Adobe E-Signature

**IMPORTANT** !

While running OAuth Process, make sure to follow below points:

1. Correct Client ID and Redirect URI is specified.  
2. Scopes provided in Authorization URL should match exactly with the scopes provided in Adobe Sign application.  
3. Use the correct shard (na1, na2, au1, eu1, jp1) as per the account is being configured.  
4. Remove any spaces provided in the authorization URL (if any).  
5. Check the Syntax of Authorization URL.

Two Way to obtain access token

OAuth (complicated): need browser prompt login (might need use headless browser eg. Selenium or casper to perform login)

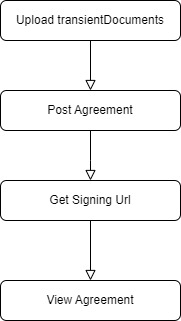
Reference: <https://secure.na1.adobesign.com/public/static/oauthDoc.jsp>

Legacy (generate token at admin account): just plug in the token in web api and can use forever

Reference: <https://secure.na4.adobesign.com/account/accountSettingsPage#pageId::ACCESS_TOKENS>

API link: <https://secure.na1.adobesign.com/public/docs/restapi/v6>

Basic Workflow for eSignature



1. Every agreement sign, we must generate a pdf programatically and perform upload the document (aka. transientDocuments in adobe) through web api.
2. After uploaded transientDocument, the api will return a transientDocID to us, we must use this transientDocID to perform Post Agreement step. We have to define all the data such as agreement name, signature type, state (must be ‘IN\_PROCESS’ in order to send out the agreement for signature etc (refer the json object below)
3. After posted (send out as email) the agreement, the api will return an agreement ID as response, we can use this agreement ID to access the signing Url.

{

"fileInfos": [

{

"transientDocumentId": "CBSCTBABDUAAABACAABAARM8ndFIfvrOW\_dI1hAmOUnEMcw2H8rvjhypm-ypiC6DcwrKIxkFBByCBaJMIiHKNM9TjRqYmIOgCAANSJkdhtfEOP6H\_3ueZGkSphnG283d2HoNJAvvRfCxEr4DwqIMRHbV5yi3TrRX5qufZMbFxioUl1kLWwyz3Ap8l15RoSYfFpE\_YAsLExAqGHsgK05N7LPW\_kZdqV6hor90Lz2v9G\_jnFU7xzE3kuC35EZ90k4pTONvLdn-8jq-W1uMZHv4b252Qiln\_sIy1yydrp2gW4G\_e3mv7w2oI9tNAIENu4dqpwB1I6TBCf1bUOtuFyRhSyMNmp9GJHLZhoxQYyb-Tpa1I5lYecjPw\_x\_2HsTR\_il2-m50scjggJoCVimZk7\_k"

}

],

"name": "Calerie Invoice",

"participantSetsInfo": [

{

"order": 1,

"role": "SIGNER",

"memberInfos": [

{ "email": "jex\_calerie@outlook.com" }

]

}

],

"signatureType": "ESIGN",

"state": "IN\_PROCESS"

}

1. After user signed the agreement, we can call the View Agreement API endpoint to view the signed document (just open the Url under DOUCMNENT type name, not MANAGE).

Pros:

Adobe E-signature provide more robust API, which is good if we use a lot of documentation/function under Adobe ecosystem.

Adobe eSignature offer cheaper pricing plan which required USD 14.99 monthly per business license.

Cons:

Adobe E-Sign API required pre-build agreement in PDF format for upload transientDocument function. Besides, throughout the post agreement step, it must send out the agreement to user email.

For authentication part the OAuth will be difficult to implement in backend code, you may refer the link <https://community.adobe.com/t5/adobe-acrobat-sign-discussions/help-with-oauth2/td-p/7063211>. But we still able to use the legacy method to generate an access token to perform all the api execution.