## Contents:

1 Features ............................................................... 3

2 Installation ............................................................ 5
   2.1 OS X & Linux: ................................................. 5
   2.2 Windows: ...................................................... 5

3 Creating EWS Object .................................................. 7
   3.1 Exchange Search Multi-Threaded: ......................... 8

4 Using Provided Methods ............................................... 9

5 Access Classes Directly ............................................ 11

6 Release History ....................................................... 13

7 Meta ........................................................................... 15

8 Contributing ............................................................. 17
   8.1 Core ............................................................... 17
   8.2 Endpoints ......................................................... 18
   8.3 Service Classes .................................................. 25
   8.4 Utilities ............................................................ 28

Python Module Index ...................................................... 29

Index ............................................................................. 31
**py-ews**

A Python package to interact with Exchange Web Services

**py-ews** is a cross platform python package to interact with both Exchange 2010 to 2019 on-premises and Exchange Online (Office 365).

This package will wrap all Exchange Web Service endpoints, but currently is focused on providing eDiscovery endpoints.
Features

**py-ews** has the following notable features in it’s current release:

- Autodiscover support
- Delegation support
- Impersonation support
- OAUth2 support
- Retrieve all mailboxes that can be searched based on credentials provided
- Search a list of (or single) mailboxes in your Exchange environment using all supported search attributes
- Delete email items from mailboxes in your Exchange environment
- Retrieve mailbox inbox rules for a specific account
- Find additional hidden inbox rules for a specified account
- Plus more supported endpoints

Currently this package supports the following endpoint’s:

- `AddDelegate`
- `ConvertId`
- `CreateFolder`
- `CreateItem`
- `DeleteFolder`
- `DeleteItem`
- `ExecuteSearch`
- `ExpandDL`
- `FindFolder`
- `FindItem`
• GetAttachment
• GetHiddenInboxRules
• GetInboxRules
• GetItem
• GetSearchableMailboxes
• GetServiceConfiguration
• GetUserSettings
• ResolveNames
• SearchMailboxes
• SyncFolderHierarchy
• SyncFolderItems
CHAPTER 2

Installation

2.1 OS X & Linux:

```
pip install py-ews
```

2.2 Windows:

```
pip install py-ews
```
When instantiating the \texttt{EWS} class you will need to provide credentials which will be used for all methods within the \texttt{EWS} class.

```python
from pyews import EWS
ews = EWS(
    'myaccount@company.com',
    'Password1234'
)
```

If you would like to use an alternative \texttt{EWS} URL then provide one using the \texttt{ews_url} parameter when instantiating the \texttt{EWS} class.

```python
from pyews import EWS
ews = EWS(
    'myaccount@company.com',
    'Password1234',
    ews_url='https://outlook.office365.com/autodiscover/autodiscover.svc'
)
```

If you would like to specify a specific version of Exchange to use, you can provide that using the \texttt{exchange_version} parameter. By default \texttt{pyews} will attempt all Exchange versions as well as multiple static and generated \texttt{EWS} URLs.

Finally, if you would like to \texttt{impersonate_as} a specific user you must provide their primary SMTP address when instantiating the \texttt{EWS} class object:

```python
from pyews import EWS
ews = EWS(
    'myaccount@company.com',
    'Password1234',
    impersonate_as='myotheraccount@company.com'
)
```
3.1 Exchange Search Multi-Threading

You can also specify `multi_threading=True` and when you search mailboxes we will use multi-threading to perform the search.
Using Provided Methods

Once you have instantiated the EWS class with your credentials, you will have access to pre-exposed methods for each endpoint. These methods are:

- `get_service_configuration`
- `get_searchable_mailboxes`
- `get_user_settings`
- `resolve_names`
- `execute_ews_search`
- `execute_outlook_search`
- `get_inbox_rules`
- `get_hidden_inbox_rules`
- `get_item`
- `get_attachment`
- `sync_folder_hierarchy`
- `sync_folder_items`
- `create_item`
- `delete_item`
- `search_and_delete_message`
- `get_domain_settings`
- `find_items`
- `search_mailboxes_using_find_item`
- `create_search_folder`
- `find_search_folder`
• delete_search_folder
In some cases you may want to skip using the EWS interface class and build your own wrapper around py-ews. To do this, you must first import the Authentication class and provide credential and other details before invoking a desired endpoint. Below is an example of this:

```python
from pyews import Authentication, GetSearchableMailboxes

Authentication(
    'myaccount@company.com',
    'Password1234'
)

reference_id_list = []
for mailbox in GetSearchableMailboxes().run():
    reference_id_list.append(mailbox.get('reference_id'))
    print(mailbox)
```

As you can see, you must instantiate the Authentication class first before calling an endpoint. By the way, you can import all endpoints directly without using the EWS interface.

For more examples and usage, please refer to the individual class documentation

• Endpoint
CHAPTER 6

Release History

• 1.0.0
  – Initial release of py-ews and it is still considered a work in progress
• 2.0.0
  – Revamped logic and overhauled all endpoints and classes
• 3.0.0
  – Refactored completely - this can be considered a new version
Meta

Josh Rickard – @MSAdministrator
Distributed under the MIT license. See LICENSE for more information.
8.1 Core

8.1.1 Core

The Core class is inherited by all other classes within py-ews. This class controls logging as well as parsing of EWS SOAP request responses.

class pyews.core.core.Core
    The Core class inherits logging and defines required authentication details as well as parsing of results

    camel_to_snake(s)

    parse_response(soap_response, namespace_dict=None)
        parse_response is standardized to parse all soap_responses from EWS requests

        Args: soap_response (BeautifulSoup): EWS SOAP response returned from the Base class namespace_dict (dict, optional): A dictionary of namespaces to process. Defaults to None.

        Returns: list: Returns a list of dictionaries containing parsed responses from EWS requests.

8.1.2 Authentication

The Authentication class is used to configure all communications with Exchange. Additionally, this class contains settings which are used when generating and making SOAP requests.
This class defines the authentication credentials, ews_url(s) to attempt, exchange versions, impersonation details, and more.

class pyews.core.authentication.Authentication

8.1.3 ExchangeVersion

This documentation provides details about the ExchangeVersion class within the pyews package.

class pyews.core.exchangeversion.ExchangeVersion(version)

Used to validate compatible Exchange Versions across multiple service endpoints

Examples: To determine if a version number is a valid ExchangeVersion then would pass in the value when instantiating this object:


static valid_version(version)

Determines if a string version name is in list of accepted Exchange Versions

Args: version (str): String used to determine if it is an acceptable Exchange Version

Returns: bool: Returns either True or False if the passed in version is an acceptable Exchange Version

8.1.4 Endpoints

This documentation provides details about the set and generated API endpoints used within the pyews package.

class pyews.core.endpoints.Endpoints(domain=None)

Endpoints provides and generates endpoints to attempt EWS requests against.

get()

8.2 Endpoints

This documentation provides details about the available Exchange Web Services endpoints within the pyews package.

All endpoints inherit from either the Autodiscover or Operation classes. These classes make extensibility much easier and allows users of this package to define new endpoints easily.

8.2.1 AddDelegate

This documentation provides details about the AddDelegate class within the pyews package.

This class is used to add delegates to a provided mailbox.

class pyews.endpoint.adddelegate.AddDelegate(target_mailbox, delegate_to, inbox_permissions=None, calendar_permissions=None, contacts_permissions=None)

Adds the delegate_to to the provided target_mailbox with specific permission sets.
Args:  
- target_mailbox (str): The mailbox in which permissions are added  
- delegate_to (str): The account that the permissions are associated with  
- inbox_permissions (str, optional): Mailbox permission set value. Defaults to None.  
- calendar_permissions (str, optional): Calendar permission set value. Defaults to None.  
- contacts_permissions (str, optional): Contacts permission set value. Defaults to None.

RESULTS_KEY = 'Items'

8.2.2 ConvertId

This documentation provides details about the ConvertId class within the pyews package.

This class is used to convert item and folder ids from one format to another.

class pyews.endpoint_convertid.ConvertId(user, item_id, id_type, convert_to)
Takes a specific user, item_id, id_type, and the desired format to convert to as inputs.

Args:  
- user (str): The mailbox that the ID is associated with  
- item_id (str): The item ID  
- id_type (str): The Item ID type  
- convert_to (str): The format to convert the Item ID to

Raises:  
- UnknownValueError: One or more provided values is unknown

ID_FORMATS = ['EntryId', 'EwsId', 'EwsLegacyId', 'HexEntryId', 'OwaId', 'StoreId']

RESULTS_KEY = '@Id'

8.2.3 CreateFolder

This documentation provides details about the CreateFolder class within the pyews package.

This class is used to create folders (specifically focused on search folders) in the specified user mailbox.

class pyews.endpoint_createfolder.CreateFolder(search_string, search_folder=True, display_name='Search Folder', search_all_folders=True, base_folder='inbox', traversal='Deep')
Creates a Folder. Default behavior is to create a Search Folder (eDiscovery Search Folder).

Args:  
- search_string (str): A search string which is converted to a Search Filter  
- search_folder (bool, optional): Creates a Search Folder. Defaults to True.  
- display_name (str, optional): The display name of the search folder. Defaults to 'Search Folder'.  
- search_all_folders (bool, optional): The display of the search folder should look at. Defaults to True.  
- base_folder (str, optional): The base folder the search folder should look at. Defaults to 'inbox'.  
- traversal (str, optional): The traversal type. Options are Deep and Shallow. Defaults to 'Deep'.

BASE_SHAPES = ['IdOnly', 'Default', 'AllProperties']

BODY_TYPES = ['Best', 'HTML', 'Text']

RESULTS_KEY = 'CreateFolderResponseMessage'

SEARCH_FILTERS = ['Contains', 'Excludes', 'Exists', 'IsEqualTo', 'IsNotEqualTo', 'IsGreaterThan', 'IsGreaterThanOrEqualTo', 'IsLessThan', 'IsLessThanOrEqualTo']

soap()
### 8.2.4 CreateItem

This documentation provides details about the CreateItem class within the pyews package.

This class is used to create items in a users mailbox.

```python
class pyews.endpoint.createitem.CreateItem(type='message', message_disposition='SendAndSaveCopy', save_item_to='sentitems', **kwargs)
```

Creates an e-mail message

**Args:**
- `type` (str, optional): The type of item to create. Defaults to 'message'.
- `message_disposition` (str, optional): The action to take when the item is created. Defaults to 'SendAndSaveCopy'.
- `save_item_to` (str, optional): Where to save the created e-mail message. Defaults to 'sentitems'.

**BODY_TYPES =** ['Best', 'HTML', 'Text']

**MESSAGE_DISPOSITION =** ['SaveOnly', 'SendOnly', 'SendAndSaveCopy']

**RESULTS_KEY =** 'Items'

### 8.2.5 DeleteFolder

This documentation provides details about the DeleteFolder class within the pyews package.

This class is used to delete folders (specifically focused on search folders) from the specified user mailbox.

```python
class pyews.endpoint.deletefolder.DeleteFolder(folder_id)
```

**RESULTS_KEY =** 'Folders'

### 8.2.6 DeleteItem

This documentation provides details about the DeleteItem class within the pyews package.

This class is used to delete items (typically mailbox items) from the specified user mailbox.

```python
class pyews.endpoint.deleteitem.DeleteItem(item_id, delete_type='MoveToDeletedItems')
```

Deletes the provided Item ID from an Exchange store.

**Args:**
- `item_id` (str): The Item ID to delete
- `delete_type` (str, optional): The delete type when deleting the item. Defaults to 'MoveToDeletedItems'.

**DELETE_TYPES =** ['HardDelete', 'SoftDelete', 'MoveToDeletedItems']

### 8.2.7 ExecuteSearch

This documentation provides details about the ExecuteSearch class within the pyews package.

This class is used to execute a search as the authenticated user but acting like an Outlook mail client.

```python
class pyews.endpoint.executesearch.ExecuteSearch(query, result_row_count='25', max_results_count='-1')
```

Executes a search as an Outlook client for the authenticated user.
Args: query (str): The query to search for. result_row_count (str, optional): The row count of results. Defaults to '25'. max_results_count (str, optional): The max results count. -1 equals unlimited. Defaults to '-1'.

RESULTS_KEY = 'Items'

8.2.8 ExpandDL

This documentation provides details about the ExpandDL class within the pyews package. This class is used to expand a provided distribution list / group.

class pyews.endpoint.expanddl.ExpandDL(user)
    Expands the provided distribution list
    Args: user (str): The distribution list to expand
    RESULTS_KEY = 'Items'

8.2.9 FindFolder

This documentation provides details about the FindFolder class within the pyews package. This class is used to find (search) for a search folder in a mailbox.

class pyews.endpoint.findfolder.FindFolder(folder_id='searchfolders', traversal='Shallow')

BASE_SHAPES = ['IdOnly', 'Default', 'AllProperties']
BODY_TYPES = ['Best', 'HTML', 'Text']
RESULTS_KEY = 'Folders'
SEARCH_FILTERS = ['Contains', 'Excludes', 'Exists', 'IsEqualTo', 'IsNotEqualTo', 'IsGreaterThan', 'IsGreaterThanOrEqualTo', 'IsLessThan', 'IsLessThanOrEqualTo']

8.2.10 FindItem

This documentation provides details about the FindItem class within the pyews package. This class is used to find (search) an item from a mailbox.

class pyews.endpoint.finditem.FindItem(query_string, distinguished_folder_name='inbox', base_shape='AllProperties', include_mime_content=True, body_type='Best', traversal='Shallow', reset_cache=False, return_deleted_items=True, return_highlight_terms=True)

Retrieves results from a query string

Args: item_id (str): The item id you want to get information about. change_key (str, optional): The change key of the item. Defaults to None. base_shape (str, optional): The base shape of the returned item. Defaults to 'AllProperties'. include_mime_content (bool, optional): Whether or not to include MIME content. Defaults to True. body_type (str, optional): The item body type. Defaults to 'Best'.

8.2. Endpoints
**8.2.11 GetAttachment**

This documentation provides details about the GetAttachment class within the pyews package. This class is used to retrieve an attachment object using a provided attachment id.

```python
class pyews.endpoint.getattachment.GetAttachment(attachment_id)
    Retrieves the provided attachment_id
    Args: attachment_id (str): The attachment_id to retrieve.
    RESULTS_KEY = 'Attachments'
```

**8.2.12 GetHiddenInboxRules**

This documentation provides details about the GetHiddenInboxRules class within the pyews package. This class is used to get hidden mailbox rules from the specified user mailbox.

```python
class pyews.endpoint.gethiddeninboxrules.GetHiddenInboxRules
    FindItem EWS Operation attempts to find hidden inbox rules based on ExtendedFieldURI properties.
    RESULTS_KEY = 'Items'
```

**8.2.13 GetInboxRules**

This documentation provides details about the GetInboxRules class within the pyews package. This class is used to get mailbox rules from the specified user mailbox.

```python
class pyews.endpoint.getinboxrules.GetInboxRules(user=None)
    Retrieves inbox rules for the authenticated or specified user.
    Args: user (str, optional): The user to retrieve inbox rules for. Defaults to authenticated user.
    RESULTS_KEY = 'GetInboxRulesResponse'
```

**8.2.14 GetItem**

This documentation provides details about the GetItem class within the pyews package. This class is used to retrieve an item from a mailbox.
class pyews.endpoint.getitem.GetItem(item_id, change_key=None, base_shape='AllProperties', include_mime_content=True, body_type='Best')

Retrieves details about a provided item id

Args:
item_id (str): The item id you want to get information about. change_key (str, optional): The change key of the item. Defaults to None. base_shape (str, optional): The base shape of the returned item. Defaults to 'AllProperties'. include_mime_content (bool, optional): Whether or not to include MIME content. Defaults to True. body_type (str, optional): The item body type. Defaults to 'Best'.

BASE_SHAPES = ['IdOnly', 'Default', 'AllProperties']
BODY_TYPES = ['Best', 'HTML', 'Text']
RESULTS_KEY = 'Items'

8.2.15 GetSearchableMailboxes

This documentation provides details about the GetSearchableMailboxes class within the pyews package.

This class is used to retrieve all searchable mailboxes that the user configuration has access to search.

class pyews.endpoint.getsearchablemailboxes.GetSearchableMailboxes(search_filter=None, expand_group_membership=True)

Retrieves all searchable mailboxes the authenticated user has access to search.

Args:
search_filter (str, optional): A search filter. Typically used to search specific distribution groups. Defaults to None. expand_group_membership (bool, optional): Whether or not to expand group memberships. Defaults to True.

RESULTS_KEY = 'SearchableMailbox'

8.2.16 GetServiceConfiguration

This documentation provides details about the GetServiceConfiguration class within the pyews package.

This class is used to retrieve one or more service configurations.

class pyews.endpoint.getserviceconfiguration.GetServiceConfiguration(configuration_name=None, acting_as=None)

Retrieves service configuration details. Default will attempt to retrieve them all.

Args:
configuration_name (list, optional): The name of one or more configuration items. Defaults to 'MailTips', 'UnifiedMessagingConfiguration', 'ProtectionRules'. acting_as (str, optional): If provided, will attempt to make call using provided user. Defaults to None.

Raises: UnknownValueError: Unknown value was provided

CONFIGURATION_NAMES = ['MailTips', 'UnifiedMessagingConfiguration', 'ProtectionRules']
RESULTS_KEY = 'ServiceConfigurationResponseMessageType'

8.2. Endpoints
8.2.17 GetUserSettings

This documentation provides details about the GetUserSettings class within the pyews package.
This class is used to retrieve user settings.

```python
class pyews.endpoint.getusersettings.GetUserSettings(user=None)
    Retrieves the user settings for the authenticated or provided user.
    Args:
        user (str, optional): A user to retrieve user settings for. Defaults to None.
    RESULTS_KEY = 'UserSettings'
```

8.2.18 ResolveNames

This documentation provides details about the ResolveNames class within the pyews package.
This class is used to resolve a mailbox email address to retrieve details about a user.

```python
class pyews.endpoint.resolvenames.ResolveNames(user=None)
    Resolves the authenticated or provided name
    Args:
        user (str, optional): A user to attempt to resolve. Defaults to None.
    RESULTS_KEY = 'ResolutionSet'
```

8.2.19 SearchMailboxes

This documentation provides details about the SearchMailboxes class within the pyews package.
This class is used to search users mailboxes for a specific item.

```python
class pyews.endpoint.searchmailboxes.SearchMailboxes(query, reference_id, search_scope='All')
    Searches one or more reference id’s using the provided query and search_scope.
    Args:
        query (str): The Advanced Query Syntax (AQS) to search with.
        reference_id (list): One or more mailbox reference Id’s.
        search_scope (str, optional): The search scope. Defaults to ‘All’.
    Raises: ValueError: The provided search scope is unknown.
    RESULTS_KEY = 'SearchPreviewItem'
    SEARCH_SCOPES = ['All', 'PrimaryOnly', 'ArchiveOnly']
```

8.2.20 SyncFolderHierarchy

This documentation provides details about the SyncFolderHierarchy class within the pyews package.
This class is used to retrieve a users mailbox folder hierarchy.

```python
class pyews.endpoint.syncfolderhierarchy.SyncFolderHierarchy(well_known_folder_name=None)
    Retrieve the authenticated users mailbox folder hierarchy.
```
Args: well_known_folder_name (str, optional): The well known folder name. Defaults to all known folder names.

FOLDER_LIST = ['msgfolderroot', 'calendar', 'contacts', 'deleteditems', 'drafts', 'inbox', ... 'quickcontacts', 'conversationhistory', 'todosearch', 'mycontacts', 'imcontactlist', 'peopleconnect', 'favorites']

RESULTS_KEY = 'Changes'

8.2.21 SyncFolderItems

This documentation provides details about the SyncFolderItems class within the pyews package.

This class is used to retrieve individual items from a users mailbox.

class pyews.endpoint.syncfolderitems.SyncFolderItems (folder_id, change_key=None)

Retrieves details about a provided folder id.

Args: folder_id (str): The folder id to retrieve details about. change_key (str, optional): The version key of the folder id. Defaults to None.

FIELD_ACTION_TYPE_MAP = {'create': ['Item', 'Message', 'CalendarItem', 'Contact', 'DistributionList', 'MeetingMessage', 'MeetingRequest', 'MeetingResponse', 'MeetingCancellation', 'Task']}

FIELD_URI_PROPERTIES = ['folder:FolderId', 'folder:ParentFolderId', 'folder:DisplayName', 'folder:UnreadCount', ... 'persona:Children', 'persona:Locations', 'persona:ExtendedProperties', 'persona:PostalAddress', 'persona:Bodies']

RESULTS_KEY = 'Changes'

8.3 Service Classes

The Service sub-package defines the structure and ultimately makes it easier to build SOAP requests for all classes inherited from these modules.

All endpoints inherit from either the Autodiscover or Operation classes. These classes make extensibility much easier and allows users of this package to define new endpoints easily.

8.3.1 Autodiscover

The Autodiscover class is used by any endpoints that want to use the Microsoft Exchange Autodiscover service.

This class defines the SOAP Envelope, namespaces, headers, and body definitions for any class that inherits from it.

Autodiscover class inherits from the Base class and defines namespaces, headers, and body of an Autodiscover SOAP request

pyews.service.autodiscover.Autodiscover.raw_xml
pyews.service.autodiscover.Autodiscover.to

8.3.2 Base

The Base class is used by both the Autodiscover and Operation classes. Each of these classes inherit from this class which defines static values for namespaces and other elements using the ElementMaker factory pattern.

This class defines the SOAP Envelope, namespaces, headers, and body definitions for Autodiscover and Operation classes.
Additionally, the Base class performs all HTTP requests and response validation. The Base class is used by all endpoints and network communications. It defines the base structure of all namespaces. It is inherited by the Autodiscover & Operation classes.

```
pyews.service.base.Base
```

### 8.3.3 Operation

The Operation class is used by all EWS Operations except ones inherited by from Autodiscover. This class defines the SOAP Envelope, namespaces, headers, and body definitions for any class that inherits from it. Operation class inherits from the Base class and defines namespaces, headers, and body of EWS Operation SOAP request.

```
pyews.service.operation.Operation
```

### 8.3.4 Adding Additional EWS Service Endpoints

As I stated above I will continue to add additional EWS Service Endpoints, but I wanted to share documentation on how to add your own support for additional service endpoints.

All endpoints inherit from either `:doc:` autodiscover or `:doc:` operation classes. In order to define a new endpoint you will need to import one of these classes.

```
from pyews.service import Operation
```

Once you have imported the appropriate class (typically this will be Operation) you will then create a new class and inherit from it. In this example I will demonstrate how to build a endpoint for the `GetAppManifests` operation:

```
from pyews.service import Operation

class GetAppManifests (Operation):
    def soap(self):
        pass
```

In order to inherit from `Operation` you must define the class name (which should be the name of the EWS operation) and a single method called `soap`.

The `soap` method will return the Body of a SOAP request using the provided elements.

- M_NAMESPACE
- T_NAMESPACE
- A_NAMESPACE
- WSA_NAMESPACE

By far the most common namespaces you will use will be the M_NAMESPACE and T_NAMESPACE properties.

If we look at the example SOAP request on this page you will see this:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:t="https://schemas.microsoft.com/exchange/services/2006/types"
   xmlns:m="https://schemas.microsoft.com/exchange/services/2006/messages">
```

(continues on next page)
We will only worry about this portion of the XML SOAP request. All other data is managed by the inherited classes:

```xml
<soap:Body >
  <m:GetAppManifests>
    <m:ApiVersionSupported>1.1</m:ApiVersionSupported>
    <m:SchemaVersionSupported>1.1</m:SchemaVersionSupported>
  </m:GetAppManifests>
</soap:Body>
</soap:Envelope>
```

To define this using the provided namespaces will use the provided namespace attributes (e.g. `m:` or `t:`) and build our return object.

This means that our `GetAppManifests` class and `soap` method will look like this:

```python
from pyews.service import Operation
class GetAppManifests(Operation):
    def soap(self):
        return self.M_NAMESPACE.GetAppManifests(
            self.M_NAMESPACE.ApiVersionSupported('1.1'),
            self.M_NAMESPACE.SchemaVersionSupported('1.1')
        )
```

That’s it! Seriously, pretty easy huh?

**Additional details**

If you see a SOAP request element on Microsoft’s site that looks like this:

```xml
<t:AlternateId Format="EwsId" Id="AAMkAGZhN2IxYTA0LWNiNzItN=" Mailbox="user1@example.com"/>
```

Then using our namespaces you would write this as:

```python
self.T_NAMESPACE.AlternateId(Format="EwsId", Id="AAMkAGZhN2IxYTA0LWNiNzItN=", Mailbox="user1@example.com")
```

**Running your class**

Now that we have our newly defined endpoint we can instantiate it and then just call the `run` method.
from pyews import Authentication
from getappmanifests import GetAppManifests

Authentication('username', 'password')
print(GetAppManifests().run())

And we’re done! I hope this helps and if you have any feedback or questions please open a pull request or an issue.

8.4 Utilities

8.4.1 Exceptions

This documentation provides details about Exceptions defined within the pyews package.
Raised when the provided value is unknown or is not in a specified list or dictionary map
pyews.utils.exceptions.UnknownValueError.args

8.4.2 Logger

This documentation provides details about the Logger class within the pyews package.
type(object_or_name, bases, dict) type(object) -> the object’s type
type(name, bases, dict) -> a new type
p
pyews.service.autodiscover.Autodiscover, 25
pyews.service.base.Base, 26
pyews.service.operation.Operation, 26
pyews.utils.exceptions.UnknownValueError, 28
pyews.utils.logger.LoggingBase, 28
Index

A
AddDelegate (class in pyews.endpoint.adddelegate), 18
API_VERSION_MAP (pyews.core.exchangeversion.ExchangeVersion
attribute), 18
args (in module pyews.utils.exceptions.UnknownValueError), 28
Authentication (class in pyews.core.authentication), 18
CreateFolder (class in pyews.endpoint.createfolder), 19
CreateItem (class in pyews.endpoint.createitem), 20

B
BASE_SHAPES (pyews.endpoint.createfolder.CreateFolder
attribute), 19
BASE_SHAPES (pyews.endpoint.findfolder.FindFolder
attribute), 21
BASE_SHAPES (pyews.endpoint.finditem.FindItem attribute), 21
BASE_SHAPES (pyews.endpoint.getitem.GetItem attribute), 23
BODY_TYPES (pyews.endpoint.createfolder.CreateFolder
attribute), 19
BODY_TYPES (pyews.endpoint.createitem.CreateItem
attribute), 20
BODY_TYPES (pyews.endpoint.findfolder.FindFolder
attribute), 21
BODY_TYPES (pyews.endpoint.finditem.FindItem
attribute), 22
BODY_TYPES (pyews.endpoint.getitem.GetItem attribute), 23

C
camel_to_snake () (pyews.core.Core method), 17
CONFIGURATION_NAMES
(pyews.endpoint.getserviceconfiguration.GetServiceConfiguration
attribute), 23
ConvertId (class in pyews.endpoint.convertid), 19
Core (class in pyews.core.core), 17

D
DELETE_TYPES (pyews.endpoint.deleteitem.DeleteItem
attribute), 20
DeleteFolder (class in pyews.endpoint.deletefolder), 20
DeleteItem (class in pyews.endpoint.deleteitem), 20

E
Endpoints (class in pyews.core.endpoints), 18
EXCHANGE_VERSIONS
(pyews.core.exchangeversion.ExchangeVersion
attribute), 18
ExchangeVersion (class in pyews.core.exchangeversion), 18
ExecuteSearch (class in pyews.endpoint.executesearch), 20
ExpandDL (class in pyews.endpoint.expanddl), 21

F
FIELD_ACTION_TYPE_MAP
(pyews.endpoint.syncfolderitems.SyncFolderItems
attribute), 25
FIELD_URI_PROPERTIES
(pyews.endpoint.syncfolderitems.SyncFolderItems
attribute), 25
FindFolder (class in pyews.endpoint.findfolder), 21
FindItem (class in pyews.endpoint.finditem), 21
FOLDER_LIST (pyews.endpoint.syncfolderhierarchy.SyncFolderHierarchy
attribute), 25

G
get () (pyews.core.endpoints.Endpoints method), 18
GetAttachment (class in pyews.endpoint.getattachment), 22
<table>
<thead>
<tr>
<th>Class/Method</th>
<th>Module/Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHiddenInboxRules</td>
<td>pyews.endpoint.gethiddeninboxrules</td>
<td>22</td>
</tr>
<tr>
<td>GetInboxRules</td>
<td>pyews.endpoint.getinboxrules</td>
<td>22</td>
</tr>
<tr>
<td>GetItem</td>
<td>pyews.endpoint.getitem</td>
<td>22</td>
</tr>
<tr>
<td>GetSearchableMailboxes</td>
<td>pyews.endpoint.getsearchablemailboxes</td>
<td>23</td>
</tr>
<tr>
<td>GetServiceConfiguration</td>
<td>pyews.endpoint.getserviceconfiguration</td>
<td>23</td>
</tr>
<tr>
<td>GetUserSettings</td>
<td>pyews.endpoint.getusersettings</td>
<td>24</td>
</tr>
<tr>
<td>ID_FORMATS</td>
<td>pyews.endpoint.convertid.ConvertId</td>
<td>19</td>
</tr>
<tr>
<td>MESSAGE_DISPOSITION</td>
<td>pyews.endpoint.createitem.CreateItem</td>
<td>20</td>
</tr>
<tr>
<td>parse_response()</td>
<td>pyews.core.core.Core</td>
<td>17</td>
</tr>
<tr>
<td>pyews.service.autodiscover.Autodiscover</td>
<td>pyews.service.autodiscover</td>
<td>25</td>
</tr>
<tr>
<td>pyews.service.base.Base</td>
<td>module</td>
<td>26</td>
</tr>
<tr>
<td>pyews.service.operation.Operation</td>
<td>module</td>
<td>26</td>
</tr>
<tr>
<td>pyews.utils.exceptions.UknownValueError</td>
<td>module</td>
<td>28</td>
</tr>
<tr>
<td>pyews.utils.logger.LoggingBase</td>
<td>module</td>
<td>28</td>
</tr>
<tr>
<td>raw_xml</td>
<td>pyews.service.autodiscover</td>
<td>25</td>
</tr>
<tr>
<td>raw_xml</td>
<td>pyews.service.base.Base</td>
<td>26</td>
</tr>
<tr>
<td>raw_xml</td>
<td>pyews.service.operation.Operation</td>
<td>26</td>
</tr>
<tr>
<td>ResolveNames</td>
<td>pyews.endpoint.resolvenames</td>
<td>24</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.addelegate.AddDelegate</td>
<td>20</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.executesearch.ExecuteSearch</td>
<td>21</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.expanddl.ExpandDL</td>
<td>21</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.findfolder.FindFolder</td>
<td>21</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.finditem.FindItem</td>
<td>22</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.getattachment.GetAttachment</td>
<td>22</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.gethiddeninboxrules.GetHiddenInboxRules</td>
<td>22</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.getinboxrules.GetInboxRules</td>
<td>22</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.getitem.GetItem</td>
<td>23</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.getsearchablemailboxes.GetSearchableMailboxes</td>
<td>23</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.getserviceconfiguration.GetServiceConfiguration</td>
<td>23</td>
</tr>
<tr>
<td>RESULTS_KEY</td>
<td>pyews.endpoint.getusersettings.GetUserSettings</td>
<td>24</td>
</tr>
<tr>
<td>SEARCH_FILTERS</td>
<td>pyews.endpoint.createfolder.CreateFolder</td>
<td>19</td>
</tr>
<tr>
<td>SEARCH_FILTERS</td>
<td>pyews.endpoint.findfolder.FindFolder</td>
<td>21</td>
</tr>
<tr>
<td>SEARCH_SCOPES</td>
<td>pyews.endpoint.searchmailboxes.SearchMailboxes</td>
<td>24</td>
</tr>
<tr>
<td>SEARCH_SCOPES</td>
<td>in</td>
<td>24</td>
</tr>
<tr>
<td>soap()</td>
<td>pyews.endpoint.addelegate.AddDelegate</td>
<td>19</td>
</tr>
<tr>
<td>soap()</td>
<td>pyews.endpoint.convertid.ConvertId</td>
<td>19</td>
</tr>
<tr>
<td>soap()</td>
<td>pyews.endpoint.createfolder.CreateFolder</td>
<td>19</td>
</tr>
<tr>
<td>soap()</td>
<td>pyews.endpoint.createitem.CreateItem</td>
<td>20</td>
</tr>
<tr>
<td>soap()</td>
<td>pyews.endpoint.deletefolder.DeleteFolder</td>
<td>20</td>
</tr>
<tr>
<td>soap()</td>
<td>pyews.endpoint.deleteitem.DeleteItem</td>
<td>20</td>
</tr>
</tbody>
</table>
soap() (pyews.endpoint.executesearch.ExecuteSearch method), 21
soap() (pyews.endpoint.expanddl.ExpandDL method), 21
soap() (pyews.endpoint.findfolder.FindFolder method), 21
soap() (pyews.endpoint.finditem.FindItem method), 22
soap() (pyews.endpoint.getattachment.GetAttachment method), 22
soap() (pyews.endpoint.gethiddeninboxrules.GetHiddenInboxRules method), 22
soap() (pyews.endpoint.getinboxrules.GetInboxRules method), 22
soap() (pyews.endpoint.getitem.GetItem method), 23
soap() (pyews.endpoint.getsearchablemailboxes.GetSearchableMailboxes method), 23
soap() (pyews.endpoint.getserviceconfiguration.GetServiceConfiguration method), 23
soap() (pyews.endpoint.getusersettings.GetUserSettings method), 24
soap() (pyews.endpoint.resolvenames.ResolveNames method), 24
soap() (pyews.endpoint.searchmailboxes.SearchMailboxes method), 24
soap() (pyews.endpoint.syncfolderhierarchy.SyncFolderHierarchy method), 25
soap() (pyews.endpoint.syncfolderitems.SyncFolderItems method), 25
SyncFolderHierarchy (class in pyews.endpoint.syncfolderhierarchy), 24
SyncFolderItems (class in pyews.endpoint.syncfolderitems), 25

T

to (in module pyews.service.autodiscover.Autodiscover), 25

V

valid_version() (pyews.core.exchangeversion.ExchangeVersion static method), 18