Integrating Single Sign-On

Last Modified on 02/27/2018 5:01 pm EST

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 and with Active Directory via OAuth2. This page will walk you through how DevResults-SSO integration works and provide instructions on how to set it up.

Jump to:

User Administration

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

The two accounts are linked but are independent, so it is possible to have Active Directory configured for your site for your own organization's users, but to still have local partner or field staff use DevResults-only accounts to log in.

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 Active Directory, it will prevent them from logging in. We recommend still having a designated DevResults site administrator oversee permissions and account deactivation within DevResults.

SSO Integration Setup

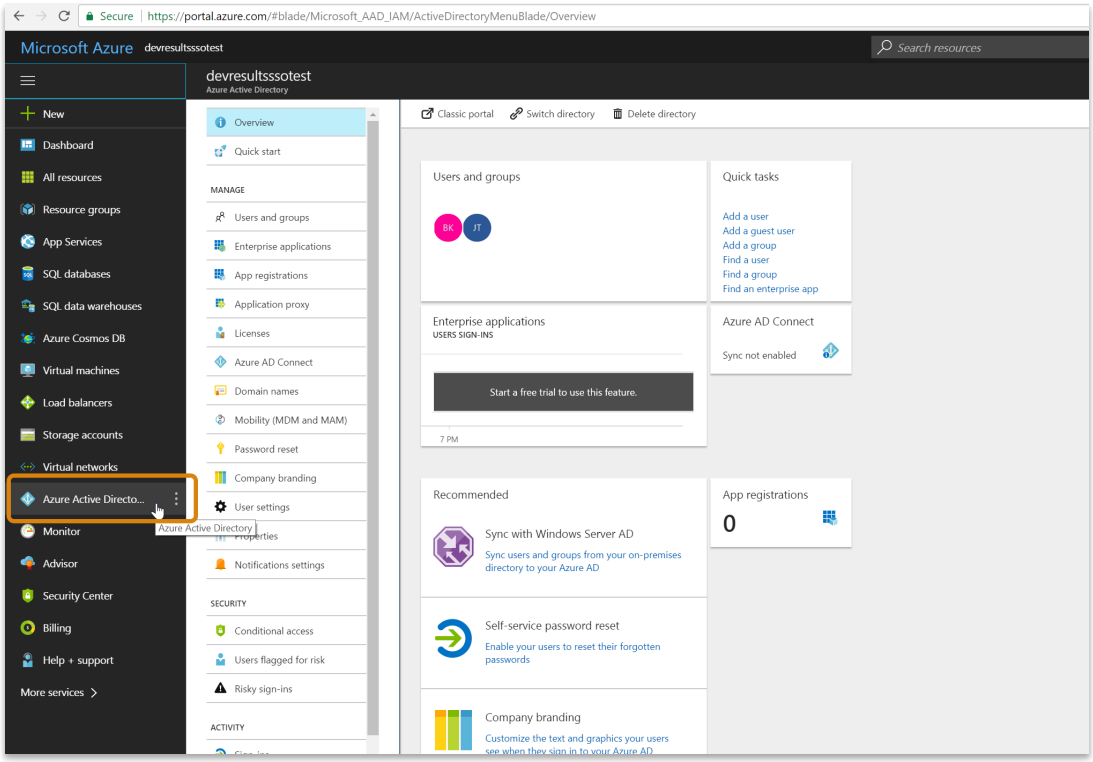
We support SSO integration with Azure Active Directory and with Active Directory via OAuth2. For both setups, the overall process is:

- 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.

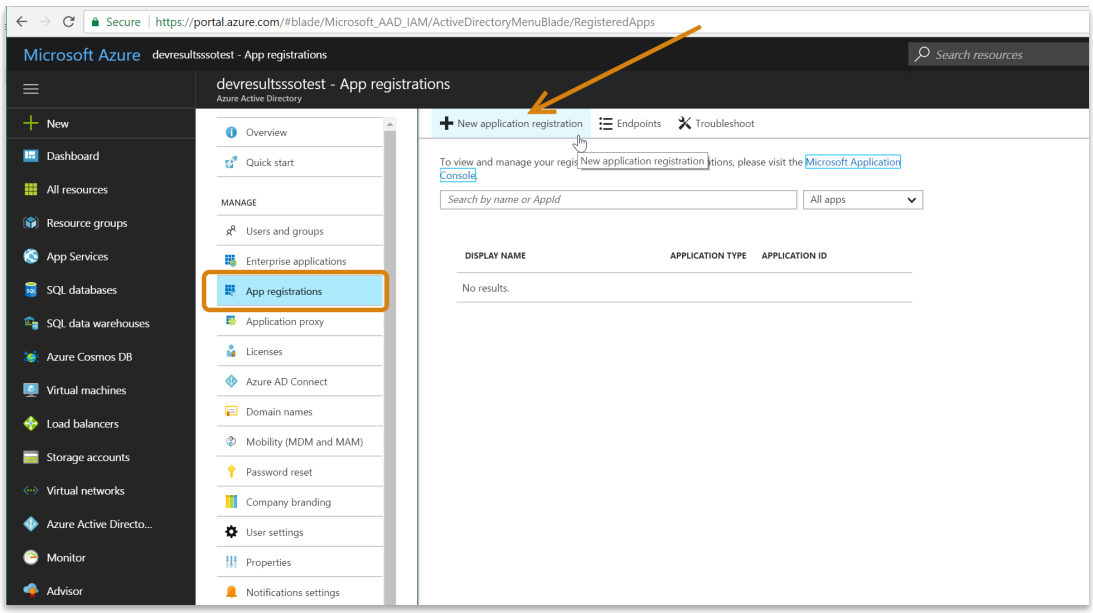
Azure Active Directory

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.

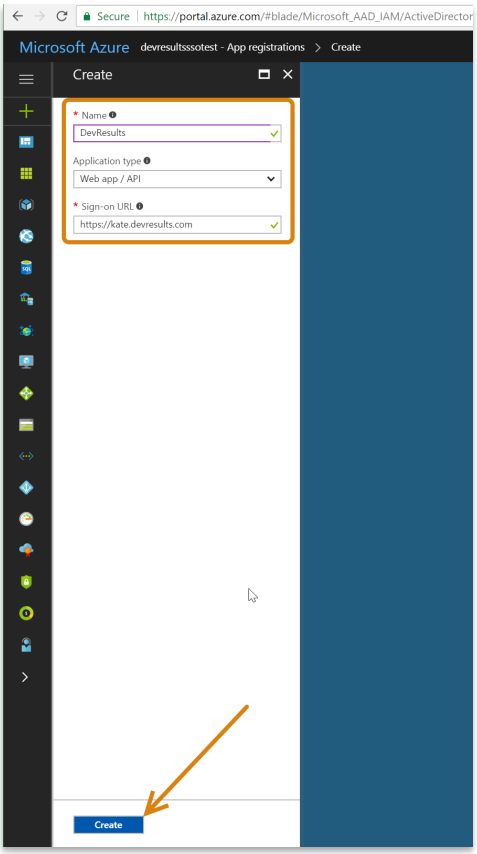
In the Azure Portal, go to Azure Active Directory.



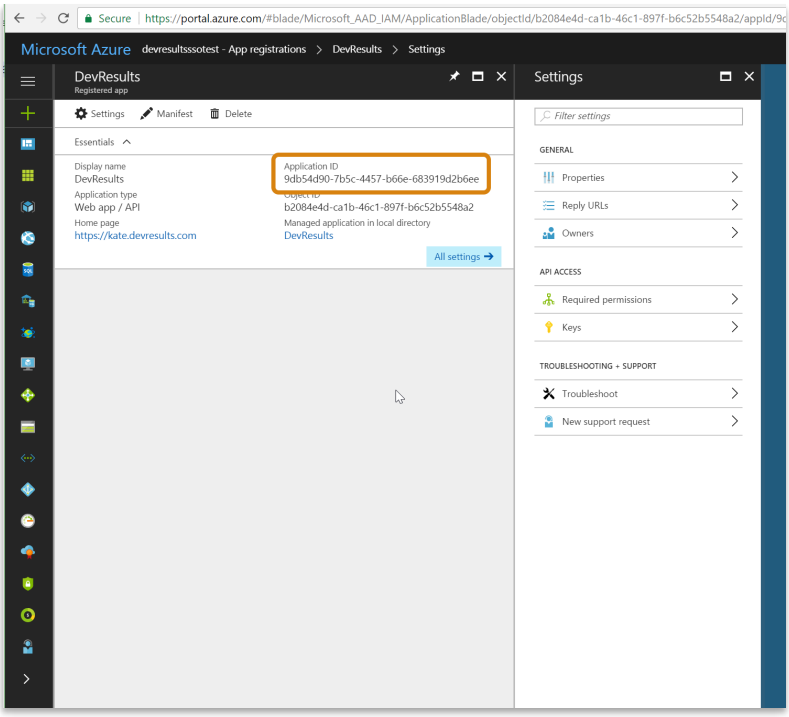
Select **App registrations**. Click on **New application registration** in the right-hand pane:



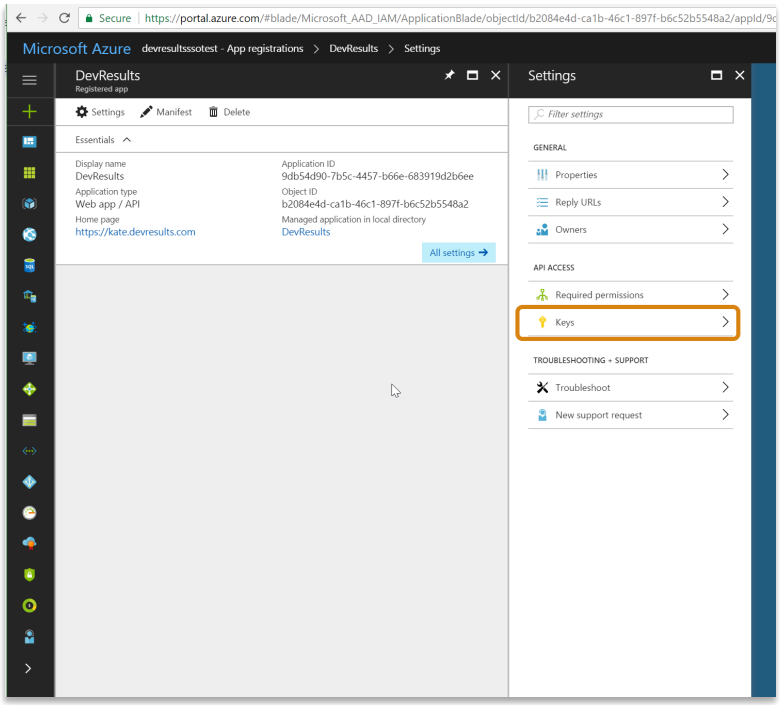
Enter a **Name** that makes sense, select **Web app / API** as the **Application type**, and use your DevResults site's url as the **Sign-on URL** (here we're using my site: kate.devresults.com). Click **Create**.



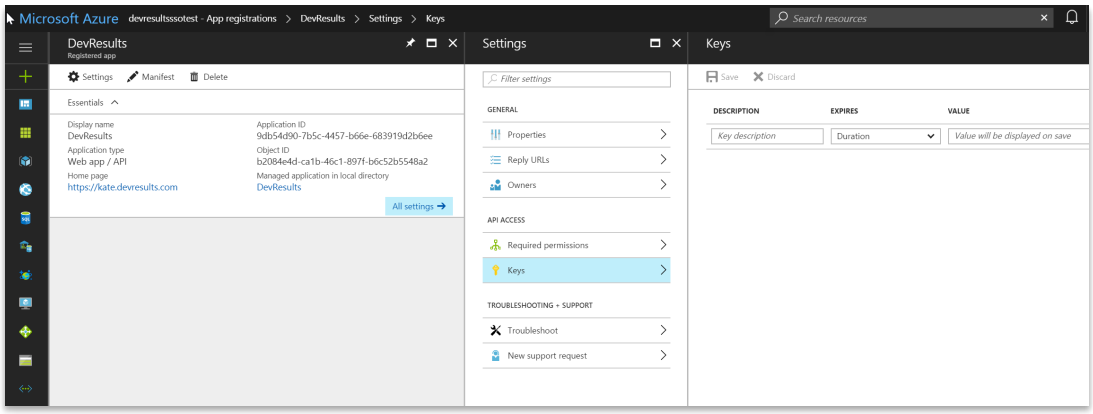
Once the application is created, click on it to open details. Copy the **Application ID**--you'll need this for the DevResults side of configuration.



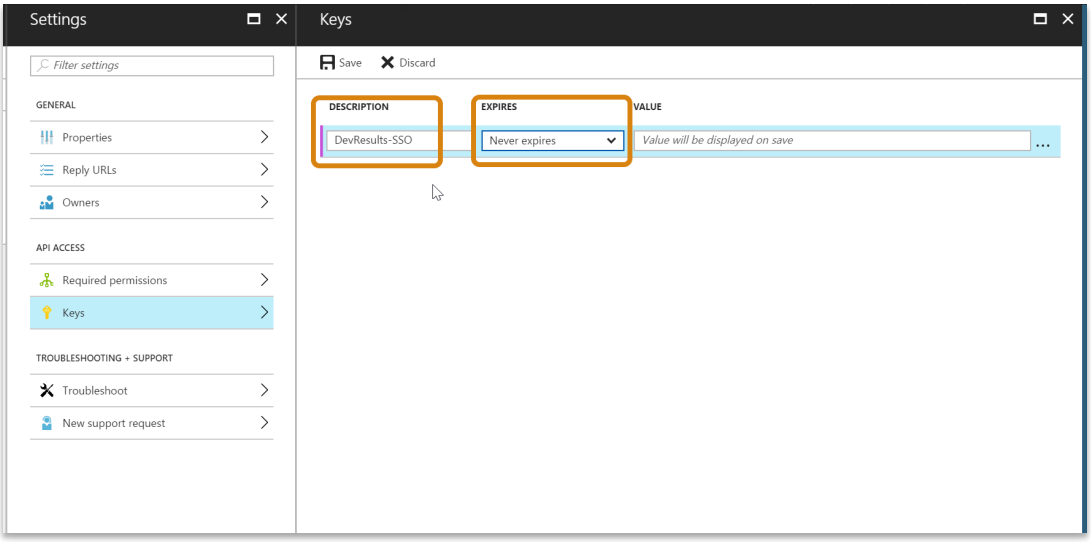
Under **Settings**, look for the **API Access** section and click on **Keys**.



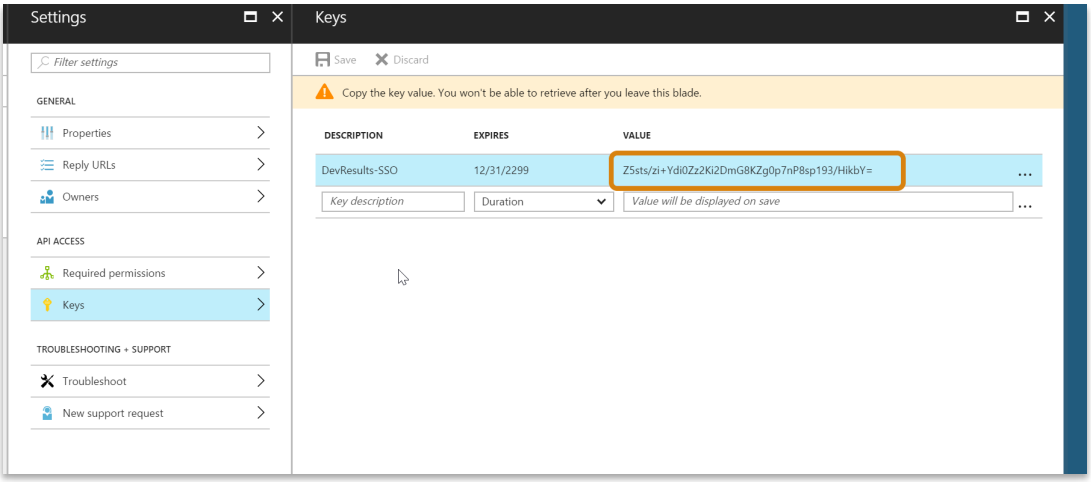
This should open a screen with an empty key:



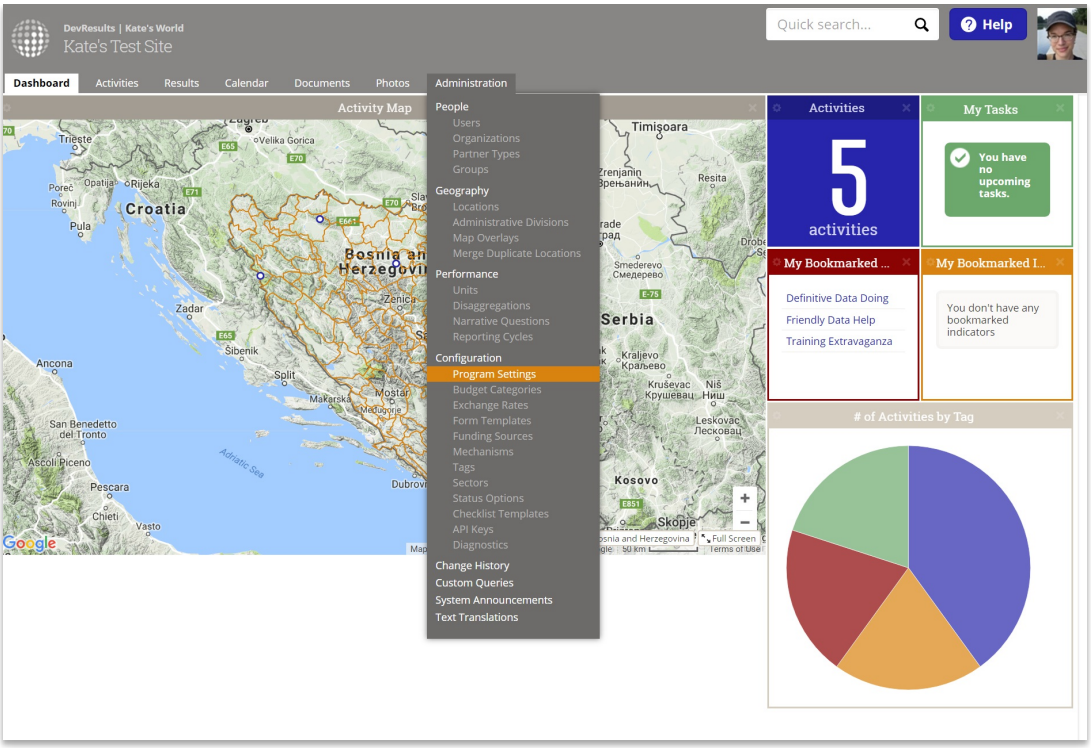
Enter DevResults-SSO (or something else useful) in the **Description**. Select a **Duration** consistent with your organization's security policies.



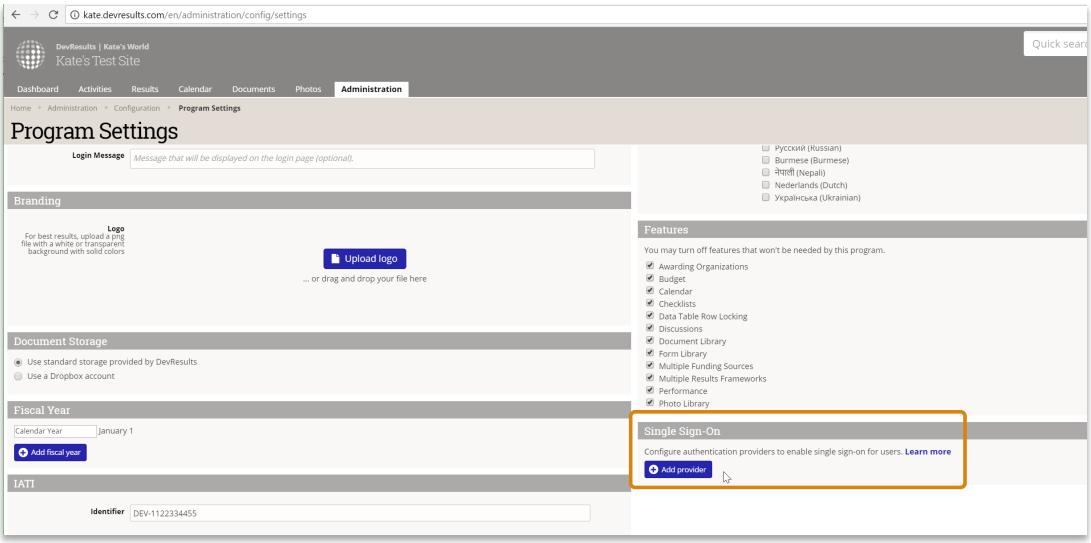
When you click **Save**, Azure will generate a Key in the Value. Copy that **Value**.



Now that there's an application in Azure Active Directory, we can set up the DevResults portion. In your DevResults site, go to **Administration > Program Settings**.



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

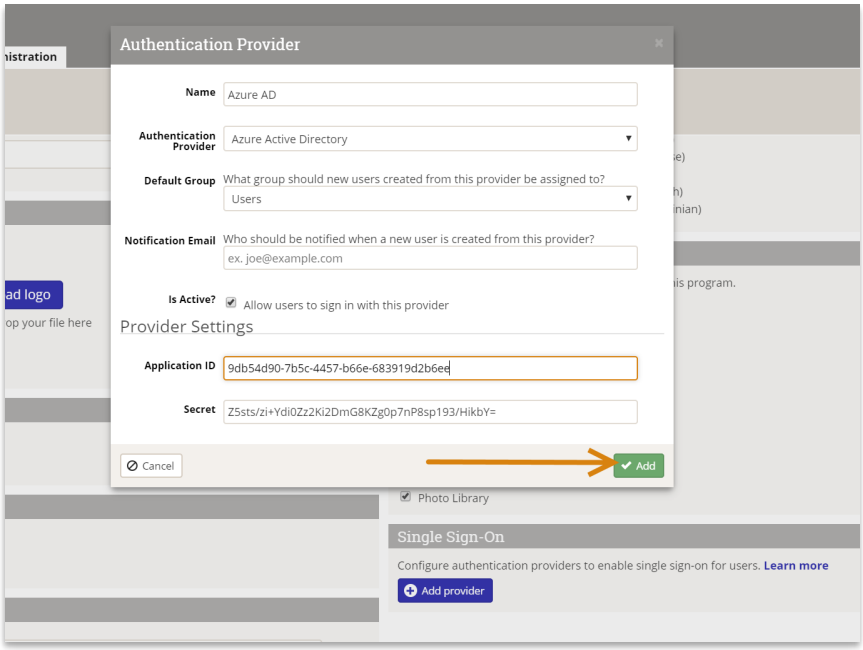


This will open the Authentication Provider pop-up. You'll need to complete these sections to configure the integration:

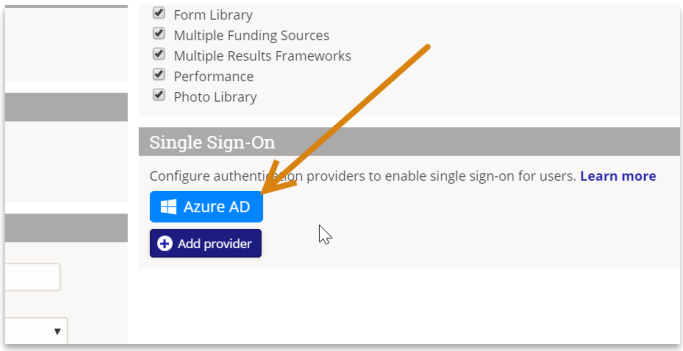
- **Name**: Provide a name for this SSO. We used Azure AD in our example. This label will appear to end-users on their login screen ("Use my {Name} account") so be sure it's something your users will understand!
- **Authentication Provider**: Currently DevResults supports Azure Active Directory and Active 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.
- **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

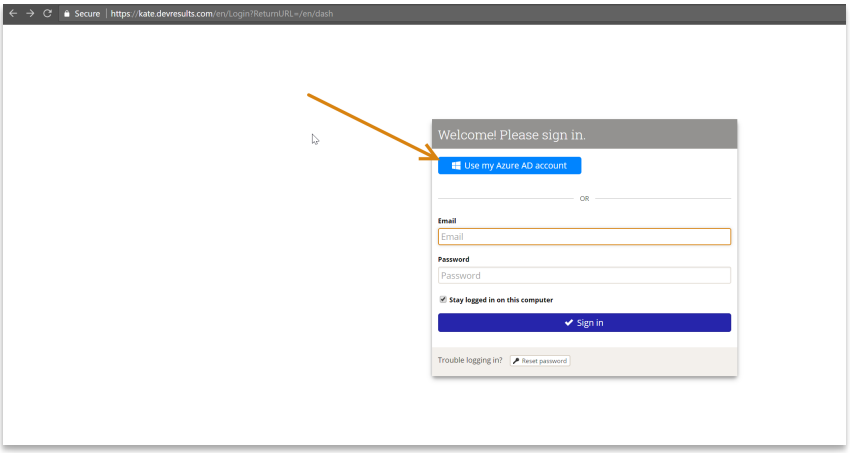
Once these fields are filled out, click the **Add** button to save your configuration settings.



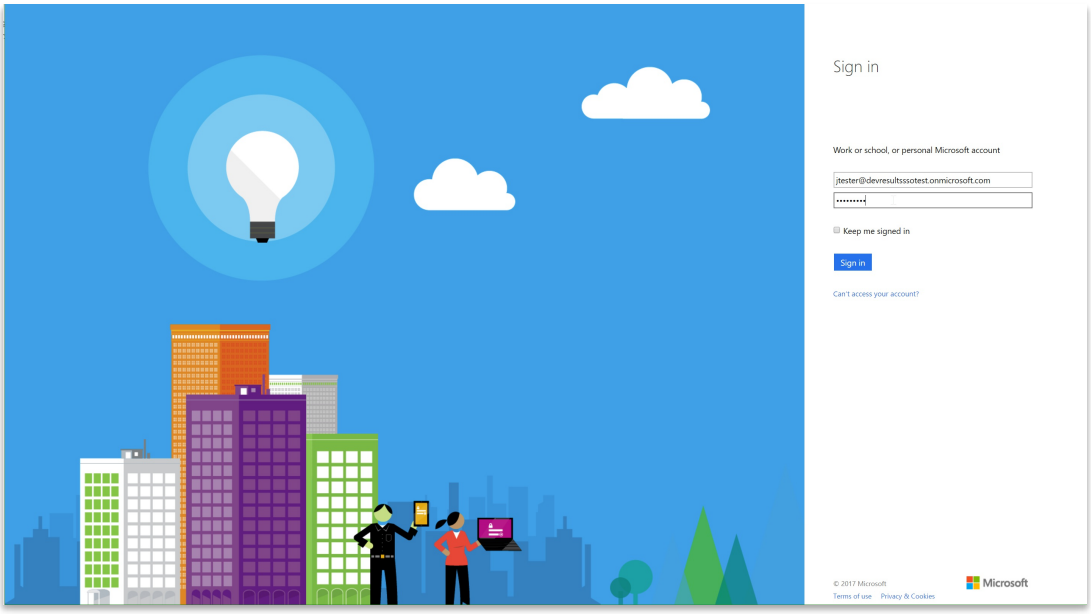
Once it's added, you'll see the provider appear on Program Settings. You can click to edit or delete it.



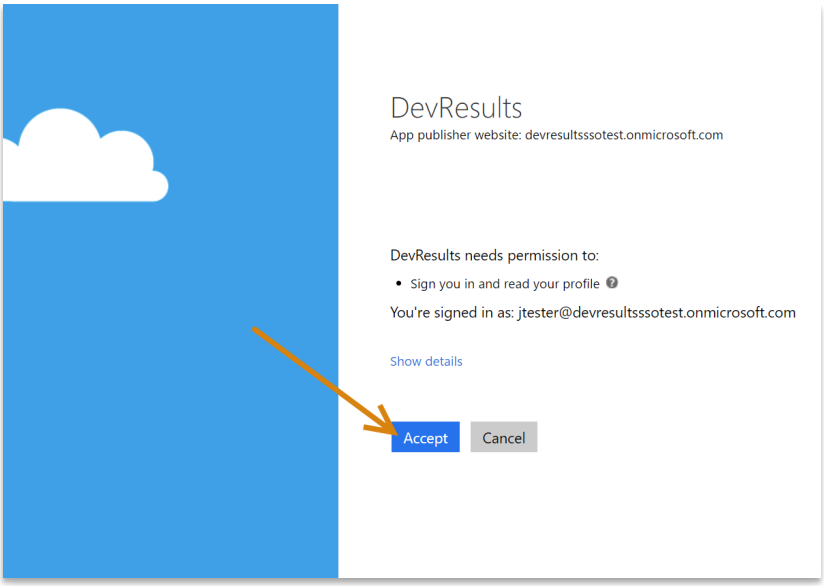
With the SSO set up and enabled, your DevResults Login page will look slightly different. It will now show a "Use my {Name} account" option as well as the regular login. Users can either use their DevResults username and password (if they already have one) or their SSO.



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.



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.

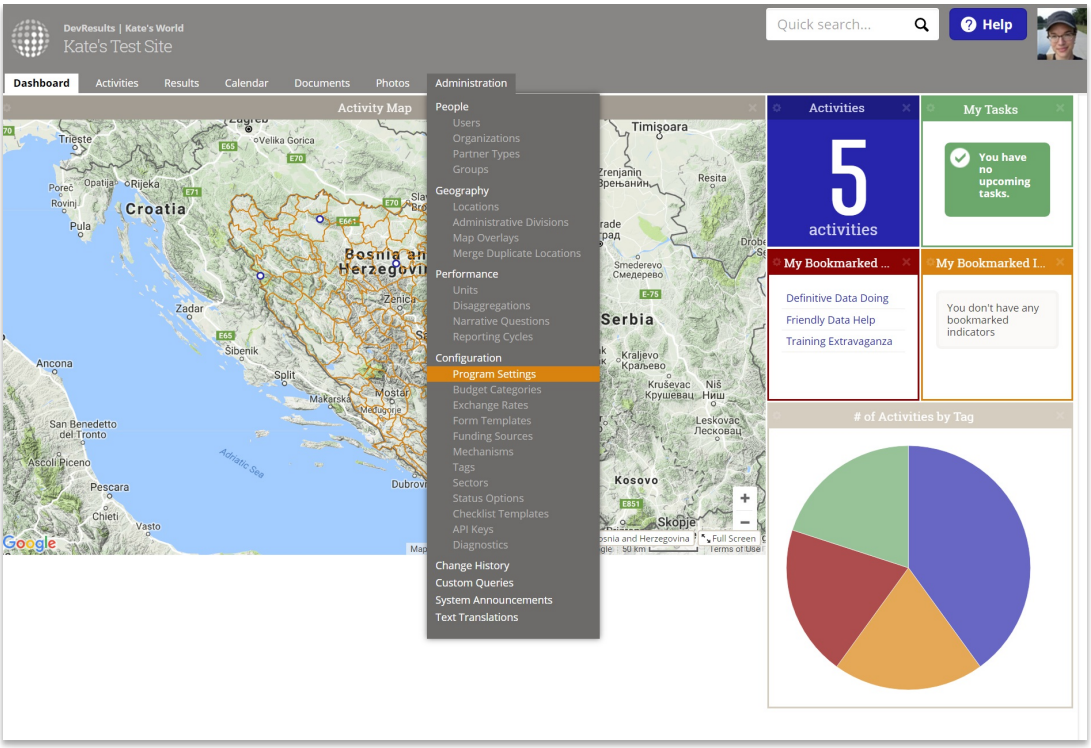


Once that's done, all future sign-ins using the Use my Azure AD account should just work. The page should now redirect to the DevResults site.

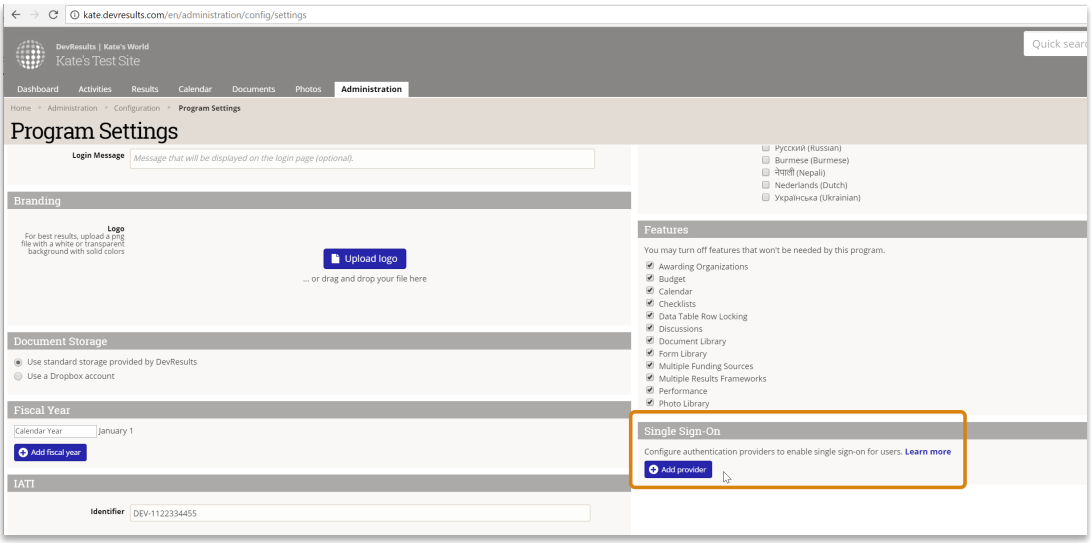
Active Directory via OAuth2

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 > Program Settings**.



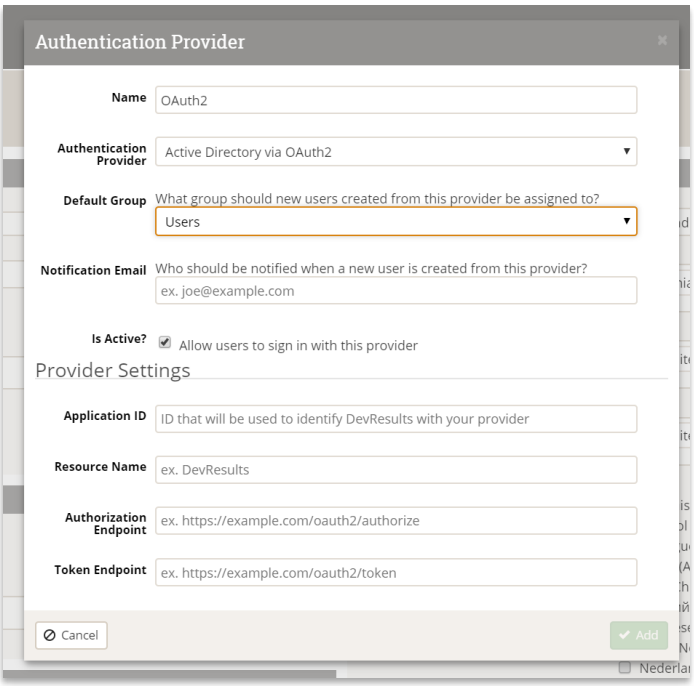
In the **Single Sign-On** section, click the **Add provider** button.



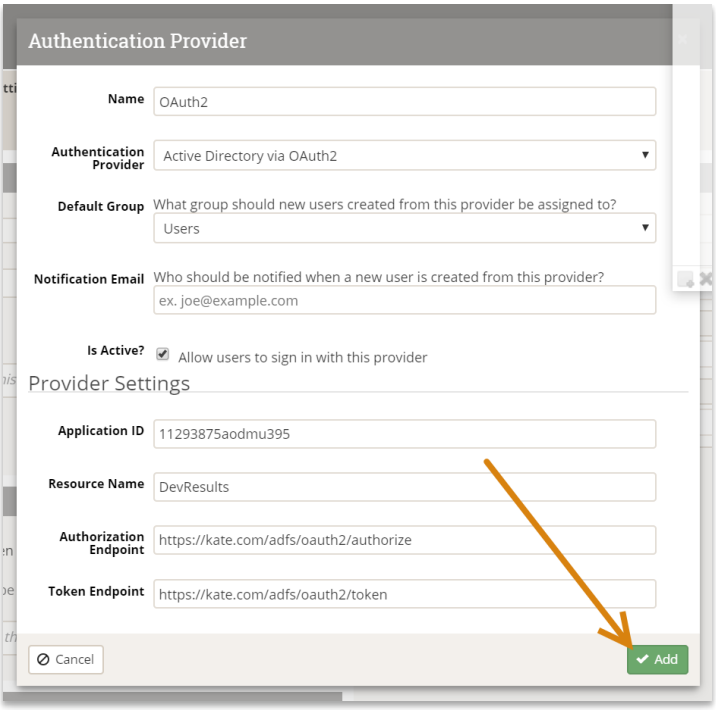
This will open the Authentication Provider pop-up. You'll need to complete these sections to configure the integration:

- **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.

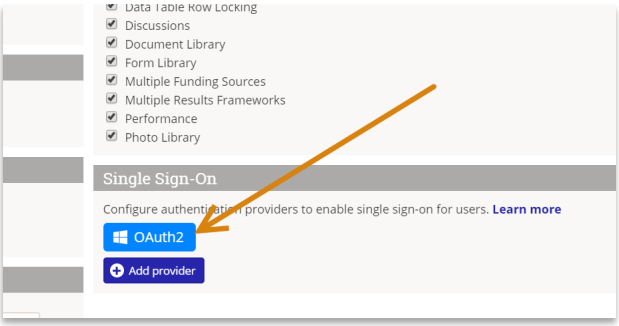
- **Notification Email**: If you want anyone to be notified when a new DevResults user is created from Active Directory, enter their email address here. (Optional)
- **Is Active?**: You can uncheck this box if you don't want this provider to be currently used; otherwise, check the box to make sure it's 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



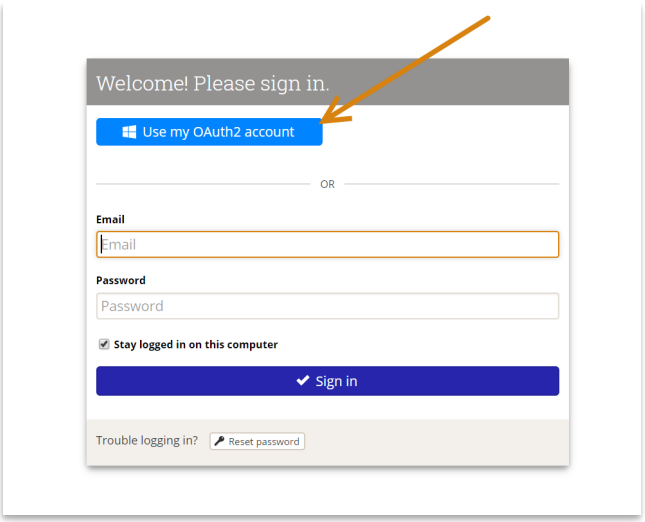
Once these fields are filled out, click the **Add** button to save your configuration settings.



Once it's added, you'll see the provider appear on Program Settings. 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 "Use my {Name} 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.

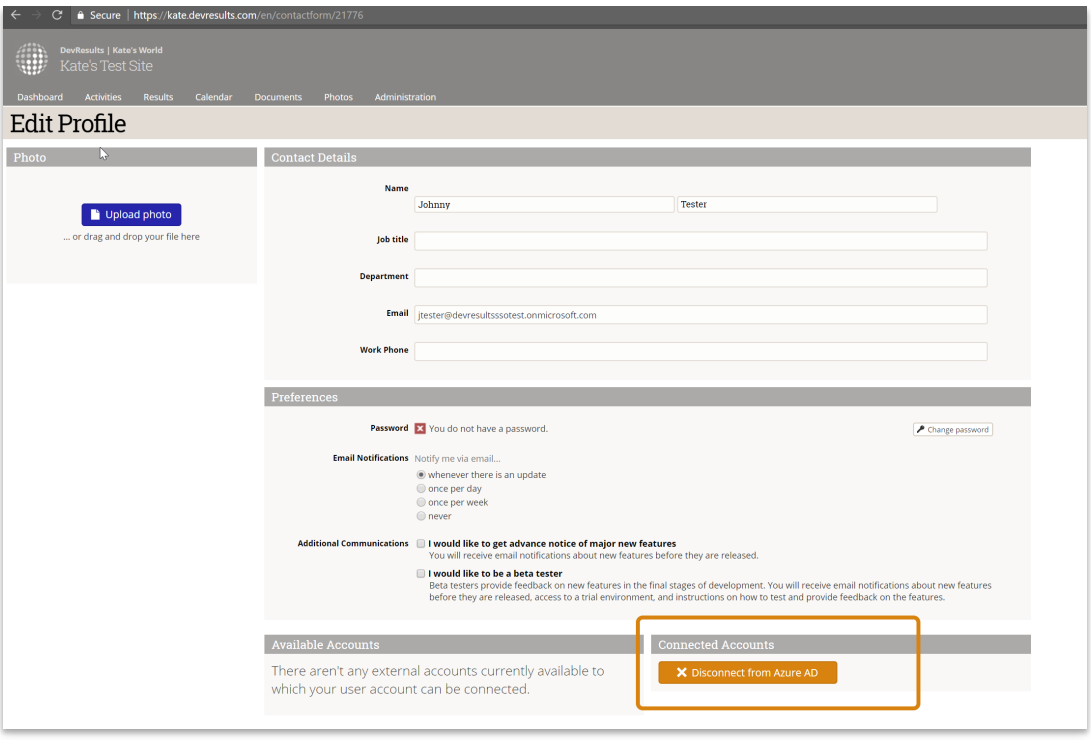


The first time they log in using the OAuth2 account, they'll be redirected to page based on your OAuth2 settings to enter credentials and grant DevResults access.

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:



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

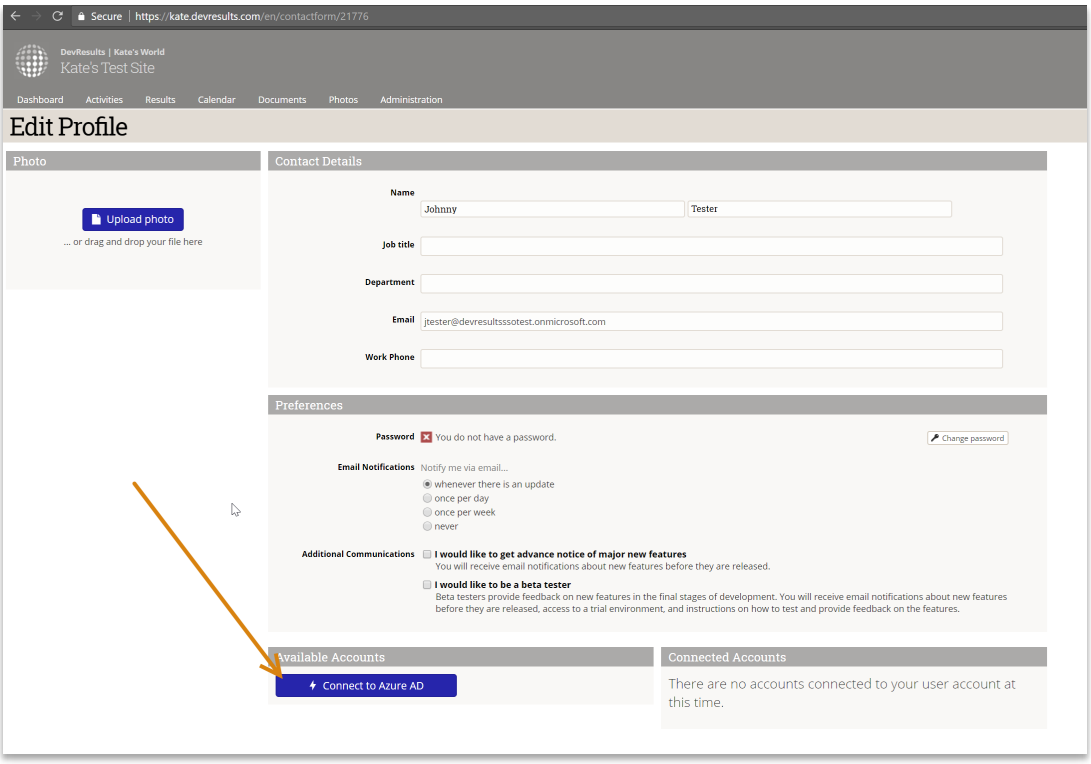
If you're a site administrator and you'd like to be able to see a list of users and whether they're using a connected SSO account or not, let us know--we'll be happy to **create a custom query** that meets your needs.

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.



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.

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

Related Articles