



DAM API Documentation

Document Version 1.6

Table of Contents

[Table of Contents](#)

[1 Overview](#)

[2 If you are new to 1Worldsync's DAM API](#)

[3 Breaking Changes](#)

[4 Using the 1Worldsync DAM API](#)

[4.1 API Versioning](#)

[4.2 Overview](#)

[5 Asset API](#)

[5.1 Add Digital Asset](#)

[5.2 Get Metadata for Digital Asset](#)

[5.3 Delete Digital Asset](#)

[6 Validation API](#)

[6.1 Get Validation for Digital Asset](#)

[7 Content API](#)

[7.1 Get Content of Digital Asset](#)

[7.2 Get and Transform Content](#)

[8 Utility API](#)

[8.1 Get Service Status](#)

[8.2 Get Service Version](#)

[9 Authorization Mechanism](#)

[APPENDIX - Supported File Types](#)

[Contact Us](#)

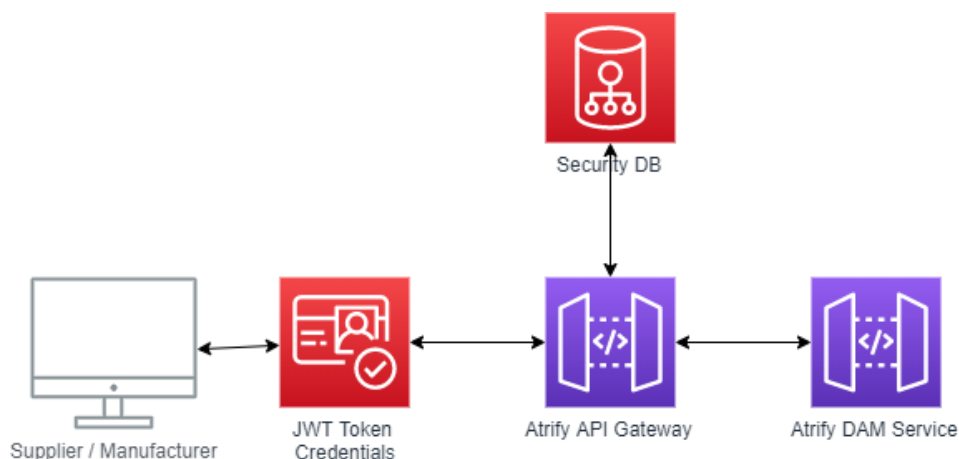
REVISION HISTORY

Date	API Version	Document Version	Changes	Author
2019-11-22	v1	1 - BETA	Initial Version	
2019-12-04	v1	1.0	Set document to final version	Daniel Haupt
2019-12-16	v1	1.1	Added missing request header info for DELETE	Kevin Bartsch
2019-12-17	v1	1.2	Updated list of supported file formats	Daniel Haupt
2020-01-14	v1	1.3	Fixed minor mistakes	Daniel Haupt
2020-11-04	v2	1.4	Description of v2 API	Arno Erpenbeck
2022-09-19	v2	1.5	Description of validation profiles and metadata extraction	Mladen Hrkac
2025-01-23	v2	1.6	Updated document to 1WS Company Design	Heike Brodda

1 Overview

1Worldsync offers Digital Asset Management capabilities for data sources to manage and exchange digital content. Digital Assets are any files about a product - such as product images, videos, and PDFs to support Safety Data Sheets and Warranties. These digital assets can be uploaded to 1Worldsync, where a URL is assigned for hosting the asset.

Example: DAM Service Authentication / Process



1Worldsync's Digital Asset Management (DAM) Application Programming Interface (API) allows you to upload, host, and manage your digital assets. By using this API, you can:

- Upload images, PDFs, and other related digital content
- Host your digital assets (have a URL assigned to your digital asset)
- Associate the digital asset with one of your existing products in Publishing, enabling the content for distribution
- Get the metadata for a given digital asset
- Validate digital assets
- Delete digital assets

This document describes how to use the APIs to manage digital assets within 1Worldsync. It details the various parameters required to access and use the service, such as web service URLs, and request and response formats.

The intended audience of this document is any 1Worldsync partner or customer who will use our services to manage digital assets.

2 If you are new to 1Worldsync's DAM API

To get started with using the DAM APIs, please get in touch with our Service Desk team (servicedesk@1worldsync.atlassian.net).

DAM API Documentation



After registration you will receive:

- Your bucket prefix
- Company GLN
- Your JWT access token, which grants you access to the service

Example:

Bucket prefix	dam-fmcg-pub
Company GLN	25252500000001
JWT access token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiIxMjM0NTY3ODkwIn0.fc7FE90Bsir9F9e74qbvaCd0F3JnS46l8ifEGSeFoRc

If you would like to test our APIs in a non-production environment, you can use our Pre-Production environment. You will need a separate client id and jwt token. Please contact our Service Desk Team (servicedesk@1worldsync.atlassian.net) if you need this additional access established.

3 What's New?

With this release we introduce versioning to our public DAM API. The new API version v2 contains the following changes and improvements:

- The JSON response objects for storage and retrieval have been restructured and contain additional details about the asset metadata (e.g. color space, resolution, clipping path information).
- A new validation API (not yet public) will offer the functionality to validate the asset metadata against a defined set of validation rules.
- Error messages have been improved.
- The API documentation has been updated and enhanced with more details and examples for all API methods.

See the sections below for further details.

4 Using the 1Worldsync DAM API

This chapter provides a listing of all available API requests in a short overview. All HTTP GET methods work without any authentication. For all other HTTP methods, you will need a valid JWT access token, which is sent as an Authorization Bearer header.

4.1 API Versioning

The 1Worldsync DAM API uses a versioning approach where the API version is indicated in the URL. All HTTP requests **must** specify the API version in the request URL. When backwards-incompatible changes are made to the API, a new version is released.

The URL pattern for all HTTP requests (GET, POST, DELETE) is:

```
https://dam-api.atrify.com/{version}/{method}/{path_param_1}/{path_param_2}?{request_param_1}&{request_param_2}&...
```

Examples:

<https://dam-api.atrify.com/v1/version>
<https://dam-api.atrify.com/v2/version>

The latest DAM API version is: **v2**

Currently supported versions:

Version	GA date	EOL date
---------	---------	----------

v1	2019-12-16	TBD
v2	2020-11-07	

There will always be a time frame where multiple versions are active in parallel so that API customers have sufficient time to switch their client applications to the new API version. However, every customer will have to choose which version they want, i.e. you cannot mix calls to different API versions. The JWT authentication token will be tied to a specific version.

For backwards compatibility we also support requests *without* specifying a version. These unversioned calls will always point to the oldest available version v1. It is strongly recommended to specify an exact version as 1Worldsync remains the right to discontinue the support for unversioned API calls.

Will everything remain completely unchanged in a version?

1Worldsync does reserve the right to make changes in any API in a short period of time for issues related to security or privacy. These changes don't happen often, but they do happen.

4.2 Overview

Take a look at the automatically generated API documentation provided via Swagger UI <https://dam-api.atrify.com/swagger-ui.html> for a first impression.

Action	Action Details	Description	Path
POST	Add Digital Asset	This is used to add a digital asset to 1Worldsync digital asset repository. When an asset is uploaded it will be hosted at a 1Worldsync assigned URL. This URL will be returned in the response. Certain other asset information such as assetId, type, size, height and width are also returned.	/{version}/ assets /{bucketPrefix}/{companyGln} QueryParams
GET	Get Metadata for Digital Asset	This is used to get the metadata for a given asset identified by the asset name.	/{version}/ assets /{bucketPrefix}/{companyGln}/{assetName}
DELETE	Delete Digital Asset	This is used to delete a digital asset. This will remove the digital asset at the URL. Caution should be used when deleting a digital asset, as the URL will result in a broken link if the URL is	/{version}/ assets /{bucketPrefix}/{companyGln}/{assetName}

		in use (e.g., by one of your trading partners).	
GET	Get Content of Digital Asset	This is used to get the digital asset as a file.	<code>/ {version} / content / {bucketPrefix} / {companyGln} / {assetName}</code>
GET	Get Content and Transform File	This is used to get the digital asset with a defined scale (height, width) and file type (jpg, jpeg, png, gif).	<code>/ {version} / content / transform / {bucketPrefix} / {companyGln} / {assetName}</code> QueryParams
GET	Get Content of External Asset	Stream the content of the given asset. Get the external asset as a file	<code>/ dam / {version} / content</code> QueryParams
GET	Get Thumbnail of External Asset	Create a thumbnail for external assets. Get thumbnail as a file	<code>/ dam / {version} / content / thumbnail</code> QueryParams
GET	Validate Asset	This is used to perform a validation of the asset's metadata against a validation profile. (v2)	<code>/ {version} / validations / {bucketPrefix} / {companyGln} / {assetName}</code> QueryParams
GET	Get Service Status	This can be used to test the connection to the DAM web service.	<code>/ {version} / status</code>
GET	Get Service Version	This can be used to test the connection to the DAM web service.	<code>/ {version} / version</code>

5 Asset API

This chapter describes the methods of the asset API.

5.1 Add Digital Asset

Description: Upload an asset into the digital asset management system.

Note: The maximum file size is 50 megabytes.

POST	<code>/{version}/assets/{bucketPrefix}/{companyGln}</code>
-------------	--

Request Headers:

KEY	VALUE
Content-Type (required)	multipart/form-data
Authorization (required)	Bearer [YOUR-JWT]
Accept	application/json

Path Parameters:

KEY	VALUE : TYPE
bucketPrefix (required)	[YOUR-BUCKET] :string
companyGln (required)	[YOUR-GLN] :string

Query Parameters:

KEY	VALUE : TYPE
fileName (required)	:string
mimeType (required)	:MIME type See APPENDIX - Supported File Types

Body Form Data:

KEY	VALUE : TYPE
file (required)	:file

Response Headers:

KEY	VALUE
Content-Type	application/json

Responses:

CODE	Description / Response
200	<p>OK</p> <pre>v1: { "assetName": "string", "assetUrl": "string", "bucketPrefix": "string", "companyGln": "string", "contentType": "string", "fileName": "string", "fileSize": "integer", "height": "integer", "width": "integer" } v2: { "object": "asset", "ok": true, "timestamp": "date-time", "data": { "bucketPrefix": "string", "companyGln": "string", "assetName": "string", "fileName": "string", "assetUrl": "string", "metadata": { "size": "integer", "contentType": "string", "width": "integer", "height": "integer", "colorSpace": "string", "colorDepth": "integer", "compressionType": "string", "horizontalResolution": "integer", "verticalResolution": "integer",</pre>

	<pre> "metadataDateTime": "date-time", "updated": "boolean", "pathData": [{ "name": "string", "numPoints": "integer", "closed": "boolean" }] } } } </pre>
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Example Request (v1):

```

POST
/v1/assets/dam-fmcg-pub/9300001000018?fileName=devil_128.png&mimeType=ima
ge%2Fpng HTTP/1.1
Host: dam-api.atrify.com
Accept: application/json
Content-Length: 16336
Content-Type: multipart/form-data;
boundary=-----d0fb0f9d6844e751

```

Example Response (v1):

```

HTTP/1.1 200
Content-Type: application/json

{"bucketPrefix":"dam-fmcg-pub","companyGln":"9300001000018","assetName":"
ae284c29-f6da-491d-acab-b8c0145a4e3e-devil_128.png","fileName":"devil_128
.png","assetUrl":"https://dam-api.atrify.com/content/dam-fmcg-pub/9300001
000018/ae284c29-f6da-491d-acab-b8c0145a4e3e-devil_128.png","fileSize":161
46,"contentType":"image/png","height":128,"width":128}

```

Example Request (v2):

```
POST
/v2/assets/dam-fmcg-pub/9300001000018?fileName=angel_128.png&mimeType=image%2Fpng HTTP/1.1
Host: dam-api.atrify.com
Accept: application/json
Content-Length: 21800
Content-Type: multipart/form-data;
boundary=-----d0fb0f9d6844e751
```

Example Response (v2):

```
HTTP/1.1 200
Content-Type: application/json

{"object":"asset","ok":true,"timestamp":"2020-08-24T13:33:32.962334Z","data":{"bucketPrefix":"dam-fmcg-pub","companyGln":"9300001000018","assetName":"a5df3da0-a5cc-430e-aaca-fe2afff3de85-angel_128.png","fileName":"angel_128.png","assetUrl":"https://dam-api.atrify.com/content/dam-fmcg-pub/9300001000018/a5df3da0-a5cc-430e-aaca-fe2afff3de85-angel_128.png","metadata":{"size":21610,"contentType":"image/png","width":128,"height":128,"colorSpace":"RGB","colorDepth":32,"compressionType":"deflate","horizontalResolution":96,"verticalResolution":96}}}
```

5.2 Get Metadata for Digital Asset

Description: Get the metadata for a given asset identified by the asset name

```
GET /{version}/assets/{bucketPrefix}/{companyGln}/{assetName}
```

Request Headers:

KEY	VALUE
Accept	application/json
If-Modified-Since (optional)	date + time in GMT

Path Parameters:

KEY	VALUE : TYPE
bucketPrefix (required)	[YOUR-BUCKET] :string
companyGln (required)	[YOUR-GLN] :string
assetName (required)	[ASSET-NAME] :string

Response Headers:

KEY	VALUE
Content-Type	application/json

Responses:

CODE	Description / Response
200	<p>OK</p> <pre>v1: { "assetName": "string", "assetUrl": "string", "bucketPrefix": "string", "companyGln": "string", "contentType": "string", "fileName": "string", "fileSize": "integer", "height": "integer", "width": "integer" } v2: { "object": "asset", "ok": true, "timestamp": "date-time", "data": { "bucketPrefix": "string", "companyGln": "string", "assetName": "string", "fileName": "string", "assetUrl": "string", "metadata": { "size": "integer", "contentType": "string", "width": "integer", "height": "integer", "colorSpace": "string", "colorDepth": "integer", "compressionType": "string", "horizontalResolution": "int",</pre>

	<pre> "verticalResolution": "integer", "metadataDateTime": "date-time", "updated": "boolean", "pathData": [{ "name": "string", "numPoints": "integer", "closed": "boolean" }] } } } </pre>
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Example Request (v1):

```

GET
/v1/assets/dam-fmcg-pub/9300001000018/ae284c29-f6da-491d-acab-b8c0145a4e3e-devil_128.png HTTP/1.1
Host: dam-api.atrify.com
Accept: application/json

```

Example Response (v1):

```

HTTP/1.1 200
Content-Type: application/json

{"bucketPrefix":"dam-fmcg-pub","companyGln":"9300001000018","assetName":"ae284c29-f6da-491d-acab-b8c0145a4e3e-devil_128.png","fileName":"devil_128.png","assetUrl":"https://dam-api.atrify.com/content/dam-fmcg-pub/9300001000018/ae284c29-f6da-491d-acab-b8c0145a4e3e-devil_128.png","fileSize":16146,"contentType":"image/png","height":128,"width":128}

```

Example Request (v2):

```

GET
/v2/assets/dam-fmcg-pub/9300001000018/a5df3da0-a5cc-430e-aaca-fe2afff3de8

```

```
5-angel_128.png HTTP/1.1
Host: dam-api.atrify.com
Accept: application/json
```

Example Response (v2):

```
HTTP/1.1 200
Content-Type: application/json

{"object":"asset","ok":true,"timestamp":"2020-08-24T13:33:32.962334Z","data":{"bucketPrefix":"dam-fmcg-pub","companyGln":"9300001000018","assetName":"a5df3da0-a5cc-430e-aaca-fe2afff3de85-angel_128.png","fileName":"angel_128.png","assetUrl":"https://dam-api.atrify.com/content/dam-fmcg-pub/9300001000018/a5df3da0-a5cc-430e-aaca-fe2afff3de85-angel_128.png","metadata":{"size":21610,"contentType":"image/png","width":128,"height":128,"colorSpace":"RGB","colorDepth":32,"compressionType":"deflate","horizontalResolution":96,"verticalResolution":96}}}
```

5.3 Delete Digital Asset

Description: This will remove the digital asset at the URL. Caution should be used when deleting a digital asset, as the URL will result in a broken link if the URL is in use (e.g., by one of your trading partners).

DELETE `/v{version}/assets/{bucketPrefix}/{companyGln}/{assetName}`

Request Headers:

KEY	VALUE
Accept	*/*; application/json
Authorization (required)	Bearer [YOUR-JWT]

Path Parameters:

KEY	VALUE : TYPE
bucketPrefix (required)	[YOUR-BUCKET] :string
companyGln (required)	[YOUR-GLN] :string
assetName (required)	[ASSET-NAME] :string

Response Headers:

KEY	VALUE
Content-Type	application/json

Responses:

CODE	Description / Response
200	OK When the asset has been deleted.
401	Unauthorized
403	Forbidden
404	Not Found When the asset could not be found and therefore was not deleted.
500	Internal Server Error

Example Request:

```
DELETE
/v2/assets/dam-fmcg-pub/9300001000018/ae284c29-f6da-491d-acab-b8c0145a4e3
e-devil_128.png HTTP/1.1
Host: dam-api.atrify.com
Accept: */*
```

Example Response:

```
HTTP/1.1 200
Content-Length: 0
```

5.4 Metadata extraction details

5.4.1 Color space extraction mapping

ColorSpace	return Value	ColorSpace	return Value
ColorSpace.TYPE_XYZ	XYZ	ColorSpace.TYPE_4CLR	4CLR
ColorSpace.TYPE_Lab	Lab	ColorSpace.TYPE_5CLR	5CLR

ColorSpace.TYPE_Luv	Luv	ColorSpace.TYPE_6CLR	6CLR
ColorSpace.TYPE_YCbCr	YCbCr	ColorSpace.TYPE_7CLR	7CLR
ColorSpace.TYPE_Yxy	Yxy	ColorSpace.TYPE_8CLR	8CLR
ColorSpace.TYPE_RGB	RGB	ColorSpace.TYPE_9CLR	9CLR
ColorSpace.TYPE_GRAY	GRAY	ColorSpace.TYPE_ACLR	ACLR
ColorSpace.TYPE_HSV	HSV	ColorSpace.TYPE_BCLR	BCLR
ColorSpace.TYPE_HLS	HLS	ColorSpace.TYPE_CCLR	CCLR
ColorSpace.TYPE_CMYK	CMYK	ColorSpace.TYPE_DCLR	DCLR
ColorSpace.TYPE_CMY	CMY	ColorSpace.TYPE_ECLR	ECLR
ColorSpace.TYPE_2CLR	2CLR	ColorSpace.TYPE_FCLR	FCLR
ColorSpace.TYPE_3CLR	3CLR		

6 Validation API

This chapter describes the methods of the validation API.

6.1 Get Validation for Digital Asset

Description: Performs a validation of the asset metadata and returns the validation results for a given asset identified by the asset name.

The asset metadata is validated against the specified validation profile. The current supported profiles are 'FMCG' and 'GS1AT'.

Note: This method is only available in version v2 or later.

Note: This method is currently not available publicly.

GET	<code>/{version}/validations</code>
-----	-------------------------------------

Request Headers:

KEY	VALUE
Accept	application/json
If-Modified-Since (optional)	date + time in GMT

Path Parameters:

None

Query Parameters:

KEY	VALUE : TYPE
url (required)	:url
profile (required)	:string Supported values: FMCG, GS1AT
language	:string Supported values: de, en

Response Headers:

KEY	VALUE
Content-Type	application/json

Responses:

CODE	Description / Response
200	OK v2: { "object": "validation", "ok": true, "timestamp": "date-time", "data": { "profile": "string",

	<pre> "result": <VALID INVALID FAILED SKIPPED>, "messages": [{ "id": "string", "severity": <INFO WARNING ERROR FATAL>, "field": "string", "message": "string" }, ...] } </pre>
400	<p>Bad Request</p> <pre> { "object": "error", "ok": false, "timestamp": "YYYY-MM-DDTHH:mm:ss.SSSZ", "errorCode": 400, "message": "string", "additionalMessage": "string" } </pre>
401	Unauthorized
403	Forbidden
404	Not Found
422	<p>Unprocessable Entity</p> <pre> { "object": "error", "ok": false, "timestamp": "date-time", "errorCode": 422, "message": "string", "additionalMessage": "string" } </pre>
500	Internal Server Error

Example Request:

```

GET
/v2/validations?profile=GS1AT&url=https://dam-api.atrify.com/assets/dam-p
ub-fmcg/9300001000018/5add067e-a619-45c5-aac7-98252a059fba-07702018451548

```

```
A004xxx0102.jpg&language=de HTTP/1.1
Host: dam-api.atrify.com
Accept: application/json
```

Example Responses:

Asset is valid:

```
HTTP/1.1 200
Content-Type: application/json

{"object":"validation","ok":true,"timestamp":"2020-08-24T14:05:55.392275Z",
"data":{"profile":"GS1AT","result":"VALID","messages":[]}}
```

Asset is invalid:

```
HTTP/1.1 200
Content-Type: application/json

{"object":"validation","ok":true,"timestamp":"2020-08-24T14:20:53.748105Z",
"data":{"profile":"GS1AT","result":"INVALID","messages":[{"id":"BD6","severity":"ERROR",
"field":"verticalResolution","message":"Wert ist 0 aber muss größer oder gleich 300
sein"}, {"id":"BP4","severity":"ERROR","field":"clippingPathAvailable","message":
"Wert ist false aber muss wahr sein"}, {"id":"BD7","severity":"ERROR","field":"dimension",
"message":"Wert ist (width=151, height=600) aber Breite muss größer oder gleich 2000
sein"}, {"id":"BD5","severity":"ERROR","field":"horizontalResolution","message":
"Wert ist 0 aber muss größer oder gleich 300 sein"}, {"id":"BD1","severity":"ERROR",
"field":"mimeType","message":"Wert ist image/png aber muss einer sein aus [image/tiff,
image/jpeg]}]}}
```

Validation failed because of invalid request parameters:

```
HTTP/1.1 400
Content-Type: application/json

{"object":"error","ok":false,"timestamp":"2020-08-24T15:34:07.245802Z",
"errorCode":400,"message":"request could not be processed due to
unsupported parameters","additionalMessage":"Unsupported validation
profile"}
```

6.2 Digital Asset Validation Profiles

Currently there are two digital asset validation profiles FMCGProfile and GS1ATProfile. Each asset metadata with appropriate target market is validated against the profile.

DE Profile (related to DQG Rules):

- id = "FMCG.BD4" - The file format is $\{\text{validatedValue}\}$, but it must be one of {"image/tiff", "image/tif", "image/jpeg", "image/jpg"}.
- id = "FMCG.BD6" - The color space is $\{\text{validatedValue}\}$, but it must be {"RGB", "YCbCr"}.
- id = "FMCG.BD7" - $\{\text{validatedValue.width}\}$ px in width and $\{\text{validatedValue.height}\}$ px in height are indicated. However, the image size must be at least {2401} px in width $\{\text{conjunctive} == \text{true} ? \text{'and'} : \text{'or'}\}$ {2401} px in height.
- id = "FMCG.BD8-H" - The horizontal resolution is $\{\text{validatedValue}\}$ {unit} is, but it must be at least {300} {DPI}.
- id = "FMCG.BD8-V" - The vertical resolution is $\{\text{validatedValue}\}$ {unit} is, but it must be at least {300} {DPI}.
- id = "FMCG.BD8" - The file size is $\{\text{formatter.format}(\%1\$2f, \text{validatedValue})\}$ {unit} is, but it must not be larger than {25} {MB}.
- id = "FMCG.BD9" - The color depth is $\{\text{validatedValue}\}$ {unit}, but it must be at least {8} {bit}.

If there are available clipping paths then ClippingPathChecks :

- id = "FMCG.BD1" - A clipping path must be defined.
- id = "FMCG.BD3" - The clipping path must be closed.

GS1ATProfile :

- id = "BD1" - The file format is $\{\text{validatedValue}\}$, but it must be one of {image/tiff or image/jpeg}.
- id = "BD3" - The color depth is $\{\text{validatedValue}\}$ {unit}, but it must be at least {24} {bit}.
- id = "BD4" - The color space is $\{\text{validatedValue}\}$, but it must be {"RGB", "YCbCr"}.
- id = "BD5" - The horizontal resolution is $\{\text{validatedValue}\}$ {unit} is, but it must be at least {300} {DPI}.
- id = "BD6" - The vertical resolution is $\{\text{validatedValue}\}$ {unit} is, but it must be at least {300} {DPI}.

- id = "BD7" - $\{\text{validatedValue.width}\}$ px in width and $\{\text{validatedValue.height}\}$ px in height are indicated.

However, the image size must be at least {2401} px in width $\{\text{conjunctive} == \text{true} ? \text{'and'} : \text{'or'}\}$ {2401} px in height.

- id = "BD8" - The file size is $\{\text{formatter.format}(\text{'\%1$.2f'}, \text{validatedValue})\}$ {unit} is, but it must not be larger than {25} {MB}.

If there are available clipping paths then ClippingPathChecks :

- id = "BP4" - A clipping path must be defined.
- id = "BP2" - The clipping path must be closed.
- id = "BP3" - The clipping path consists of $\{\text{validatedValue}\}$ points. However, it must only consist of a maximum of {600} points.

7 Content API

This chapter describes the methods of the content API.

7.1 Get Content of Digital Asset

Description: Stream the content of the given asset. Get the digital asset as a file.

GET	<code>/\{version\}/content/\{bucketPrefix\}/\{companyGln\}/\{assetName\}</code>
-----	---

Request Headers:

KEY	VALUE
Accept (required)	image/* :MIME type
If-Modified-Since (optional)	:date

Path Parameters:

KEY	VALUE : TYPE
bucketPrefix (required)	[YOUR-BUCKET] :string
companyGln (required)	[YOUR-GLN] :string
assetName (required)	[ASSET-NAME] :string

Response Headers:

KEY	VALUE
Content-Type	image/*
Last-Modified	:date
Content-Disposition	inline; filename="[filename]"
Cache-Control	max-age=2592000, public

Responses:

CODE	Description / Response
200	OK
304	Not Modified
404	Not Found
500	Internal Server Error

Example Request:

```
GET
/v2/content/dam-fmcg-pub/9300001000018/2edb8a7c-de6e-48a4-90f6-57d8f52f63
ac-queen_128.png HTTP/1.1
Host: dam-api.atrify.com
Accept: image/*
```

Example Response:

```
HTTP/1.1 200
Last-Modified: Tue, 23 Jun 2020 08:05:25 GMT
Content-Disposition: inline; filename="queen_128.png"
Cache-Control: max-age=2592000, public
Content-Type: image/png
```

7.2 Get and Transform Content

Description: This is used to get the digital asset with a defined scale (height, width) and file type (jpg, jpeg, png, gif).

The asset name as provided in the metadata after the upload and the bucket prefix and company GLN have to be passed to identify the resource. The transformed content of the found image is streamed to the client, the If-Modified-Since header is respected and the asset is

stored with max-age=one month and as a public resource, which allows proxies to store it. The given asset must be an image. Valid values for type are 'jpg', 'jpeg', 'png' or 'gif'.

GET	<code>/{version}/content/transform/{bucketPrefix}/{companyGln}/{assetName}</code>
-----	---

Request Headers:

KEY	VALUE
Accept (required)	image/* :MIME type
If-Modified-Since (optional)	:date

Path Parameters:

KEY	VALUE : TYPE
bucketPrefix (required)	[YOUR-BUCKET] :string
companyGln (required)	[YOUR-GLN] :string
assetName (required)	[ASSET-NAME] :string

Query Parameters:

KEY	VALUE : TYPE
height (required)	:integer
width (required)	:integer
type (required)	:string Supported values : jpg, jpeg, png, gif

Response Headers:

KEY	VALUE
Content-Type	image/*
Last-Modified	:date
Content-Disposition	inline; filename="[filename]"
Cache-Control	max-age=2592000, public

Responses:

CODE	Description / Response
200	OK
304	Not Modified
404	Not Found
500	Internal Server Error

Example Request:

```
GET
/v2/content/transform/dam-fmg-pub/9300001000018/2edb8a7c-de6e-48a4-90f6-57d8f52f63ac-queen_128.png?width=50&height=50&type=png HTTP/1.1
Host: dam-api.atrify.com
Accept: image/*
```

Example Response:

```
HTTP/1.1 200
Last-Modified: Tue, 23 Jun 2020 08:05:25 GMT
Content-Disposition: inline; filename="queen_128.png"
Cache-Control: max-age=2592000, public
Content-Type: image/png
```

7.3 Get Content of External Asset

Description: Stream the content of the given asset. Get the external asset as a file

This will take an url of an external asset and save it into the digital asset management system. External assets stay in our cache for a limited time so this same logic is applied to asset content saved in the digital asset management.

Optional parameters:

- isBase64 - when set to true will decode url in Base64 (default false)
- Cache-Control - if it contains *no-cache* asset will not be cached

GET	/dam/{version}/content
-----	------------------------

Request Headers:

KEY	VALUE
-----	-------

Accept (required)	image/* :MIME type
-------------------	--------------------

Query Parameters:

KEY	VALUE : TYPE
url (required)	:string
isBase64	:boolean
Cache-Control	:string

Response Headers:

KEY	VALUE
Content-Type	image/*
Content-Disposition	inline
Cache-Control	max-age=0, public

Responses:

CODE	Description / Response
200	OK
304	Not Modified
404	Not Found
500	Internal Server Error

Example Request:

```
GET /dam/v1/content?url=urlPathToImage
Host: dam-api.atrify.com
Accept: image/*
```

Example Response:

```
Content-Disposition: inline
Cache-Control: max-age=0, public
Content-Type: image/png
```

7.4 Get Thumbnail of External Asset

Description: Create thumbnail for external asset. Get thumbnail as a file

This will take an url of an external asset, download it, scale it to an desired scale and save it into the digital asset management system as a .jpg. External assets stay in our cache for a limited time so this same logic is applied to asset content saved in the digital asset management.

Thumbnails can only be created for assets that are the type of:

- .jpeg
- .png
- .tiff
- .gif
- .pdf

Optional parameters:

- isBase64 - when set to true will decode url in Base64 (default false)
- Cache-Control - if it contains *no-cache* asset will not be cached
- xres - width of the thumbnail (default 100)
- yres - height of the thumbnail (default 100)

GET	/dam/{version}/content/thumbnail
-----	----------------------------------

Request Headers:

KEY	VALUE
Accept (required)	image/jpeg

Query Parameters:

KEY	VALUE : TYPE
url (required)	:string
isBase64	:boolean
Cache-Control	:string
xres	:integer
yres	:integer

Response Headers:

KEY	VALUE
Content-Type	image/jpeg
Content-Disposition	inline
Cache-Control	max-age=0, public

Responses:

CODE	Description / Response
200	OK
304	Not Modified
404	Not Found
500	Internal Server Error

Example Request:

```
GET /dam/v1/content/thumbnail?xres=300&yres=300&url=urlPathToImage
Host: dam-api.atrify.com
Accept: image/jpeg
```

Example Response:

```
Content-Disposition: inline
Cache-Control: max-age=0, public
Content-Type: image/jpeg
```

8 Utility API

This chapter describes the methods of the utility API.

8.1 Get Service Status

Description: Get the status to test the connection to the DAM web service

GET /{version}/status

Request Headers:

No headers required

Parameters:

No parameters required

Response Headers:

KEY	VALUE
Content-Type	text/plain; charset=UTF-8

Responses

CODE	Description / Response
200	OK "ok"
500	Service not available "error"

Example Request:

```
GET /v2/status HTTP/1.1
Host: dam-api.atrify.com
Accept: */*
```

Example Response:

```
HTTP/1.1 200
Content-Type: text/plain; charset=UTF-8

ok
```

8.2 Get Service Version

Description: Get the version of the DAM web service

GET /{version}/version

Request Headers:

KEY	VALUE
Accept	application/json

Parameters:

No parameters required

Response Headers:

KEY	VALUE
Content-Type	application/json

Responses:

CODE	Description / Response
200	OK < * >: string
500	Internal Server Error

Example Request:

```
GET /v2/version HTTP/1.1
Host: dam-api.atrify.com
Accept: application/json
```

Example Response:

```
HTTP/1.1 200
Content-Type: application/json

{"artifact":"dam-rest-service","bucket":"atrify-dam-prod","buildTime":"2020-04-20T09:47:33Z","commitTime":"2020-04-20T08:58:06Z","name":"DAM REST Service","commitId":"631890e","version":"3.0.3.31","branch":"master","buildNumber":"41"}
```

9 Authorization Mechanism

The DAM POST and DELETE requests are secured via JSON Web Token (JWT). If you want to access the API, you need to set an HTTP Authorization Header with value 'Bearer [YOUR-JWT]'.

If you just have the secret and key use your JWT secret and key to generate an access token.

EXAMPLE: (www.jwt.io Debugger)

```
Header:
{
  "alg": "HS256",
  "typ": "JWT"
}

Payload:
{
  "iss": "YOUR-KEY"
}

Verify Signature:
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  YOUR-SECRET
)
```

This configuration will generate an access token like this:

```
“eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJZT1VSLUthFWSJ9.qoPYMLGhudASYWr7RyT6MtYdffXAM6d9iPBVS65Jt0I”
```

Please change “YOUR-KEY” and “YOUR-SECRET” with your personal information. If you have any questions please contact support.

EXAMPLE REQUEST:

Inside the HTTP Request Header

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJZT1VSLUthFWSJ9.qoPYMLGhudASYWr7RyT6MtYdffXAM6d9iPBVS65Jt0I
```

Here's a sample curl invocation how to do a DAM file upload:

```
curl -ksSL \  
  -H "Accept: application/json" \  
  -H "Authorization: Bearer <your-jwt-token>" \  
  -X POST \  
  -F "file=@your-file.png" \  
  "https://dam-api-test.atrify.com/assets/dam-fmcg-pub-test/<your-gln>?file  
Name=your-file.png&mimeType=image%2Fpng"
```

APPENDIX - Supported File Types

This section lists the supported file types along with the MIME type that should be used in API calls.

Name	MIME Type	File Extension
JPEG images	image/jpeg	.jpeg, .jpg
Graphics Interchange Format	image/gif	.gif
Bitmap Image File	image/bmp	.bmp
Icon Image	image/vnd.microsoft.icon	.ico
Portable Network Graphics (PNG)	image/png	.png
Tagged Image File Format	image/tiff	.tiff
Waveform Audio File Format (WAV)	audio/x-wav	.wav
Microsoft Windows Media Audio	audio/x-ms-wma	.wma
MIDI - Musical Instrument Digital Interface	audio/midi	.midi
MPEG Audio	audio/mpeg	.mpga
MPEG-4 audio	audio/mp4	.m4a
Real Audio Sound	audio/x-pn-realaudio	.ram
RealMedia	application/vnd.rn-realmedia	.rm
Flash Video	video/x-flv	.flv
3GP/3GP2	video/3gpp; video/3gpp2	.3gp, 3g2
Audio Video Interleave (AVI)	video/x-msvideo	.avi
Microsoft Windows Media Video	video/x-ms-wmv	.wmv
MPEG Video	video/mpeg	.mpeg
MPEG4 Video	video/mp4	.mp4, .mov
Adobe Portable Document Format	application/pdf	.pdf
Digital multimedia file	video/webm	.webm
Zip Archive	application/zip	.zip

Contact Us

If you need assistance, please contact us at technical-support@1worldsync.com.