Cortex

API Documentation

Data Tables API v2.2

Last revision date: August 12, 2020

Table of Contents

Change Log

DATE	CHANGE
24 November 2015	Latest document date
20 January 2016	Introduced Change Log Added warning about curly inverted commas vs. straight inverted commas (section: “Assigning keywords with Roles in a multi-valued field”)
2 August 2016	Update the list of DataTables
29 September 2016	Typo correction page 3 (Documents.Audio.Default)
17 November 2016	Add information about the 'Read Only' DataTables: Contact and Document (irrespective of type)
21 December 2016	Add section on utility parameters (for Advanced Users/Developers)
29 November 2017	Corrections in Documents.All Read-Only Resource
30 January 2018	Update CoreField.Mother to CoreField.Parent-folder in “Moving a folder or an asset into another folder”
21 February 2018	Add section and use case examples on the datatable “Documents-links“
13 March 2018	Add date format including Time in section ‘READ’ and example in Use Case Scenarios for “Reading records created or edited before/after a given date/time”
21 January 2019	Add parameter “IndexInBackground”
09 July 2020	Added “Remove Relationships from Document to Keyword” and “Remove Relationships from Keyword to Keyword”
12 August 2020	Added “Assign Representative Image”
17 August 2020	Added “Specify the KeyType to Search for or Create Keywords”
27 August 2021	Added Identify Assets Based on Single Value Authority List Fields and Paired-Value Fields.

Introduction

The Data Table API is designed to map Cortex “objects” structure and allow CRUD (Create/Add Read Update Delete) operations on objects individually or by batches.

Objects are defined as Assets (Images, Videos, Multimedia), Keywords, Folders, virtual folders, Links between Assets and virtual folders, Links between Assets and Folders.

DataTable v2.2 are used with entities which have the custom forms activated (currently available on documents: assets, folders, etc.).

DataTable v.2.1 remains available for managing data in other DataTables: Keywords, Contacts, etc.

Authentication

Authentication to use the Data Table API can use a cookie or a token requested through the Authentication API.

DataTables (Resources)

https://www.sitename.com/API/DataTable/v2.2
→ lists all Data Table resources available on your installation.

IMPORTANT NOTE:

When some calls/DataTable are not yet available in v2.2, it is possible to mix DataTable API v2.1 and DataTable API v2.2 in the same query, by making a subcall (using bracket) within the main call.

Example:

- https://www.sitename.com/API/DataTable/V2.1/Contact.Client:Update?
  Contact.CoreField.OldNumber=12345
  &Contact.CoreField.Country:=[DataTable/v2.2/ReferenceTable.Country:Read?ReferenceTable.CoreField.OldID=BELG&ReturnField=ReferenceTable.CoreField.CountryCode]

DataTables for managing Documents

Examples:

- Managing Assets of type “**Image**” → https://www.sitename.com/API/DataTable/v2.2/Documents.Image.Default
  This resource allows you to manage (Create, Read, Update, Delete) Image records

- Managing Assets of type “**Video**” → https://www.sitename.com/API/DataTable/v2.2/Documents.Video.Default
  This resource allows you to manage (Create, Read, Update, Delete) Video records

- Managing Assets of type “**Audio”** → https://www.sitename.com/API/DataTable/v2.2/Documents.Audio.Default
  This resource allows you to manage (Create, Read, Update, Delete) Audio records

- Managing Assets of type “**Multimedia**” → https://www.sitename.com/API/DataTable/v2.2/Documents.Multimedia.Default
  This resource allows you to manage (Create, Read, Update, Delete) Multimedia records

- Managing Records of type “**Folders**” → https://www.sitename.com/API/DataTable/v2.2/Documents.Folder.Default
  This resource allows you to manage (Create, Read, Update, Delete) Folder records

- Managing Records of type “**Virtual Folder”** → https://www.sitename.com/API/DataTable/v2.2/Documents.Virtual-folder.Default
  This resource allows you to manage (Create, Read, Update, Delete) Virtual Folders records

Document Subtypes

If you have multiple subtype for any of the above assets.records, all subtypes will be listed. For example, if you have Folders of subtypes Event, Shoot, and Upload Folder, you will have the following list of DataTables for Folders:

- https://www.sitename.com/API/DataTable/v2.2/Documents.Folder.Event
- https://www.sitename.com/API/DataTable/v2.2/Documents.Folder.Shoot
- https://www.sitename.com/API/DataTable/v2.2/Documents.Folder.Upload-Folder

Documents.All Read-Only Resource

Security Function required: *Family API: Datatable API - “Document (Read only)”* (Code: APITableDocument)
WARNING: This security point gives access to all documents as it does not take into account Permissions/Restrictions/Embargo dates etc

https:/www.sitename.com/API/DataTable/v2.2/Documents.All:Read?CoreField.Identifier=[ID12345]

The Document.All resource is 'read only'. It is useful to retrieve information about a document, without knowing the document type (i.e. Image, Video, Folder, etc.).

You can use the following fields to search for the document: RecordID, CoreField.Identifier, CoreField.Id_Client, CoreField.OtherNum and CoreField.OriginalFileName.

If you know the document type, you can use the dedicated DataTable.

DataTables for managing Contacts

Examples:

- Managing Contacts of type “**Client**” → https://www.sitename.com/API/DataTable/v2.2/Contacts.Client.Default
- Managing Contacts of type “**Staff**” → https://www.sitename.com/API/DataTable/v2.2/Contacts.Staff.Default
- Managing Contacts of type “**Source**” → https://www.sitename.com/API/DataTable/v2.2/Contacts.Source.Default
- Managing Contacts of type “**Source agent**” → https://www.sitename.com/API/DataTable/v2.2/Contacts.Source-agent.Default
- Managing Contacts of type “**Agent**” → https://www.sitename.com/API/DataTable/v2.2/Contacts.Agent.Default
- Managing Contacts of type “**Billing account**” → https://www.sitename.com/API/DataTable/v2.2/Contacts.Billing-account.Default
- Managing Contacts of type “**Company**” → https://www.sitename.com/API/DataTable/v2.2/Contacts.Company.Default

Contacts.All Read-Only Resource

Security Function required: *Family API: Datatable API - Contact (Read only)*

https://www.sitename.com/API/DataTable/v2.2/Contacts.All:Read?Contact.CoreField.Email1=[EMAIL_ADDRESS]

The Contact.All resource is 'read only'. It is useful to retrieve information about a contact, without knowing the contact type (i.e. Client, Staff, Source, etc.).

If you know the contact type, you can use the dedicated DataTable.

DataTables for managing Keywords

Examples:

- Managing Keywords → https://www.sitename.com/API/DataTable/v2.2/Keywords
- Managing Links between Keywords (Thesaurus hierarchy) → https://www.sitename.com/API/DataTable/v2.2/Keywords-links

DataTables for managing Links between Documents

Example:

- Managing Links between assets and virtual folders → https://www.sitename.com/API/DataTable/v2.2/Documents-links

GET and POST requests

For API testing purposes, every action can be performed using POST or GET HTTP requests.

- The maximum URL length for a GET request is 260 characters.
- For POST requests, the parameters are passed in the HTTP header or in the URL. If the same parameter is present in both, the value in HTTP header will override the URL.

Data Table APIs operations and parameters

Each of these resources / data tables supports the following operations:

● ListFields	List the fields that are available for further read, update or add operations
● Read	Can perform a search based on a given set of criteria and return the requested fields for each matching object
● Update	To update the given fields with the given values for a given object based on its RecordID value
● Create	To create an object with the given fields and the given values. The RecordID of the newly created object is returned
● CreateOrUpdate	To create an object with the given fields and the given values or update an existing object if found in the data table
● Delete	To delete the object that has the given RecordID value

The operations are prefixed by a colon (**:**)

For example:
www.sitename.com/API/DataTable/V2.2/Documents.Image.Default:ListFields

www.sitename.com/API/DataTable/V2.2/Contacts.Client.Default:ListFields

LIST - Listing fields available for a given data table

ListFields will return the list of fields that are available for the given resource.

For example:
www.sitename.com/API/DataTable/V2.2/Documents.Image.Default:ListFields

will list all fields available for Assets of type "Default Image"

Documents.Image.Default is the default image type. Administrator of Cortex can create additional custom Asset Types of images. For example, if an custom asset type of Images is created — for example: 'Born Analog Image' — The syntax will then be: Documents.Image.Born-Analog-Image

The call to list its fields would then be:
www.sitename.com/API/DataTable/V2.2/Documents.Image.Born-Analog-Image:ListFields

The following attributes are included in the response:

Read-Only	<CanBeUsedAsAReadCriterion type="Boolean">True</CanBeUsedAsAReadCriterion>
Multi-Valued	<IsMultiValue type="Boolean">True</IsMultiValue>
Label	<Label> → The field label, as it appears in Cortex Administration screens
The field type	<DataType> → e.g. Text, DateTime, Boolean, Numeric, etc.
The field length in number of characters (for text fields)	<DataLength>
Whether the field is **based on a reference table**	IsBasedOnAuthorityTable <type="Boolean">False</IsBasedOnAuthorityTable>
Whether the **field is searchable**	<IsSearchable type="Boolean">True</IsSearchable>
Whether the **field is visible**	<IsVisible type="Boolean">True</IsVisible>

READ - Reading records

Read will return all fields for the requested objects.

Specifying filter conditions =

For example:
www.sitename.com/API/DataTable/V2.2/Documents.Image.Default:Read?CoreField.Identifier=DM123

Blank fields are not included in the responses unless ***&Verbose=1*** *has been specified in the query (see Verbose parameter).*

The following operators are available. You can use a field value, or a field name.

=	**Is equal to** Examples: www.sitename.com/API/DataTable/v2.2/Contact.Client:**Read?** Contact.CoreField.Last_Name=a* → will return clients with Last Name starting with the letter 'a' www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:**Read?** CoreField.System-ID=DM123&Verbose=1 → will return image DM123, with all fields including blank fields www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read? CoreField.CreateDate=CoreField.EditDate → will return assets where the CreateDate is the same as the EditDate
>	**Is greater than / Is posterior to** Examples: www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read? CoreField.CreateDate>2000-02-19 → will return assets created on or after 19th February 2000 Date format: Use one of these 2 ISO 8601 date formats: YYYY-MM-DD or YYYY-MM-DDTHH:MM:SS www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read? CoreField.EditDate>CoreField.ExpirationDate → will return assets where the EditDate is posterior to the Expiration Date
<	**Is smaller than / Is anterior to** Examples: www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read? CoreField.CreateDate<2000-02-19 → will return assets created on or before 19th February 2000 Date format: Use one of these 2 ISO 8601 date formats: YYYY-MM-DD or YYYY-MM-DDTHH:MM:SS www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read? CoreField.CreateDate<Document.CoreField.DocDate → will return assets created in the system before the date specified in the field “Document Date”

<>	**Is different from** Examples: www.sitename.com/API/DataTable/v2.2/Contacts.Client.Default:Read?CoreField.Country<>France → will return clients with a Country other than France www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read?CoreField.ModelReleaseExpirationDate<>CoreField.PropertyReleaseExpirationDate → return assets where the Model Release Expiration Date and the Property Release Expiration Date are different
!=	**Is different from** works the same way as <> above

UPDATE - Updating records

Update will update a record for given condition.

For example:
https://www.sitename.com/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123

Assigning and Removing operators

:= → The operator for assigning a value to a single value field

For example:
&CoreField.Title:=My new title

/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123&CoreField.Title:=My new title

To remove the value, make a call in which you do not assign anything:

&CoreField.Title:=

/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123&CoreField.Title:=

+:= → The operator for assigning a value to a single valued field based on an Authority List

For example:
&CustomNameSpace.Video-Format+:=PAL

/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123&CustomNameSpace.Video-Format+:=PAL

To remove the value, make a call in which you do not assign anything:

For example:
&CustomNameSpace.Video-Format+:=

/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123&CustomNameSpace.Video-Format+:=

+= → The operator for assigning a value to a multi valued field based on an Authority List (if value already exists in the Authority List)

For example:
&CoreField.Keywords+=Flower

/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123&CoreField.Keywords+=Flower

Using the above example, if the keyword “Flower” does not exist in your thesaurus you will get the following response from the API call:

```
<Error>
   The requested value was not found in the authority list and you did not request to create it if needed (operator ++=)
</Error>
```

-= → The operator to remove the value to a multi valued field based on an Authority List

For example:
&CoreField.Keywords-=Flower

/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123&CoreField.Keywords-=Flower

++= → The operator to create and assign a value to a multi valued field based on an Authority List

For example:
&CoreField.Keywords++=Flower

/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123&CoreField.Keywords++=Flower

Note:

- *The keyword type for keyword created with the operator ++= is “NewlyCreated”*
- *The application is setup by default to only add or remove keywords of type 'NewlyCreated' or ‘Common’*

--= → The operator to remove ALL the value assigned to a multi valued field based on an Authority List

For example:
&CoreField.Keywords--=

/API/DataTable/V2.2/Documents.Image.Default:Update?CoreField.Identifier=DM123&CoreField.Keywords--=

* See important information about Enable / Disable Automated Email when Creating/Updating User Accounts via API

CREATE - Creating records

Create will create a new record.

If the operation is successful, then the RecordID of the newly created object is returned.

If the operation failed, the response will include details of the fields with unexpected or required values.

For example:
/API/DataTable/V2.2/Documents.Image.Default:Create?CoreField.Title:=An API Created Record

* See important information about Enable / Disable Automated Email when Creating/Updating User Accounts via API

CREATEORUPDATE - Create Or Update records

CreateOrUpdate allows you to perform the Update and Add operations in one single operation.

The object is updated if found in the database. If it is not found, the object is created.

For example:
API/DataTable/V2.2/Documents.Image.Default:CreateOrUpdate?CoreField.Title=A Record Created by API&CoreField.Keywords+=Flower

Will Create Or Update image with Title = A Record Created by API & Add the Keyword Flower

* See important information about Enable / Disable Automated Email when Creating/Updating User Accounts via API

DELETE - Deleting records

Delete the object with the given RecordID or condition

For example:
/API/DataTable/V2.2/Documents.Image.Default:Delete?CoreField.Identifier=FD110408

Will delete image with Identifier FD110408

Encoding special characters in parameter values

Special characters in parameter values must be URL encoded using percent-encoding.

Refer to http://en.wikipedia.org/wiki/Percent-encoding for the percent-encoding of common characters, such as:

Special character	Percent-encoding
&	%26
=	%3D

NOTE: The percent-encoding of the “space” characters into %20 is optional.

For example

/API/DataTable/V2.2/Documents.Image.Default:Read?Title=Ben %26 Jerry&ShowAllFields=1

or

/API/DataTable/V2.2/Documents.Image.Default:Read?Title=Ben%20%26%20Jerry&ShowAllFields=1

where Ben %26 Jerry

or Ben%20%26%20Jerry

is Ben & Jerry

For a list of percent-encoding, visit http://www.w3schools.com/tags/ref_urlencode.asp (Column “From UTF-8”)

MaxRecordsAffected parameter

When trying to create or update more than 1 object in a single operation without using the MaxRecordsAffected parameter, you will get the following failure response : "*If executed, this operation would have updated [RECORD_COUNT] records. The current syntax authorizes the update of up to 1 record.*"

For example if you are trying to updated metadata on images based on Original filename, and there are multiple assets with this filename in your database.

To upgrade more than 1 record, use the parameter MaxRecordsAffected=100.

The maximum number of objects that can be created or updated in a single operation is 100.

For example:

www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:CreateorUpdate?CoreField.OriginalFileName=beach20150402.jpg&CoreField.DocMark:=E&MaxRecordsAffected=100

FieldLengthSafety parameter

If you create or update fields with values that exceed the maximum length for the field, you will get a failure response indicating the field concerned by the error and its maximum length. Example:

<Message>Update failed for the given records</Message>
<ErrorList>
<Document.CoreField.Title>This field cannot contain more than 200 characters</Document.CoreField.Title>
</ErrorList>

Note: Information about the max length for a given field can be retrieved using a ListFields call (attribute: <DataLength>)

Adding the parameter FieldLengthSafety=1 to an API call will:
- Update the fields by truncating the values provided to the max length.
- Append the full values provided to a Notes field

For example:

www.sitename.com/API/DataTable/v2.2/Documents.Video.Default:CreateorUpdate?CoreField.System-identifier=DMO1234&CoreField.Title:=This is such a long title that it exceeds 200 characters and would make the API call fail. A title really should be succinct, further details about the video would be better added to the Caption field. To truncate this title after the first 200 characters, use the Field Length Safety parameter&FieldLenghtSafety=1

Verbose parameter

Adding the parameter Verbose=1 to an API call will cause all fields of a record to be included in the response even if they are blank.

IndexInBackground parameter

By default, Cortex indexes newly created and updated items at the time of the API call, which means there will be a delay to account for processing time. Adding the parameter IndexInBackground=1 to an API call will speed up the API call by delaying item indexing.

For example:

https://www.sitename.com/API/Datatable/v2.2/Documents.Folder.Default:Create?CoreField.Title=My new folder &IndexInBackground=1

Data Table APIs Response

Standard response codes

The Data Table API returns the following response codes:

- Listing Fields:
    - If successful, the response returns the fields for the specified data table

- When Reading:
    - If successful, the response returns the fields for the specified object(s)
    - If the object/record is not found, the response returns 0 results

- When Updating:
    - Success

```
Response>
<RecordsAffected type="List">
<Result>
<RecordID>KRFQ7UHR39</RecordID>
<Code>SUCCESS</Code>
</Result>
</RecordsAffected>
```

- Record not found

```
<Response>
<RecordsAffected type="List"/>
<Code>RECORD_NOT_FOUND</Code>
<Message>Found no record with the given criteria</Message>
</Response>
```

- Update failed

```
<Response><RecordsAffected type="List"><Result>
<RecordID>KRFQ7UHR39</RecordID>
<Code>UPDATE_FAILED</Code>
<Message>Update failed for the given records</Message>
<ErrorList>
<Contact.CoreField.Email1>Email address is not valid</Contact.CoreField.Email1>
</ErrorList>
```

- When Creating:
    - Success

```
Response>
<RecordsAffected type="List">
<Result>
<RecordID>KRFQ7UHR39</RecordID>
<Code>SUCCESS</Code>
```

```
</Result>
</RecordsAffected>
```

- Creation failed

```
<Response>
<Code>CREATION_FAILED</Code>
<Message>Creation failed for the given record</Message>
<ErrorList>
<Contact.CoreField.Last_Name>This field must be filled</Contact.CoreField.Last_Name>
</ErrorList>
</Response>
```

- When Deleting:
  - Success

Failure messages

Error codes common to all Cortex REST APIs

Http return code	Message	Notes
401	Security violation	Your Security Profile does not give you access to the given API
403	This API cannot connect using a Secure Sockets Layer (SSL). Please change in your URL, HTTPS to HTTP	
403	This API can only connect using a Secure Sockets Layer (SSL). Please change in your URL, HTTP to HTTPS	
503	The server is currently in a state that does not allow API calls... [Client specific message]	
500	The server encountered an unexpected condition which prevented it from fulfilling the request. [Client specific message]	This is the fallback error message/code for all unhandled errors

Errors codes specific to the DataTable APIs

Http return code	Message	Notes
412	Update failed for the given records	
404	Found no record with the given criteria	
400	Wrong value set for parameter MaxRecordsAffected. Should be less than...	When trying to create or update 100 objects in a single operation using the 100MaxRecordsAffected parameter, but there are more than 100 objects concerned with the query condition
412	Creation failed for the given record	
400	Some parameters were missing while trying to call the requested function	
400	Some of the parameters don't have the expected type	
400	If executed, this operation would have updated [RECORD_COUNT] records. The current syntax authorizes the update of up to 1 record. To upgrade more than 1 record, please use the parameter MaxRecordsAffected	When trying to create or update more than 1 object in a single operation without using the MaxRecordsAffected parameter (note: the maximum number of objects that can be created or updated in a single operation is 100)
400	Some errors were found in the requested assignments or requested criteria. Please check messages.	A detailed description of the error for each parameter is included in the request summary. Including: ● The given field is unknown ● The given field is not searchable for Read operations

		● The given operator is not compatible with a single-value field. Please use assignment operator ":=" instead ● The given operator is not compatible with a multi-value field. Please use multi-value operators instead (+=, ++=, -=, --=) ● The given field is not supported ● The requested value was not found in the authority list and you did not request to create it if needed (operator ++=) ● The requested value was not found in the authority list and you don't have the right to create it ● An error occurred during the creation of the reference value
500	Failed	This is the fallback error message/code for all DataTable API unhandled errors

Additional information in the response

In addition to the HTTP code, the expected results, with the following additional information:

- API Request Information:
    - Resource requested (see section 3- Example Data Tables (Resources))
    - Boolean tag indicating the authentication state (IsLoggedIIn : True or False)
    - User Login
    - Timeout Period (in minutes)

- API Request Interpretation
    - Query field
    - Operator used with description of operator
    - Field value

- API Response Summary
    - Current page number
    - Number of results per page
    - Total results count
    - Link to the next page of results

For example:

```
<Result>
    <APIRequestInfo>
    <Module>DataTable</Module>
    <APIVersion>V2.2</APIVersion>
    <Resource>Documents.Image.Default:Create</Resource>
    <IsLoggedIn type="Boolean">True</IsLoggedIn>
    <ProviderVersion>5.5.20.036</ProviderVersion>
    <ProviderIdentity>OL-SERV-NAM</ProviderIdentity>
    <Status>LoggedIn</Status>
    <UserLogin>MyLoginName</UserLogin>
    <TimeoutPeriodMinutes type="Numeric">20</TimeoutPeriodMinutes>
    </APIRequestInfo>
    <RequestInterpretation>
    <Updates type="List">
        <Update>
            <Field>CoreField.Title</Field>
            <Operator>:=</Operator>
            <OperatorDescription>Assign a value to a single-value field</OperatorDescription>
            <Value>An API Created Record</Value>
        </Update>
        </Updates>
        </RequestInterpretation>
        <Response>
        <RecordID>29XDHUFLFLF</RecordID>
        <Code>SUCCESS</Code>
        </Response>
</Result>
```

JSON Parameter

Every CRUD operation has the same XML schema as a result. Only specific operations (such as creating links in the “Documents.Link” data table) will have a different result format.

The output is in XML by default, but can be turned into JSON using the URL parameter format=JSON.

For example:
www.sitename.com/API/DataTable/V2.2/Documents.Image.Default:Read?
CoreField.Identifier=DM123&format=JSON

SortField Parameter

Results are ordered by RecordID by default.

They can be sorted by any CoreField using the API parameter SortField.

Add **SortField=[FIELD_API_NAME]** to the URL.
Replace [FIELD_API_NAME] with the API name of the field to sort by.

For example:
www.sitename.com/API/DataTable/V2.2/Documents.Image.Default:Read?
CoreField.EditDate>2018-01-31
&SortField=CoreField.CreateDate

Use Case Scenarios

Listing fields

For Data Table of assets of type “Image” data table (Documents.Image.Default)

www.sitename.com/API/DataTable/V2.2/Documents.**Image**.Default:**ListFields**

Reading records created or edited before/after a given date/time

To get the list of Client record last edited at or after 10th March 2018 8pm:

www.sitename.com/API/V2.2/DataTable/Contacts.**Client**.Default:Read?
CoreField.EditDate>2018-03-10T20:00:00

To sort the results by EditDate, add &SortField=CoreField.EditDate:

www.sitename.com/API/V2.2/DataTable/Contacts.**Client**.Default:Read?
CoreField.EditDate>2018-03-10T20:00:00
&SortField=CoreField.EditDate

To get a list of Client records created on or before 31st December 2014:

www.sitename.com/API/V2.2/DataTable/Contacts.**Client**.Default:Read?
CoreField.CreateDate<2014-12-31

Create or Update an Image record based on your internal system identifier

www.sitename.com/API/DataTable/V2.2/Documents.**Image**.Default:CreateOrUpdate?
Document.CoreField.Id_Client**=**OL123456
&Document.CoreField.Title**:=**This is the title for this image
&Document.CoreField.CaptionLong**:=**This is the description for this image
&Document.CoreField.Title**:=**Title

Add keywords (multi-valued field) to an image (only if keyword already exists in authority list)

www.sitename.com/API/DataTable/V2.2/Documents.**Image**.Default:Update?
Document.CoreField.Id_Client=OL123456
&CoreField.Keywords**+=**Keyword1|Keyword2|Keyword3

Add keywords (multi-valued field) to an image (create keyword if does not exist

and assign)

www.sitename.com/API/DataTable/V2.2/Documents.**Image**.Default:Update?
Document.CoreField.Id_Client=OL123456
&CoreField.Keywords++=Keyword1|Keyword2|Keyword3

Remove one keyword from an image

www.sitename.com/API/DataTable/V2.2/Documents.**Image**.Default:Update?
Document.CoreField.Id_Client=OL123456&
&CoreField.Keywords-=Keyword1

Remove all keywords from an image

www.sitename.com/API/DataTable/V2.2/Documents.**Image**.Default:Update?+=
Document.CoreField.Id_Client=OL123456&
&CoreField.Keywords--=

Remove all keywords and add a keyword

www.sitename.com/API/DataTable/V2.2/Documents.**Image**.Default:Update?
Document.CoreField.Id_Client=OL123456
&CoreField.Keywords--=
&CoreField.Keywords+=Keyword4

Remove relationships from document to keyword

Note: This API may require an update to Cortex. If this API does not work, please submit a support request here to schedule a Cortex update.

https://www.sitename.com/API/DataTable/v2.1/Document.Keywords.Link:Delete?Document.CoreField.DO_RecordID=[DataTable/v2.2/Documents.Image.Default:Read?CoreField.Identifier=DMW116470]
&Document.CoreField.KW_RecordID=[DataTable/v2.1/Tags.Keyword:Read?Tags.CoreField.Keyword_English=Computer]

Remove relationships from keyword to keyword

Note: This API may require an update to Cortex. If this API does not work, please submit a support request here to schedule a Cortex update.

Keyword to Keyword (Tags,Keyword.Link) → https://www.sitename.com/API/DataTable/v2.1/Tags.Keyword.Link:Delete?
Tags.CoreField.KeyidFather=[DataTable/v2.1/Tags.Keyword:Read?Tags.CoreField.Keyword_English=Computer]
&Tags.CoreField.KeyidSon=[DataTable/v2.1/Tags.Keyword:Read?Tags.CoreField.Keyword_English=PC

Specify the KeyType to Search for or Create Keywords

To specify the KeyType when searching for or creating Keywords (via API and referencing a CSV column), use a call like this:

https://www.sitename.com/API/v2.2/DataTable/Documents.All:Update?Corefield.OriginalFileName=[File Name]
&CoreField.Keywords++=[Keywords]

You need to edit the values in the CSV to specify the KeyType for search and creation. For example, find “|” in the Keywords column, and replace it with:

{"SearchTypes":"CustomCommon","CreationType":"CustomCommon"}|

In this situation, you would also add this to the last Keyword in each row:

{"SearchTypes":"CustomCommon","CreationType":"CustomCommon"}

Moving a folder or an asset into another folder

- Specify which Folder/Asset you want to Update or CreateAndUpdate (i.e. which Folder/Asset you want to move to another Parent Folder)
- Specify the new Parent Folder using the field: CoreField.Mother

Example for moving an image to another parent folder:

```
www.sitename.com/API/DataTable/V2.2/Documents.Image.Default:Update?
CoreField.Identifier=DM5555
&CoreField.Parent-folder:=[Documents.Folder.Default:CoreField.Unique-Identifier=DMSTO122]
```

Example for moving a folder to another parent folder:

```
www.sitename.com/API/DataTable/V2.2/Documents.Folder.Default:Update?
CoreField.Identifier=DMSTO123
&CoreField.Parent-folder:=[Documents.Folder.Default:CoreField.Unique-Identifier=DMSTO122]
```

The call in between the [brackets] is a sub call to locate the Parent Folder in the database based on its Identifier(CoreField.Identifier) or other criteria.

Adding assets into virtual folders

- Specify the asset you want to add into the virtual folder (ChildRecords)
- Specify the Virtual Folder (ParentRecord)

Example for adding an image to a virtual folder:

```
www.sitename.com/API/DataTable/v2.2/Documents-links:ParentChildLink?
ParentRecord=[Documents.Folder.Default:CoreField.Unique-Identifier=DMSTO122]
&ChildRecords=[Documents.Image.Default:CoreField.Unique-Identifier=123456789]
```

Example for adding a video to a virtual folder:

```
www.sitename.com/API/DataTable/v2.2/Documents-links:ParentChildLink?
ParentRecord=[Documents.Folder.Default:CoreField.Unique-Identifier=DMSTO122]
```

&ChildRecords=[Documents.Video.Default:CoreField.Unique-Identifier=9876543]

The calls in between the [brackets] are sub calls to locate the asset and Virtual Folder in the database, based on their Unique Identifier(CoreField.Unique-Identifier) or other criteria.

If you want to add a large number of assets to the Virtual Folder you can use iAPI for batch applying the API call to all records in a CSV file.

Designating a Lead asset and relating other assets as alternative versions of this Lead asset

AssignLeadDocument can be used to assign the “Lead asset” field to several assets at once.

It takes 2 parameters:

- LeadIdentifier: the identifier of the Lead asset
- TrailIdentifiers: the identifiers of the assets to assign as related versions, separated by commas.

For example:
www.sitename.com/API/DataTable/V2.2/Documents.Image.Default:AssignLeadDocument?
LeadIdentifier=DM123
&TrailIdentifiers=DM456

Assign Representative Image

The Assign Representative Image API call allows you to assign a Representative Image to a specific folder. This call uses two parameters:

- CoreField.ID_Client: The folder’s Legacy Identifier. This is located on the Details tab of a folder.
- CoreField.Identifier: The asset’s Unique Identifier. This is located on the Details tab of an asset.

www.sitename.com/API/DataTable/v2.2/Documents.Folder.Default:Update?CoreField.ID_Client=CTLARCH0150&CoreField.Representative_DO:=[DataTable/v2.2/Documents.Image.Default:Read?CoreField.Identifier=CTL39449]

The call between the brackets is a sub call to locate the image based on the image’s Unique Identifier (CoreField.Identifier).

Assign the Source to an asset (Contact of type Photographer / Creator / Source)

www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:CreateorUpdate?
CoreField.OriginalFileName=beach 20150215.jpg
&CoreField.CT_RecordidSource:=[Contacts.Source.Default:CoreField.Identifier=DMOCT12]

The call in between the [brackets] is a sub call to locate the Photographer in the database based on its FastID (CoreField.Identifier) or other criteria (e.g. email)

Assign Keywords with Roles in a multi-valued field

www.sitename.com/API/DataTable/v2.2/Documents.Video.Default:CreateorUpdate?
CoreField.System-Identifier=DM23478
&Entities.Cast++=Chantal Muller{"Role":"Animation"}

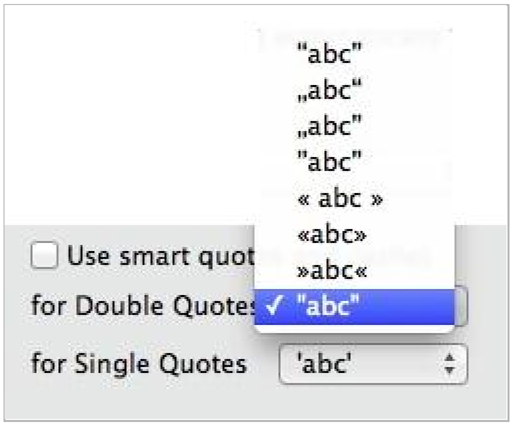
Example with multiple keywords/roles:

www.sitename.com/API/DataTable/v2.2/Documents.Video.Default:CreateorUpdate?
CoreField.System-Identifier=DM23478
&Entities.Cast++=Elsa{"Role":"Actor"}|Olaf{"Role":"Special effects"}

This will populate both the keyword list and roles list on the fly (operator ++=)

If you only want to add the keyword if the role already exists, use the operator +=

Warning: the above calls will NOT work with the curly quotation mark/inverted comma (“). Make sure you are using the straight quotation mark (").
Your keyboard may be set up to automatically convert double and single quotes to the curly characters. On a Mac, the feature “Use smart quotes and dashes” is found under System Preferences > Keyboard > Text.



□ Use smart quot

for Double Quotes ✓ "abc"

for Single Quotes 'abc'

Set Purposes on Large amount of Image assets using DataTable and iAPI

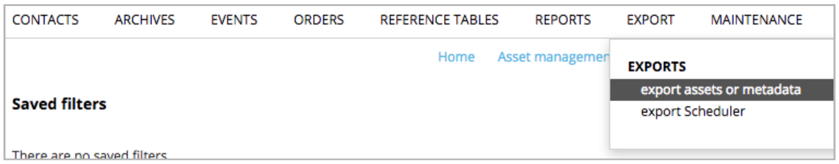
Scenario: Thousands of Images in the DAM need to be repurposed - too many to do through front end AI

- (Windows Only!) - Before you begin you will need the iAPI tool (download from the “Cortex Applications” page on your website)

- iAPI documentation: here

Step 1 - Export the Metadata BASED on a saved search

Go to Cortex Administration -> Exports -> export assets or metadata



Click ->

Which will bring up:

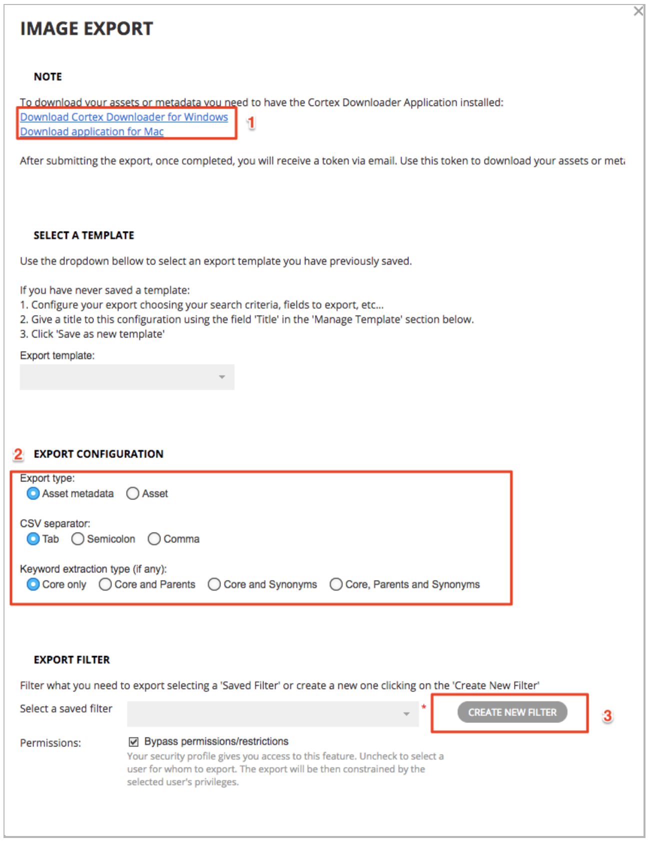


IMAGE EXPORT

NOTE

To download your assets or metadata you need to have the Cortex Downloader Application installed:
Download Cortex Downloader for Windows
Download application for Mac

1

After submitting the export, once completed, you will receive a token via email. Use this token to download your assets or met

SELECT A TEMPLATE

Use the dropdown bellow to select an export template you have previously saved.

If you have never saved a template:
1. Configure your export choosing your search criteria, fields to export, etc...
2. Give a title to this configuration using the field 'Title' in the 'Manage Template' section below.
3. Click 'Save as new template'

Export template:

2 EXPORT CONFIGURATION

Export type:
● Asset metadata ○ Asset

CSV separator:
● Tab ○ Semicolon ○ Comma

Keyword extraction type (if any):
● Core only ○ Core and Parents ○ Core and Synonyms ○ Core, Parents and Synonyms

EXPORT FILTER

Filter what you need to export selecting a 'Saved Filter' or create a new one clicking on the 'Create New Filter'

Select a saved filter *

CREATE NEW FILTER 3

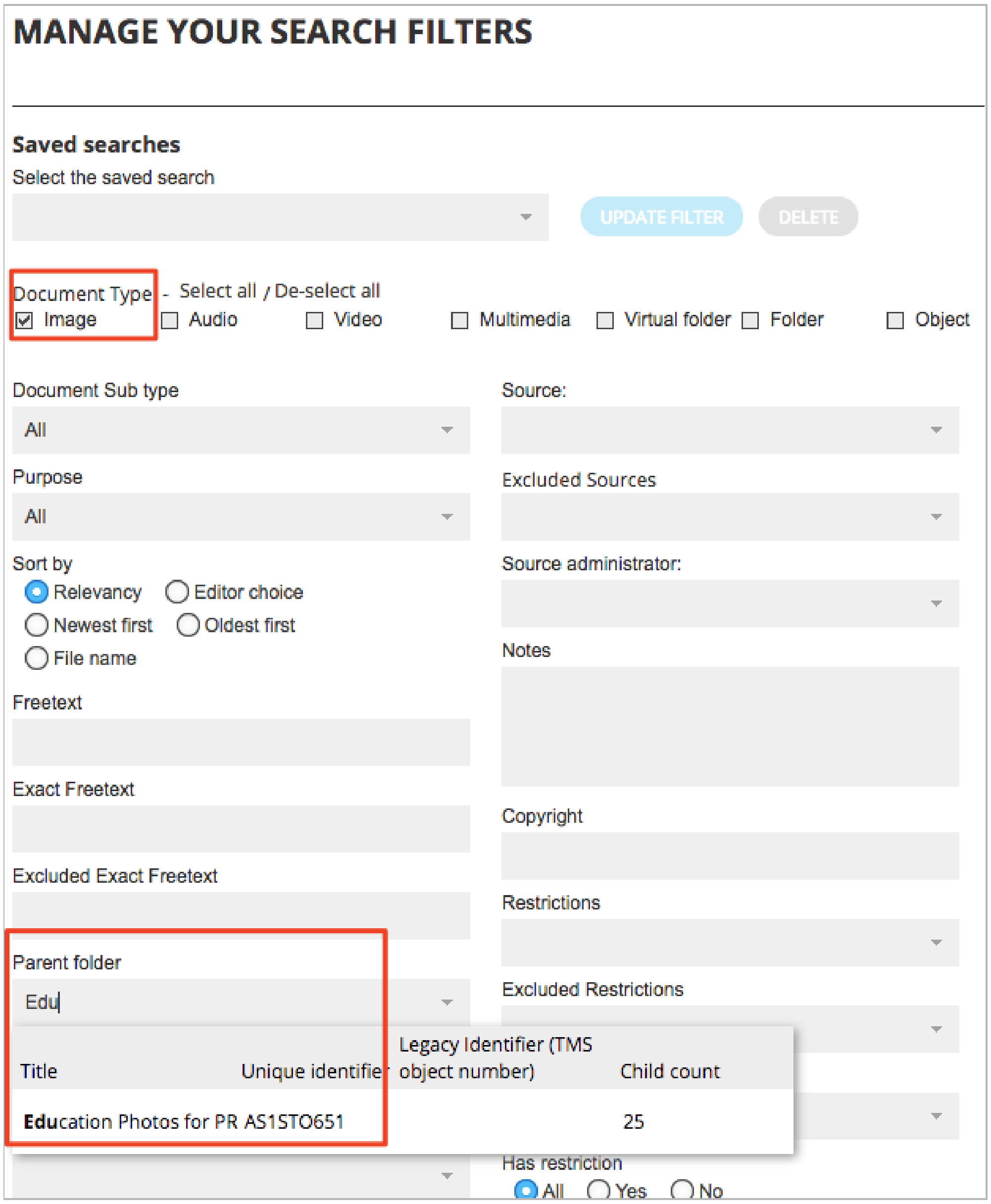
Permissions: ☑ Bypass permissions/restrictions
Your security profile gives you access to this feature. Uncheck to select a user for whom to export. The export will be then constrained by the selected user's privileges.

For above screenshot:
1 - Be sure to have the Cortex Downloader installed on your machine. (download from the “Cortex Applications” page on your website)

2 - Make sure that the "Export Type" is set to "Asset Metadata" under Export Configuration. (Leave other radio buttons set to default).

3 - "Export Filter" - Click CREATE NEW FILTER and do the following:

Configure a Filter to grab ALL *image* assets contained in the top level folder you want to capture asset metadata.



MANAGE YOUR SEARCH FILTERS

Saved searches

Select the saved search

Document Type - Select all / De-select all

☑ Image ☐ Audio ☐ Video ☐ Multimedia ☐ Virtual folder ☐ Folder ☐ Object

Document Sub type: All

Source:

Purpose: All

Excluded Sources

Sort by

● Relevancy ○ Editor choice
○ Newest first ○ Oldest first
○ File name

Source administrator:

Notes

Freetext

Exact Freetext

Copyright

Excluded Exact Freetext

Restrictions

Parent folder: Edu

Excluded Restrictions

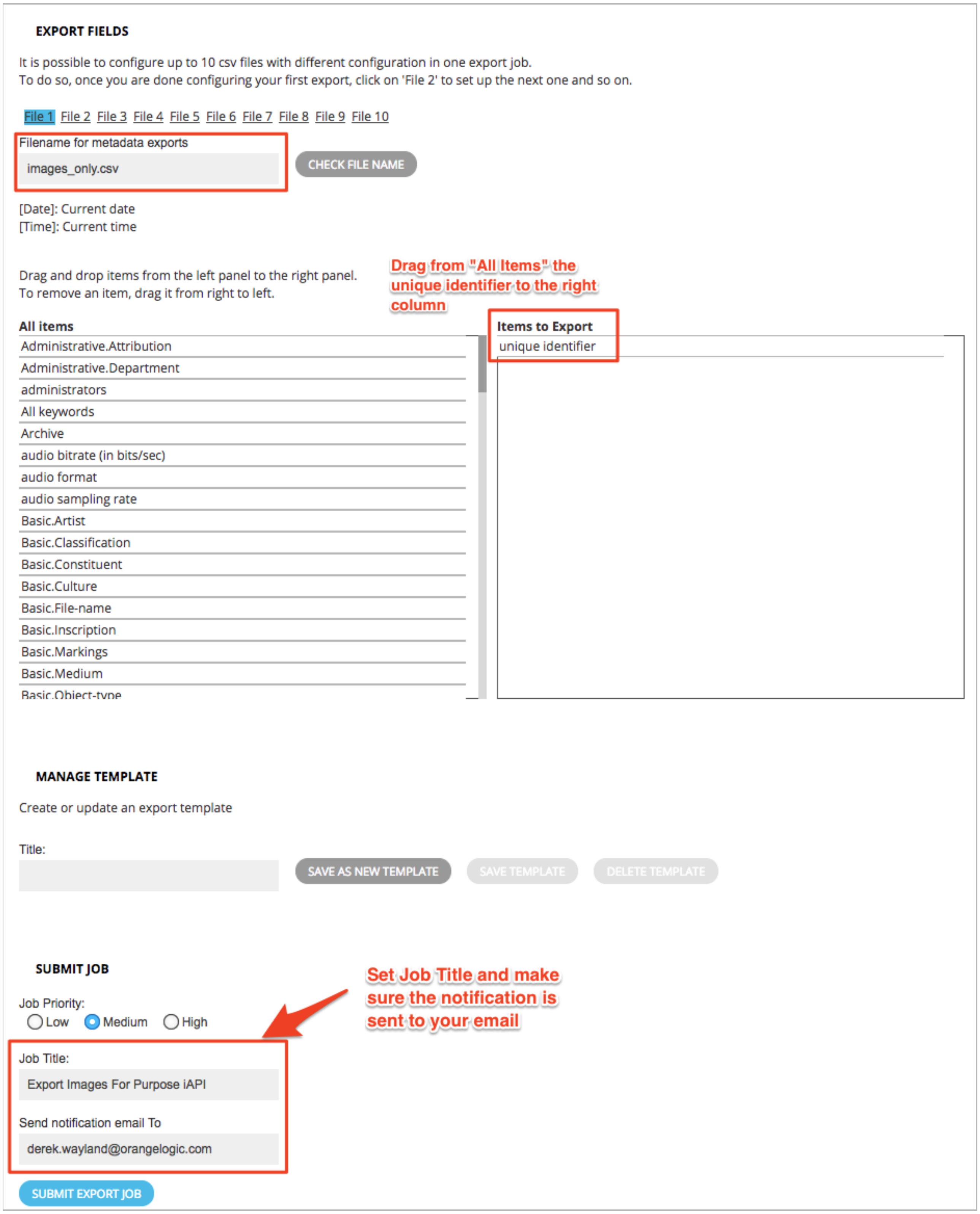
Title	Unique identifie	Legacy Identifier (TMS object number)	Child count
Education Photos for PR AS1STO651			25

Has restriction

● All ○ Yes ○ No

Click CREATE NEW FILTER at the bottom and save your filter.

4 - Complete the "Export metadata as follows"



EXPORT FIELDS

It is possible to configure up to 10 csv files with different configuration in one export job.
To do so, once you are done configuring your first export, click on 'File 2' to set up the next one and so on.

File 1 File 2 File 3 File 4 File 5 File 6 File 7 File 8 File 9 File 10

Filename for metadata exports: images_only.csv

CHECK FILE NAME

[Date]: Current date
[Time]: Current time

Drag and drop items from the left panel to the right panel.
To remove an item, drag it from right to left.

Drag from "All Items" the unique identifier to the right column

All items	Items to Export
Administrative.Attribution	unique identifier
Administrative.Department	
administrators	
All keywords	
Archive	
audio bitrate (in bits/sec)	
audio format	
audio sampling rate	
Basic.Artist	
Basic.Classification	
Basic.Constituent	
Basic.Culture	
Basic.File-name	
Basic.Inscription	
Basic.Markings	
Basic.Medium	
Basic.Object-type	

MANAGE TEMPLATE

Create or update an export template

Title:

SAVE AS NEW TEMPLATE SAVE TEMPLATE DELETE TEMPLATE

SUBMIT JOB

Job Priority:
○ Low ● Medium ○ High

Set Job Title and make sure the notification is sent to your email

Job Title: Export Images For Purpose iAPI

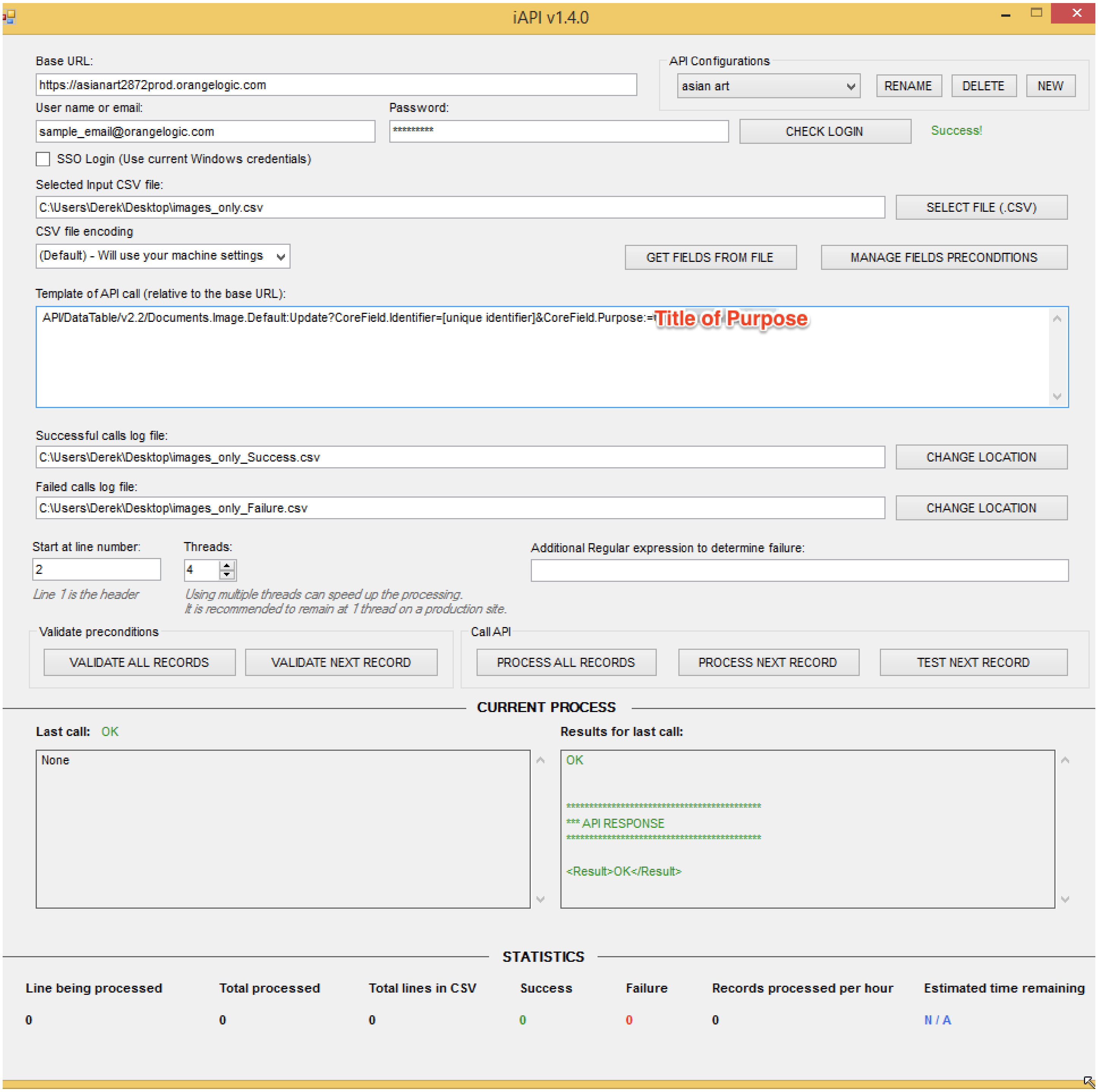
Send notification email To: derek.wayland@orangelogic.com

SUBMIT EXPORT JOB

This will be put into a queue which will generate a CSV file - Once it is complete it will email you (may take a few mins -> longer depending on asset count

Step 2 - Populate iAPI call with template API Call and account details.

Fill in the URL and credentials for your user for your site.
Take the “images_only.csv” metadata file and import it into API.



Base URL: https://asianart2872prod.orangelogic.com

API Configurations: asian art

User name or email: sample_email@orangelogic.com

Password: *********

Success!

□ SSO Login (Use current Windows credentials)

Selected Input CSV file: C:\Users\Derek\Desktop\images_only.csv

CSV file encoding: (Default) - Will use your machine settings

Template of API call (relative to the base URL):

```
API/DataTable/v2.2/Documents.Image.Default:Update?CoreField.Identifier=[unique identifier]&CoreField.Purpose:=
```

Successful calls log file: C:\Users\Derek\Desktop\images_only_Success.csv

Failed calls log file: C:\Users\Derek\Desktop\images_only_Failure.csv

Start at line number: 2

Line 1 is the header

Threads: 4

Using multiple threads can speed up the processing.
It is recommended to remain at 1 thread on a production site.

Additional Regular expression to determine failure:

Last call: OK

None

Results for last call:

```
OK

*********************************************
*** API RESPONSE
*********************************************

<Result>OK</Result>
```

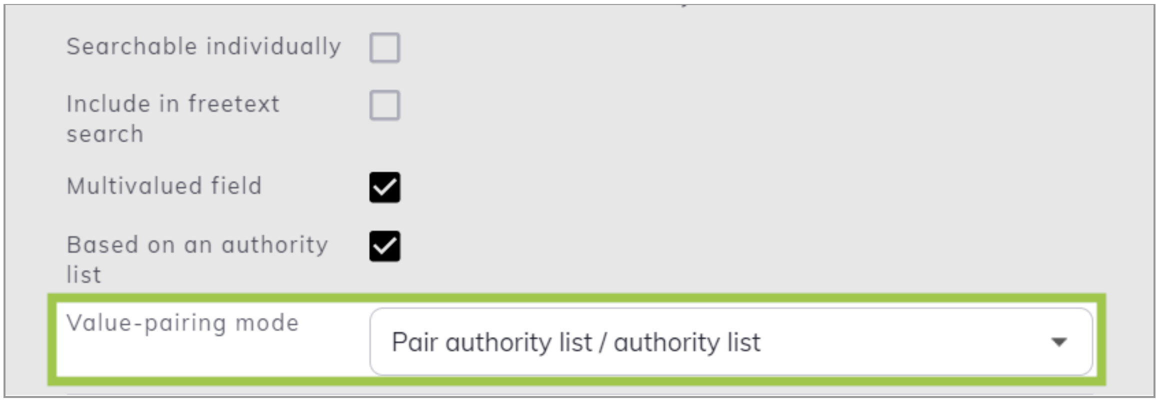
Line being processed	Total processed	Total lines in CSV	Success	Failure	Records processed per hour	Estimated time remaining
0	0	0	0	0	0	N / A

Replace the API Template with the following (In this example we will be using Pending Process)
/API/DataTable/v2.2/Documents.Image.Default:Update?CoreField.Identifier=[unique identifier]&CoreField.Purpose:=Pending Process

Identify Assets Based on Single Value Authority List Fields and Paired-Value Fields

In Cortex, the items in drop-down menus (on the Details tab, for example) are sets of Tags. A set of Tags in a drop-down menu is called an Authority List. Authority Lists can have one set of metadata (Single Value Authority List field) or two sets of metadata (Paired-Value field). For more information on these fields, go here.

Beginning in Cortex Yokohama, you can call an API to identify an asset based on the values in Single Value Authority List fields or Paired-Value fields. These fields are determined by the Value-Pairing Mode setting. These APIs only support returning a single asset. This means that the values in that asset must be unique.



Searchable individually □

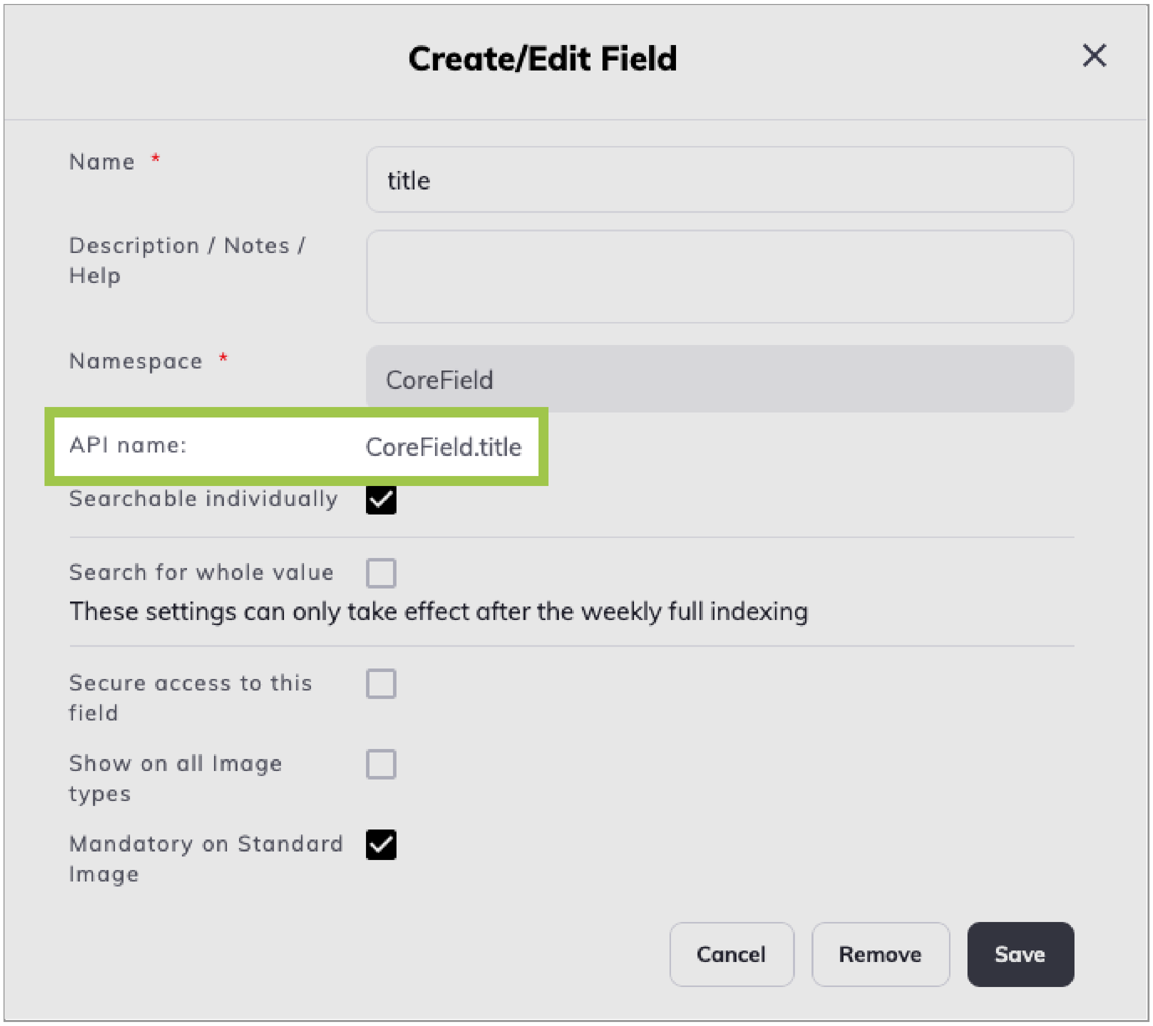
Include in freetext search □

Multivalued field ☑

Based on an authority list ☑

Value-pairing mode: Pair authority list / authority list

The Paired-Value APIs use the **Field API Name** parameter. This parameter is the API Name that Cortex generates based on the field's Namespace and Name. This is located in the **Create/Edit Field** window when you are in **Edit Layout Mode**.



Create/Edit Field

Name * title

Description / Notes / Help

Namespace * CoreField

API name: CoreField.title

Searchable individually ☑

Search for whole value □

These settings can only take effect after the weekly full indexing

Secure access to this field □

Show on all Image types □

Mandatory on Standard Image ☑

Query a Paired Authority List / Role Paired-Value Field

Examples for querying a **Paired Authority List / Role** Paired-Value field:

- For folders: www.sitename.com/API/DataTable/v2.2/Documents.Folder.Default:Read?[Field API Name]=keyword1{“Role”:”Value”}
- For images: www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read?[Field API Name]=keyword1{“Role”:”Value”}

Replace the **Value** parameter with the Role you want to query. For example, if you want to search for the **Director** role, the end of the API would look like this: **=keyword1{“Role”:Director”}**.

Query a Paired Authority List / Paired Authority List Paired-Value Field

Examples for querying a **Paired Authority List / Paired Authority List** Paired-Value field:

- For folders: www.sitename.com/API/DataTable/v2.2/Documents.Folder.Default:Read?[Field API Name]=keyword1{“LinkedKeyword”:”Keyword2”}
- For images: www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read?[Field API Name]=keyword1{“LinkedKeyword”:”Keyword2”}

Replace the **Keyword2** parameter with the Tag you want to query. For example, if you want to search for the Tag **In Lockdown**, the end of the API would look like this: **=keyword1{“LinkedKeyword”:“In Lockdown”}**.

Query a Paired Authority List / Free-Text Field or a Free-Text Field / Paired Authority List Paired-Value Field

Examples for querying a **Paired Authority List / Free-Text Field** or a **Free-Text Field / Paired Authority List** Paired-Value field:

- For folders: www.sitename.com/API/DataTable/v2.2/Documents.Folder.Default:Read?[Field API Name]=keyword1[Field API Name]=keyword{“LinkedText”:”Text”}
- For images: www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read?[Field API Name]=keyword1[Field API Name]=keyword{“LinkedText”:”Text”}

Replace the **Text** parameter with the free-text content you want to query. For example, if you want to search for the Director ID **9258**, the end of the API would look like this: **=keyword{“LinkedText”:”9258”}**.

Query a No Pairing Paired-Value or a Single Value Authority List Field

Examples for querying a **No Pairing** Paired-Value field or a **Single Value Authority List** field:

- For folders: www.sitename.com/API/DataTable/v2.2/Documents.Folder.Default:Read?[Field API Name]=Keyword1
- For images: www.sitename.com/API/DataTable/v2.2/Documents.Image.Default:Read?[Field API Name]=Keyword1

Replace **Keyword1** with the Tag you want to query. For example, if you want to search for the Tag **Embargo**, the end of the API would look like this: **=Embargo**.

Enable / Disable Automated Email when Creating/Updating User Accounts via API

If you are using APIs to create accounts on a Test or Pre-prod site using CSVs, the events that are generated by your calls are - by default - NOT sending email notifications.

If you DO want emails to be sent, add ***&AllowNotifications=1*** in your API call.

Examples :

1. You are creating new accounts by API : Event type “Administrator created a new Contact" will be triggered and logged, but will not send email notifications to new Contacts. To enable emails to be sent, add ***&AllowNotifications=1*** in your API call.
   Emails are sent to the email addresses specified in the field “Email to” field of your Event Type.

2. You are batch updating the “Editor’s Rating” for asset : Event type “Document level changed" will be triggered and logged, but will not send email notifications. To enable emails to be sent, add ***&AllowNotifications=1*** in your API call.
   Emails are sent to the email addresses specified in the field “Email to” field of your Event Type.
   Sample call:
   API/Datatable/v2.1/Document.Asset.Image:Update?Document.CoreField.Identifier=IMAGEID1&Document.CoreField.DocMark:=I&AllowNotifications=1

Utility Parameters (for Advanced Users/Developers)

UseSystemNames Parameter

Including the API parameter **UseSystemNames=1** will return the system name of all fields rather than the user defined names.

ReturnField Parameter

Including the API parameter **ReturnField=[FIELD_API_NAME]** will only return in the response that field and the encrypted recordID for matching records.

Only one field can be declared in one API call.

For example: To set the country at a contact level.

www.sitename.com/API/DataTable/V2.1/Contacts.Client.Default:Update?
Contact.CoreField.Email1=john.doe@orangelogic.com
&Contact.CoreField.Country:=
[DataTable/v2.1/ReferenceTable.Country:Read?ReferenceTable.CoreField.Country_English=France&**ReturnField=**ReferenceTable.CoreField.CountryCode]