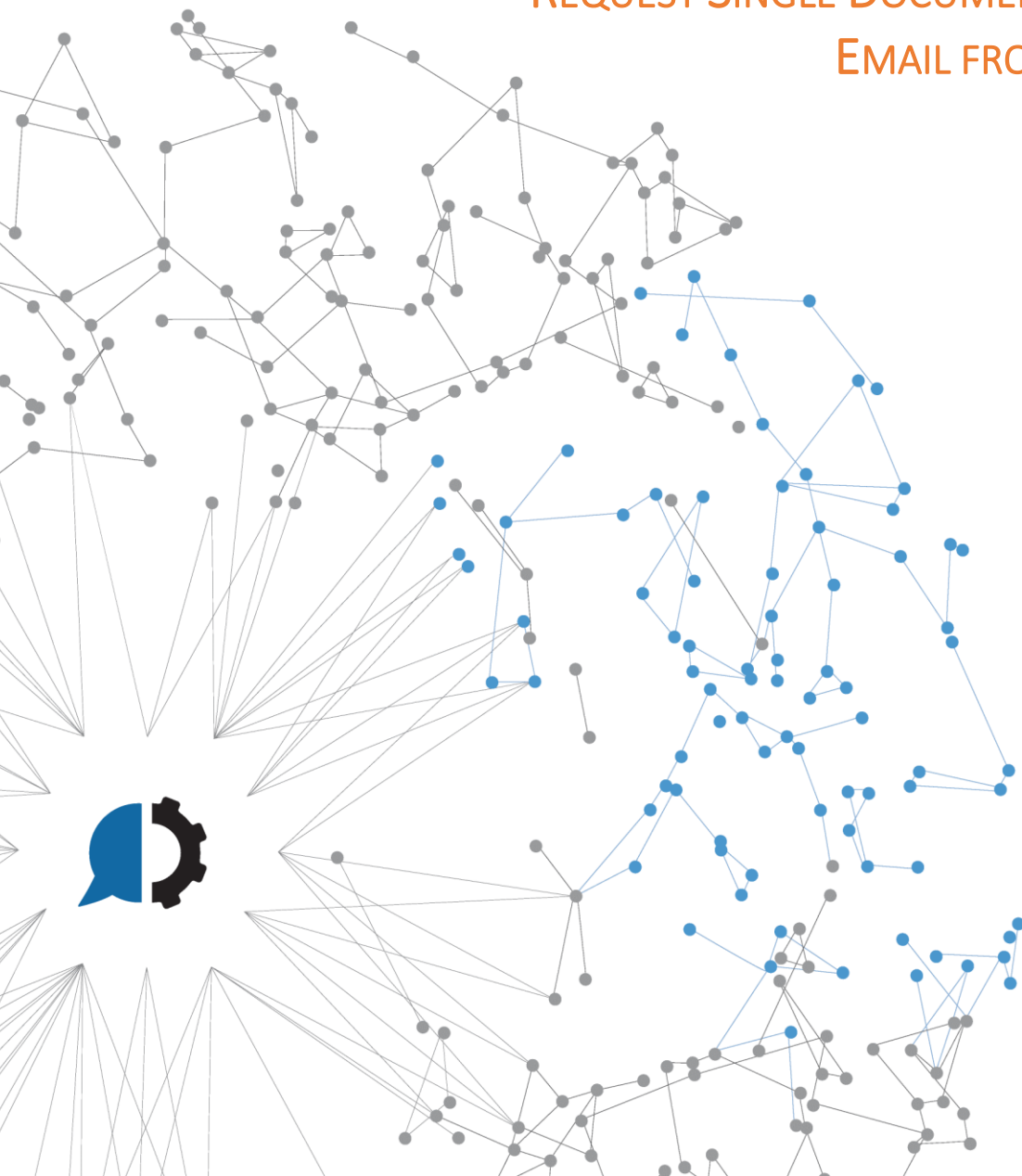




ADOBE SIGN BOT

REQUEST SINGLE DOCUMENT SIGNATURE BY
EMAIL FROM EXCEL INPUT





Content

| | |
|---|----|
| OVERVIEW | 3 |
| PREREQUISITES | 4 |
| HOW TO IMPORT AN AUTOMATE TASK | 7 |
| HOW TO GENERATE THE INPUT FILE | 8 |
| HOW TO EXECUTE THE ADOBE SIGN BOT - REQUEST SINGLE DOCUMENT SIGNATURE BY EMAIL FROM EXCEL INPUT | 9 |
| APPENDIX A – EXCEL INPUT FILE WITH RESULTS AND ADOBE SIGN VIEW RESULT | 12 |
| APPENDIX B - TROUBLESHOOTING | 13 |



OVERVIEW

This how to guide introduces the *Adobe Sign Bot - Request Single Document Signature by Email from Excel INPUT*. Instead of spending precious time and resources on repetitive, manual tasks, let Automate go to work and streamline your IT and business processes with robotic process automation. Automate can take over the manual steps which greatly reduces repetitive activities and improves the quality and consistency of the work. The top processes being automated are report generation, file movement, data import and export, and scheduling batch processing.

The ***Adobe Sign Bot - Request Single Document Signature by Email from Excel INPUT*** is based in the Adobe Sign REST API. It creates an ***Agreement***, sends it out for signatures and returns the agreement ID in Adobe Sign by executing the ***“/agreements”*** HTTP **POST** method with the data provided in the INPUT Excel file and updates it with the resulting of the execution. Bot uses the API Token authentication method to avoid using username and password. *The Token is associated to the user account used for getting the Authorization Code*, so you will need a BOT for each user account that needs to send signature requests. Carefully check the [PREREQUISITES](#) and the [HOW TO GENERATE THE INPUT FILE](#) sections to get detailed information.

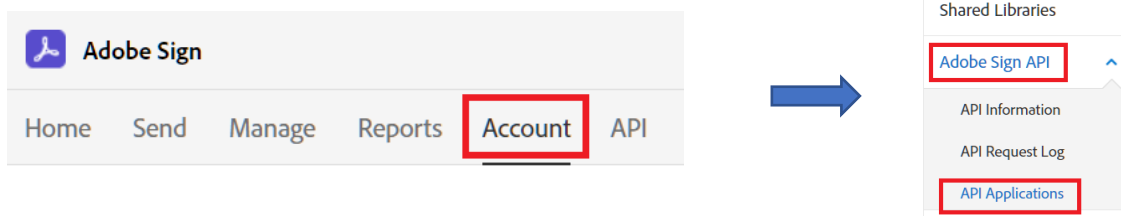
*Is important to note that **Adobe Sign only supports APIs Integration with its “Enterprise Plan”**.*

There is also available an [ADOBE SIGN BOT - GENERATE AND REFRESH TOKEN](#) (based in the OAuth Authorization Code Workflow) that will provide an easy way for first creation and refreshing of the Token for this bot to work. By default, the Token expires after 1 hour.

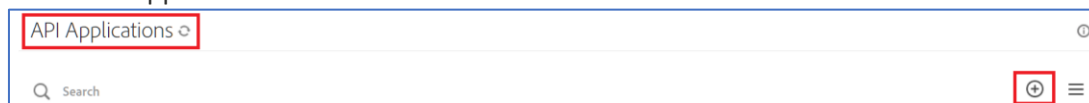


PREREQUISITES

- **Automate:** The Adobe Sign Bot depends on Automate software in order to work. The minimal supported versions are:
 - [Automate Ultimate 11.2](#)
 - [Automate Plus 11.2](#)
 - [Automate Desktop 11.2](#)
- **Automate Markup Language file (.AML):** The primary file type used in Automate which contains the steps of our Adobe Sign Bot task
- **Adobe Sign Graph:** Our minimal requirements are
 - **Create an application in Adobe Sign on the Application List Page:** You will need to create an application within the Adobe Sign platform to be able to configure the proper scopes and requests for the token that will be used for the connections of other Adobe Sign Bots. You will find it by signing into your Adobe Sign account going to the **ACCOUNT** tab > **Adobe Sign API** > **API Applications** section. You will need to be an administrator, so if you do not have permissions, ask your Adobe Sign Administrator. More information on how to create the App can be find in this [Adobe Sign Link](#).

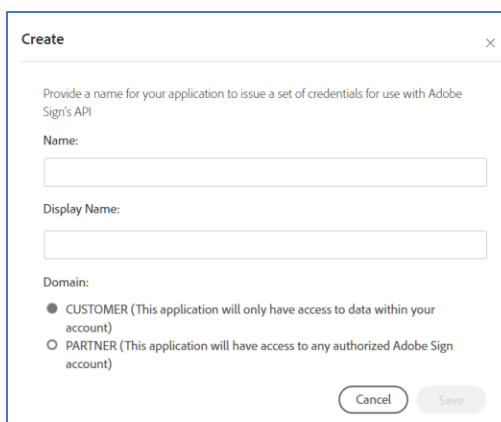


Add an API Application:





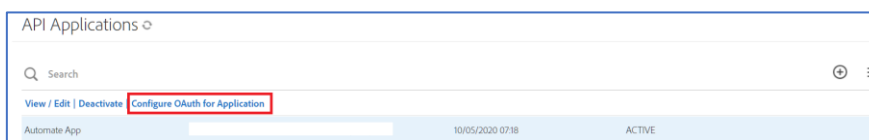
Complete the Name, Display Name
and select the Domain according to
your needs:



The 'Create' dialog box contains the following fields and options:

- Name:** A text input field.
- Display Name:** A text input field.
- Domain:** Two radio button options:
 - ☒ **CUSTOMER** (This application will only have access to data within your account)
 - ☐ **PARTNER** (This application will have access to any authorized Adobe Sign account)
- Buttons:** 'Cancel' and 'Save' buttons at the bottom right.

- **Configure OAuth for your application:** you create your Application, it will appear in the API Applications list. Select yours and click in **"Configure OAuth for Application"**. A configuration form will be opened. Complete it according to your needs. Within this configuration you will find the **Client ID** and **Client Secret** that will be needed for the Token generation (keep them safely). You will find more information about the Authorization Code Grant Workflow [here](#).



The 'API Applications' list shows a table with one application:

| API Applications | | |
|---|------------------|--------|
| Search | | |
| View / Edit / Deactivate Configure OAuth for Application | | |
| Automate App | 10/05/2020 07:18 | ACTIVE |

OAuth config: Set the **Redirect URL** (you can use <https://localhost>) and select the **Scopes** that you will need:



Configure OAuth

Client ID:

Client Secret:

Note: You must keep your Client Secret confidential.

Redirect URI:

Note: The redirectUri specified in your OAuth requests must belong to this list of uris. You can mention multiple uris as comma separated list.

Enabled Scopes

You must enable the scopes that you intend to request through the OAuth protocol. Please limit the scopes that you enable to the minimum set necessary for your application, which is one of the requirements for Certification.

Please [contact support](#) if you need to change which scopes are enabled for your application. ?

Note that only Group Admins can approve OAuth requests that use the ".group" scope modifier, and only Account Admins can approve OAuth requests that use the ".account" scope modifier.

| Enabled? | Scope | Modifier | Description |
|--------------------------|-----------------|----------|---|
| <input type="checkbox"/> | user_read | account | View users in your account |
| <input type="checkbox"/> | user_write | account | Create or manage users within your account |
| <input type="checkbox"/> | user_login | account | Login on behalf of any user in your account |
| <input type="checkbox"/> | agreement_read | account | Access documents & data on behalf of any user in your account |
| <input type="checkbox"/> | agreement_write | account | Manage the status of documents on behalf of any user in your account |
| <input type="checkbox"/> | agreement_send | account | Send documents on behalf of any user in your account |

- **Adobe Sign Token generated with proper permissions scopes for the Bot execution:** Once the application is created and configured you will be able to generate the token for connection. This token must be generated with the proper permissions scopes, for this bot to work you will need the following scopes: **user_login:self**, **agreement_send:account** and **agreement_write:account**.

The scope **"user_login:self"** scope is required for login on behalf of any user in your account. The **"agreement_send:account"** is for sending documents on behalf of any user in your account. And the scope **"agreement_write:account"** is for managing the status of documents on behalf of any user in your account. We also provide in the our [Bot Store](#) an **Adobe Sign Bot - Generate and Refresh Token** that's based in the **Adobe Sign OAuth Authorization Code Workflow** that you can use for this matter. It also has a user guide that better explains this Workflow.



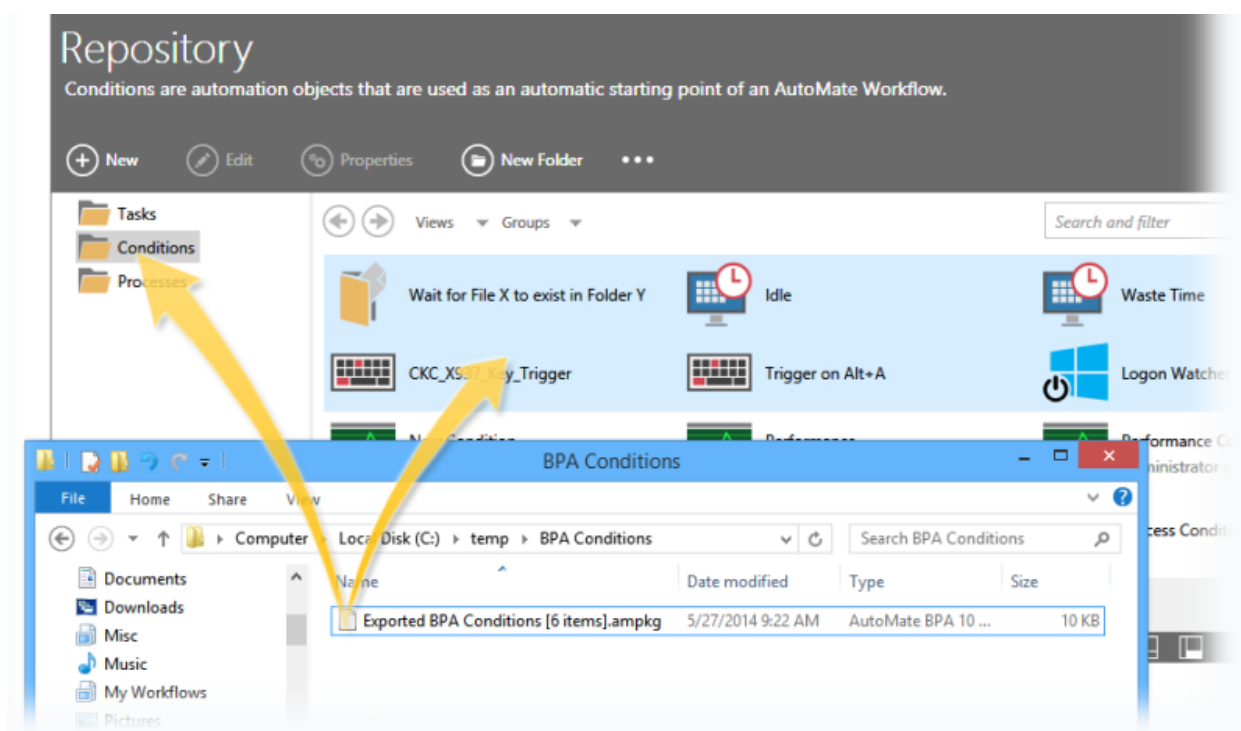
- [Adobe Sign eSignature API Version](#): This bot was created using Adobe Sign REST API v.6.

HOW TO IMPORT AN AUTOMATE TASK

Compatible file types can be imported to the repository via drag-and-drop

From the [Server Management Console](#), navigate to the Repository section

Drag the desired file(s) from its original location and drop them into the folder in the [SMC](#). Files can be dropped into the folder icon or the main panel (as shown below). Imported object(s) are automatically placed into their corresponding repository location.



HOW TO GENERATE THE INPUT FILE

The input file ***“Adobe Sign Request Document Signature INPUT.xlsx”*** shipped with this Adobe Sign Bot provides a self-explanatory guide to complete all the fields in your transaction automatically.



The Excel INPUT file

| A | B | C | D | E |
|--|----------------------|---------------|---------|--|
| Adobe Sign Request Single Document Signature by Email fields | | | | |
| Document Name Name and extension | Documents Foder Path | Email Subject | Message | Recipients Format (list with semicolon separator): Recipient1@domain.com;Recipient2@domain.com |

| F | G | H |
|-------------------|---------|--------------------------------------|
| Created and Sent? | Details | |
| Yes/No | Date | Agreement ID or Error Result Message |

NOTES:

- Fields/Columns in **Red** are mandatory.
- **Column A** has the document **name** and **extension** and in the **column B** you will set the folder path in which the documents are stored. They are automatically converted to PDF files by Adobe Sign.
- **Column E** has the sign request **Recipients**. For multiple recipients is important to keep the format (List of emails separated by semicolon “;”). The signing requests will be sent to the recipient in *order of appearance in the list*, first recipient will receive the request and after signing the document, the second recipient will receive the signing request.
- **Columns F to H** in the INPUT file will be filled with the result of the execution and the Agreement ID. See [APPENDIX A](#) for an example of the results.

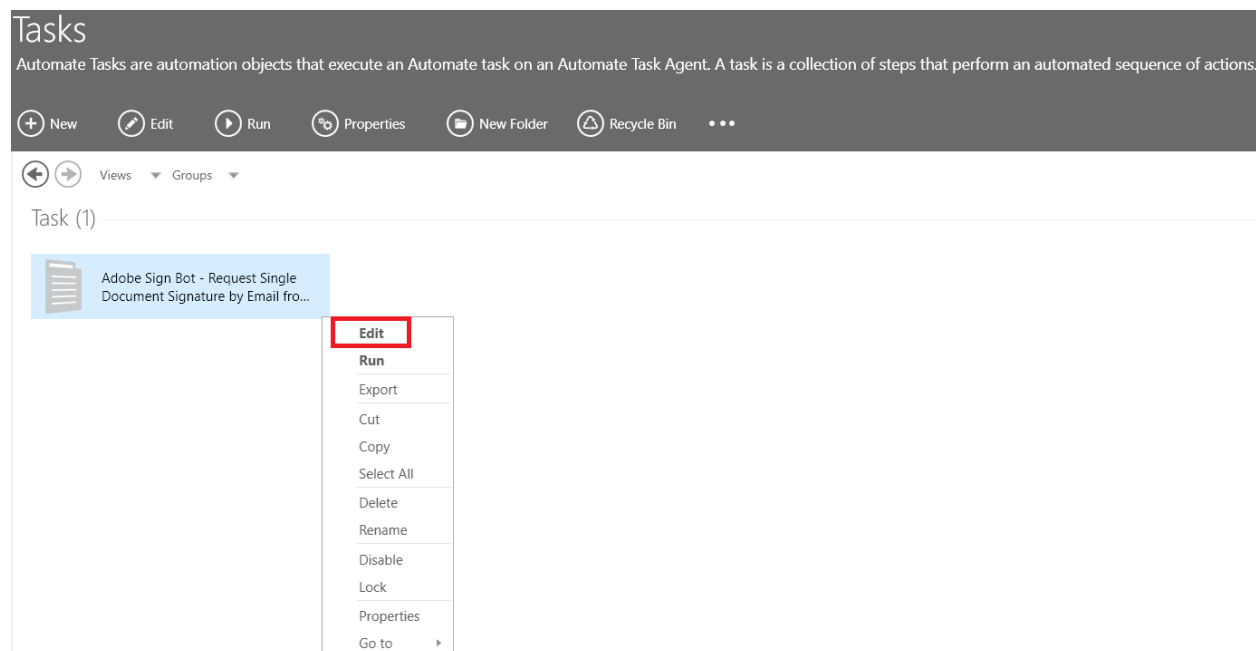
HOW TO EXECUTE THE ADOBE SIGN BOT - REQUEST SINGLE DOCUMENT SIGNATURE BY EMAIL FROM EXCEL INPUT

If this is the first time running this task, we will need to set some parameters



Open the [Server Management Console](#) and locate the imported task

Edit the imported task by right clicking on the task and selecting [edit](#)



From [step 2](#) edit the next variables according to you desired output:

- **var_ExcelInputFile:** Set the location path and name of the ***“Adobe Sign Request Document Signature INPUT.xlsx”*** file. Example: *D:\Automate\AdobeSign\Adobe Sign Request Document Signature INPUT.xlsx*
- **var_Token:** Set your token for connection or the path of the file with the token information. Remember that the token will be associated to the account used for Token Generation.

Example: ***C:\Automate\AdobeSignToken.txt***

In the case of using a .txt file, the token must have the next format/structure:
access_token:TokenString

Example of the file:



Note that there is a refresh_token value, this is managed by Adobe Sign Bot - Generate and Refresh Token

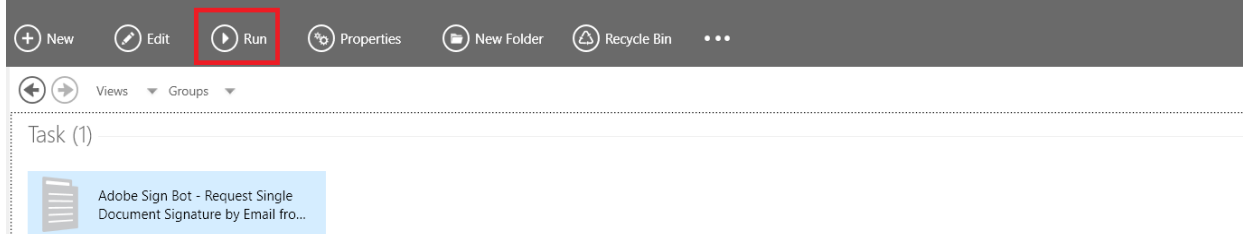
[Save and close](#) the task

Select the task and click on [Run](#)

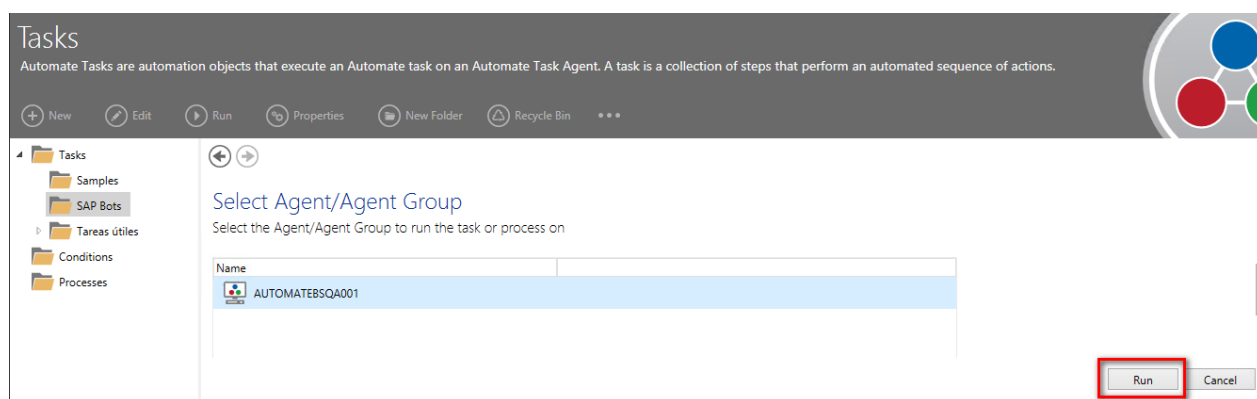


Tasks

Automate Tasks are automation objects that execute an Automate task on an Automate Task Agent. A task is a collection of steps that perform an automated sequence of actions.



Select your [Agent](#) and click on [Run](#) again





APPENDIX A – EXCEL INPUT FILE WITH RESULTS AND ADOBE SIGN VIEW RESULT

Excel INPUT file with results:

| A | B | C | D | E |
|--|-----------------------|--|----------------------------------|--|
| Adobe Sign Request Single Document Signature by Email fields | | | | |
| Document Name Name and extension | Documents Foder Path | Email Subject | Message | Recipients Format (list with semicolon separator): Recipient1@domain.com;Recipient2@domain.com |
| PDF File.pdf | D:\Automate\AdobeSign | Automate AdobeSign Bot Request Signing | Sent from Automate AdobeSign Bot | Recipient1@domain.com |
| CSV File.csv | D:\Automate\AdobeSign | Automate AdobeSign Bot Request Signing | Sent from Automate AdobeSign Bot | Recipient1@domain.com ; Recipient2@domain.com |

Note that the execution result is written from column F to H. Column H will have the Agreement ID in case of success.

| F | G | H |
|-------------------|------------------------|--|
| Created and Sent? | | Details |
| Yes/No | Date | Agreement ID or Error Result Message |
| YES | 02/10/2020 12:03:24 PM | CBJCHBCAABAAL7Q30VzvO9_oBsbftngL0mjLbQ3lhS |
| YES | 02/10/2020 12:03:30 PM | CBJCHBCAABAAGzGzHYLHdLGCfVXDjHG8-nFgLVqKdzMk |

Adobe Sign View result:

| Home Send <u>Manage</u> Reports Account API | | |
|--|-------------------------|-------------------|
| Your agreements | | |
| <div>Filters</div> <div>Search for agreements and users...</div> | | |
| STATUS | | |
| In Progress (1) | | |
| Waiting for You (0) | | |
| Completed | | |
| In Progress | | |
| RECIPIENTS | TITLE | STATUS |
| recipient1@domain.com 0 of 2 completed | [DEMO US... est Signing | Out for Signature |



APPENDIX B - TROUBLESHOOTING

- **Logs:** Each iteration of our bot creates a log file for troubleshooting. You can locate the log file under C:\Automate\Tasks\<<TASK NAME>>. By Default, the task name is *Adobe Sign Bot - Request Single Document Signature by Email from Excel INPUT*.



About HelpSystems

Organizations around the world rely on HelpSystems to make IT lives easier and keep business running smoothly. Our software and services monitor and AutoMate processes, encrypt and secure data, and provide easy access to the information people