory account

OR

Email

Email

Password

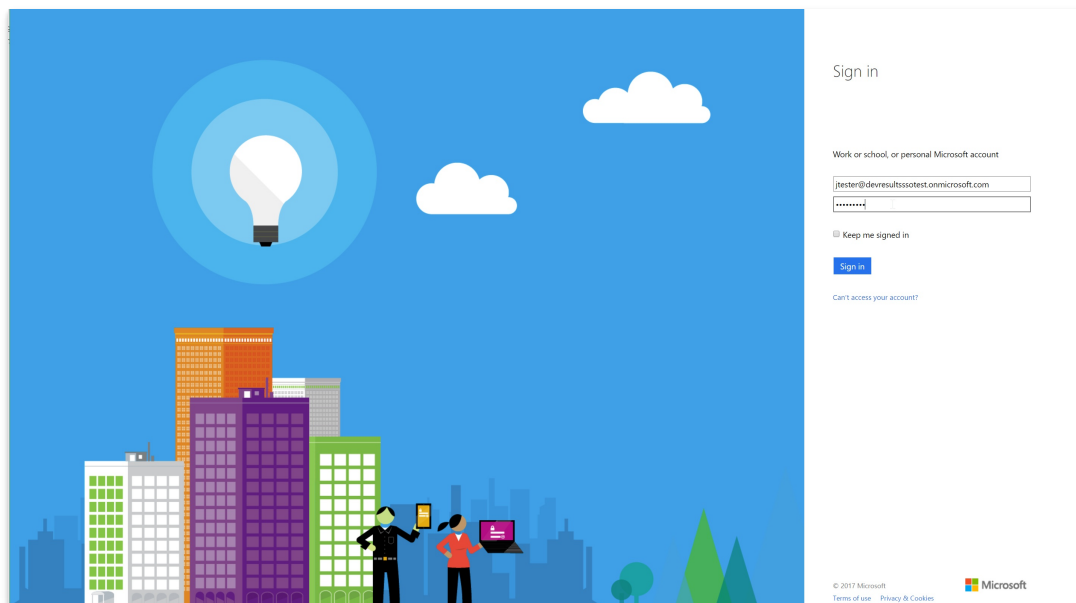
Password

☒ Stay logged in on this computer

✓ Log in

Trouble logging in? [Reset password](#)

The first time they log in using the Azure AD account, they'll be redirected to a Microsoft login screen where they'll need to enter their credentials for Active Directory.



Sign in

Work or school, or personal Microsoft account

☐ Keep me signed in

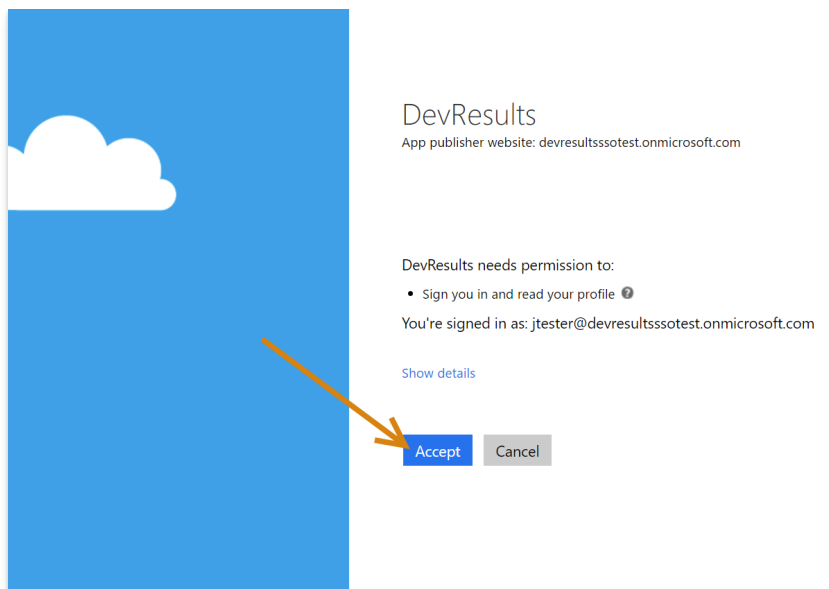
Sign in

Can't access your account?

© 2017 Microsoft
[Terms of use](#) [Privacy & Cookies](#)

Microsoft

Once they've entered credentials, they will need to grant DevResults permission to "sign you in and read your profile". This is only necessary the first time the user logs in using this method.



Once that's done, all future sign-ins using the Use my Azure AD account should just work. The page should now redirect to the DevResults site.

[back to top](#)

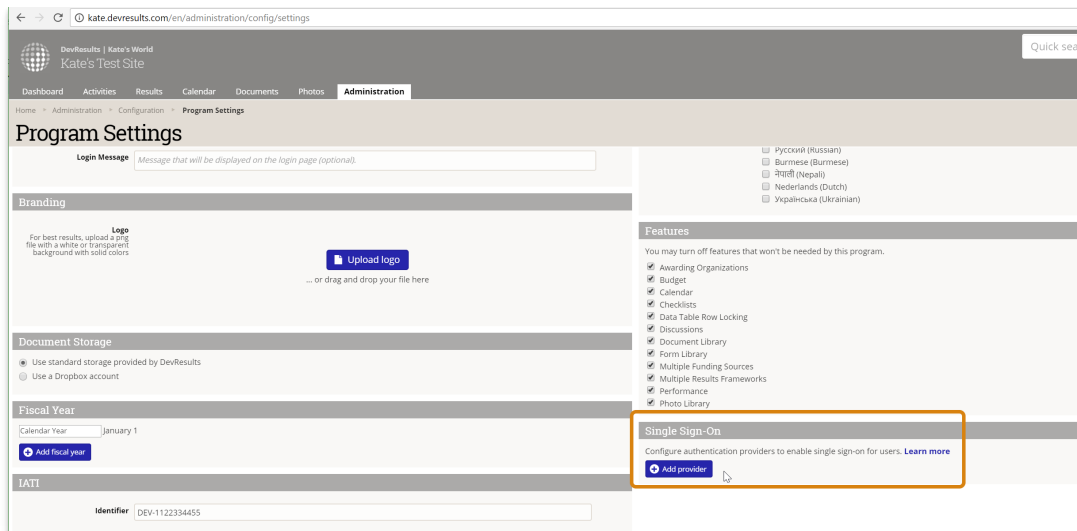
Active Directory via OAuth2

Configuring an Active Directory authentication via OAuth2 is fairly similar to the Azure Active Directory setup, except the configuration on the AD side must be done via Powershell script. [Here's](#) a good general set of instructions on how to do so. You'll need to specify the endpoint, a resource name, and the full Redirect URI.

Once these are set up, you can set up the DevResults portion. In your DevResults site, go to **Administration > Settings**.



In the **Single Sign-On** section, click the **Add provider** button.



This will open the Authentication Provider pop-up. You'll need to complete these sections to configure the integration:

- **Name:** Provide a name for this SSO. We used OAuth2 in our example. This label will appear to end-users on their login screen ("Use my {Name} account") so be sure it's something your users will understand!
- **Authentication Provider:** Currently DevResults supports Azure Active Directory and Active Directory via OAuth2. For this example, select Active Directory via OAuth2.
- **Default Group:** When new users are created in DevResults from Active Directory, what group should they be added to by default? For our example, we used our standard **Users** group, but you can choose any existing group in your DevResults site.
- **Notification Email:** If you want anyone to be notified when a new DevResults user is created from Active Directory, enter their email address here. (Optional)
- **Is Active?:** You can uncheck this box if you don't want this provider to be currently used; otherwise, check the box to make sure it's going to be used.
- **Provider Settings: Application ID:** Paste in the Application ID you configured in your PowerShell script
- **Provider Settings: Resource Name:** Use the Resource Name you configured in your PowerShell script
- **Provider Settings: Authorization Endpoint:** Use the Endpoint you configured in your PowerShell script
- **Token Endpoint:** Use the token endpoint you configured in your PowerShell script

The 'Authentication Provider' pop-up form is shown. It contains the following fields and options:

- Name:** OAuth2
- Authentication Provider:** Active Directory via OAuth2
- Default Group:** Users
- Notification Email:** ex. joe@example.com
- Is Active?:** ☒ Allow users to sign in with this provider
- Provider Settings:**
 - Application ID:** ID that will be used to identify DevResults with your provider
 - Resource Name:** ex. DevResults
 - Authorization Endpoint:** ex. https://example.com/oauth2/authorize
 - Token Endpoint:** ex. https://example.com/oauth2/token

At the bottom, there are 'Cancel' and 'Add' buttons.

Once these fields are filled out, click the **Add** button to save your configuration settings.

Authentication Provider

Name: OAuth2

Authentication Provider: Active Directory via OAuth2

Default Group: What group should new users created from this provider be assigned to?
Users

Notification Email: Who should be notified when a new user is created from this provider?
ex. joe@example.com

Is Active? ☒ Allow users to sign in with this provider

Provider Settings

Application ID: 11293875aodmu395

Resource Name: DevResults

Authorization Endpoint: https://kate.com/adfs/oauth2/authorize

Token Endpoint: https://kate.com/adfs/oauth2/token

Once it's added, you'll see the provider appear on the DevResults Administration / Settings page. You can click to edit or delete it.

☒ Data Table KOW Locking
☒ Discussions
☒ Document Library
☒ Form Library
☒ Multiple Funding Sources
☒ Multiple Results Frameworks
☒ Performance
☒ Photo Library

Single Sign-On

Configure authentication providers to enable single sign-on for users. [Learn more](#)

With the SSO set up and enabled, your DevResults Login page will look slightly different. It will now show a "Log in with my ___ account" option as well as the regular login. Users can either use their DevResults username and password (if they already have one) or their SSO.

Welcome! Please log in.

OR

Email:

Password:

☒ Stay logged in on this computer

Trouble logging in?

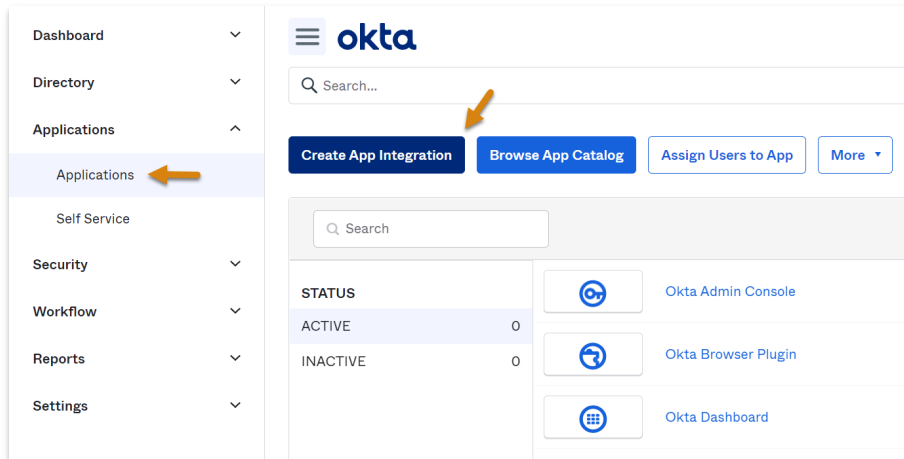
The first time they log in using the OAuth2 account, they'll be redirected to page based on your OAuth2 settings to enter credentials and grant DevResults access.

[back to top](#)

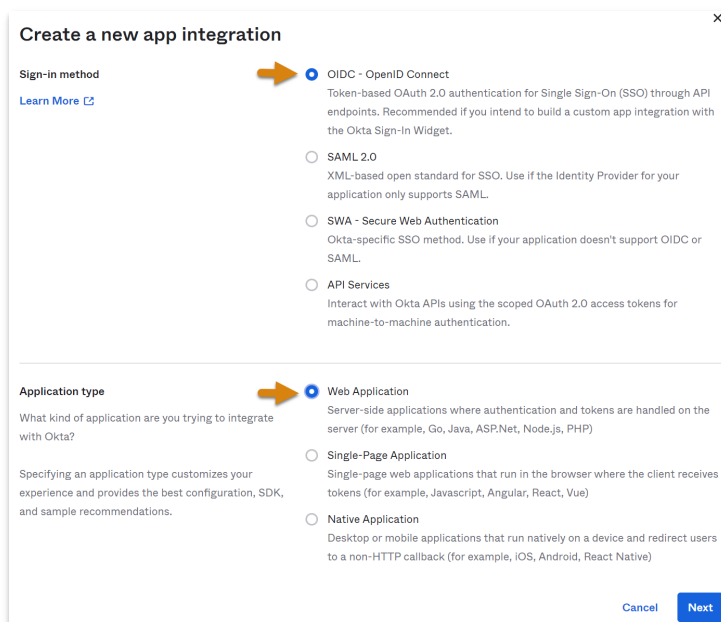
Okta Setup

DevResults can be integrated with your existing Okta environment in just a few easy steps. The first part of this process is also documented in [Okta's developer documentation](#), replicated here for your convenience.

1. Sign into your Okta Admin Console and go to **Applications->Applications** (from the left hand side menu) and click on **Create App Integration**.



2. Select either **ODIC - OpenID Connect** or **SAML 2.0** as the sign-in method and **Web Application** as the application type, then click **Next**.




3. If this is an ODIC connectionn, specify the **App integration name** (e.g. DevResults), upload the DevResults logo (optional), leave the **Grant type** set to **Authorization Code**, and click **Save**. Later, you'll need to revise the **Sign-in redirect URI**, but you'll have to finish the next step in DevResults before you'll be able to change these fields.

New Web App Integration

General Settings

App integration name → DevResults

Logo (Optional) → 

Grant type

[Learn More](#)

☐ Client acting on behalf of itself
☐ Client Credentials
→ ☒ Client acting on behalf of a user
☒ Authorization Code
☐ Refresh Token
☐ Implicit (hybrid)

Sign-in redirect URIs

Okta sends the authentication response and ID token for the user's sign-in request to these URIs

[+ Add URI](#)

[Learn More](#)

http://localhost:8080/authorization-code/callback

Sign-in redirect URIs

Note: if you have already established access groups, you can **limit access to select groups** at the bottom of this page. If not, you can **allow everyone in your organization to access** DevResults via Okta SSO.

Assignments

Controlled access

Select whether to assign the app integration to everyone in your org, only selected group(s), or skip assignment until after app creation.

☒ Allow everyone in your organization to access
☐ Limit access to selected groups
☐ Skip group assignment for now

[Save](#) [Cancel](#)

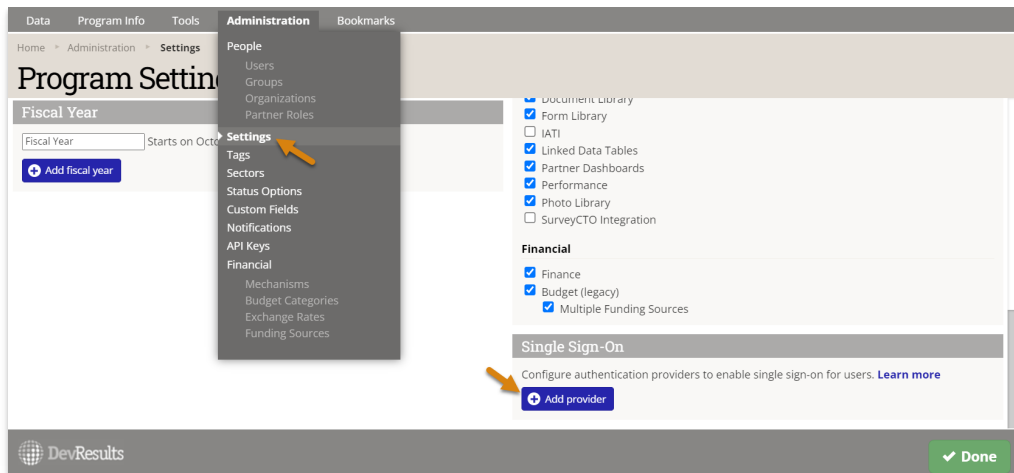
If you're connecting using SAML 2.0:

- Okta's **Single Sign-On URL** -> DevResults **Redirect URI**
- Okta's Audience URI -> https://{your instance subdomain}. devresults.com/saml-sp

Add an Attribute Statement with Name "DevResultsEmail" and Value "user.email" and a second Attribute Statement with Name "DevResultsFullName" and Value String.join(" ", user.firstName, user.lastName) as below. Be careful, these are case sensitive!

ATTRIBUTE STATEMENTS		
Name	Name Format	Value
DevResultsEmail	Unspecified	user.email
DevResultsFullName	Unspecified	String.join(" ", user.firstName, user.lastName)

4. In DevResults, go to **Administration > Settings**. In the **Single Sign-On** section, click the **Add provider** button.



5. In the following screen, select **Okta** as the **Authentication Provider** and fill out the other fields accordingly. If connecting using ODIC, you'll need to get the following fields from Okta (after saving your configuration in step 3 above) to fill in the **Provider Settings** section:

- Okta's **Client ID** -> DevResults **Application ID**
- Okta's **Client secret** -> DevResults **Secret**
- Okta **domain** -> **Authorization Endpoint**, in the format **https://{Okta domain goes here}/oauth2**, e.g. **https://okta-server/oauth2**.

If using **SAML 2.0**:

- Okta's **Single Sign-On URL** -> DevResults **Authorization Endpoint**
- Okta's **Entity ID** -> DevResults **Application ID**
- Okta's **X.509 Certificate** -> DevResults **Secret** (NOTE: include **-----BEGIN CERTIFICATE-----** and **-----END CERTIFICATE-----**)
- **Default RelayState** to any string value you want, just remember to add it to Okta's **Default Relay State**

Once every field is populated, click **Add**.

Authentication Provider

Name

Okta SSO

Authentication Provider

Okta

Default Group

What group should new users created from this provider be assigned to?

Viewers

Notification Email

Who should be notified when a new user is created from this provider?

owner@devresultsdemo.com

Is Active?

☒ Allow users to log in with this provider

Provider Settings

Application ID

ID that will be used to identify DevResults with your provider

Secret

Secret or private key for the authentication provider

Authorization Endpoint

ex. https://example.com/oauth2/authorize

Default RelayState

ex. RandomString12345

Cancel

Add

6. You should see a new authentication provider, with the name you provided in the previous step (e.g. Okta SSO). Click this button and copy the **Redirect URI** that has been created.

Data

Program Info

Tools

Admin

Home

Administration

Settings

Program Settings

For user results, upload a png file with a white or transparent background with solid colors

Document Storage

Storage Provider

DevResults

Change

Fiscal Year

Fiscal Year

Starts on October 1

Add fiscal year

Authentication Provider

Name

Okta SSO

Authentication Provider

Okta

Default Group

What group should new users created from this provider be assigned to?

Viewers

Notification Email

Who should be notified when a new user is created from this provider?

owner@devresultsdemo.com

Is Active?

☒ Allow users to log in with this provider

Provider Settings

Redirect Uri

https://[REDACTED]/api/oauth2/validate

Application ID

ID that will be used to identify DevResults with your provider

Secret

Secret or private key for the authentication provider

Authorization Endpoint

ex. https://example.com/oauth2/authorize

Cancel

Delete

Save

Single Sign-On

Configure authentication providers to enable single sign-on for users. [Learn more](#)

Okta SSO

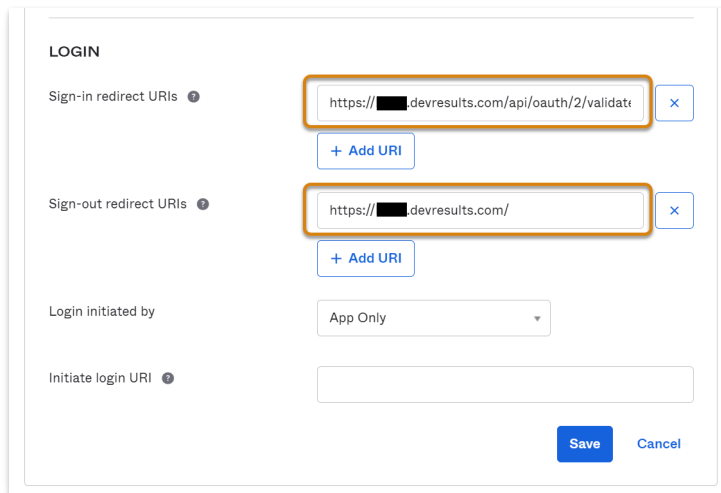
Disable password-based login for users with a single sign-on account?

Add provider

DevResults

Done

7. Back in Okta, click on **Edit** in the **General Settings** box and paste the Redirect URI from DevResults into the **Sign-in redirect URI** field. In the **Sign-out redirect URI** field, paste the same text but take off everything after devresults.com (in otherwords, just include the root domain).



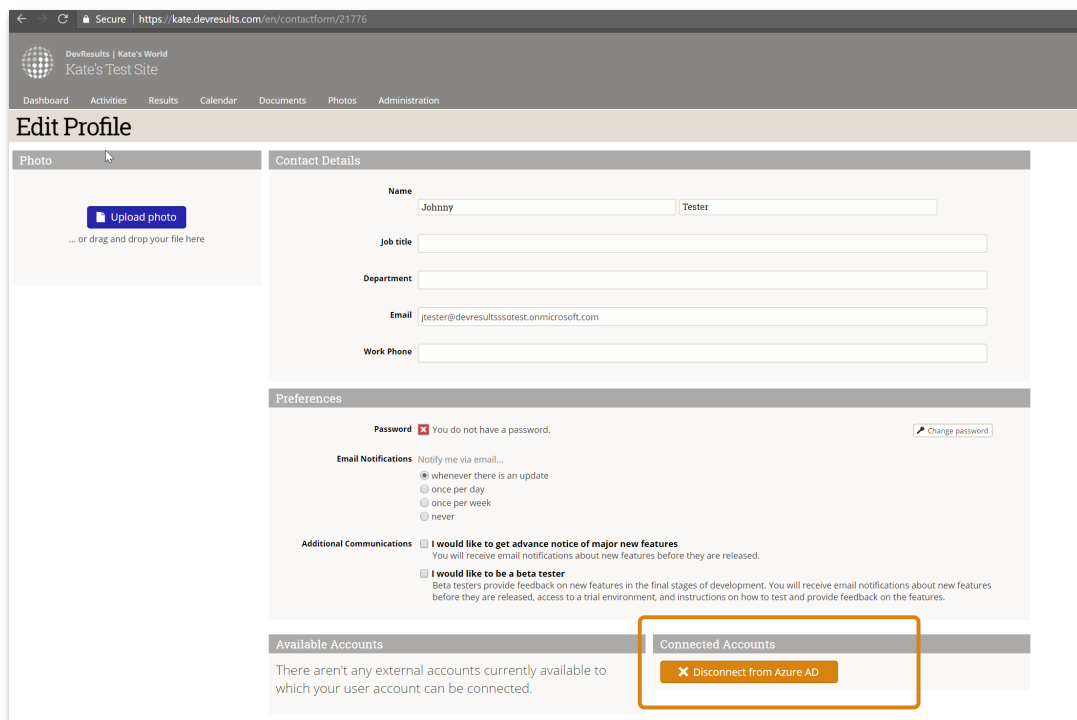
The image shows the 'General Settings' dialog box in Okta. It has four sections: 'Sign-in redirect URIs', 'Sign-out redirect URIs', 'Login initiated by', and 'Initiate login URI'. The 'Sign-in redirect URIs' field contains 'https://[redacted].devresults.com/api/oauth/2/validate' and has a '+ Add URI' button below it. The 'Sign-out redirect URIs' field contains 'https://[redacted].devresults.com/' and also has a '+ Add URI' button below it. The 'Login initiated by' dropdown is set to 'App Only'. The 'Initiate login URI' field is empty. At the bottom right are 'Save' and 'Cancel' buttons.

[back to top](#)

Frequently Asked Questions

How can I tell if I'm using a DevResults account or their Active Directory Account?

A user can tell if their DevResults account is related to an Active Directory account in their Profile details. Click on your profile picture in the upper right and select **Edit Profile**. The Connected Accounts section will have an entry if you're using Active Directory:



The image shows the 'Edit Profile' page in the DevResults application. The page has a header with the DevResults logo and navigation links. The main content area is divided into two columns. The left column has a 'Photo' section with an 'Upload photo' button. The right column has 'Contact Details' and 'Preferences' sections. The 'Contact Details' section has fields for Name, Job title, Department, Email, and Work Phone. The 'Preferences' section has a 'Password' field, 'Email Notifications' (with radio buttons for 'whenever there is an update', 'once per day', 'once per week', and 'never'), and 'Additional Communications' (with checkboxes for 'I would like to get advance notice of major new features' and 'I would like to be a beta tester'). At the bottom, there are two sections: 'Available Accounts' (which says 'There aren't any external accounts currently available to which your user account can be connected.') and 'Connected Accounts' (which has a button labeled 'X Disconnect from Azure AD').

You can disconnect this relationship by clicking the **Disconnect from...** button here.

If you're a site administrator and you'd like to be able to see a list of users and whether they're using a connected SSO account or not, let us know--we'll be happy to **create a custom query** that meets your needs.

I already had a DevResults user account before we added SSO. Can I link those accounts somehow?

Individual users can link their DevResults account to an Active Directory account, provided the email addressees are the same.

Click on your profile picture in the upper right and select **Edit Profile**.

If you don't currently have an Active Directory account linked to your DevResults account, you'll have nothing in the **Connected Accounts** section and the **Available Accounts** section will have a **Connect to {Name}** button.

The screenshot shows the 'Edit Profile' page for a user named Johnny Tester. The page is divided into several sections:

- Photo:** Includes an 'Upload photo' button and a note to drag and drop a file.
- Contact Details:** Fields for Name (Johnny Tester), Job title, Department, Email (tester@devresultssstest.onmicrosoft.com), and Work Phone.
- Preferences:** Includes a Password field (with a note 'You do not have a password.' and a 'Change password' link), Email Notifications (with radio buttons for 'whenever there is an update', 'once per day', 'once per week', and 'never'), and Additional Communications (with checkboxes for 'I would like to get advance notice of major new features' and 'I would like to be a beta tester').
- Available Accounts:** A section with a blue button labeled 'Connect to Azure AD'. An orange arrow points to this button.
- Connected Accounts:** A section with the text 'There are no accounts connected to your user account at this time.'

Clicking that button will take you to the Microsoft login screen where you can enter their Active Directory credentials and grant DevResults permission to use them. Once you've done this, moving forward you'll click the **Use my {SSO} account** when you go to log into DevResults.

The screenshot shows the 'Register an application' page in the Microsoft Azure portal. The page is divided into several sections:

- Name:** A text box containing 'DevResults'.
- Supported account types:** A section with the heading 'Who can use this application or access this API?'. It includes three radio buttons: 'Accounts in this organizational directory only (DevResults)' (selected), 'Accounts in any organizational directory', and 'Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)'. A link 'Help me choose...' is also present.
- Redirect URI (optional):** A section with the heading '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.' It includes a dropdown menu set to 'Web' and a text box containing 'https://testaccount@devresults.com'.
- Footer:** A link to 'By proceeding, you agree to the Microsoft Platform Policies' and a 'Register' button.

How do I configure my SSO on the training site after it has been refreshed with data from my live site?

There are a few different options for ensuring that the training site remains properly configured with your SSO after a training site refresh:

- In most cases, you only need to provide your SSO application with two "Sign-in redirect URIs," one that has <your-site>.devresults.com as the host and one that has <your-site>.training-devresults.com as the host. The protocol and path should be the same for both.
- If you'd like to have different groups of people have access to training and live sites, use a separate SSO provider/application, or just have different settings between training and production, the solution is a bit more complicated. One solution would be to use the DevResults API to re-create the login settings on the training site after the refresh completes. The general idea is as follows:
 - Create an API Key with Owner permissions on the production site (this way it is always present in training after the refresh)
 - After the training refresh, use the API key to delete the live site login and re-create the training site login
 - On the SSO provider side, you can either use a wildcard in the "Sign-in redirect URI" to handle the fact that the number path may change over time, or you can update the SSO app with the correct URI after creating the settings on the DevResults side.

[back to top](#)

Didn't answer your question? Please email us at help@devresults.com.

Related Articles
