GMail Connected System

The Connected System allows you to authenticate using a service account against a GMail server inbox. With that connection you have the following features available:

- Get a list of messageIds for messages that are in the Inbox and are Unread
- Get a message by its Id and get all data (like body, recipients, threadId, etc.)
- Send a message (either new or as a reply to an existing message)

The purpose is to provide an easy-to-use way to collect emails from a GMail system using an API instead of an email poller. It is not intended for processing high volumes of emails. The target when building this Connected System was up to 250 emails per hour.

It is highly recommended using these integrations in conjunction with the Transaction Manager. Use the ListInboxMessages to get the list of messages to process and create a transaction for each message. Then have a transaction Job Type that gets the message itself. As this can be a resource intensive operation, as a rule of thumb do not plan more than 5 per transaction manager cycle. This obviously depends on the way your infrastructure is set up and what other processes are running on the system.

Sending messages through this mailbox is expected to be low volume as well; the Google account settings might even block the account if high volumes are used. The purpose of the send email functionality os to send emails directly from the mailbox, without having to go through an smtp, which might cause messages to be sent 'on behalf of'.

The connected system

The Google Mail API Connected System allows you to provide all service account credentials to connect to the GMail inbox of your choice.

As a prerequisite, you must set up a service account on the Google platform. Make sure this service account has the permissions/scope 'https://www.googleapis.com/auth/gmail.modify https://www.googleapis.com/auth/gmail.labels' and is allowed to impersonate the user/email address for the mailbox you want Appian to access. It is recommended to download the service_account json file, as it contains all the data needed to set up the connected system.

Connected System Properties

Input	Data Type	Required	Multiple	Description
Name	Boolean	Yes	No	The name for the connected system
Description	Text	No	No	Your description for the connected system
Application Name	Text	No	No	The name of your application. This can be used on the Google Monitoring to identify which application is accessing GMail
Google Project ID	Text	Yes	No	The project ID for the project in the Google Platform; you can find it in the service account JSON file
Service Account Private Key ID	Text	Yes	No	The Service Account Private Key ID for the project in the Google Platform; you can find it in the service account JSON file
Private Key	Text	Yes	No	The Private Key itself; you can find it in the service account JSON file. You can include the full string, including -----BEGIN

Private Key	Text	Yes	No	PRIVATE KEY----- and -----END PRIVATE KEY-----. The \n characters you find in the JSON do not have to be removed
Client Email	Text	Yes	No	The email address for the service account. This is not the sane as the email address for the inbox you want to connect to; that can be provided as 'GMail Impersonation User' property
Client ID	Text	Yes	No	The Client ID for the service account in the Google Platform; you can find it in the service account JSON file
GMail Impersonation User	Text	No	No	The name for the mailbox you want to read and/or send emails from in case it is different than the one for the service account
Scope	Text	No	No	The scope(s) you need to connect with. If you use more than one scope then add a [space] between the scopes, e.g. https://www.googleapis.com/auth/gmail.modify https://www.googleapis.com/auth/gmail.labels

TEST CONNECTION

The TEST CONNECTION button allows you to validate that you can authorize successfully with the service account

GMail Connected System Integrations

The GMail connected system provides a number of integrations that will use the service account to connect to the mailbox (of the impersonated user) and run the provided integration.

ListInboxMessages

The ListInboxMessages function extracts a list of messageIds from the connected mailbox and returns it as a list of strings in the Result. It has no input parameters.

The proposed way to save the list of messageIds is to add a custom output on the Integration Data Output Tab with the following expression: `index(ac!Result,"messageIds",null)` and save the result in a variable of type Text-Multiple. This list of message IDs can then be used to queue transactions for a job in the transaction manager.

GetMessageById

The GetMessageById function extracts all information from an email given its messageId and returns it in the Integration Result. Any attachments, as well as an EML version of the email (that can be downloaded and displayed separately) will be saved in the provided Storage Folder. After processing the email, it will be relabeled with the provided GMail Success Label in case processing was successful. If processing was unsuccessful, it will be labeled with the provided Failure label. If the Label(s) do not yet exist, they will be created by the Connected system.

Inputs

Input	Data Type	Required	Multiple	Description
MessageId	Text	Yes	No	The Id for the message to be extracted, typically provided via a rule input

Storage Folder	Text	Yes	No	The folder where attachments will be stored
GMail Success Label	Text	Yes	No	The label to use after processing the mail; labels INBOX and UNREAD will be removed. This can only be a single label
GMail Failure Label	Text	Yes	No	The label to use after processing the mail and running into a failure doing so; labels INBOX and UNREAD will be removed. This can only be a single label

Returns

The result goes into a dictionary with all properties of an email. You can access these by creating a variable of type Map in your process model and map the Result part of the Results to it (using index(ac!Result,"", null) in Custom Outputs). The following properties can be expected in this result dictionary:

- attachments - a dictionary of attachments for the mail. Each attachment has the following properties
  - filename - The name of the attachment
  - size - The size of the attachment in KBytes
  - docId - The Appian Document as saved in the target folder
- Subject - Subject for the email
- From - Sender of the email
- To - The recipient(s) of the email
- Cc - The CC recipient(s) of the email
- Bcc - The BCC recipient(s) of the email (only visible if you are on the bcc list).
- internalDate - The internal message creation timestamp (As Appian Date and Time), which determines ordering in the inbox. For normal SMTP-received email, this represents the time the message was originally accepted by Google.
- References - The list of messageIds that this message refers to. This is used to build threads in the GMail client and should be included again when sending a reply.
- message-ID - The unique messageId for the message (same as input)
- threadId - The threadId for the message. It is the same as the first message in the thread and can be used to group messages together
- historyId - The ID of the last history record that modified this message.
- received - String with the date when the email was received, format DDD, dd Mon yyyy hh:mm:ss +TimezoneOffset. This comes from the header and is less secure than the internalDate.
- labels - Labels on the message before processing it
- bodyText - The plain text version of the email body
- bodyHtml - The HTML version of the email body
- EMLDocId - The Appian Document that contains the email in an EML format (including its attachments)

SendMessage

The SendMessage function creates an email and sends it directly from the GMail account, storing the message in the Sent folder afterwards. Attachments can be added, as well as the list of Message-IDs that this email is a reply to, in order to make threading work properly in GMail clients.

Inputs

Input	Data Type	Required	Multiple	Description
Email body	Text	Yes	No	The body content to send; this can be HTML or plain text
Is HTML Body?	Text	Yes	No	Indicator to flag if the Email Body is HTML based or not
Email sender	Text	Yes	No	The email address of the sender of the email
Email recipient	Text	Yes	No	One or more email recipients. If you want to define

				multiple recipients, separate them with a ,
Email CC recipient	Text	No	No	Zero or more email CC recipients. If you want to define multiple CC recipients, separate them with a ,
Email BCC recipient	Text	No	No	Zero or more email BCC recipients. If you want to define multiple BCC recipients, separate them with a ,
Email Subject	Text	Yes	No	The subject for the email
In-Reply-To Message-ID	Text	No	No	The Message ID for the message that should be replied to
Reference	Text	No	No	All Message-IDs that this email relates to. The In-Reply-To Message-ID will automatically be added to this. Take the references from the mail you received and include it again when replying to it
Thread ID	Text	No	No	The Thread ID the message belongs to
Documents	Document	No	Yes	The Document(s) to include in the email as attachment

In-Reply-To Message-ID, Reference and Thread-ID all relate to replies only. The In-Reply-To Message-ID is the most important one to include, but depending on the client, Reference and Thread ID might also be needed. As a rule of thumb, use all three for the best results.

Returns

The Result will have the messageId of the message that was created when sending was succesful. Otherwise an integration error will be populated with the reason for the error.