

# Sprinklr + AEM

AEM Integration Project API Specifications

Version 1.0 A



29 West 35th Street, 7th Floor  
New York, NY 10001 USA

PH: (917) 933-7800  
Twitter @sprinklr

[www.sprinklr.com](http://www.sprinklr.com)  
[info@sprinklr.com](mailto:info@sprinklr.com)

The information contained herein is confidential and proprietary and can be neither disclosed nor duplicated without the prior written consent of Sprinklr, Inc.

Copyright © 2019 Sprinklr, Inc. All rights reserved.

**About Sprinklr:** <https://www.sprinklr.com/about-sprinklr/>

### **Notice**

Reasonable efforts are made to ensure document information integrity, but errors will happen, and Sprinklr, Inc. maintains the right to change management with release revisions and assumes no responsibility for the information contained herein.

### **Your Responsibility for Your Systems' Administration**

You are responsible for the administration of your Sprinklr & integrated systems. Your administrator(s) should consume all product content to mitigate risks not limited to authorization controls, terms of use, data management, licensed use, etc.

### **Trademarks**

Sprinklr and the Sprinklr logo are registered trademarks of Sprinklr, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders.

# Table of Content

<b>Table of Content</b>	<b>3</b>
<b>Project Description</b>	<b>4</b>
<b>Selecting Assets from AEM</b>	<b>5</b>
Search Asset API Endpoint	6
Service Endpoint	6
Headers:	6
Request Parameter Description:	7
Nested Filters Parameter Description:	8
SAM Filters:	9
Sample Request Payload:	9
Response:	10
Response Definition	12
Download Asset API Endpoint	15
Service Endpoint	15
Sample Request:	15
Sample Response:	15

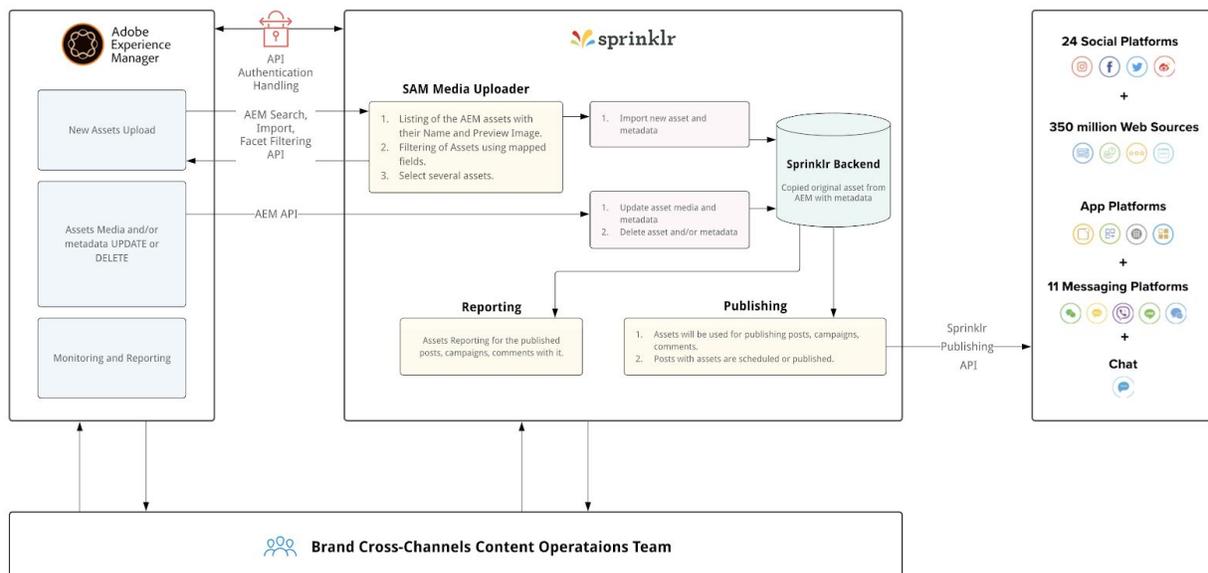
# 1. Project Description

Using multiple disconnected DAMs lead to inefficiencies and cause potential governance issues

## 2. Selecting Assets from AEM

In order to get the Asset details, Sprinklr needs two API endpoint on the AEM side i.e:

1. *Asset Category API Endpoint - HTTPS (POST): To fetch all the asset categories.*
2. *Search Assets API Endpoint - HTTPS (POST): Sprinklr will call this endpoint to fetch the assets and facets information from AEM.*
3. *Download Asset API Endpoint - HTTPS (GET): Sprinklr will call this endpoint to download the Asset.*



**NOTE:** Everything that comes under content (Photo, Video, etc.) is termed as an Asset.

### 2.1. Search Asset API Endpoint

Below, are the details of the API Endpoint, Request params, and Response Params along with other objects that we need in require in the request body and need in the response body:

- Service Endpoint
- Headers

- Request Parameter and Descriptions
- SAM Filters
- Sample Request Payload
- Response

## Service Endpoint

Sprinklr needs an Https POST API endpoint on Ali Express side to read the details of Assets.

Sample:

**POST** <https://baseURL/search/>

**baseURL:** The base Url represents the server address or the Host.

## Headers:

The following set of HTTP header fields provide required information about the request or response, or about the object sent in the message body. Both request headers and response headers can be controlled using these endpoints

Key	Value	Description
Content-Type	application/json	The request format should be JSON as the endpoint expects a JSON body.
Authorization	Access token	The credential used by an application to access an API.
Key	API-key	Indicates the App/API key, which is obtained during application creation.

## Request Parameter Description:

The parameter that Sprinklr can pass in the request body while calling the endpoint to get the desired results.

Parameters	Sub Parameters	Required/ Optional	Description	Type
------------	----------------	--------------------	-------------	------

q		Optional	Here you can pass the keyword to search for within AEM.	String
filter		Required	Filter to apply within the entity type.	Object
	type	Required	<p>For Phase 1 : Filter type will be AND (all criteria will need to be matched)</p> <p>For future Phases : this can be extended so the user can choose from the available filter types.</p> <p>AND OR NOT IN GT GTE LT LTE NIN EQUALS NOT_EQUALS</p>	FilterTypes
	filters	Required	Filters represent the list of nested filters. <a href="#">See the table "Nested Filters" below for filter details.</a>	List of Filter, String
sort		Required	Sorting information.	Sorting Object
	key	Required	<p>Key on the basis of which you want to sort the query. Key types are given below:</p> <ul style="list-style-type: none"> <li>• monthOrders,</li> <li>• productRating,</li> <li>• sellerRating,</li> <li>• gCommentRatio</li> </ul>	String
	order	Optional	Short in ascending or descending order. i.e. ASC and DESC.	String

Page		Required	Pagination information.	Pagination object
	size	Required	Size of the page to fetch in response.	Integer
	cursor	optional	The number from which you want to start fetching data.	Integer

### Nested Filters Parameter Description:

Parameters	Sub Parameters	Required/Optional	Description	Type
filters			Filters represent the list of nested filters.	List of Filter, String
	type	Required	Type of filter should match the provided key and values. It can be a list of filter types	String
	key	Required	The key on which you want to apply the filter.	String
	values	Required	The value of the key. It should match with the key.	String

### SAM Filters:

Customer to provide the details on Filters that can be used.

### Sample Request Payload:

```
curl --location --request POST 'https://https//baseURL/search/SAM' \
--header 'Authorization: Bearer {{Enter your Access Token}}' \
--header 'Key: {{Enter your API KEY}}' \
--header 'accept: application/json' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode '{
  "q": "String",
  "filter": {
```

```

"type": "AND",
"filters": [
  {
    "type": "EQUALS",
    "key": "assetType",
    "values": [
      "IMAGE"
    ]
  },
  {
    "type": "EQUALS",
    "key": "categoryId",
    "values": [
      "500901"
    ]
  },
  {
    "key": "assetStatus",
    "values": [
      "APPROVED"
    ],
    "type": "IN"
  }
]
},
"sort": {
  "key": "productRating",
  "order": "DESC"
},
"page": {
  "size": 10
  "cursor": 0
}
}'

```

## Response:

```

{
  "assets": [
    {
      "id": {{The Id of an asset with AEM}},
      "name": "wueste-metadaten-v1.jpg",
      "description": "Beschreibung-d",
      "assetType": "IMAGE",
      "status": "APPROVED",
      "attachment": {
        "type": "IMAGE",
        "attachmentOptions": null
        "title": "asd, Titel_de",
        "description": "Beschreibung-de",
        "previewUrl":
          "https://qa4-sprcdn-assets.sprinklr.com/400002/title-f6bfb54a-1105-478d-b2cc-bcf9a975f8af-2136884405.jpg",
        "mimeType": "image/jpeg"
      },
      "taxonomy": [null]
      "validity": [
        {

```

```
"expiryTime": null,
"availableFrom": null,
"neverExpire": null
}
],
"categoryId": "500901",
"categoryLevel": null,
"assetSource": "AEM",
"actionStats": null,
"shareConfigs": null,
"restricted": false
}
],
"facets": [
{
"field": "REGIONS",
"label": "REGIONS",
"values": [
{
"value": "countries:Spain",
"count": 11,
"label": "Spain"
},
{
"value": "countries:Colombia",
"count": 7,
"label": "Colombia"
},
{
"value": "countries:Spain/Madrid",
"count": 2,
"label": "Madrid "
},
{
"value": "countries:de",
"count": 1,
"label": "Denmark "
}
]
},
{
"field": "LANGUAGES",
"label": "LANGUAGES",
"values": [
{
"value": "lang:de_de",
"count": 775,
"label": "Denmark "
},
{
"value": "lang:en",
"count": 435,
"label": "English "
},
{
"value": "lang:es_es",
"count": 1,
"label": "LANGUAGES3 "
}
]
}
]
```

```

    },
    {
      "field": "KEYWORDS",
      "label": "KEYWORDS",
      "values": [
        {
          "value": "de",
          "count": 1,
          "label": "KEYWORDS1 "
        },
        {
          "value": "monument",
          "count": 1,
          "label": "KEYWORDS2 "
        },
        {
          "value": "stichwörter",
          "count": 1,
          "label": "KEYWORDS3 "
        }
      ]
    }
  ],
  {
    "field": "CATEGORIES",
    "label": "CATEGORIES",
    "values": [
      {
        "value": "900",
        "count": 14,
        "label": "Category label - Electronics"
      },
      {
        "value": "90050",
        "count": 7,
        "label": "Sub-category label - Wearables"
      },
      {
        "value": "9005011",
        "count": 2,
        "label": "Sub-Sub-category - Connected watch"
      }
    ]
  }
],
"count": 1,
"start": 40
}

```

## Response Definition

Parameter	Sub Parameter	Description	Type
id		Id of the asset.	String

name		Name of the asset.	String
description		Description of asset.	String
assetType		The type of asset. ENUM:[PHOTO, VIDEO, TEXT]  <i>Can be extended after Phase 1, for example : ENUM:[PRESENTATION, PHOTO, VIDEO, AUDIO, FLASH, HTML, PDF, DOC, DOCX, EXCEL, ZIP, PSD, SKETCH, FONT, COUPON, ...]</i>	String
status		Status of the asset. ENUM:[APPROVED, DRAFT]	String
attachment		Information about the attachment.	String
	type	The type of attachment. ENUM:[IMAGE, Video, Text]	String
	attachmentOptions	Array of attachment properties per channel and/or account. ChannelSpecificMediaProperties.	
taxonomy		Taxonomy related to an asset	
	CustomProperties	custom properties for the asset.	String
validity		Information regarding the validity of an asset.	
	expiryTime	Asset expiry time. This option is not valid if neverExpire is true.	Integer
	availableFrom	Asset is available from time.	Integer
	neverExpire	If True, asset will never expire.	Boolean

parentId		Parent Asset Id.	String
assetSource		Source of the Asset. e.g. AEM	String
actionStats		Stats of the asset actions. e.g Download, Published.	Double
shareConfigs		Asset share configuration.	
	shareLevel	Asset share level. e.g. USER, GLOBAL, CLIENT.	String
	sharedWithIds	Id based on shareLevel.	String
restricted		If True, assets are restricted from publishing.	Boolean
facets		Object containing facets details	
	field	Name of the field on which filtration is done on facets level.	String
	label	Label given to a segmentation in facet.	String
	values	Values containing :  Value is the name given to an asset segmentation. Count is the count of total assets in the segment.	Object
Count		Represents the total number of assets that matches the filter query	Integer

## 2.2. Download Asset API Endpoint

Sprinklr will use this API endpoint to download the assets from either a public Url or from private via using access token and app key.

### Service Endpoint

Sprinklr needs an Https GET API endpoint on AEM side to download the Asset.

Sample:

**GET** {{Using previewURL from the Search API response }}

Example:<https://qa4-sprcdn-assets.sprinklr.com/400002/title-f6bfb54a-1105-478d-b2cc-bcf9a975f8af-2136884405.jpg>

Note: oAuth is not required for this endpoint.

### Sample Request:

```
curl --location --request GET
```

```
'https://qa4-sprcdn-assets.sprinklr.com/400002/title-f6bfb54a-1105-478d-b2cc-bcf9a975f8af-2136884405.jpg'
```

### Sample Response:

Image download using the previewUrl.

The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: <https://qa4-sprcdn-assets.sprinklr.com/400002/title-f6bfb54a-1105-478d-b2cc-bcf9a975f8af-2136884405.jpg>
- Send button
- Params, Authorization, Headers (7), Body, Pre-request Script, Tests, Settings tabs
- Query Params table:

KEY	VALUE	DESCRIPTION
Key	Value	Description

- Body, Cookies, Headers (13), Test Results tabs
- Status: 200 OK
- Time: 414 ms
- Size: 89.08 KB
- Save button

The main content area displays a downloaded image of two translucent, faceted geometric shapes (one orange and one red) on a grid background.