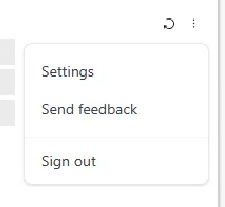
## Folder Pane

In this pane, the user is able to see their folders and items stored in their OTCS server after they authenticated. There is also a settings page that accessible from clicking the 3 dots icon, in here the user can edit their addin settings such as their OTCS API url and mandatory filing configuration.



There also a queue page where it will store all the pending emails which will be send to their OTCS server. The user can retry the upload by clicking the retry button. It also have the functionality to change the upload path by double clicking an email then select the desired location.

## Filing Pane

This pane will allow the user to select multiple folder, then after they click send the composed email will be stored inside the queue with their corresponding filing location.

## Filing Features

### File Email via Folder Pane

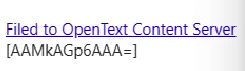
The user can upload both single or multiple selected emails by clicking the upload icon when hovering a folder.



### File Email via Send

If mandatory filing enables, user need to select a filing location before email send. Else the addin will block the email sending out.

Users need to select at least one filing location before email send. After the verification is completed, the email will be processed with a footer inserted at the bottom of the email body. The footer contains a hyperlink where the email filed to and an GUID(Combination of 6 first chars and 6 last chars of MessageId) for each filing records.



# Back-End Architecture

## Authentication

Everytime the client send a request (except login) the backend will check if theres any OTCSTicket in the header, if no then return "NEED\_REAUTHENTICATE" message response, if yes then check if it stored in the DB and if no then return the same message "NEED\_REAUTHENTICATE", if found then call the OTCS API url with corresponding user configuration along with the ticket.

## Additional Tables

* Emailfiling\_ClientConfiguration  
  To store each of the user configuration
* Emailfiling\_ClientInfo  
  Store each of the registered client
* Emailfiling\_EmailProperties  
  Store each of the uploaded email properties (for filing purpose)
* Emailfiling\_FeedbackEmail  
  Store the recipient email when the user want to send a feedback
* Emailfiling\_FilingQueue  
  Store the properties of pending emails
* Emailfiling\_FilingRecords  
  Store the pending email
* Emailfiling\_OTCSTicket  
  Store the OTCS ticket for each user

## API Response

For each response will be returning both **Message** and **Data** property.

* Message (String)  
  This is for to let the client know whether the request is succes or not, need to reauthenticate, and so on.
* Data (dynamic)  
  This can be null depending the request. It will hold the data that the request ask to.