Mail receive with OAuth2.0

Description

The *Mail receive with OAuth2.0* app can retrieve mails from mailboxes that require an OAuth 2.0 login.

It largely repeats the functionality of the *Mail receive* element available in Switch Designer, but there are some differences (see at the end).

Compatibility

Switch 2020 Spring and higher.

Connections

There are no incoming connections, and the app has a single outgoing connection.

Properties detailed info

Flow elements properties

- **Name** Simple name of your element in the design area. Choose a name that is easy to understand, e.g. the name of the mail server/service.
- **Description** Short description of the flow element. Usually contains information about the goals of this component in the flow. This shows up in the page generated by *Document flow*
- **User name (login)** the user's email address. E.g. switch-user@gmail.com. This address will be used for connecting to the mailbox and extracting the mails' contents.
- **Server** Information about the mail server connection that will be used to retrieve the data. It is a dropdown list with three possible values: *GMail*, *Outlook.com* and *Other*
    - ***Server address*** URL or IP address of the IMAP-enabled server to establish the connection to (accessible only if server set to *Other*).
      For *GMail* the default is *imap.gmail.com*, for *Outlook* it is *outlook.office365.com*.
    - ***Port*** The port to use for communication with the server (accessible only if the server set to *Other*, for *GMail* and *Outlook 993* port is used). Default value: *143*.
    - ***Authentication type*** A dropdown with the authentication method to use to connect to the IMAP server. The values are *Password* (for user/password authentication) or *OAuth 2.0* (when an OAuth2 token is to be used)
        - ***Password*** The user password which is used to access the mail account (accessible only if *Authentication type* is set to *Password*). Default value: *Password*
        - **Server requires secure connection** Defines whether the mail server requires a secure connection using the TLS protocol or not. When using the OAuth 2.0 authorization method or when connecting to Gmail or Outlook, this property is not available because in those cases always require a secure connection.

- **OAuth 2.0 authorization token** The OAuth 2.0 token the user gets for his account to access the mailbox. The user should provide at least 5 fields to create a token:
    - *Application ID*
    - *Application password*
    - *Authorization URL*
    - *Token URL*
    - *Scope*

  The app simplifies this for *GMail* and *Outlook* because these services always use the same URL's and scopes regardless of the account. For *Gmail, Token URL: https://oauth2.googleapis.com/token, Authorization URL: https://accounts.google.com/o/oauth2/v2/auth, Scope: https://mail.google.com/*. More info about token creation can be found here: https://developers.google.com/identity/protocols/oauth2.

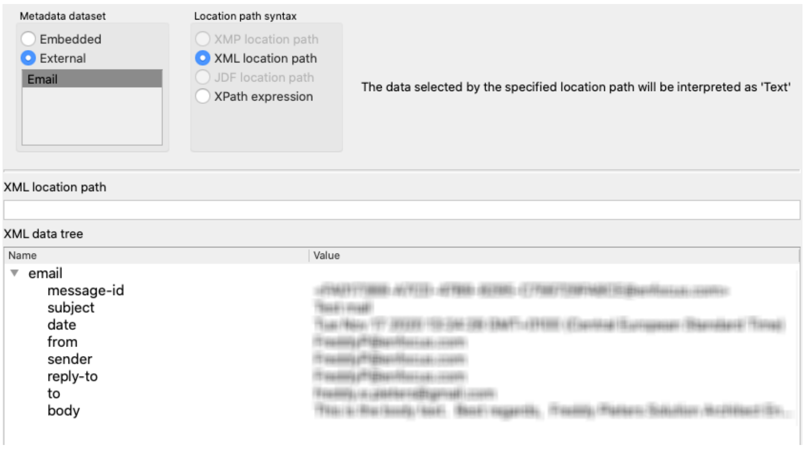
- **Leave originals on server** If set to *No*, mails will be removed from the server's inbox, otherwise they will be moved to another mailbox folder
    - ***Move to IMAP folder*** The IMAP folder where to move mails that were already processed by Switch (only accessible if *Leave originals on server* property set to *Yes*)
- **Check every (minutes)** The frequency in minutes with which the email account will be checked for new mail
- **Time-of-day window** If set to yes, the app checks for new mail only during a certain period of the day
    - ***Allow from (hh:mm)***
    - ***Allow to (hh:mm)***
- **Day-of-week window** If set to yes, the app checks for new mail only on certain days of the week
    - ***Allow from***
    - ***Allow to***
- **Day-of-month window** If set to yes, the app checks for new mail only on a certain day of the month
    - ***Day*** A number in the range [1 . . 31]
    - ***Relative to*** Determines whether the day of the month is relative to *Start of the month* or *End of the month*
- **Attach metadata** Response for attaching additional mail info as job metadata. If set to *Yes*, an XML dataset with mail data will be attached to the generated job.
    - ***Dataset name*** The name of the dataset that will be created for the job. Accessible only if *Attach metadata* set to *Yes*
    - ***Convert HTML to plain text*** If set to *Yes*, the HTML message body will be converted to plain text before it will be written in dataset. If set to *No*, the message body will remain in HTML format. Accessible only if *Attach metadata* set to *Yes*
- **Scan for nested attachments** Set to *Yes* to scan all nested emails until only real assets are found.

- **Collect attachments** Set to *Yes* to collect all attachments in a folder. Single file attachments will also end up in a job folder
    - **Assemble message and attachments** Set to *Yes* to assemble the injected message and the attachment(s) in the same job folder. If there is no message to inject, this property will not affect the attachment handling. Accessible only if *Collect attachments* is set to *Yes*.
- **Inject message as file** Determines whether the email message itself (as opposed to its attachments) is injected into the flow as a separate file. Choices are:
    - *No*: only attachments are injected in the flow
    - *If no attachments*: the mail message is injected as a separate file only if it has no attachments
    - *Always*: the mail message is always injected as a separate file. Depending on the value of the *Convert HTML to plain text* property the message is saved as a .html file or as a .txt file with UTF8 encoding. The file will be named according to the subject line of the e-mail message. If there is no subject line, the file will be named *No subject.txt* or *No subject.html*.
- **Convert HTML to plain text** If set to *Yes*, the HTML message body will be converted to plain text before using it for generating output data. If set to *No*, the message body will remain in HTML format, if applicable of course. Accessible only if *Inject message as file* set to *Yes*.

Differences with *Mail receive*

This app only supports IMAP and not POP.

Mail receive has three properties that allow adding information about the mail to the job ticket. Concretely speaking, it can populate the variables of the Email group of the Switch variables, and it can also add the name of the element to the job's hierarchy info. This app is based on the new NodeJS-based scripting environment and as not all functions to fill in the job ticket are available yet, this app cannot do that yet. However, there is a property that allows you to add the metadata of the mail as an XML dataset to the job. This is what the XML looks like:



Metadata dataset
Embedded [✓ External]
Email

Location path syntax
XMP location path [✓ XML location path] JDF location path XPath expression

The data selected by the specified location path will be interpreted as 'Text'

XML location path

XML data tree

Name	Value
email	
message-id	[redacted]
subject	[redacted]
date	[redacted]
from	[redacted]
sender	[redacted]
reply-to	[redacted]
to	[redacted]
body	[redacted]