
pyEchosign Documentation

Release 0.3.1dev

Jens Astrup

Jul 08, 2017

Contents

1	Documentation	3
1.1	pyEchosign package	3
2	Indices and tables	11

WIP - A module to connect to the Echosign API to create, send, and retrieve agreements.

Documentation is hosted at [RTD](#)

pyEchosign package

EchosignAccounts

class EchosignAccount (*access_token: str, **kwargs*)

Saves OAuth Information for connecting to Echosign

access_token

The OAuth Access token to use for authenticating to Echosign

user_id

The ID of the user to specify as the API caller, if not provided the caller is inferred from the token

user_email

The email of the user to specify as the API caller, if not provided the caller is inferred from the token

api_access_point

The API endpoint used as a base for all API calls

get_agreements (*query=None*)

Gets all agreements for the EchosignAccount

Keyword Arguments **query** – (str) A search query to filter results by

Returns: An iterator of *Agreement* objects

get_library_documents ()

Gets all Library Documents for the EchosignAccount

Returns: A list of *Agreement* objects

Agreements

class Agreement

Represents either a created agreement in Echosign, or one built in Python which can be sent through, and created in Echosign.

Parameters `account` (`EchosignAccount`) – An instance of `EchosignAccount`. All Agreement actions will be conducted under this account.

Keyword Arguments

- **fully_retrieved** (`bool`) – Whether or not the agreement has all information retrieved, or if only the basic information was pulled (such as when getting all agreements instead of requesting the specific agreement)
- **echosign_id** (`str`) – The ID assigned to the agreement by Echosign, used to identify the agreement via the API
- **name** (`str`) – The name of the document as specified by the sender
- **status** (`Agreement.Status`) – The current status of the document (OUT_FOR_SIGNATURE, SIGNED, APPROVED, etc)
- **users** (`list[DisplayUser]`) – The users associated with this agreement, represented by `EchosignAccount`
- **files** (`list`) – A list of `TransientDocument` instances which will become the documents within the agreement. This information is not provided when retrieving agreements from Echosign.

account

`EchosignAccount` – An instance of `EchosignAccount`. All Agreement actions will be conducted under this account.

fully_retrieved

`bool` – Whether or not the agreement has all information retrieved, or if only the basic information was pulled (such as when getting all agreements instead of requesting the specific agreement)

echosign_id

`str` – The ID assigned to the agreement by Echosign, used to identify the agreement via the API

name

`str` – The name of the document as specified by the sender

status

`Agreement.Status` – The current status of the document (OUT_FOR_SIGNATURE, SIGNED, APPROVED, etc)

users

`list[DisplayUser]` – The users associated with this agreement, represented by `EchosignAccount`

files

`list` – A list of `TransientDocument` instances which will become the documents within the agreement. This information is not provided when retrieving agreements from Echosign.

class Status

Possible status of agreements

Note: Echosign provides ‘WAITING_FOR_FAXIN’ in their API documentation, so pyEchosign has also included ‘WAITING_FOR_FAXING’ in case that’s just a typo in their documentation. Once it’s

determined which is used, the other will be removed.

ACCEPTED = 'ACCEPTED'

APPROVED = 'APPROVED'

ARCHIVED = 'ARCHIVED'

DELIVERED = 'DELIVERED'

EXPIRED = 'EXPIRED'

FORM = 'FORM'

FORM_FILLED = 'FORM_FILLED'

OTHER = 'OTHER'

OUT_FOR_ACCEPTANCE = 'OUT_FOR_ACCEPTANCE'

OUT_FOR_APPROVAL = 'OUT_FOR_APPROVAL'

OUT_FOR_DELIVERY = 'OUT_FOR_DELIVERY'

OUT_FOR_FORM_FILLING = 'OUT_FOR_FORM_FILLING'

OUT_FOR_SIGNATURE = 'OUT_FOR_SIGNATURE'

RECALLED = 'RECALLED'

SIGNED = 'SIGNED'

WAITING_FOR_AUTHORIZING = 'WAITING_FOR_AUTHORIZING'

WAITING_FOR_FAXIN = 'WAITING_FOR_FAXIN'

WAITING_FOR_FAXING = 'WAITING_FOR_FAXING'

WAITING_FOR_MY_ACCEPTANCE = 'WAITING_FOR_MY_ACCEPTANCE'

WAITING_FOR_MY_ACKNOWLEDGEMENT = 'WAITING_FOR_MY_ACKNOWLEDGEMENT'

WAITING_FOR_MY_APPROVAL = 'WAITING_FOR_MY_APPROVAL'

WAITING_FOR_MY_DELEGATION = 'WAITING_FOR_MY_DELEGATION'

WAITING_FOR_MY_FORM_FILLING = 'WAITING_FOR_MY_FORM_FILLING'

WAITING_FOR_MY_SIGNATURE = 'WAITING_FOR_MY_SIGNATURE'

WIDGET = 'WIDGET'

`Agreement.cancel()`

Cancels the agreement on Echosign. Agreement will still be visible in the Manage page.

`Agreement.delete()`

Deletes the agreement on Echosign. Agreement will not be visible in the Manage page.

Notes

This action requires the 'agreement_retention' scope, which doesn't appear to be actually available via OAuth

Agreement.documents

Retrieve the *AgreementDocuments* associated with this agreement. If the files have not already been retrieved, this will result in an additional request to the API.

Returns: A list of *AgreementDocument*

```
Agreement.send_agreement (agreement_name: str, recipients: typing.List[pyEchosign.classes.users.Recipient],
                          ccs=None, days_until_signing_deadline=0, external_id='',
                          sender_signature_required=False, merge_fields: typing.List[typing.Dict[str, str]] = None, message='')
```

Sends this agreement to Echosign for signature

Parameters

- **agreement_name** – A string for the document name which will appear in the Echosign Manage page, the email to recipients, etc.
- **recipients** – A list of *Recipients*. The order which they are provided in the list determines the order in which they sign.
- **ccs** – (optional) A list of email addresses to be CC'd on the Echosign agreement emails (document sent, document fully signed, etc)
- **days_until_signing_deadline** – (optional) “The number of days that remain before the document expires. You cannot sign the document after it expires” Defaults to 0, for no expiration.
- **external_id** – (optional) “A unique identifier for your transaction... You can use the ExternalID to search for your transaction through [the] API”
- **sender_signature_required** – (optional) (bool) Whether or not a sender signature is required. Defaults to False. If true, the signer will sign first. The additional options of the signer signing last, or sequentially isn't currently supported (because I haven't thought of a clean way to handle providing that info).
- **merge_fields** – (optional) A list of dictionaries, with each one providing the 'fieldName' and 'defaultValue' keys. The field name maps to the field on the document, and the default value is what will be placed inside.
- **message** – (optional) A message which will be displayed to recipients of the agreement

Returns

A namedtuple representing the information received back from the API. Contains the following attributes

agreement_id “The unique identifier that can be used to query status and download signed documents”

embedded_code “Javascript snippet suitable for an embedded page taking a user to a URL”

expiration “Expiration date for autologin. This is based on the user setting, API_AUTO_LOGIN_LIFETIME”

url “Standalone URL to direct end users to”

Raises `ApiError` – If the API returns an error, such as a 403. The exact response from the API is provided.

Documents

Agreement Documents

class AgreementDocument (*echosign_id: str, mime_type: str, name: str, page_count: int, supporting_document=False, field_name=None*)

Represents a document used in an Agreement.

echosign_id

The ID of the Document which can be used to retrieve its file stream

mime_type

The MIME type of the document

name

The name of the document

page_count

The number of pages in the document

supporting_document

Whether or not this document is a “supporting document” as specified by the API

field_name

If a supporting document, what the name is of the supporting document field

Library Documents

class LibraryDocument

Represents a Library Document in Echosign. When pulling all Library Documents, only the echosign_id, template_type, modified_date, name, and scope are available. Accessing all other attributes results in an HTTP request to pull the full document information.

account

EchosignAccount – An instance of *EchosignAccount*. All Agreement actions will be conducted under this account.

echosign_id

str – The ID for this document in Echosign

document

bool – If this LibraryDocument is a document in Echosign

form_field_layer

bool – If this LibraryDocument is a form field layer

modified_date

datetime – The day on which the LibraryDocument was last modified

name

str – The name of the LibraryDocument in Echosign

scope

str – The visibility of this LibraryDocument, either ‘PERSONAL’, ‘SHARED’, or ‘GLOBAL’

GLOBAL = ‘GLOBAL’

PERSONAL = ‘PERSONAL’

SHARED = ‘SHARED’

delete ()

Deletes the LibraryDocument from Echosign. It will not be visible on the Manage page.

document = False

form_field_layer = False

fully_retrieved = False

locale

retrieve_complete_document ()

Retrieves the remaining data for the LibraryDocument, such as locale, status, and security options.

scope = None

Transient Documents

class TransientDocument

A document which can be used in Agreements - is deleted by Echosign after 7 days. The TransientDocument is created in Echosign on instantiation.

Parameters

- **account** – The *EchosignAccount* to be associated with this document
- **file_name** (*str*) – The name of the file
- **file** – The actual file object to upload to Echosign, accepts a stream of bytes.
- **mime_type** – (optional) The MIME type of the file. Echosign will infer the type from the file extension if not provided.

file_name

The name of the file

file

The actual file object to upload to Echosign

mime_type

The MIME type of the file

document_id

The ID provided by Echosign, used to reference it in creating agreements

expiration_date

The date Echosign will delete this document (not provided by Echosign, calculated for convenience)

Users

class DisplayUser (*email: str, **kwargs*)

Maps to the DisplayUserInfo provided by Echosign for agreements fetched in bulk

class Recipient (*email: str, **kwargs*)

Bases: *pyEchosign.classes.users.DisplayUser*

Provides additional attributes to facilitate sending documents to recipients, such as Security Options.

authentication_method

str – “The authentication method for the recipients to have access to view and sign the document” (Echosign API Docs). Available options are ‘NONE’ (string), ‘INHERITED_FROM_DOCUMENT’ or ‘PASSWORD’ or ‘WEB_IDENTITY’ or ‘KBA’ or ‘PHONE’.

password

str – Optional - “The password required for the recipient to view and sign the document”

Any content in between double quotes (“like this”) is taken from the [Echosign API documentation](#).

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

A

ACCEPTED (Agreement.Status attribute), 5
 access_token (EchosignAccount attribute), 3
 account (Agreement attribute), 4
 account (LibraryDocument attribute), 7
 Agreement (class in pyEchosign.classes.agreement), 4
 Agreement.Status (class in pyE-
 chosign.classes.agreement), 4
 AgreementDocument (class in pyE-
 chosign.classes.documents), 7
 api_access_point (EchosignAccount attribute), 3
 APPROVED (Agreement.Status attribute), 5
 ARCHIVED (Agreement.Status attribute), 5
 authentication_method (Recipient attribute), 8

C

cancel() (Agreement method), 5

D

delete() (Agreement method), 5
 delete() (LibraryDocument method), 7
 DELIVERED (Agreement.Status attribute), 5
 DisplayUser (class in pyEchosign.classes.users), 8
 document (LibraryDocument attribute), 7
 document_id (TransientDocument attribute), 8
 documents (Agreement attribute), 5

E

echosign_id (Agreement attribute), 4
 echosign_id (AgreementDocument attribute), 7
 echosign_id (LibraryDocument attribute), 7
 EchosignAccount (class in pyEchosign.classes.account), 3
 expiration_date (TransientDocument attribute), 8
 EXPIRED (Agreement.Status attribute), 5

F

field_name (AgreementDocument attribute), 7
 file (TransientDocument attribute), 8

file_name (TransientDocument attribute), 8
 files (Agreement attribute), 4
 FORM (Agreement.Status attribute), 5
 form_field_layer (LibraryDocument attribute), 7, 8
 FORM_FILLED (Agreement.Status attribute), 5
 fully_retrieved (Agreement attribute), 4
 fully_retrieved (LibraryDocument attribute), 8

G

get_agreements() (EchosignAccount method), 3
 get_library_documents() (EchosignAccount method), 3
 GLOBAL (LibraryDocument attribute), 7

L

LibraryDocument (class in pyE-
 chosign.classes.library_document), 7
 locale (LibraryDocument attribute), 8

M

mime_type (AgreementDocument attribute), 7
 mime_type (TransientDocument attribute), 8
 modified_date (LibraryDocument attribute), 7

N

name (Agreement attribute), 4
 name (AgreementDocument attribute), 7
 name (LibraryDocument attribute), 7

O

OTHER (Agreement.Status attribute), 5
 OUT_FOR_ACCEPTANCE (Agreement.Status at-
 tribute), 5
 OUT_FOR_APPROVAL (Agreement.Status attribute), 5
 OUT_FOR_DELIVERY (Agreement.Status attribute), 5
 OUT_FOR_FORM_FILLING (Agreement.Status at-
 tribute), 5
 OUT_FOR_SIGNATURE (Agreement.Status attribute), 5

P

page_count (AgreementDocument attribute), 7

password (Recipient attribute), 8
PERSONAL (LibraryDocument attribute), 7

R

RECALLED (Agreement.Status attribute), 5
Recipient (class in pyEchosign.classes.users), 8
retrieve_complete_document() (LibraryDocument method), 8

S

scope (LibraryDocument attribute), 7, 8
send_agreement() (Agreement method), 6
SHARED (LibraryDocument attribute), 7
SIGNED (Agreement.Status attribute), 5
status (Agreement attribute), 4
supporting_document (AgreementDocument attribute), 7

T

TransientDocument (class in pyEchosign.classes.documents), 8

U

user_email (EchosignAccount attribute), 3
user_id (EchosignAccount attribute), 3
users (Agreement attribute), 4

W

WAITING_FOR_AUTHORIZING (Agreement.Status attribute), 5
WAITING_FOR_FAXIN (Agreement.Status attribute), 5
WAITING_FOR_FAXING (Agreement.Status attribute), 5
WAITING_FOR_MY_ACCEPTANCE (Agreement.Status attribute), 5
WAITING_FOR_MY_ACKNOWLEDGEMENT (Agreement.Status attribute), 5
WAITING_FOR_MY_APPROVAL (Agreement.Status attribute), 5
WAITING_FOR_MY_DELEGATION (Agreement.Status attribute), 5
WAITING_FOR_MY_FORM_FILLING (Agreement.Status attribute), 5
WAITING_FOR_MY_SIGNATURE (Agreement.Status attribute), 5
WIDGET (Agreement.Status attribute), 5