Create a new SMTP service: Microsoft OAuth2

Last Modified on 02/07/2024 3:30 pm EST

Many email providers are moving away from basic authentication in favor of an OAuth2 implementation. Where basic auth uses a username/password combination, OAuth2 has a more involved process involving a client ID and a client secret, plus an additional authorization step that generates a token that gets used behind the scenes. It's considered a more secure form of authentication.

Because OAuth2 is more involved, we're building out SMTP support for it based on customer request.

> **KnowledgeOwl currently supports Microsoft OAuth2 SMTP. If you need us to set up another SMTP provider's OAuth2 authorization flow, please contact us and let us know the provider--we will add new providers based on customer interest.**

We've tested our OAuth2 configuration with the Azure Portal.

You will need your IT administrator to set up the Microsoft/Azure side of SMTP. We'd also recommend either giving them direct access to configure our side of the SMTP setup, or making sure you have time set aside to work with them to get things set up in KnowledgeOwl properly.

Resources for the Microsoft/Azure admin

1. You'll need to create an app on the Microsoft side to handle these requests. We found this Microsoft documentation helpful as a starting place:

   - **Quickstart: Register an application with the Microsoft identity platform**

   - **Authenticate an IMAP, POP or SMTP connection using OAuth**

2. You'll also need to have at least one active email account that can be used by that app. This user account must have the **Authenticated SMTP** permission in **Mail > Email Apps**.

3. Here are the settings we used for our app, though you may want to tweak for your setup:

   - **Single tenant app registration OR Multi-tenant app registration**

   - **Under redirect URI, add a Web URI for** `https://app.knowledgeowl.com/oauth2/custom-smtp-token`

   - **In API permissions, we used Microsoft Graph > delegated permissions and used these permissions:**

- **Mail.Send**

- **Mail.Send.Shared**

- **SMTP.Send**

4. **The app you create will need to go through an authorization process within KnowledgeOwl, initiated by a user of your organization. You will need to use Power Shell to set that user's value for** `SmtpClientAuthenticationDisabled` **to** `false` **. (See the official Microsoft documentation for more details:** Enable or disable authenticated client SMTP submission (SMTP AUTH) in Exchange Online**.)**

5. **On the KnowledgeOwl side, you'll need to provide:**

   ○ **The email address that should be used--this should be the email account mentioned in step 1 above.**

   ○ **The Client ID, listed as the Application or Client ID**

     ■ **If you're using Azure Portal, this is the Application (client) ID listed on the App registration's Overview page**

   ○ **The Client Secret Value**

     ■ **If you're using Azure Portal, you can generate this in the App registration's Certificates & secrets**

     ■ **In our testing, we were only able to have one active secret connecting to KO within the app; adding additional secrets failed.**

   ○ **If using a Single Tenant app, the Tenant ID listed on the App registration's Overview page**

> **If you are using Azure SMTP within an app, it will only send to email addresses within your Azure organization (single tenant) or with a valid Microsoft email (multitenant). This setting can be configured in Authentication > Supported account types in Azure.**

KnowledgeOwl setup

Once you have the Azure app and email account set up and a client secret generated, you can begin setting up your new SMTP service in KnowledgeOwl. To do so:

1. **Click on your profile icon/name in the upper right.**

2. **Select SMTP from the dropdown.**

3. **Select the + Configure New Service button.**

4. **This will open the Create SMTP Service page:**

Create SMTP Service

Friendly Name * **5** `My Custom SMTP`

SMTP Host * **6** `smtp.myorg.com`

Port * **7** `587`

Send From Address * **8** `support@myorg.com`

Reply To Address **9**

Send From Name **10** `MyOrg Support`

Authentication ○ Unauthenticated

○ Username/Password

11 ● OAuth2

Once this is created you'll need to authorize the service before it can be used

Provider **12** `Microsoft` ⌄

Login Email * **13**

Client ID * **14**

Client Secret * **15**

Tenant ID **16**

Only required for single tenant applications.

17

[Back]   [Create]

5. Add a **Friendly Name**. This is the name that will be displayed in SMTP dropdowns for individual features, so it should describe the SMTP service in some way.

6. Add the **SMTP Host**. This is the address your email server provides for the SMTP host. For Microsoft, use `smtp.office365.com` unless your IT admin tells you otherwise.

7. Add the **Port**. Most providers will have specific recommended ports. Unless your IT admin tells you otherwise, use `587` .

8. Add the **Send From Address**. The email address that will show up in the "From" field for all emails sent using this SMTP service. This should be a valid email address in your Microsoft organization.

9. *Optional:* Add a **Reply To Address**. This allows you to specify a Reply-to address different from the Send From Address.

10. *Optional:* Add a **Send From Name**. This is the name that will show up in the "From" field for any emails sent using this SMTP service.

11. In the **Authentication** section, select **OAuth2** to configure your SMTP using OAuth2.

12. Select **Microsoft** as the provider.

13. Add the **Login Email** that will be used to authenticate. This needs to be an account that has permissions to send email.

14. Add the **Client ID** from your app. In Azure Portal, this is the Application (client) ID listed on the App registration's Overview page.

15. Add the **Client Secret** from your app. In Azure Portal, you can generate this in the App registration's Certificates & secrets page.

16. If you are using a Single Tenant app, add the **Tenant ID** for your app. In Azure Portal, this is the Tenant ID listed on the App registration's Overview page.

> ⚠️ **DO NOT ENTER A TENANT ID IF YOU ARE USING A MULTI-TENANT APP.**

17. Once you've finished completing all fields, click **Create**.

18. Once you have created the service, you must **Authorize** that service. This uses the information you previously entered to perform the OAuth2 authorization flow and get the token we need to authenticate. To begin authorization, select the **Authorize** button.

Edit SMTP Service
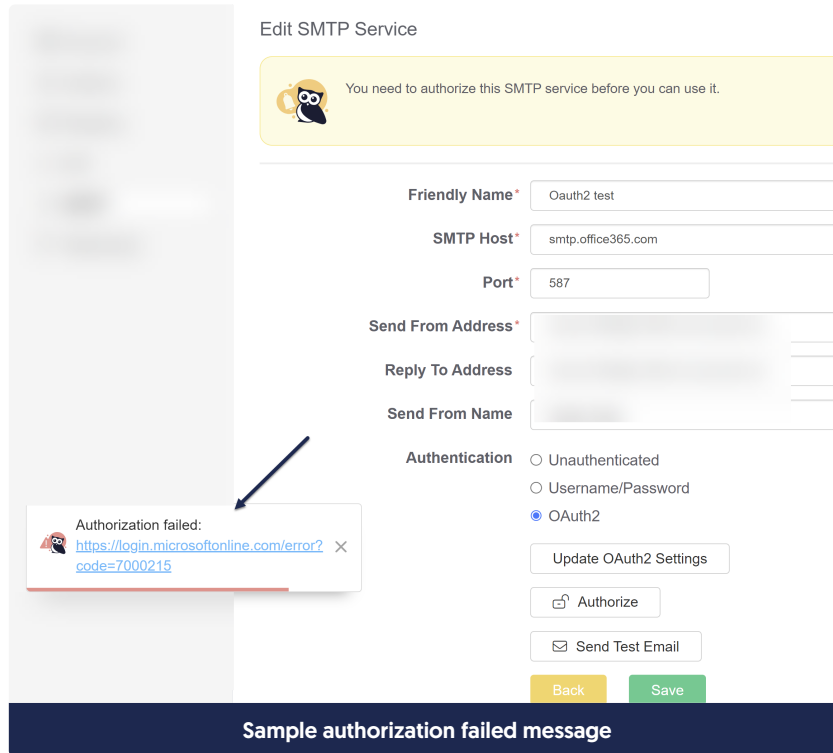
Friendly Name*	Azure Oauth2
SMTP Host*	smtp.office365.com
Port*	587
Send From Address*	██████.onmicrosoft.com
Reply To Address	██████.onmicrosoft.com
Send From Name	KO Azure
Authentication	○ Unauthenticated
	○ Username/Password
	● OAuth2

Update OAuth2 Settings

🔓 Authorize

✉ Send Test Email

Back   Save

Once you've created your OAuth2 SMTP service, the Authorize button appears

a. If there are any issues authorizing your SMTP service, an error message will appear in the lower left. This
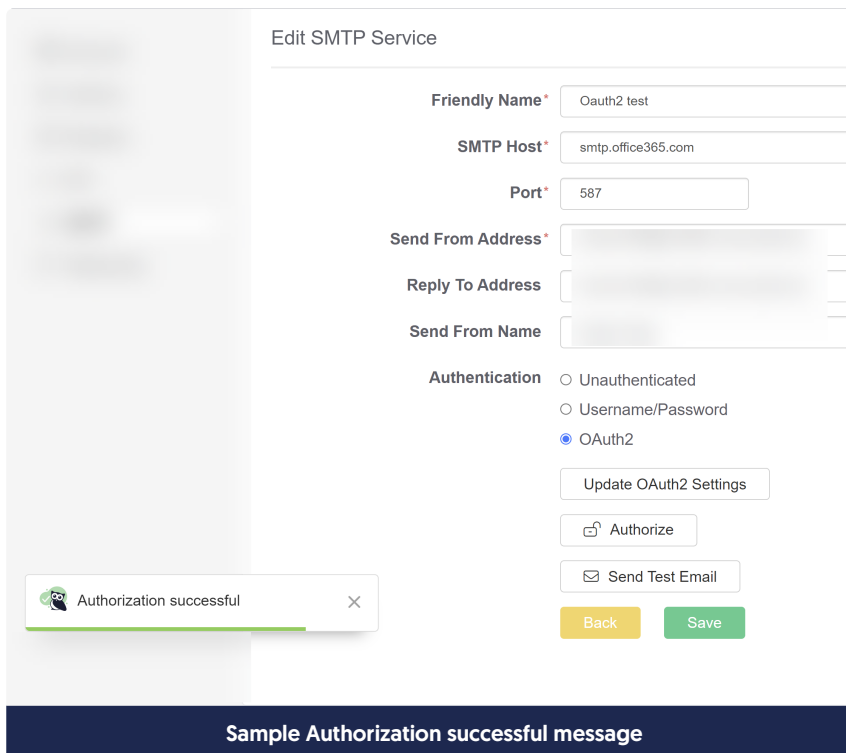
message will contain a link to the error code we received back from Microsoft; you can click that link to get more details on what caused your authorization to fail. You'll want to review this error code with your IT admin:

Sample authorization failed message

19. When the authorization completes successfully, a confirmation message will slide out in the lower left. Once you see that message, continue to the next step:

Sample Authorization successful message

20. Once you get the authorization successful message, click the **Send Test Email** to fully confirm the SMTP is configured properly. We **strongly recommend** doing this to confirm that emails are sending properly. (Note that Azure will expect you to send this to an email within your own organization.)

   a. If your authorization completed successfully, the test email should go through and show you a confirmation message. At this point, your SMTP is ready to use!

   b. If for any reason your test email fails:

      i. Try re-authorizing the service. Make sure you get a confirmation message.

      ii. Be sure the test email you're sending is part of your organization's Azure platform. Azure SMTP cannot be used to send emails to external services.

      iii. Be sure that your Microsoft admin has run the Power Shell command to set the account you're authorizing the app with to have `SmtpClientAuthenticationDisabled` **set to** `false` .

Once you have sent a successful test email, your SMTP service should now be available for use.

You can add it to:

- Any knowledge base's **Subscription** settings
- Your account's **Reader emails** for reader welcome and password reset emails

KnowledgeOwl IP Address

If you need to set up an exception for these communications with your SMTP provider, see **Required IP addresses** for more details.