

# IMAP Connector

The word “IMAP” is referred to as – Internet Mail Access Protocol. So, IMAP is an internet protocol which is responsible for the handling of email operations that are carried over a network. The **IMAP Connector** is used for performing various kind of email operations. Basically, It creates an IMAP channel that enables you to retrieve email from an IMAP email server. IMAP cannot send email, for that, you must use an SMTP outbound connector. You can choose the required operations from the dropdown using templates from your BPMN process.

## Prerequisites

To start working with the IMAP, user required host name of email server, port number, username and password, domain etc. User need to specify a folder path in email server.

- Host- A hostname is a distinct name or label assigned to any device connected to a computer network, in this case its location where email server is hosted.
- Username and Password: Username and password of respective server
- Port Number: Port Number of IMAP server

## Creating IMAP connector task

Currently, the IMAP Connector supports six types of operations: list emails, search emails, delete email, download email, move email, read email.

To use a IMAP Connector in your process, either change the type of existing task by clicking on it and using the wrench-shaped Change type context menu icon or create a new Connector task by using the Append Connector context menu. Follow our [guide on using Connectors](#) to learn more.

## Making IMAP Connector executable

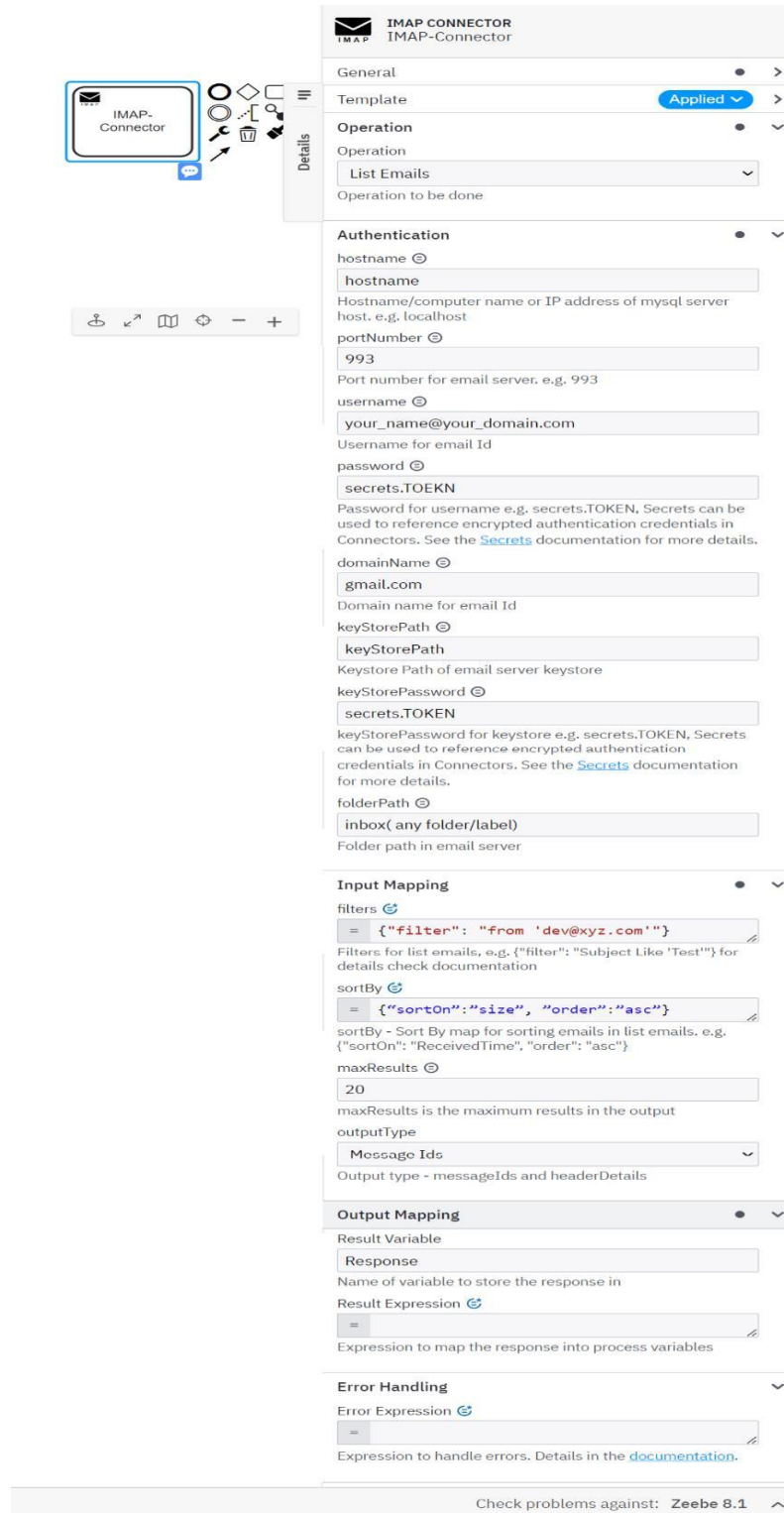
To make the **IMAP Connector** executable, fill out the mandatory fields highlighted in red in the properties panel.

## Authentication for IMAP Connector

**IMAP Connector** authentication object takes – host, port, username and password(as secrets Token i.e.secrets.Token) domainName, keyStorePath, keyStorePassword(as secrets Token e.g. secrets.Token) and folderPath. folderPath is nothing but folders/labels in email(e.g inbox).

There are six important email operations to be performed by using IMAP Connector.

# 1. OPERATION – LIST EMAILS



**IMAP CONNECTOR**  
IMAP-Connector

General

Template **Applied**

Operation

Operation  
List Emails

Operation to be done

Authentication

hostname  
hostname  
Hostname/computer name or IP address of mysql server  
host, e.g. localhost

portNumber  
993  
Port number for email server, e.g. 993

username  
your\_name@your\_domain.com  
Username for email Id

password  
secrets.TOKEN  
Password for username e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

domainName  
gmail.com  
Domain name for email Id

keyStorePath  
keyStorePath  
Keystore Path of email server keystore

keyStorePassword  
secrets.TOKEN  
keyStorePassword for keystore e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

folderPath  
inbox( any folder/label)  
Folder path in email server

Input Mapping

filters  
= {"filter": "from 'dev@xyz.com'"}  
Filters for list emails, e.g. {"filter": "Subject Like 'Test'"} for details check documentation

sortBy  
= {"sortOn": "size", "order": "asc"}  
sortBy - Sort By map for sorting emails in list emails. e.g. {"sortOn": "ReceivedTime", "order": "asc"}

maxResults  
20  
maxResults is the maximum results in the output

outputType  
Message Ids  
Output type - messageIds and headerDetails

Output Mapping

Result Variable  
Response  
Name of variable to store the response in

Result Expression  
=  
Expression to map the response into process variables

Error Handling

Error Expression  
=  
Expression to handle errors. Details in the [documentation](#).

Check problems against: Zeebe 8.1

## To List an emails, take the following steps:

1. In the **Operation** section, set the field value **Operation** as **List Emails**.
2. Set the required parameters and credentials in the **Authentication** section.
3. In the **Input Mapping** section, set the field filters, sortBy, maxResults and outputType as the desired. You must use FEEL expression for filters and sortBy.
4. “filters” is a map with keys – filter, filterList and logicalOperator. It will be used to create searchTerm for search messages. The value for filter key is string with space separated –searchFeild, operator and value. If value is string or date, wrap it in single quote.
5. Set **filters**, using FEEL expression as a context with key-value pairs.
  - {"filter": "subject = 'Find Attachment'" } -> Search for emails containing “Find Attachment” as a substring.
  - {"filter", "ReceivedTime <= '14/09/2022 04:08:00 PM'" } -> Search for emailswith ReceivedTime less than this date
  - {"filter", "ReceivedTime > '29/03/201 12:01:00 AM'" } -> Search for emails with ReceivedTime less than this date
  - {"filter": "body 'contains this string'" } -> Search for emails whose body contains 'this string'
  - {"filter": "from '[abhi@abc.com](#)'" } -> Search for emails which came from '[abhi@abc.com](#)' Id.
  - {"filter": "size >=73265" } -> Search for emails with size greater than or equal to 73265 bytes
6. Above filters are all simple filters. You can write more complex filters using logical Operators.

For example,

```
{
  "logicalOpeartor" : "AND",
  "filterList" : [
    {
      "filter": "body 'contains this string'"
    },
    {
      "filter": "from 'abhi@abc.com'"
    },
    {
      "filter": "size >= 72635"
    }
  ]
}
```

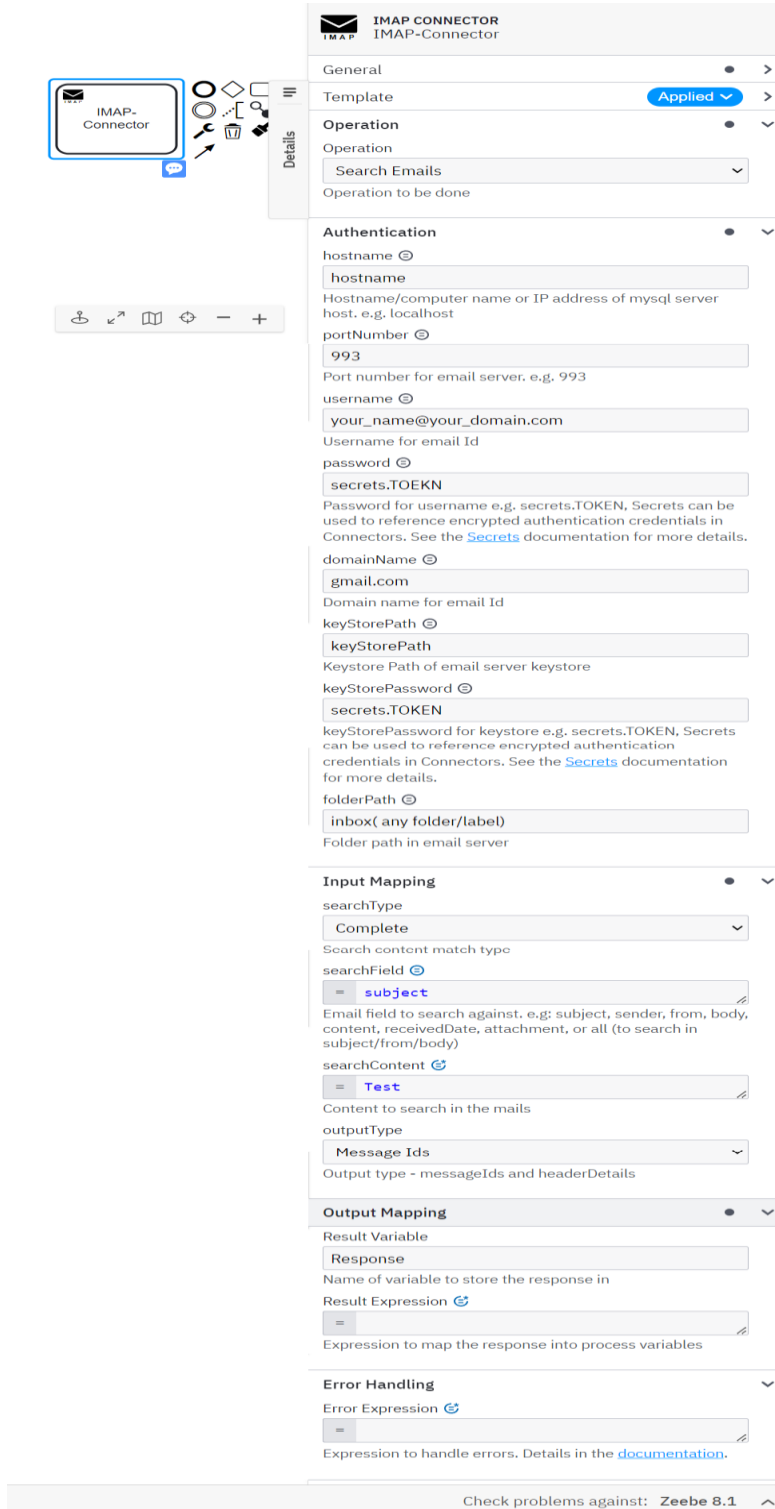
7. Set **sortBy**, using FEEL expression as a context with key-value pairs. For eg. {"sortOn": "from", "order": "asc"}. For sortOn key different values can be used like size, from, subject, message number, messageID, sent date or received date. For order key use "asc" or "dsc" which sort according to ascending and descending order.
8. Date String should be in format - "dd/MM/yyyy hh:mm:ss a", "dd/MM/yyyy HH:mm:ss", "dd/MM/yyyy", "E MMM dd HH:mm:ss Z > yyyy", "EEEE MMMM d yyyy", "MMMM d yyyy", "yyyy-MM-dd HH:mm:ss", "yyyy-MM-d HH:mm:ss", "yyyy-MM-dd", "dd MMM yyyy".
9. **maxResults** is maximum number of records in output
10. **outputType** can be selected as Message Id or Header Details as per requirements.

## List Emails operation response

You can use an output mapping to map the response:

- Use **Result Variable** to store the response in a process variable. Response is based on what one selects in outputType i.e. messageId or headerDetails.  
 Message Ids - List of email message ids.  
 headerDetails - List of maps of email header information related to - subject, size, received date. Etc.
- Use Result Expression i.e FEEL to map the response into process.

## 2. OPERATION –SEARCH EMAILS



**IMAP CONNECTOR**  
IMAP-Connector

General

Template **Applied**

**Operation**

Operation  
Search Emails

Operation to be done

**Authentication**

hostname  
hostname  
Hostname/computer name or IP address of mysql server host. e.g. localhost

portNumber  
993  
Port number for email server. e.g. 993

username  
your\_name@your\_domain.com  
Username for email Id

password  
secrets.TOKEN  
Password for username e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

domainName  
gmail.com  
Domain name for email Id

keyStorePath  
keyStorePath  
Keystore Path of email server keystore

keyStorePassword  
secrets.TOKEN  
keyStorePassword for keystore e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

folderPath  
inbox( any folder/label)  
Folder path in email server

**Input Mapping**

searchType  
Complete

Search content match type

searchField  
= subject  
Email field to search against. e.g: subject, sender, from, body, content, receivedDate, attachment, or all (to search in subject/from/body)

searchContent  
= Test  
Content to search in the mails

outputType  
Message Ids  
Output type - messageIds and headerDetails

**Output Mapping**

Result Variable  
Response  
Name of variable to store the response in

Result Expression  
=  
Expression to map the response into process variables

**Error Handling**

Error Expression  
=  
Expression to handle errors. Details in the [documentation](#).

Check problems against: **Zeebe 8.1**

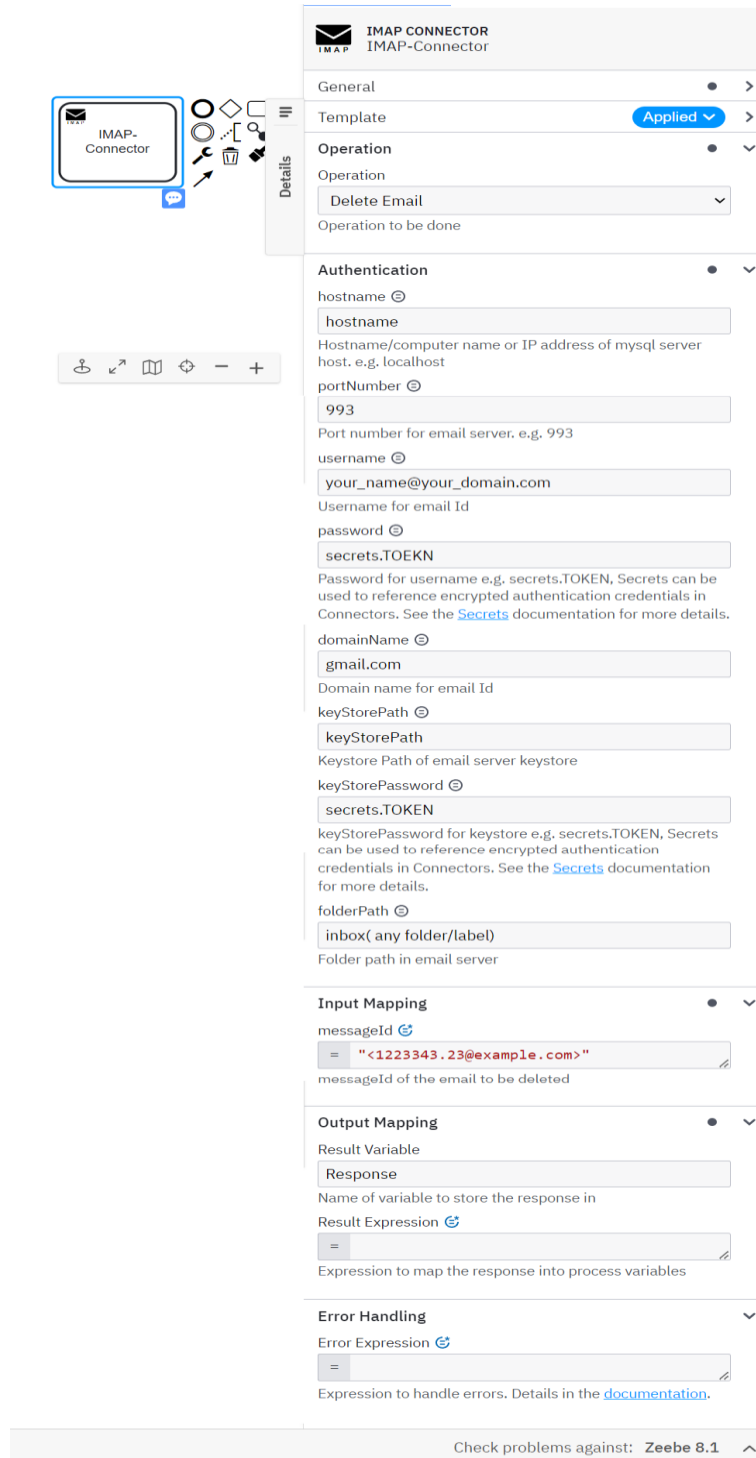
## To Search an emails, take the following steps:

1. In the **Operation** section, set the field value **Operation** as **Search Emails**.
2. Set the required parameters and credentials in the **Authentication** section. See the relevant appendix entry to find out more.
3. In the **Input Mapping** section, set the field **searchType**, **searchField**, **searchContent** and **outputType** as the desired. FEEL expression is required for **searchField**, **searchContent**.
4. **searchField** should contain email field to search against. E.g. subject, sender, from, body, content, receivedDate, attachment, or all (will search in every fields)
5. **searchContent** should contain content one is looking for in particular **searchField**.
6. **outputType** can be selected as Message Id or Header Details as per requirements.

## Search Emails operation response

- Use **Result Variable** to store the response in a process variable. Response is based on what one selects in **outputType** i.e. **messageIds** or **headerDetails**.  
Message Ids - List of email message ids.  
headerDetails - List of maps of email header information related to - subject, size, received date. Etc.
- Use Result Expression i.e FEEL to map the response into process.

### 3. OPERATION – DELETE EMAIL



**IMAP CONNECTOR**  
IMAP-Connector

General

Template **Applied**

**Operation**

Operation  
Delete Email  
Operation to be done

**Authentication**

hostname  
hostname  
Hostname/computer name or IP address of mysql server host. e.g. localhost

portNumber  
993  
Port number for email server. e.g. 993

username  
your\_name@your\_domain.com  
Username for email Id

password  
secrets.TOKEN  
Password for username e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

domainName  
gmail.com  
Domain name for email Id

keyStorePath  
keyStorePath  
Keystore Path of email server keystore

keyStorePassword  
secrets.TOKEN  
keyStorePassword for keystore e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

folderPath  
inbox( any folder/label)  
Folder path in email server

**Input Mapping**

messageId  
=  
"<1223343.23@example.com>"  
messageId of the email to be deleted

**Output Mapping**

Result Variable  
Response  
Name of variable to store the response in

Result Expression  
=  
Expression to map the response into process variables

**Error Handling**

Error Expression  
=  
Expression to handle errors. Details in the [documentation](#).

Check problems against: **Zeebe 8.1**

### To Delete email, take the following steps:

1. In the **Operation** section, set the field value **Operation** as **Delete Email**.
2. Set the required parameters and credentials in the **Authentication** section. See the relevant appendix entry to find out more.
3. In the **Input Mapping** section, set the field messageId.
4. Set **messageId**, which is single unique message id like [<37447564.88583@abc.com>](mailto:37447564.88583@abc.com)

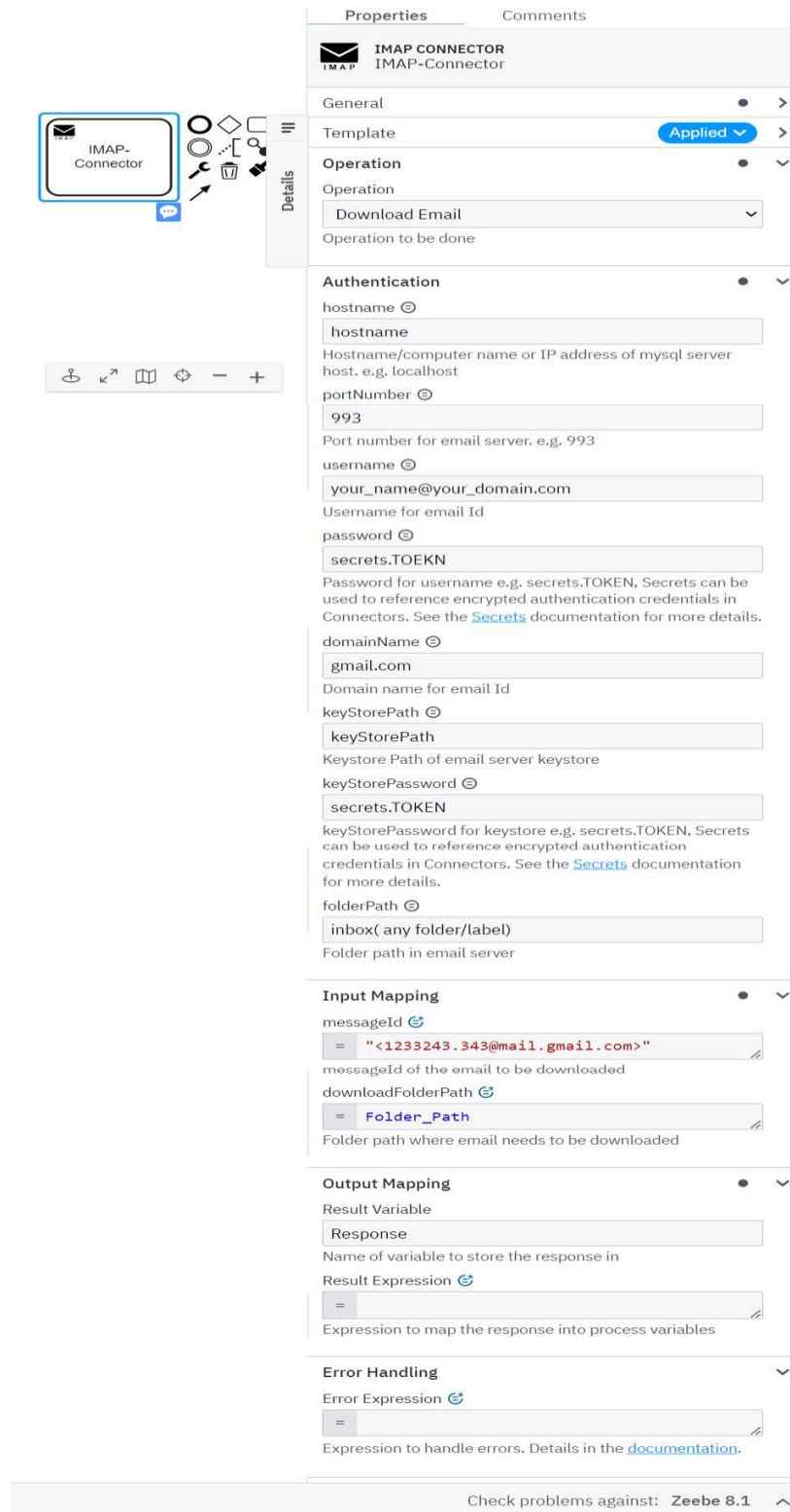
### Delete Email operation response

You can use an output mapping to map the response:

1. Use **Result Variable** to store the response in a process variable.
2. Use Result Expression i.e. FEEL to map the response into process.



## 4. OPERATION – DOWNLOAD EMAIL



**Properties** **Comments**

**IMAP CONNECTOR**  
IMAP-Connector

**General**

**Template** **Applied**

**Operation**

Operation  
Download Email  
Operation to be done

**Authentication**

hostname  
hostname  
Hostname/computer name or IP address of mysql server host. e.g. localhost

portNumber  
993  
Port number for email server. e.g. 993

username  
your\_name@your\_domain.com  
Username for email Id

password  
secrets.TOKEN  
Password for username e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

domainName  
gmail.com  
Domain name for email Id

keyStorePath  
keyStorePath  
Keystore Path of email server keystore

keyStorePassword  
secrets.TOKEN  
keyStorePassword for keystore e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

folderPath  
inbox( any folder/label)  
Folder path in email server

**Input Mapping**

messageId  
=  
"<1233243.343@mail.gmail.com>"  
messageId of the email to be downloaded

downloadFolderPath  
=  
Folder\_Path  
Folder path where email needs to be downloaded

**Output Mapping**

Result Variable  
Response  
Name of variable to store the response in

Result Expression  
=  
Expression to map the response into process variables

**Error Handling**

Error Expression  
=  
Expression to handle errors. Details in the [documentation](#).

Check problems against: **Zeebe 8.1**

### To download email operation, take the following steps:

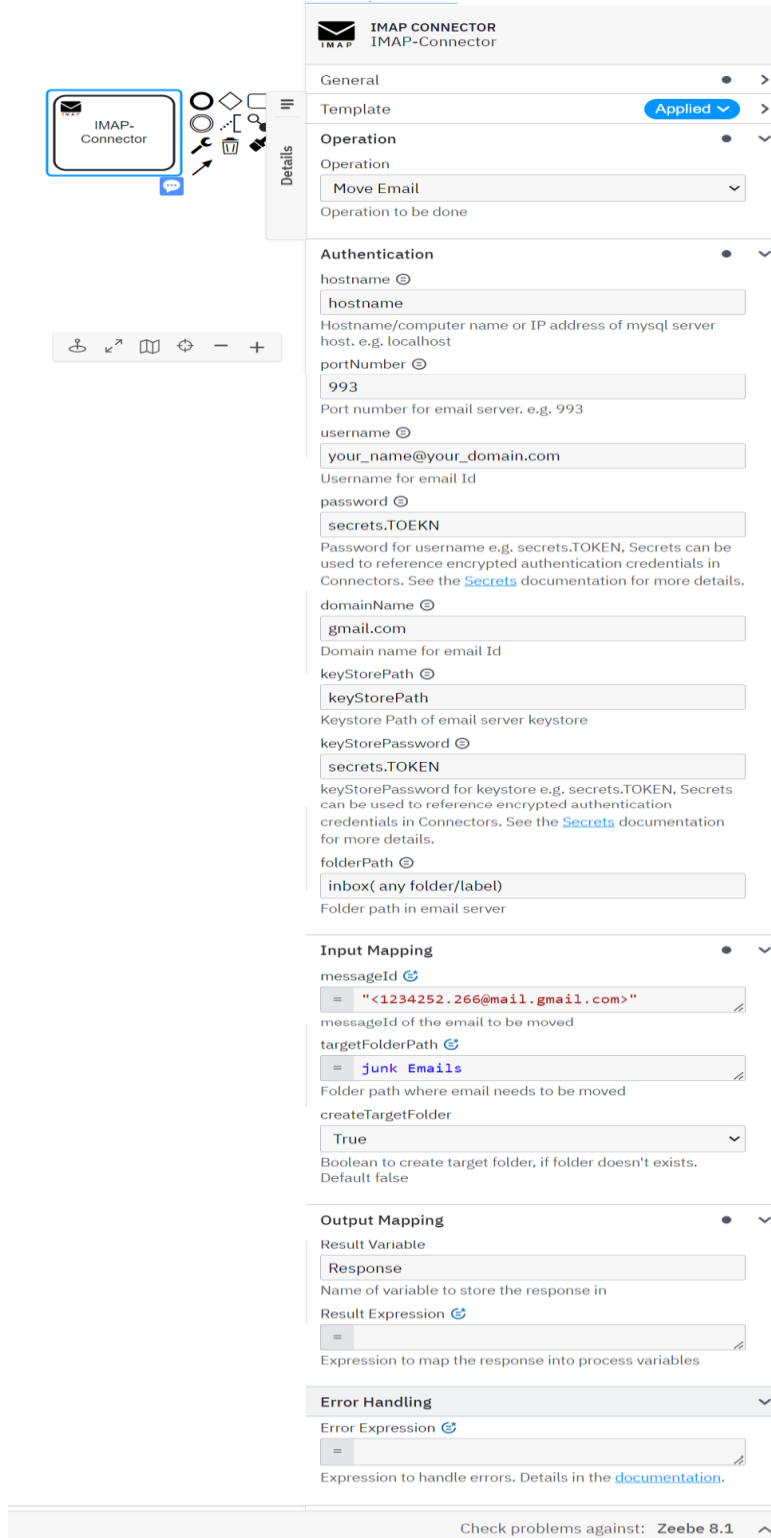
1. In the **Operation** section, set the field value **Operation** as **Download Email**.
2. Set the required parameters and credentials in the **Authentication** section. See the relevant appendix entry to find out more.
3. In the **Input Mapping** section, set the field **messageId**, **downloadFolderPath**.
4. Set **messageId**, which is single unique message id.
5. Set **downloadFolderPath**, which will be used to store downloaded email.

### Download Email operation response

You can use an output mapping to map the response:

1. Use **Result Variable** to store the response in a process variable.
2. Use Result Expression i.e., FEEL to map the response into process.

## 5. OPERATION – MOVE EMAIL



**IMAP CONNECTOR**  
IMAP-Connector

General

Template **Applied**

**Operation**

Operation  
Move Email

Operation to be done

**Authentication**

hostname  
hostname  
Hostname/computer name or IP address of mysql server host. e.g. localhost

portNumber  
993  
Port number for email server. e.g. 993

username  
your\_name@your\_domain.com  
Username for email Id

password  
secrets.TOKEN  
Password for username e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

domainName  
gmail.com  
Domain name for email Id

keyStorePath  
keyStorePath  
Keystore Path of email server keystore

keyStorePassword  
secrets.TOKEN  
keyStorePassword for keystore e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

folderPath  
inbox( any folder/label)  
Folder path in email server

**Input Mapping**

messageId  
= "<1234252.266@mail.gmail.com>"  
messageId of the email to be moved

targetFolderPath  
= junk Emails  
Folder path where email needs to be moved

createTargetFolder  
True  
Boolean to create target folder, if folder doesn't exists. Default false

**Output Mapping**

Result Variable  
Response  
Name of variable to store the response in

Result Expression  
=  
Expression to map the response into process variables

**Error Handling**

Error Expression  
=  
Expression to handle errors. Details in the [documentation](#).

Check problems against: **Zeebe 8.1**

**To move email operation, take the following steps:**

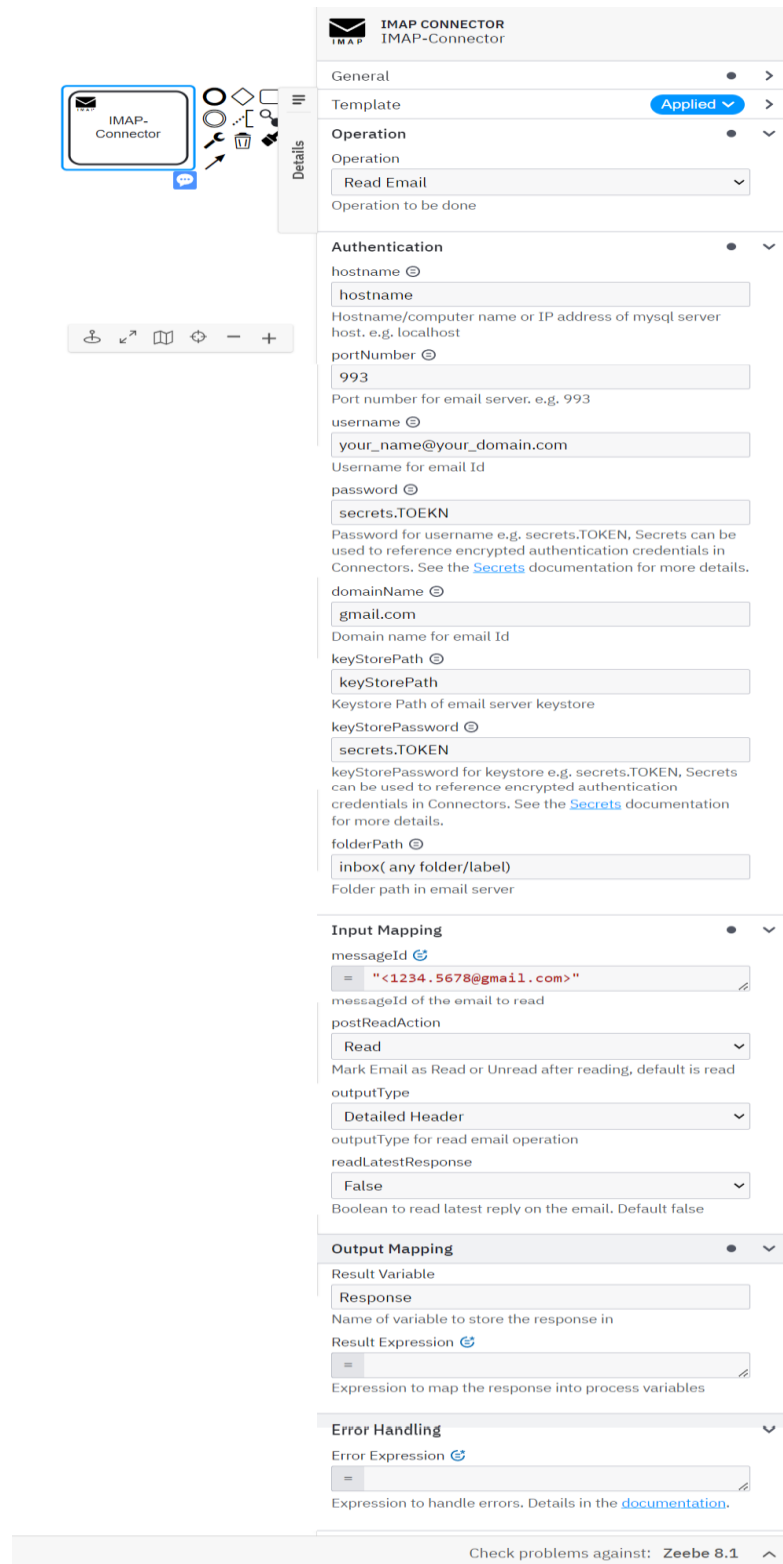
1. In the **Operation** section, set the field value **Operation** as **Move Email**.
2. Set the required parameters and credentials in the **Authentication** section. See the relevant appendix entry to find out more.
3. In the **Input** section, set the field **messageId**, **targetFolderPath**, **createTargetFolder**
4. Set **messageId**, which is unique message id.
5. Set **targetFolderPath**, which will store all moved emails.
6. Set **createTargetFolder**. Set it “True” if targetfolder doesn’t exist. It will create a folder with this name otherwise, if targetFolder exists set it as “False”.

**Move Email operation response**

You can use an output mapping to map the response:

1. Use **Result Variable** to store the response in a process variable.
2. Use Result Expression i.e. FEEL to map the response into process.

## 6. OPERATION – READ EMAIL



**IMAP CONNECTOR**  
IMAP-Connector

General

Template **Applied**

**Operation**

Operation  
Read Email  
Operation to be done

**Authentication**

hostname  
hostname  
Hostname/computer name or IP address of mysql server  
host, e.g. localhost

portNumber  
993  
Port number for email server, e.g. 993

username  
your\_name@your\_domain.com  
Username for email Id

password  
secrets.TOKEN  
Password for username e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

domainName  
gmail.com  
Domain name for email Id

keyStorePath  
keyStorePath  
Keystore Path of email server keystore

keyStorePassword  
secrets.TOKEN  
keyStorePassword for keystore e.g. secrets.TOKEN, Secrets can be used to reference encrypted authentication credentials in Connectors. See the [Secrets](#) documentation for more details.

folderPath  
inbox( any folder/label)  
Folder path in email server

**Input Mapping**

messageId  
=<1234.5678@gmail.com>  
messageId of the email to read

postReadAction  
Read  
Mark Email as Read or Unread after reading, default is read

outputType  
Detailed Header  
outputType for read email operation

readLatestResponse  
False  
Boolean to read latest reply on the email. Default false

**Output Mapping**

Result Variable  
Response  
Name of variable to store the response in

Result Expression  
=  
Expression to map the response into process variables

**Error Handling**

Error Expression  
=  
Expression to handle errors. Details in the [documentation](#).

Check problems against: **Zeebe 8.1**

**To read email operation, take the following steps:**

1. In the **Operation** section, set the field value **Operation** as **Read Email**.
2. Set the required parameters and credentials in the **Authentication** section.
3. In the **Input Mapping** section, set the field messageId, postReadAction, outputType, readLatestResponse.
4. Set **messageId**, which is unique message id.
5. Set **postReadAction** as Read if you want to flag it as a “Read” after reading an email or “Unread” otherwise. Default is “Read”.
6. **outputType** can be selected as Message Id or Header Details as per requirements.
7. Set readLatestResponse as “True” if you want to read latest emails otherwise “False”.

**Read Email operation response**

You can use an output mapping to map the response:

- Use **Result Variable** to store the response in a process variable. Response is based on what one selects in outputType i.e. messageIds or headerDetails.  
Message Ids - List of email message ids.  
headerDetails - List of maps of email header information related to - subject, size, received date. Etc.
- Use Result Expression i.e. FEEL to map the response into process.

## Appendix & FAQ

### How can I authenticate my Connector?

The **IMAP Connector** needs the credentials for connection. Hostname (host) – of the server where database is hosted, Port (port) – on which email server is running, Username (username) – User with proper privilege for operation and Password (password) – User password, which need to be saved as a Token in Secret vault and input can be provided as: secrets.TOKEN\_NAME

### What kind of attributes are needed to perform “Move Email” Operation?

- The required attributes are messageId, targetFolderPath, createTargetFolder
- Set messageId, which is unique message id.
- Set targetFolderPath, which will store all moved emails.
- Set createTargetFolder. Set it “True” if targetfolder doesn’t exist. It will create a folder with this name otherwise, if targetFolder exists set it as “False”.

### What is sortBy input parameter?

- sortBy input is map with keys- sortOn and order
- sortOn key basically indicate on what basis one want to sort there emails. It can have values like - size, from, subject, message number, messageID, sent date or received date.
- “order” key describe ascending and descending order. It can have value like “asc” or “dsc”.

### What is filters input parameter?

- Filters input is Map with keys – filter, logicalOperator and filterList.

- Filter key's value is a Map with keys- filter, operator.
- 'filter' – Specify the condition on which filter is performed. Following are some of the attributes one can apply condition on.  
"size", "flag", "body", "from", "header", "subject", "recipient", "messageNumber", "messageId", "sentDate" and "receivedDate".
- operator – supported operators are [ =, ==, equals, <>, not equals, <, less than, >, greater than, <=, less than or equals, >=, greater than or equals, like, in, is, not in, starts with, ends with ]
- logicalOperator key's value can be OR, AND or NOT.