How to setup OAuth2 Authentication with Microsoft

Global Support
08 2016

Prerequisites

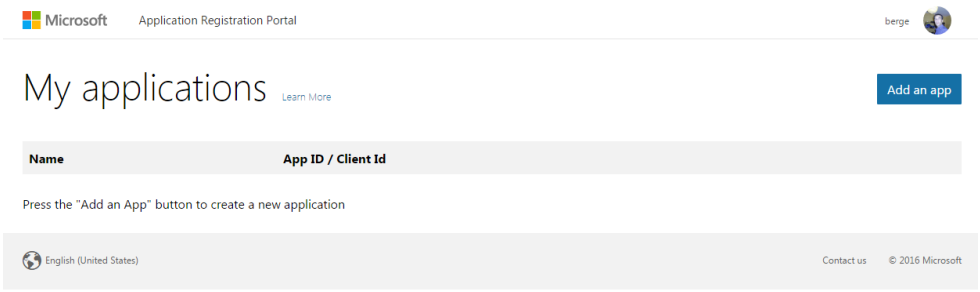
You will require the following items to set up OAuth2 with a Microsoft account:

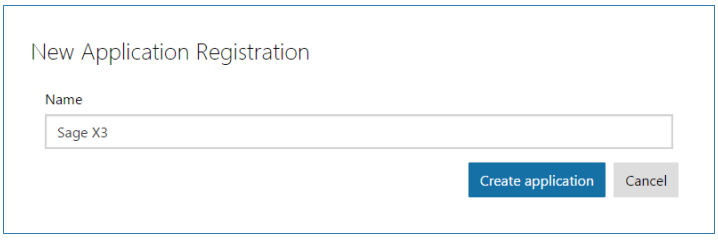
- Your server URL – public or private – is required. Examples https://www.my_server.com or https://MyServerName.
- A Microsoft account that will be used to administer the service. This can be done through following link: **https://apps.dev.microsoft.com**. In the following example we will use the mock account *my_admin_account@live.com*.
- Select a name for your OAuth2 service. It must start with a letter (A-Z or a-z) followed by any combination of letters (A-Z or a-z), digits or underscores. In the following example, the name *Microsoft* is used.
- **Oauth2** must be configured as a valid authentication method in your Sage X3 **nodelocal.js** file such as below:

```
session: {
    timeout: 30, // minutes
    checkInterval: 60, // seconds
    auth: ["basic", "oauth2"]
}
```

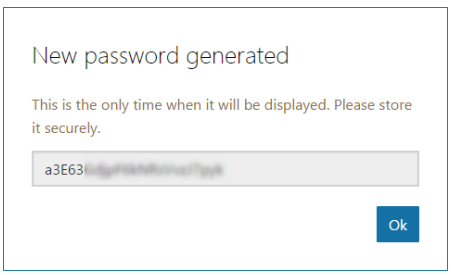
Step 1: Create a client ID

- Add an application to your Microsoft account: https://apps.dev.microsoft.com

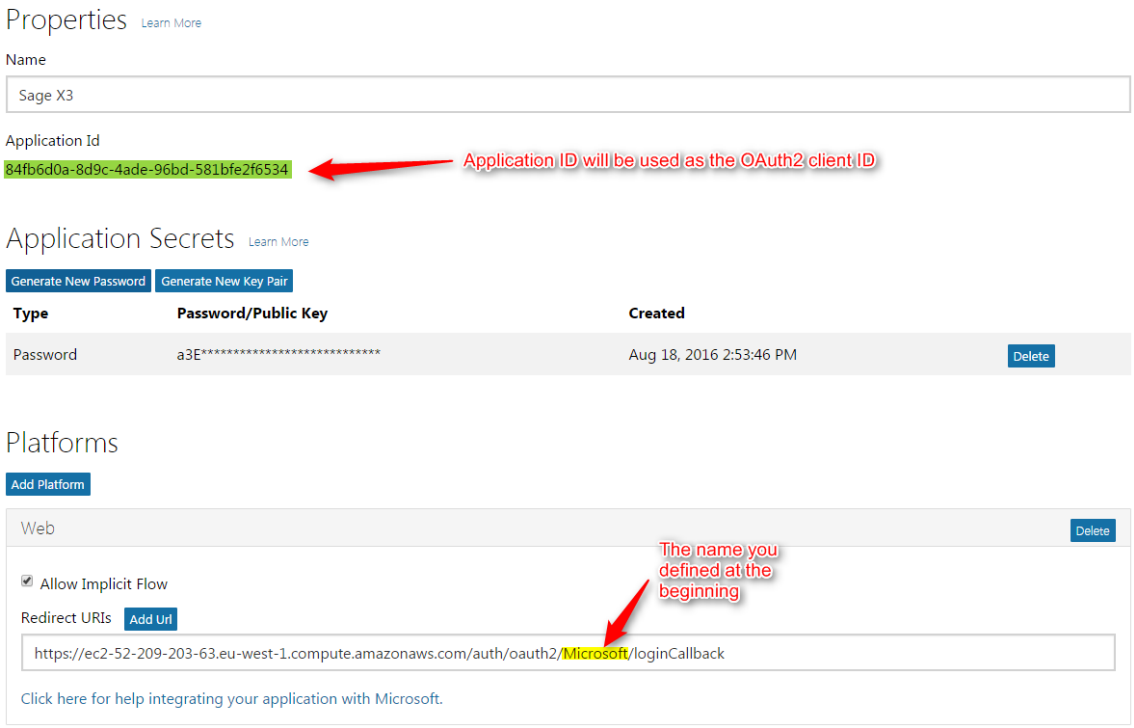




- Click **Generate a Password**, under **Application Secrets**. Copy the password to a safe place, such as a password manager (e.g. KeePass), for reference later in the process:



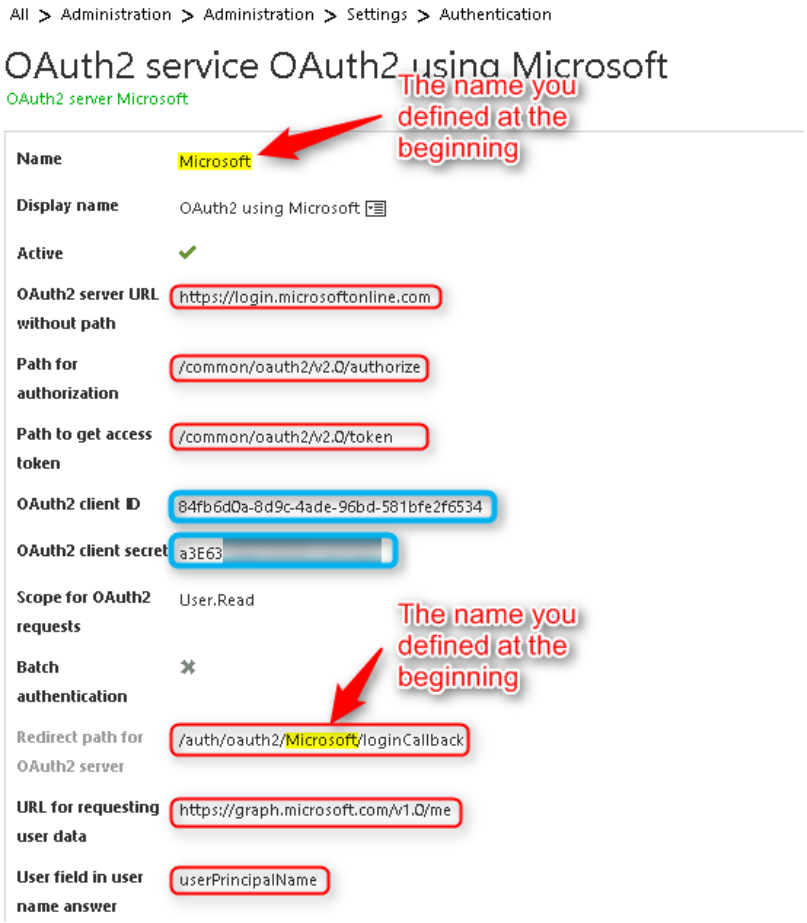
- Click **Add a platform** in the **Platforms** section. Choose **Web**.
- Enter the redirection Url: **http(s)://X3host:X3port/auth/oauth2/Microsoft/loginCallback** where *X3host:X3port* are the host and port of your Syracuse server.
- There is no need to specify the port if you are using port 80 with http (Default http port) or port 443 with https (Default https port).



- Save the application.

Step 2: Create an Oauth2 service in the Sage X3 Web Server (Product Update 9)

- Go to Administration > Settings > Authentication > OAuth2 – Create a new OAuth2 service:



- The name of the service must match the service name chosen earlier (Prerequisites) exactly. The display name can be chosen freely.
- The frames highlighted in red above hold constant values for Oauth2 **Microsoft** authentication. They must be populated exactly as shown in the picture above for any Microsoft service:
  - **Oauth2 server URL without path**: https://login.microsoftonline.com
  - **Path for authorization**: /common/oauth2/V2.0/authorize
  - **Path to get access token**: /common/oauth2/V2.0/token
  - **Scope for Oauth2 requests**: User.Read
  - **URL for requesting user data**: https://graph.microsoft.com/v1.0/me
  - **User field in user name answer**: userPrincipalName
- **Redirect path for Oauth2 server** depends on the Sage X3 (Product Update 9) Web Server and the service name selected earlier (c.f. Prerequisites). If this is editable, it must host the Web Server address followed by the **/auth/oauth2/NAME/loginCallback** segment (where **NAME** is the service name.)
- The information in blue (**Oauth2 client ID** and **Oauth2 client secret**) corresponds to the client ID shown and to the password defined in the **Client ID for web application** section on the Microsoft service site, on the page that shows the client ID.

Important security note:

You can set the Path for authorization and the Path to get access tokens depending on the types of accounts you would like to authorize:

/common/oauth2/V2.0…: All Microsoft accounts are allowed, personal or organizational
/consumers/oauth2/V2.0…: Non organizational account are allowed
/organizations/oauth2/V2.0…: Only organization accounts are allowed
/<tenant-id>/oauth2/V2.0…: Only accounts from the specified tenant-id are allowed

Best security practice for user management and traceability is to avoid the use of personal and non-organizational accounts where possible.

To obtain a tenant-id, see the following link: **http://stackoverflow.com/questions/26384034/how-to-get-the-azure-account-tenant-id**. You can try using **/common/** initially and then restrict later using **/tenant-id/** if you cannot access the tenant-id upfront (You will need Azure administration privileges to get your tenant-id.)

Step 3: Link your users to their Microsoft account

At this stage you are ready to enable oAuth2 authentication for all users. Follow the steps below to link users to their Microsoft accounts:

- Ensure that both **oauth2** and **basic** are enabled in your nodelocal.js file.
- Edit the Global settings. Change the default authentication method to **oauth2**.
- Edit the admin user. Set his/her authentication method to **DB** (this is a safety net in case your auth2 configuration does not work - you will change it later).
- Edit a test user (other than admin). Set their email to a Microsoft account for which you have valid credentials, your personal account for example.
- Log out and log back in with the test user. If you get an error, log in again as admin to fix the oauth2 configuration and try again.
- When the test is successful, log in again as admin, assign a Microsoft account email to the admin user and change the admin user to use the default authentication method (oauth2).
- Check all user emails and edit them if necessary to match the user's Microsoft account.
- Edit your nodelocal.js and only enable **oauth2**. Restart the Web server. Your server is now safely configured to authenticate all users, including Admin, with their Microsoft accounts.

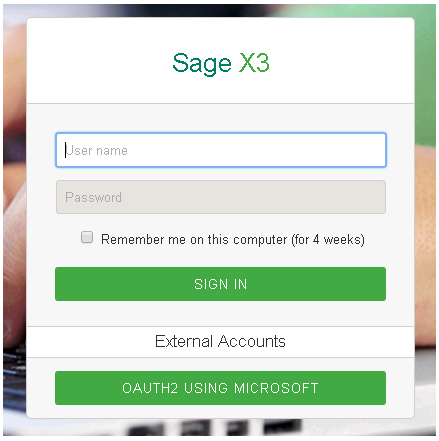
Note:

We recommend you use an external identity service such as LDAP or oAuth2 *for all users*, including special users that support web service calls. Nevertheless, you will need to adapt your web service clients to authenticate with oAuth2

If you are using web services published by Sage X3, you can temporarily activate both **basic** and **oauth2** in your nodelocal.js file and configure the special web service users to use **basic** authentication. This will allow you to keep your web services in operation while you adapt them for oauth2. Once you have upgraded your web service clients you should edit nodelocal.js again and only enable **oauth2** to tighten security.

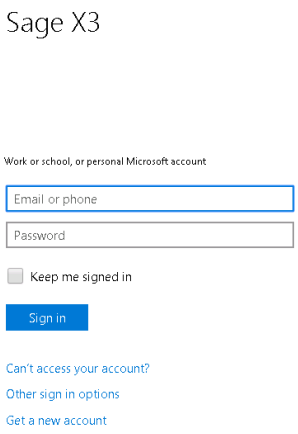
User connection

The login screen contains buttons for different authentication methods. The user must click on the button that shows the name of the service (as defined in the prerequisites above) in capital letters. There may be several buttons below the **External Accounts** headline, similar to the example below:

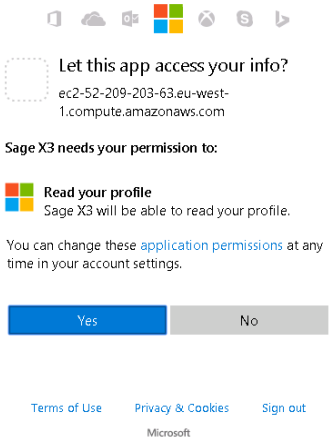


A direct link can also be typed and/or set in your browser favorites to access the Microsoft authentication directly:
http://www.my_server.com/auth/oauth2/Microsoft/loginStart

When this is done, you are redirected to the Microsoft site to authenticate using your Microsoft account if this has not already been done. The page appears as follows:



You will be prompted by Microsoft to allow Sage X3 to access the profile when logging in for the first time:



Note: Your authentication will be valid until you log out of your Microsoft account or clear your browser's cookies As a result Sage X3 may not need to prompt you to authenticate if your Microsoft login is still valid.