

# Documentation for EWS Related Scripts

Katerina Chinnappan

March 11, 2019

Currently, this script uses the **exchangelib** python library and Microsoft Outlook account: testtask123@outlook.com/123testtask. Tests are done based on the emails in this account's inbox.

## 1 Project Organization

The scripts are organized in the following way:

- unit tests in **test\_ews.py** in root directory
- actual script (**ews.py**) and helper scripts in root/src/ directory
- documentation in root/docs/ directory

## 2 EWS Script & Helper Scripts

### **start.py**

**start.py** script starts the **ews.py** script.

---

```
ews.start_ews()
```

---

This executes the main **ews.py** script.

### **initial\_setup.py**

This small script initializes some common variables which are accessed from other scripts.

### **ews.py**

This is the main script. It is divided into 5 functions. Here is a brief overview of each function:

---

```
def start_ews()
```

---

The **start\_ews** function is invoked by **start.py** script. It calls the rest of the functions in correct order.

---

```
def get_emails(account)
```

---

The **get\_emails** function accepts an *Account* type parameter which is accessed from the **initial\_setup.py** script. All emails from the inbox are fetched. Certain fields from the emails are extracted. Domains from the emails are also extracted and are appended to a list (removing all duplicated and sorting them in alphabetical order). Function **create\_json\_file** is called and is passed extracted email information.

---

```
def create_json_file(sender, subject, time_received, email_size,
                    file_name_counter)
```

---

.json files with received data are created in the same directory as script. A counter which is incremented with every round is added to the end of the filename making sure that two identical email addresses have a unique filename.

---

```
def create_folders(domains,account)
```

---

**create\_folders** function creates the processed folder and all folders with the domain extracted from emails as the folder name. If the folders already exists, they are not created and a message is returned to the console.

---

```
def copy_and_move_emails(domains,account)
```

---

**copy\_and\_move\_emails** function copies ALL emails from inbox to processed folder. All emails in inbox are moved to the created subfolders (created folder with the domain as folder name)

### **move\_emails.py**

This is a helper script. Moves all emails from processed folders back to inbox for testing purposes.

### **delete\_json\_files.py**

This is a helper script. Deletes all .json files in the same directory as script for testing purposes.

### **cleanup.py**

This is a helper script. Deletes all subfolders and emails in subfolders of inbox.

### 3 Notes

Some related notes concerning the scripts:

- currently, the processed folder accepts all email copies from inbox – if two identical emails are copied to processed - the folder stores all copies. This could be solved by comparing the contents of emails and preventing identical emails getting copied twice.
- it is possible to also store the email content in the json file, however, by default, the script doesn't store the email content due to the fact that the content of the email could be very long. There is an option in the code to add the email content to the json file.
- another way to store email content in json files could be to determine some size threshold and then check the size of the email and determine if the email content should be saved in the json file or not.
- currently, the tests are very simple, comparing mock-up data with returned data from functions. The tests could be extended to test more details like file size, file content, etc.

### 4 Consideration of how this script could be extended to a general type of email service providers

In order for this script to be ported to be used by specific email server providers, the common functionality like opening an inbox would need to be generic/common for all email server providers. Common variables will also need to be defined instead of using specific email server provider variables and those would then pin-point to the appropriate variables depending on what email server is currently used. Using if/else blocks will also help to filter and determine to which email-server to establish a connection with.