Integrating Adobe Stock with an Enterprise DAM

Contents

Introduction

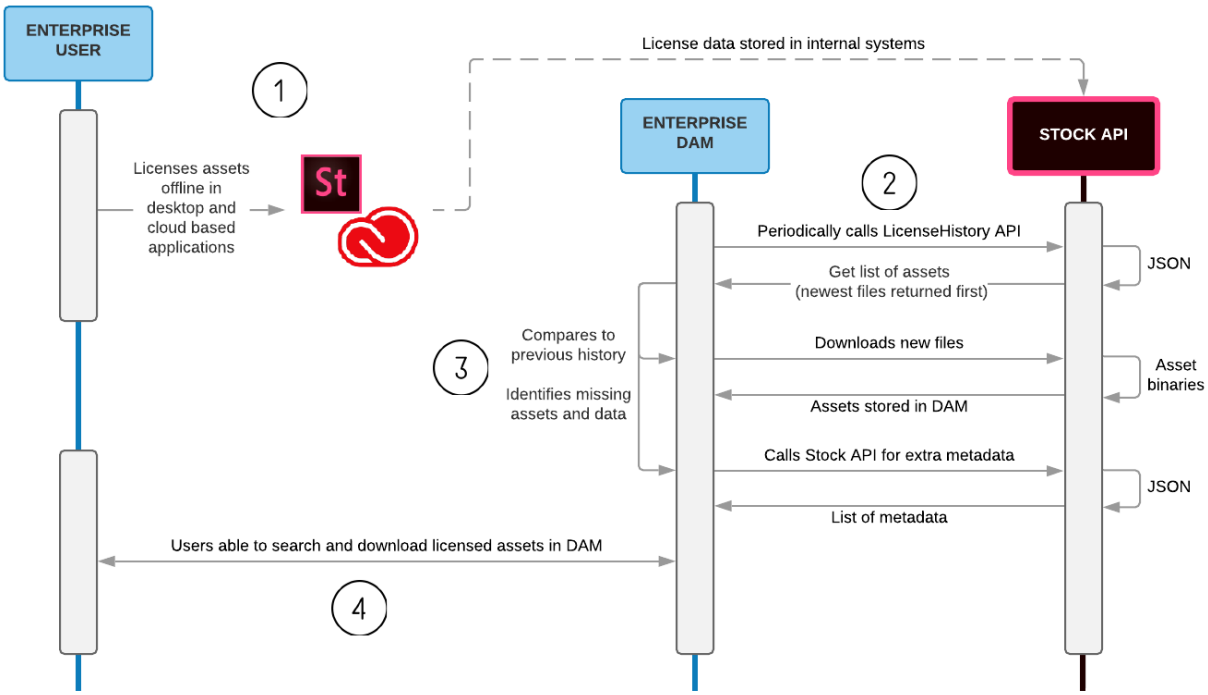
This document outlines a generic workflow for creating an integration between Adobe Stock and a Digital Asset Management (DAM) system. The goal is to allow licensed Stock assets to appear automatically in the DAM without user intervention. Note that searching and licensing Stock assets using an API is not in the scope of this document but could be added to the capabilities as well.

One challenge faced by DAM administrators is compiling a “single source of truth” for all assets owned by the enterprise. This task is especially challenging when acquisition of new assets is decentralized, such as when creative professionals license assets directly from Adobe Stock, either from its website, or as part of the in-app experience in Creative Cloud applications. Since these assets are available to anyone in the company, typically the user must then remember to manually upload and register the assets in the DAM. Or even more typically, the DAM users and administrators will not even be aware these assets exist and are available.

This workflow attempts to address the issues of discoverability and acquisition of these assets through an automated process. Please note that the approach below is generic and may or may not work with every DAM system. Furthermore, the steps outlined here illustrate only what must be done from the

Adobe Stock side, and not how to implement in the DAM. For the latter piece, the company's DAM technical team and administrators would likely need to perform some work or engage an implementation partner.

Workflow overview



This diagram illustrates the process at a high level.

1. The first step happens offline and does not involve the DAM system. This is where the end-user browses and downloads Adobe Stock content for use in creative projects. Stock for Enterprise is particularly robust and allows granting granular user permissions for who may license assets and what types they have access to and regulating how much license quota they have. Adobe Stock provides multiple touchpoints for acquiring new content, but the typical sources are via the Stock website, and via in-app experiences in Creative Cloud applications such as Photoshop, InDesign and Premiere. Users are granted access rights by their Adobe administrators and can then sign in under their own usernames and use the applications. Furthermore, Adobe can integrate with the enterprise's single-sign on (SSO) mechanism for even deeper integration with company policies.
    - The net result is that the assets, once licensed, may only exist on the user's hard drive. Or if the user deleted the original asset, it only exists as a license history record in Adobe Stock.
2. The workflow begins in earnest once an integration is created in the DAM with Adobe Stock. The DAM application authenticates itself to Adobe using an enterprise service account (see *Getting Started*, below) allowing full automation, without human intervention. When the integration is active, the DAM can fetch the enterprise Stock license history programmatically. Ideally, this process

would be run as an automatically scheduled job, perhaps once a day. The license history consists of a list of every asset licensed by the company, organized by date.

   - Stock license history is available both via an API, and via the administrator manually downloading a report from the Stock website. Best practice for DAM integrations is to perform a one-time download of *all* licensed assets and metadata from Stock, import them into the DAM, and then use a daily scheduled process to import new assets (see #3).

3. Each day (or whatever time period is defined), the DAM retrieves a list of license history using the Stock API. Because it is organized by date (newest items at the top), the DAM application would compare the list against what is already in the repository and identify only new assets. At this point it would trigger an import process, downloading ideally not only the asset itself, but any necessary metadata needed to allow the asset to be searchable in the DAM.

   - Some DAM systems can read embedded XMP metadata from the asset itself, reducing the need for fetching additional data from the Stock API. However, not all assets support this (such as Video, Templates, 3D, etc.) Furthermore, some data is added *after* the asset has been uploaded to Stock by the photographers (such as search keywords), and therefore that data will not be embedded in the final downloaded asset. In those cases, that data is *only* available via the API as separate data.

4. Once the assets and data are downloaded, the DAM would trigger import processes and workflows to store the asset in its file repository along with associated metadata. This part of the flow is *not covered* by this document because it varies from system to system. However, the net outcome is that all DAM users would now be able to search and download Stock assets that were originally licensed from outside the DAM.

   - As mentioned earlier, the DAM integration could be extended further to allow its users to search and license directly from Stock inside the DAM, but this level of integration is not always required or desired, especially if the user acquiring the assets is a creative professional who prefer to work in their Adobe tools directly.

Getting started

Enterprise service account

While is it not required to create a service account integration, it is strongly recommended. Applications can integrate with Adobe Stock both by interactive user login (called OAuth), and with technical service accounts. The latter option is available exclusively to Adobe enterprise customers and is ideal in this case because it requires no human interaction or login. Instead, the application stores secrets on with the Adobe API provider (Adobe I/O) and proves its identity programmatically with this exchange of secrets.

The process for establishing a service account is documented below. Please note that it requires that the user on Adobe I/O Developer Console either be a System Administrator in the Adobe Admin Console, or the user is delegated the Developer role. For more information, see *Manage developers*.

To create a new Stock API service account integration, *refer to the documentation here*. Once the integration account is created, it will be used internally by the DAM application to authenticate and authorize itself for retrieving company license history.

Please note that the service account integration must be linked to an Adobe Stock product profile in the Adobe Admin Console. Product profiles for Stock are used to control quota, but also are used to store license information. If this concept is not familiar, please contact your Adobe administrator and refer to the *Adobe Stock for Enterprise* documentation.

Adobe Stock APIs

The goal of the service account integration is to generate an *access token* which can be used to authorize API requests to Adobe Stock. As part of this tutorial, the License History and Files (Metadata) APIs will be used. It is recommended that the developer team test and get familiar with how each service works, as well as understand at a high level how all Stock APIs function and are accessed.

- *Getting started with the Adobe Stock API*. Provides an overview of all Stock APIs, how to register to make requests, technical requirements, and sample code and SDKs for using the APIs. Note that use of the SDKs is not required; all Stock APIs can be called directly using HTTP methods.
- *License History API reference*. The License History API does most of the work in this process, providing a comprehensive list of all assets licensed by users of the enterprise account as well as download links. This is a protected API, requiring authentication with a valid access token. Authentication is achieved using a service account (see above).
- *Bulk metadata Files API reference*. Use of the Files API is optional but recommended. This allows the application to fetch additional metadata associated with each asset, such as search keywords, title, category, etc. This API does not require an access token to use it, just an API key obtained from the Adobe I/O Developer Console. However, it would make no sense calling this API by itself without also fetching license history. Therefore, the application would typically call the Files API immediately after getting the License History API response and use the same credentials for both requests.

Workflow details

Initial import

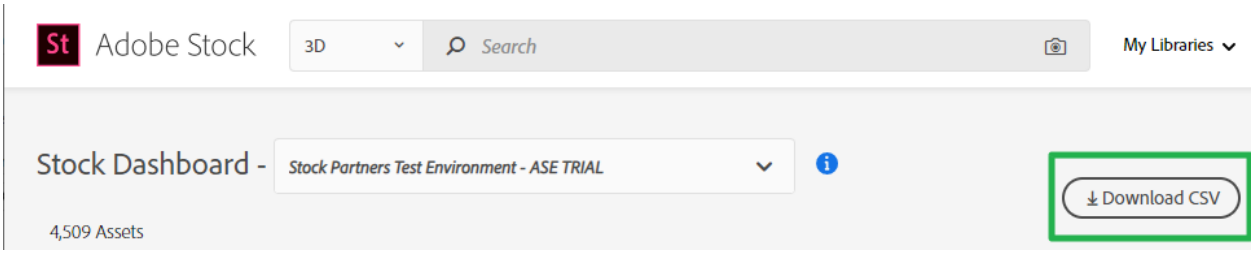
Unless the customer is new to Adobe Stock and has no previous license history, it may be desirable to perform a one-time download and import of all Stock licensed assets before scheduling a daily import job. This is especially true if the account has hundreds or thousands of assets, as this task may take several hours or even days to complete, depending on network conditions.

Using License History API

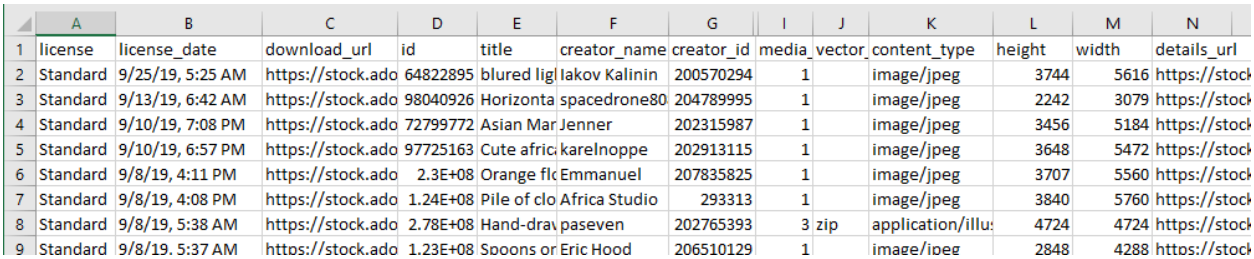
The process for fetching assets using the License History API is detailed below, under *Daily scheduled sync*, however the steps below assume that the DAM application is only retrieving the most recent assets. The application can go further and simply process all the assets in the list in the same method, followed by or in parallel with Files API requests to get accompanying metadata.

Using License History CSV report

Alternatively, the Stock administrator can start by downloading the CSV (comma-separated values) report available on the License History page on the Adobe Stock website.



Once downloaded, the file can be opened in Excel. A developer could use formulas and concatenation to create download links from this document in Excel, or a simple script could parse the CSV to get this information. The goal would be to create download URLs and Files API (metadata) requests. If more details are needed for an Excel-based process, please contact *stockapis@adobe.com*.



	A	B	C	D	E	F	G	I	J	K	L	M	N
1	license	license_date	download_url	id	title	creator_name	creator_id	media_	vector_	content_type	height	width	details_url
2	Standard	9/25/19, 5:25 AM	https://stock.ado	64822895	blured lig	Iakov Kalinin	200570294	1		image/jpeg	3744	5616	https://stock
3	Standard	9/13/19, 6:42 AM	https://stock.ado	98040926	Horizonta	spacedrone80	204789995	1		image/jpeg	2242	3079	https://stock
4	Standard	9/10/19, 7:08 PM	https://stock.ado	72799772	Asian Mar	Jenner	202315987	1		image/jpeg	3456	5184	https://stock
5	Standard	9/10/19, 6:57 PM	https://stock.ado	97725163	Cute afric	karelnoppe	202913115	1		image/jpeg	3648	5472	https://stock
6	Standard	9/8/19, 4:11 PM	https://stock.ado	2.3E+08	Orange flo	Emmanuel	207835825	1		image/jpeg	3707	5560	https://stock
7	Standard	9/8/19, 4:08 PM	https://stock.ado	1.24E+08	Pile of clo	Africa Studio	293313	1		image/jpeg	3840	5760	https://stock
8	Standard	9/8/19, 5:38 AM	https://stock.ado	2.78E+08	Hand-dra	paseven	202765393	3	zip	application/illus	4724	4724	https://stock
9	Standard	9/8/19, 5:37 AM	https://stock.ado	1.23E+08	Spoons or	Eric Hood	206510129	1		image/jpeg	2848	4288	https://stock

Daily scheduled sync

Once the initial import has completed, the integrator can check for new files each day by creating a job in their application to call the License History API each day. The Stock API offers no webhooks or push notifications when this list has changed, so the application must check and compare the current results to the previous results. Note that this API always returns the most recently licensed assets first, so in many cases the integration would only need to read the first entries and check if those files were imported.

1. **Create authentication token.** Assuming that a service account integration was created, the application will generate a signed JWT and post it to the Adobe IMS endpoint to receive back an access token for 24 hours. Best practice is to either cache the token locally and renew it each 24 hours, or simply generate a new JWT and receive a new access token for each request.
    - Sample JWT access token exchange request

```
curl --location
  --request POST 'https://ims-na1.adobelogin.com/ims/exchange/jwt/'
  --header 'Content-Type: application/x-www-form-urlencoded'
  --data-urlencode 'client_id={API_KEY}'
  --data-urlencode 'client_secret={CLIENT_SECRET}'
  --data-urlencode 'jwt_token={ENCODED_JWT}'
```

- Sample JWT exchange response

```
{
    "token_type": "bearer",
    "access_token": "{ACCESS_TOKEN}",
    "expires_in": 86399998
}
```

- For more details, see *JWT (Service Account) Authentication*.

2. **Call License History API.** The app calls the Stock license history endpoint and optionally requests all license history, or just the history of the selected profile. By default, this API returns JSON results with 100 rows. If there are more than 100 licensed assets, the API supports pagination. As mentioned above, it always returns the results in reverse chronological order, with the most recently licensed asset listed first.

   - Sample License History API request: Requesting all history for organization

```
curl --location --request GET
'https://stock.adobe.io/Rest/Libraries/1/Member/LicenseHistory?all=t
rue'
  --header 'X-Product: MyApp/1.0'
  --header 'x-api-key: {API_KEY}'
  --header 'Authorization: Bearer {ACCESS_TOKEN}'
```

     - Note that if the `` `all=true` `` command is left off, then the application may not be able to see the entire license history. This is because Adobe Stock stores license data at the product profile level, and integrations can only be linked to a single profile. By default, the application can only view the profile in which it resides. Applying this command enables the application to see all license history for the organization.

   - Sample License History API response

```
{
    "nb_results": 244,
    "files": [
        {
            "license": "Standard",
            "license_date": "1/19/21, 11:30 PM",
            "download_url":
"https://stock.adobe.com/Download/DownloadFileDirectly/TzIJUeAxAZ6jM
8sqhSJGIiMJhiYPz3OI",
            "id": 77438420,
            "title": "Eating lunch at the office",
            "creator_name": "AntonioDiaz",
            "creator_id": 201590104,
            "content_url": "https://stock.adobe.com/Rest/stock-
photo/eating-lunch-at-the-office/77438420",
```

```
            "media_type_id": 1,
            "vector_type": null,
            "content_type": "image/jpeg",
            "height": 3744,
            "width": 5616,
            "details_url":
"https://stock.adobe.com/77438420?as_channel=affiliate&as_source=api
&as_content=953592d1c4f94b32912a4af0f3689814"
        }, ... ]}
```

- Documentation: *License History API reference*

3. **Parse JSON response**. Each asset is listed in an object array as shown above. The metadata returned is only a subset of what is available, so if more data is needed, see the optional step below for retrieving additional metadata. Assuming that the goal is to download the files, the only data point required is the `download_url` property for each asset. It's assumed that the application is storing these extracted fields in the DAM database for each new asset found.

   - At a minimum, it's recommended that the application retrieves these fields for each asset. These fields are typically *not present* in the XMP data of the downloaded asset, so this API is the only way to retrieve them in bulk. Note that all the data values except the download URL below are assumed to be static—for example, the license date should never change. This can be used in the next step.

     - `id`: Stock asset ID. Required to fetch additional data.
     - `license`: The license type, e.g., "Extended" or "Standard."
     - `license_date`: Date and time that a user licensed the asset.
     - `download_url`: This will be used in the download step, below. This value may be subject to change, so it should be fetched each time a new download is needed.
     - `content_type`: Asset mime-type, to assist in downloading.

4. **Compare to files already downloaded.** The License History API returns items in descending order by license date, so it's assumed the application will either store the ID of the last asset imported, or use the current system date to determine what needs to be downloaded by iterating over the asset data parsed in the previous step. Note If the organization is large and licenses more than 100 assets a day, it may require a few pagination requests to find the last asset imported and begin the daily import process.
5. **[Optional] Download metadata.** As mentioned above, the License History API only returns partial metadata, and the downloaded asset itself may have no additional metadata embedded in its XMP data. Stock provides the Files API for fetching metadata in bulk, and since the License History response returns up to 100 assets at once, it's recommended to call the Files API to get the remaining data for those 100 assets in a single request. This step could be performed after downloading the assets, but keep in mind that the download cannot be performed with a bulk request, and will take far longer than fetching the metadata.

- Adobe Stock makes nearly no changes to the asset XMP data, and it does not embed any keywords or categories or other typical data in the file—these values will only be present if they were present in the original asset at time of upload.
  - Consult the *Search API documentation* for a complete list of response fields.
  - Note that the APIs will not return specific usage restrictions such as "editorial use only." To prevent a DAM user from misusing an asset in a project (e.g., using a photo of a politician in an ad campaign), it would be the responsibility of the DAM application to properly tag the asset. Therefore, it's recommended to fetch the field `` `is_editorial` `` as shown in the example below. If the field returns true, business rules could be applied to either skip downloading of the asset to prevent legal issues, or could apply special handling rules for such protected assets.
- Sample Files API metadata request

```
curl --location -g --request GET
'https://stock.adobe.io/Rest/Media/1/Files?locale=en_EN&ids=32099999
9, ...
{LIST_OF_IDS}&result_columns[]=id&result_columns[]=title&result_colu
mns[]=keywords&result_columns[]=content_type&result_columns[]=catego
ry&result_columns[]=is_editorial' --header 'X-Product: CFSTest/1.0'
  --header 'X-Product: MyApp/1.0'
  --header 'x-api-key: {API_KEY}'
```

- Sample Files API response

```
{
  "nb_results": 100,
  "files": [
    {
      "id": 320999999,
      "title": "year 2020 business solution concept",
      "keywords": [
        {
          "name": "2020"
        },
        {
          "name": "business"
        }, ...
      ],
      "content_type": "image/jpeg",
      "category": {
        "id": 167,
        "name": "Business"
      },
      "is_editorial": false
    }, ... ] }
```

- Documentation: *Files API reference*

6. **Download files.** This step will do the bulk of the work and take the longest to complete. Using the `download_url` property retrieved above, the application will now call the Download API endpoint for each asset. This step also requires the access token obtained at the beginning. The Download API will verify that the owner of the token has a license to download the file, generate a short-lived signed URL, and then redirect the request to the cloud storage bucket where the file resides. It's important to follow redirects or else the download might not complete.

   - Sample download URL with access token

```
curl -L "
https://stock.adobe.com/Download/DownloadFileDirectly/TzIJUeAxAZ6jM8
sqhSJGIiMJhiYPz3OI?token={ACCESS_TOKEN}"
```

   - As a best practice, it's recommended that the download process be independent and/or asynchronous of the process to fetch the history and metadata, especially if the network connection isn't stable.
     - There is no fixed limit on number of concurrent downloads, but it's recommended to keep download threads to less than five. If more bandwidth is needed, please contact *stockapis@adobe.com* for next steps.
   - Note that the format of the download URL is a generated hash in the future may use a different syntax—this URL cannot be generated programmatically and must be retrieved from the License History API.
   - Documentation: *Downloading licensed files*

References

Adobe Admin Console

Setting up Developers: *https://helpx.adobe.com/enterprise/using/manage-developers.html*

Managing Stock for Enterprise: *https://helpx.adobe.com/enterprise/using/adobe-stock-enterprise.html*

Adobe Developer Console

Adding a JWT service account: *https://www.adobe.io/apis/experienceplatform/console/docs.html#!AdobeDocs/adobeio-console/master/services-add-api-jwt.md*

JWT authentication: *https://www.adobe.io/authentication/auth-methods.html#!AdobeDocs/adobeio-auth/master/JWT/JWT.md*

Adobe Stock API

Getting started: *https://www.adobe.io/apis/creativecloud/stock/docs.html*

License History API: *https://www.adobe.io/apis/creativecloud/stock/docs.html#!adobe/stock-api-docs/master/docs/api/13-license-history.md*

Files API: *https://www.adobe.io/apis/creativecloud/stock/docs.html#!adobe/stock-api-docs/master/docs/api/19-bulk-metadata-files-reference.md*

Search API (for response fields): *https://github.com/adobe/stock-api-docs/blob/master/docs/api/11-search-reference.md#responses*

Technical API FAQ: *https://www.adobe.io/apis/creativecloud/stock/docs.html#!adobe/stock-api-docs/master/docs/15-faq.md*

Business API FAQ: *https://www.adobe.io/apis/creativecloud/stock/docs.html#!adobe/stock-api-docs/master/supplemental/stock-api-business-faq.md*

Please contact *stockapis@adobe.com* with additional technical questions.

Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110-2704
USA
www.adobe.com

5/15