



Overview

API Documentation

Quickstart Guide

C3.ai COVID-19 API Documentation (0.1)

Download OpenAPI specification: [Download](#)

This is the reference documentation for C3.ai COVID-19 HTTP RESTful API. The API request and responses are in JSON.

Please contribute your questions, answers and insights to [this Stack Overflow page for C3.ai COVID-19 Data Lake community](#) .

- Log in with your Stack Overflow credentials.
- In your posts please tag `c3ai-datalake` and mention that you are using the C3.ai COVID-19 Data Lake, so that others can view and help build on your contributions.

For support, please send email to: covid@c3.ai.

Quickstart Guide

Get started with downloadable R and Python Notebooks.

R

Before you proceed

Before proceeding, ensure that the following are installed on your computer:

- [R from CRAN](#), and
- [R Studio](#)

Download R Notebook

- [Download R Notebook](#)

Troubleshooting

While opening the R notebook (.Rmd file), if you see the error:

```
Unable to locate R binary by scanning standard locations.
```

then you probably did not install [R from CRAN](#). Make sure you install both [R Studio](#) and [R from CRAN](#).

Python

Before you proceed

Before proceeding, ensure that the following are installed on your computer (this quickstart guide is currently tested only on Mac OS):

- [Python 3 for Mac OS](#), and
- [Jupyter Notebook](#)

Download Python Jupyter Notebook

- [Download Python Jupyter Notebook](#)

Troubleshooting

- Ensure that you have Python 3 and not Python 2.7.
- While opening the Jupyter Notebook, if you see the error:

```
"Error loading notebook: An unknown error occurred while loading this notebook.  
This version can load notebook formats or earlier. See the server log for  
details."
```

then you can probably resolve this by installing [Python from Anaconda](#).

Release Notes

- [Release Notes for 0.1 \(click to expand\)](#)

Data from Multiple Sources

Using these APIs, you can pull together data from multiple COVID-19 data sources with a single API call. This is made possible by using C3.ai Types.

If you are new to the concept of a C3.ai Type, then it is easier to think of a C3.ai Type as an entity that holds the data. Using C3.ai Types makes it possible to programmatically interact with a unified, federated image of COVID-19 data.

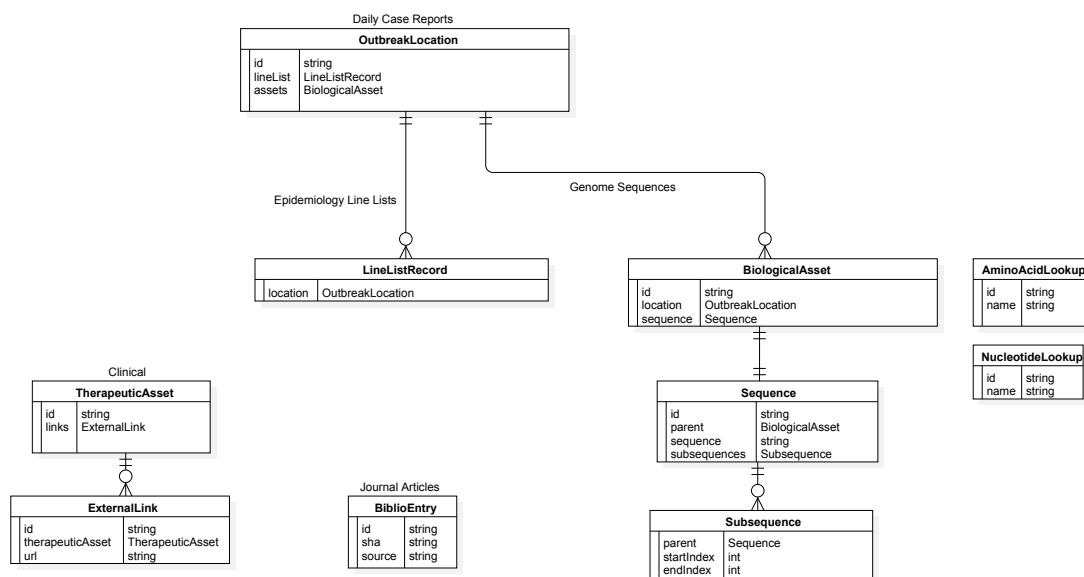
On this page, entries such as `OutbreakLocation`, `LineListRecord` are the names of C3.ai Types. Each C3.ai Type holds data of a certain kind. For example:

- `OutbreakLocation` stores location data such as countries, provinces, cities, where COVID-19 outbreaks are recorded, and
- `LineListRecord` stores individual-level information such as symptoms, travel history, reported onset, and discharge status from laboratory-confirmed COVID-19 patients.

While each such C3.ai Type holds the data of a particular kind, you can use these APIs to connect up the data from multiple C3.ai Types. For example, you can join the data

from two C3.ai Types, [BiologicalAsset](#) and [Sequence](#). This can be accomplished by using the `include` option in the fetch API call.

The following is an example entity relationship diagram showing how C3.ai Types are connected. Not all fields are shown in the below diagram. Refer to the **Fields** table for a C3.ai Type for a full listing of the fields for that C3.ai Type. See, for example, [LineListRecord](#).



Fetching from Multiple Sources

The `include` parameter is a powerful way to fetch data from multiple C3.ai Types. This parameter can also be used to fetch specific fields from a single C3.ai Type.

When you want to join data from two C3.ai Types, you make a fetch API call to one C3.ai Type, and use `include` in your request body to refer to the second C3.ai Type. The returned objects will contain fields and data from both the C3.ai Types.

See the section [Using include](#) for detailed examples showing how to use `include` to combine data from multiple C3.ai Types.

C3.ai APIs for COVID-19 Unified Data

The following APIs are currently supported:

- `fetch` (for all C3.ai Types presented here)
- `evalmetrics` (for [OutbreakLocation](#))
- `getarticlemetadata` (for [BiblioEntry](#))

NOTE: If you are new to the concept of RESTful API, this [Postman Learning Center](#) is a good place to start. All APIs described in this documentation can be verified using the Postman client.

The following table shows the APIs available for specific data sources (more data sources are being added):

Data Category	Data Source	C3.ai Types	APIs
Daily Case Reports	Johns Hopkins University	OutbreakLocation	<code>fetch</code> , <code>evalmetrics</code> .
Daily Case Reports	The COVID Tracking Project	OutbreakLocation	<code>fetch</code> , <code>evalmetrics</code> .
Daily Case Reports	The New York Times	OutbreakLocation	<code>fetch</code> , <code>evalmetrics</code> .
Daily Case Reports	European Centre for Disease Prevention and Control	OutbreakLocation	<code>fetch</code> , <code>evalmetrics</code> .
Daily Case Reports	World Health Organization (Situation Reports)	OutbreakLocation	<code>fetch</code> , <code>evalmetrics</code> .
Epidemiology Line Lists	nCoV-2019 Data Working Group	LineListRecord	<code>fetch</code> .
Epidemiology Line Lists	MOBS Lab	LineListRecord	<code>fetch</code> .
Clinical	Milken Institute	TherapeuticAsset	<code>fetch</code> .

Data Category	Data Source	C3.ai Types	APIs
Clinical	World Health Organization (COVID-19 R&D)	TherapeuticAsset	<code>fetch</code> .
Genome Sequences	National Center for Biotechnology Information Virus Database	BiologicalAsset, Sequence, Subsequence, AminoAcidLookup, NucleotideLookup	<code>fetch</code> .
Journals	CORD-19 Journal Articles	BiblioEntry	<code>fetch</code> , <code>getarticlemetad</code> <code>ata</code> .

Using C3.ai APIs

Required Headers

All C3.ai APIs described in this documentation must be used with the following header settings:

Headers	Setting
Accept	application/json
Content-Type	application/json

Using fetch

The request JSON for the `fetch` API should be used with the `filter` key. This `filter` key can be used in the fetch call to select any combination of the fields in the data. A few examples follow:

IMPORTANT: For a list of fields available for a C3.ai Type, refer to the fields section of that C3.ai Type in this document.

To fetch the data that match the specific values of the `id` field of the data:

```
{
  "spec" : {"filter": 'id == "Afghanistan"'}
}
```

A few other examples:

```
{
  "spec" : {"filter": 'id == "Afghanistan" && age == 45'}
}
```

or,

```
{ // See BiologicalAsset
  "spec": {
    "filter": "isolationSource == 'feces' && location == 'Japan'",
    "limit": -1
  }
}
```

or, using a `"contains(field, "string")"` format:

```
{ // See LineListRecord
  "spec": {
    "filter": "gender == 'male' && lineListSource == 'OPEN' && age <= 20 && cor
```

```
}  
}
```

The `fetch` API returns two main kinds of information in its response:

- The data from the C3.ai Type, fetched as an array of the objects.
- Metadata, or data describing data, such as:
 - The number of objects fetched.
 - Number of rows of information.
 - An indicator if more data exists in the C3.ai Type that was not returned.
- See the section [Limits](#) for the number of entries returned per fetch API call.

For full details on request and response JSON, see the several examples provided in the `fetch` API for all the C3.ai Types in this documentation.

Using include

In a C3.ai Type, the data type of a field can be a C3.ai Type. For example, the field `links` in `TherapeuticAsset` is of the type `ExternalLink`. This is how these two C3.ai Types are connected.

For example, to join data from these two connected C3.ai Types, `TherapeuticAsset` and `ExternalLink`, use the `include` parameter as follows:

- Make a `fetch` API call to `TherapeuticAsset`.
- The field `links` in `TherapeuticAsset` is of `ExternalLink` Type. Using the dot notation on the `links` field, you can access any field in the `ExternalLink`. For example, specifying `links.url` will resolve into `ExternalLink.url`, which will obtain the `url` field data from the `ExternalLink`.
- Notice that we have not issued a fetch call to `ExternalLink`. See the full fetch example, including the response objects, below.

Example 1: Join data from `TherapeuticAsset` and `ExternalLink` (click arrow to open)

► Example 1: Join data from `TherapeuticAsset` and `ExternalLink`

Example 2: Join data from BiologicalAsset and Sequence (click arrow to open)

In the following example, the `include` parameter is used with `this` keyword, which obtains all the fields from BiologicalAsset and the field `sequence` from Sequence.

- Join data from BiologicalAsset and Sequence

More filter Examples

- Filter on `dateTime`

```
{
  "spec" : {
    "filter": "field >= dateTime('YYYY-MM-DD')"
```

// Example: Fetch BiologicalAssets with collectionData post March 10, 2020.

```
{
  "spec" : {
    "filter": "collectionDate >= dateTime('2020-03-10')"
```

```
}
```

- Logical operators

```
// Note: Multiple filters can be applied with an "and" ("&&") or an "or" ("||")

// Operator "and".
{
```

```
"spec" : {  
  "filter": "fieldA == 'stringA' && fieldB == 'stringB'"  
}  
}  
  
// Example: Fetch BiologicalAssets with isolationSource 'feces' AND location '  
{  
  "spec": {  
    "filter": "isolationSource == 'feces' && location == 'Japan'"  
  }  
}  
  
// Operator "or".  
{  
  "spec" : {  
    "filter": "fieldA == 'stringA' || fieldB == 'stringB'"  
  }  
}  
  
// Example: Fetch BiologicalAssets with isolationSource 'feces' OR location 'J  
{  
  "spec": {  
    "filter": "isolationSource == 'feces' || location == 'Japan'"  
  }  
}
```

- Filter on **contains**

```
// Contains: Retrieve all entries, where the specified field contains matching  
{  
  "spec": {  
    "filter": "contains(field,'string')"  
  }  
}  
  
// Example: Fetch Biological Assets where genBankTitle contains "ORF10".  
{  
  "spec": {  
    "filter": "contains(genBankTitle,'ORF10')"  
  }  
}
```

- Filter on **startsWith**

```
// startsWith: Retrieve all entries, where the specified field starts with mat
{
  "spec": {
    "filter": "startsWith(field, 'string') "
  }
}

// Example: Fetch LineListRecords where location starts with "Snohomish".
{
  "spec": {
    "filter": "startsWith(location, 'Snohomish') "
  }
}
```

- Filter on `endsWith`

```
// EndsWith: Retrieve all entries where the specific field ends with matching
{
  "spec": {
    "filter": "endsWith(field, 'string') "
  }
}

//Example: Fetch LineListRecords where location ends with "UnitedStates".
{
  "spec": {
    "filter": "endsWith(location, 'UnitedStates') "
  }
}
```

- Filter on `empty`

```
// Empty: Retrieve all entries, where the specified field is NULL.
{
  "spec": {
    "filter": "empty(field) "
  }
}

//Example: Fetch LineListRecords without gender, age, or location data.
{
```

```

"spec": {
  "filter": "empty(gender) && empty(age) && empty(location)"
}
}

```

- Filter on `exists`

```

// Exists: Retrieve all entries where the specified fields exist.
{
  "spec": {
    "filter": "exists(field)"
  }
}

//Example: Fetch LineListRecords with gender, age, or location data.
{
  "spec": {
    "filter": "exists(gender) && exists(age) && exists(symptoms)"
  }
}

```

Using evalmetrics

While the `fetch` API returns the raw data, the `evalmetrics` API returns Timeseries data based on the metrics expression you provide. Metrics are instructions you can provide in the request JSON of the `evalmetrics` API for how to transform the data into Timeseries data. The `evalmetrics` API will then return the resulting Timeseries data.

The following fields are supported in the request JSON of the `evalmetrics` API:

IMPORTANT: Limits apply to these fields. See [Limits](#).

- **ids:** This is the list of source objects on which you want to evaluate the metrics on. For example: `"ids": ["King_Washington_UnitedStates", "SanDiego_California_UnitedStates"]`.

- **expressions:** Here you place a list of the metrics that you wish to evaluate. For example, `"expressions":["NYT_ConfirmedCases"]`. See the [table in Available Metrics section](#) for a list of supported metrics.
- **start** and **end:** The datetime fields where you can put the start and end dates of the period for which you want to evaluate your metrics. For example, `"start":"2020-03-01"` and `"end":"2020-03-30"`.
- **interval:** The desired interval for the Timeseries output. For example, `"interval":"DAY"`.

The response JSON for the `evalmetrics` API consists of the Timeseries data array, the timestamp array, and an array consisting of the fraction of data missing from the Timeseries.

IMPORTANT: The `evalmetrics` request body should be used with the `spec` key, just as how you would do in a `fetch` request.

Refer to the several examples provided in the `evalmetrics` API for [OutbreakLocation](#) for full details on response JSON.

Important Notes on evalmetrics

The following applies when using the `evalmetrics` API.

1. The `evalmetrics` API returns the cumulative number of daily cases for a given data source.
2. The metric `JHU_ConfirmedCasesInterpolated` (see [Available Metrics](#)) returns time series, which interpolate missing data points with the immediately previous non-missing value.
3. The `end` date field in the JSON request acts as an open interval. That is, if `end` is set to "2020-04-04", then only the data upto and including April 3rd is returned. If you need the data for April 4th, then you must set `end` date to "2020-04-05".
4. Data from a few locations, such as countries and states, are the aggregate of data from more granular locations, such as counties. For example, in the JHU Dataset, there is no data for the number of confirmed cases in California. This value is computed by adding the number of cases across all counties in California.

Using getarticlemetadata

This API is available only for [BiblioEntry](#). This API returns the full-text, in JSON, of the journal article from the CORD-19 dataset. See [GetArticleMetadata](#).

Location IDs

The location IDs that should be used in the `id` field (in `fetch` API) and `ids` field (in `evalmetric` API) for these APIs are available for download in [this Microsoft Excel document](#).

Limits

Fetch Limits

API	Limit
BiblioEntry: <code>/api/1/biblioe</code> <code>ntry/fetch</code>	If <code>limit</code> is not specified in the request body, then by default 2000 rows, or less if the available data is less than 2000 rows, are returned. This is also the maximum rows returned even if <code>limit</code> setting is higher, or is -1. Use the <code>offset</code> parameter to fetch more rows. For example, see Example 6 for Python in LineListRecord .

API	Limit
LineListRecord: <code>/api/1/linelistrecord/fetch</code>	If <code>limit</code> is not specified in the request body, then by default 2000 rows, or less if the available data is less than 2000 rows, are returned. A maximum of 5000 rows is returned per API call even if <code>limit</code> setting is > 5000 or is -1. Use the <code>offset</code> parameter to fetch more rows. For example, see Example 6 for Python in LineListRecord .

Evalmetric Limits

Setting	Limit
<code>start</code>	Must be greater than or equal to 2019-12-20 (20th December 2019).
<code>end</code>	Must be less than today.
<code>interval</code>	Only <code>DAY</code> is supported currently.
<code>ids</code>	The number of <code>ids</code> specified in this array must be less or equal to 10.
<code>expressions</code>	The number of <code>expressions</code> specified in this array must be less or equal to 4.

Available Metrics

Evaluated metrics of COVID-19 datasets are available. The following table shows the worldwide regions for which these metrics exist currently. Use the expressions from the **Evalmetrics expressions** column in the `expressions` field of the request JSON of the `evalmetrics` API. For example, `"expressions":["JHU_ConfirmedDeaths"]`.

Dataset	Evalmetrics expressions	Regions: Aggregation levels
---------	-------------------------	-----------------------------

Dataset	Evalmetrics expressions	Regions: Aggregation levels
Johns Hopkins University	<code>JHU_ConfirmedCases</code> , <code>JHU_ConfirmedDeaths</code> , <code>JHU_ConfirmedRecoveries</code> , <code>JHU_ConfirmedCasesInterpolated</code> (see Important Notes on evalmetrics)	United States: Country, State or Territory, and County level. Global: Country, Province level
The COVID Tracking Project	<code>CovidTrackingProject_ConfirmedCases</code> , <code>CovidTrackingProject_ConfirmedDeaths</code> , <code>CovidTrackingProject_ConfirmedHospitalizations</code> , <code>CovidTrackingProject_NegativeTests</code> , <code>CovidTrackingProject_PendingTests</code>	Global: Country, Province level
European Centre for Disease Prevention & Control	<code>ECDC_ConfirmedCases</code> , <code>ECDC_ConfirmedDeaths</code> ,	Global: Country level
The New York Times	<code>NYT_ConfirmedCases</code> , <code>NYT_ConfirmedDeaths</code>	United States: Country, State or Territory, and County level, Global: Country, Province level
WHO	<code>WHO_ConfirmedCases</code> , <code>WHO_ConfirmedDeaths</code>	Global: Country, Province (only applicable to China prior to March 15, 2020) level

OutbreakLocation

OutbreakLocation stores location data such as countries, provinces, cities, where COVID-19 outbreaks are recorded.

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
id	String	The location ID of the country, state or province, and county to fetch for the COVID-19 outbreak. Should be used with the key <code>filter</code> , for example: <code>"filter": 'id == "Afghanistan"'</code> . The location IDs that should be used in this <code>id</code> field (in <code>fetch</code> API) and <code>ids</code> field (in <code>evalmetric</code> API) for these APIs are available to download. See the section Location-IDs.
name	String	Actual name of the country, state or province, and county corresponding to the location ID.
fips	String	FIPS code for the country , and for the county and county-equivalents in the United States.
population2018	Integer	Population of the country, state or province, and county for the year 2018.
lineList	LineListRecord	C3.ai Type LineListRecord.
assets	BiologicalAsset	C3.ai Type BiologicalAsset.

Fetch

Examples (Click on the arrows to expand)

The following examples show how to fetch COVID-19 outbreak location data using this API.

IMPORTANT: The location IDs that should be used in the `id` field (in `fetch` API) and `ids` field (in `evalmetric` API) for these APIs are available to download. [See the section Location-IDs.](#)

- ▶ Fetch facts about Germany
- ▶ Fetch facts about Beijing, China (request example only)
- ▶ Fetch facts about Santa Clara County, California, United States (request example only)

HEADER PARAMETERS

Content-Type required	string Set this to application/json.
---------------------------------	---

Accept required	string Set this to application/json.
---------------------------	---

REQUEST BODY SCHEMA: application/json

spec ▾	object Container of query evaluation attributes
--------	--

filter	string Filter expression for which Obj instances to return. For example: <code>"filter": 'id == "Afghanistan" && age == 45'</code> . Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, and most non-timeseries functions supported by the expression engine."
--------	--

include	string Specifies which fields to bring back values for in the returned objects. A list of fields. For example: <code>"include": "productType, description, origin, links.url"</code> .
---------	---

limit	integer Maximum number of rows that should be returned, starting from offset.
-------	--

offset	integer <int32> (The Offset Schema) Number of rows to skip.
--------	--

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs >	object Container of query evaluation attributes
count	integer <int32> (The Count Schema) Number of rows returned.
hasMore	boolean (The Hasmore Schema) If set to true there were more objs that were not returned.

POST /api/1/outbreaklocation/fetch

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type

Content type
application/json

Copy Expand all Collapse all

```
{
  - "objs": {
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

Evalmetrics

Examples (Click on the arrows to expand)

The following examples show how to use the Evalmetrics API.

IMPORTANT: The location IDs that should be used in the `id` field (in `fetch` API) and `ids` field (in `evalmetric` API) for these APIs are available to download. [See the section Location-IDs.](#)

Johns Hopkins University

- ▶ Example 1: Total number of confirmed cases in the United States
- ▶ Example 2: Total number of confirmed deaths in California

- ▶ Example 3: Total number of confirmed recoveries in Hubei, China
- ▶ Example 4: Total number of confirmed cases, deaths, and recoveries in Santa Clara, California
- ▶ Example 5: Total number of confirmed cases in France and Germany
- ▶ Example 6: Total number of confirmed Cases, Deaths, and Recoveries in King County, Washington, and San Diego, California

The COVID Tracking Project

- ▶ Example 1: Total number of confirmed cases in Washington, United States
- ▶ Example 2: Total number of confirmed deaths in Washington, United States
- ▶ Example 3: Total number of confirmed hospitalizations in Washington, United States
- ▶ Example 4: Total number of negative test results in Washington, United States
- ▶ Example 5: Total number of pending test results in Washington, United States

New York Times

- ▶ Example 1: Total number of confirmed cases in the United States
- ▶ Example 2: Total number of confirmed deaths in New York
- ▶ Example 3: Total number of confirmed cases and deaths in Cook County, Illinois
- ▶ Example 4: Total number of confirmed cases and deaths in Pennsylvania and Ohio, United States

European CDC

- ▶ Example 1: Total number of confirmed cases in Italy
- ▶ Example 2: Total number of confirmed deaths in United Kingdom

WHO

- ▶ Example 1: Total number of confirmed cases in South Korea
- ▶ Example 2: Total number of confirmed deaths in France

HEADER PARAMETERS

Content-Type <i>required</i>	string Set this to application/json.
---------------------------------	---

Accept <i>required</i>	string Set this to application/json.
---------------------------	---

REQUEST BODY SCHEMA: application/json

spec ▾

object

Container of time series evaluation attributes

ids

array

Unique identifiers (IDs) for the sources to evaluate. Either IDs or a filter is required for evaluating a metric. For example, ["Hubei_China", "NewYork_US"].

expressions

array

The expressions to evaluate. For example: ["JHU_ConfirmedCases", "JHU_ConfirmedDeaths", "JHU_ConfirmedRecoveries"].

interval

string

Interval of the data to be returned. For example: 'DAY'.

start

string

Start datetime of the time range. For example: '2020-03-01'.

end

string

End datetime of the time range. For example: '2020-03-30'. IMPORTANT: This `end` date field acts as an open interval. That is, if `end` is set to "2020-04-04", then only the data upto and including April 3rd is returned. If you need the data for April 4th, then you must set `end` date to "2020-04-05".

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

result ▾
required

object

Returned object containing Timeseries data.

id ▾

object

Container of time series evaluation attributes.

expression ▾

object

type	string Name of the metric.
count	integer Number of data elements in the returned array.
dates	array Array of timestamps corresponding to the returned Timeseries data.
start	datetime Timestamp indicating the start of the returned Timeseries data.
end	datetime Timestamp indicating the start of the returned Timeseries data.
data	array Timeseries data returned.
missing	array Data that is missing from the returned Timeseries.
interval	string Interval of the data to be returned. Same as it is in the request. For example: "DAY".
timeZone	string Time zone where the returned Timeseries data was originally recorded.

POST /api/1/outbreaklocation/evalmetrics

Request samples

Payload

Content type

application/json

Copy

Expand all

Collapse all

```
{
  - "spec": {
    "ids": [ ],
    "expressions": [ ],
    "interval": "string",
    "start": "string",
    "end": "string"
  }
}
```

Response samples

200

Content type

application/json

Copy

Expand all

Collapse all

```
{
  - "result": {
    + "id": { ... }
  }
}
```

LineListRecord

LineListRecord stores individual-level crowdsourced information from laboratory-confirmed COVID-19 patients. Information includes gender, age, symptoms, travel history, location, reported onset, confirmation dates, and discharge status.

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
location	OutbreakLocation	C3.ai Type OutbreakLocation.
locationType	String	Specific location where the individual was assessed. For example: Yokohama Port, National Centre for Infectious Diseases.
isGroundZero	Boolean	Is the patient located in Wuhan? Allowed values are: <code>true</code> , <code>false</code> .
livesInGroundZero	Boolean	Does the patient live in Wuhan? Allowed values are: <code>true</code> , <code>false</code> .
traveledToGroundZero	Boolean	Did the patient travel to Wuhan? Allowed values are: <code>true</code> , <code>false</code> .
relevantTravelHistoryLocation	String	Details of patient travel history including locations travelled.
recordSource	String	Public source from which this patient information collected.
gender	String	Gender of the patient. Allowed values are: <code>male</code> , <code>female</code> , <code>other</code> .

Field	Data type	Description
age	Integer	Age of the patient.
ageRange	String	Age range, if age is provided as age range.
groundZeroExposure	String	Description regarding whether the individual had exposure to the Wuhan marketplace.
chronicDisease	String	Medical history of chronic disease symptoms.
symptomStartDate	Datetime	Date COVID-19 symptoms appeared.
symptoms	String	Description of the symptoms.
caseConfirmationDate	Datetime	Date of the confirmation of the case.
caseInCountry	Integer	Ordinal number indicating whether this case is, for example, 20th or 32nd, in that country.
hospitalAdmissionDate	Datetime	Date admitted to hospital.
relevantTravelHistoryDates	String	Dates when the patient travelled in recent past.
outcome	String	Outcome of the treatment. Allowed values are: <code>stable</code> , <code>discharge</code> , <code>death</code> , <code>""</code> (null value).
didDie	Boolean	Did patient die? Allowed values are: <code>true</code> , <code>false</code> .
didRecover	Boolean	Did patient recover? Allowed values are: <code>true</code> , <code>false</code> .
outcomeDate	Datetime	Date of the outcome.

Field	Data type	Description
notes	String	Clinical notes.
lineListSource	String	Data source for the crowdsourced line list records. Allowed values are: OPEN (nCoV2019 Data Working Group), DXY (MOBS Lab).

Fetch

Examples (Click on the arrows to expand)

The following examples show how to use this API.

- ▶ Example 1: Fetch all Line List Records of males of age under 20 with travel history to Wuhan, tracked by the nCoV-2019 Data Working Group
- ▶ Example 2: Fetch first two thousand (2000) LineList records (request only)
- ▶ Example 3: Fetch the first two thousand (2000) Line List Records tracked in the nCoV-2019 Data Working Group (request only)
- ▶ Example 4: Fetch the first two thousand (2000) Line List Records tracked by MOBS Lab (request only)
- ▶ Example 5: Fetch the first two thousand (2000) male-patient Line List Records tracked by MOBS Lab (request only)
- ▶ Example 6: Python example using "offset" parameter

HEADER PARAMETERS

Content-Type
required string
Set this to application/json.

Accept
required string
Set this to application/json.

REQUEST BODY SCHEMA: application/json

spec ▾

object

Container of query evaluation attributes

filter

string

Filter expression for which Obj instances to return. For example: `"filter": 'id == "Afghanistan" && age == 45'`. Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, || and most non-timeseries functions supported by the expression engine."

include

string

Specifies which fields to bring back values for in the returned objects. A list of fields. For example: `"include": "productType, description, origin, links.url"`.

limit

integer

Maximum number of rows that should be returned, starting from offset.

offset

integer <int32> (The Offset Schema)

Number of rows to skip.

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs >

object

Container of query evaluation attributes

count

integer <int32> (The Count Schema)

Number of rows returned.

hasMore

boolean (The Hasmore Schema)

If set to true there were more objs that were not returned.

POST /api/1/linelistrecord/fetch

Request samples

Payload

Content type

application/json

Copy Expand all Collapse all

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  - "objs": {
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

```
}
```

BiologicalAsset

BiologicalAsset stores the metadata of the genome sequences collected from SARS-CoV-2 samples in the National Center for Biotechnology Information Virus Database. See also [Sequence](#).

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
id	String	Genome sequence ID.
assetType	String	Biological molecule type. Allowed values are: <code>protein sequence</code> , <code>nucleotide sequence</code> .
genus	String	Genus. Allowed values are: <code>Betacoronavirus</code> .
family	String	Family. Allowed values are: <code>Coronaviridae</code> .
species	String	The species that the BiologicalAsset relates to.
authors	String	Individuals cited as source of genome sequences.

Field	Data type	Description
genBankTitle	String	Sequence description in the National Center for Biotechnology Information Virus Database.
publications	String	Public source from which this patient information collected.
name	String	Name of the patient.
location	OutbreakLocation	C3.ai Type OutbreakLocation.
sequence	Sequence	C3.ai Type Sequence.
host	String	The host organism from which the sample was taken. Allowed values are: <code>Homo sapiens</code> .
isolationSource	String	Source from which the sample was taken. Allowed values are: <code>oronasopharynx</code> , <code>blood</code> , <code>feces</code> , <code>lung</code> , <code>swab</code> , <code>lung</code> , <code>oronasopharynx</code> .
collectionDate	Datetime	Date when samples were collected.

Fetch

Examples (Click on the arrows to expand)

The following examples show how to use this API.


- ▶ Example 1: Fetch metadata for all BiologicalAsset sequences (request only)
- ▶ Example 2: Fetch metadata for all BiologicalAsset protein sequences (request only)
- ▶ Example 3: Fetch metadata for all BiologicalAsset nucleotide sequences (request only)
- ▶ Example 4: Fetch metadata for all BiologicalAsset sequences sampled from blood (request only)
- ▶ Example 5: Fetch metadata for all BiologicalAsset sequences taken in Japan and sampled from feces

HEADER PARAMETERS

Content-Type
required string
Set this to application/json.

Accept
required string
Set this to application/json.

REQUEST BODY SCHEMA: application/json

spec  object
Container of query evaluation attributes

filter string
Filter expression for which Obj instances to return. For example: `"filter": 'id == "Afghanistan" && age == 45'`. Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, || and most non-timeseries functions supported by the expression engine."

include string
Specifies which fields to bring back values for in the returned objects. A list of fields. For example: `"include": "productType, description, origin, links.url"`.


limit integer
Maximum number of rows that should be returned, starting from offset.

offset integer <int32> (The Offset Schema)
Number of rows to skip.

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs  object
Container of query evaluation attributes

count	integer <int32> (The Count Schema) Number of rows returned.
hasMore	boolean (The Hasmore Schema) If set to true there were more objs that were not returned.

POST /api/1/biologicalasset/fetch

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
  - "objs": {
    "type": "string".
  }
}
```

```
    type: "string",  
    "location": { },  
    "name": "string",  
    "id": "string",  
    "version": "string",  
    "meta": { },  
    "typeIdent": "string"  
  },  
  "count": 0,  
  "hasMore": true  
}
```

Sequence

Sequence stores the genome sequences collected from SARS-CoV-2 samples in the National Center for Biotechnology Information Virus Database. See also [BiologicalAsset](#).

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
id	String	Genome sequence ID.
parent	BiologicalAsset	C3.ai Type BiologicalAsset.
sequence	String	Actual genomic sequence. Should be in uppercase.

Field	Data type	Description
sequenceType	String	Biological molecule type. Allowed values are: <code>protein</code> , <code>complete</code> , <code>partial</code> , <code>GenBank</code> .
length	Int	Length of the sequence, as in <code>length(sequence)</code> .

Fetch

Examples (Click on the arrows to expand)

The following examples show how to use this API.


- ▶ Example 1: Fetch all sequences from Sequence (request only)
- ▶ Example 2: Fetch all protein sequences (request only)
- ▶ Example 3: Fetch all complete genome sequences (request only)
- ▶ Example 4: Fetch all Sequences with ≤ 100 Base Pairs (or Amino Acids) (request only)
- ▶ Example 5: Fetch sequence 5R7Y_A

HEADER PARAMETERS

Content-Type
required string
Set this to application/json.

Accept
required string
Set this to application/json.

REQUEST BODY SCHEMA: application/json

spec  object
Container of query evaluation attributes

filter string
Filter expression for which Obj instances to return. For example: `"filter": 'id == "Afghanistan" && age == 45'`. Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. `==`, `<`, `<=`, `>`, `>=`, `!=`), arithmetic operators (e.g. `+`, `-`, `/`), `&&`, `||` and most non-timeseries functions supported by the expression engine."

include	string Specifies which fields to bring back values for in the returned objects. A list of fields. For example: <code>"include": "productType, description, origin, links.url"</code> .
limit	integer Maximum number of rows that should be returned, starting from offset.
offset	integer <int32> (The Offset Schema) Number of rows to skip.

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs >	object Container of query evaluation attributes
count	integer <int32> (The Count Schema) Number of rows returned.
hasMore	boolean (The Hasmore Schema) If set to true there were more objs that were not returned.

POST /api/1/sequence/fetch

Request samples

Payload

Content type
application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type
application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{
  - "objs": {
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

Subsequence

Subsequence stores indices of critical segments within a nucleotide or amino acid sequence, for example, introns, exons, and proteins. Indices are 1-indexed.

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
parent	Sequence	The Sequence this Subsequence references.
startIndex	Int	The position where the value of this Subsequence begins in relation to the parent's sequence.
endIndex	Int	The position where the value of this Subsequence ends in relation to the parent's sequence.

Fetch

Examples (Click on the arrows to expand)

The following examples show how to use this API.

- ▶ Example 1: Fetch all Subsequences (request only)
- ▶ Example 2: Fetch all subsequences of genome MN975266 (request only)

HEADER PARAMETERS

Content-Type
required

string
Set this to application/json.

Accept
required

string
Set this to application/json.

REQUEST BODY SCHEMA: application/json

spec ▾

object

Container of query evaluation attributes

filter

string

Filter expression for which Obj instances to return. For example: `"filter": 'id == "Afghanistan" && age == 45'`. Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, || and most non-timeseries functions supported by the expression engine."

include

string

Specifies which fields to bring back values for in the returned objects. A list of fields. For example: `"include": "productType, description, origin, links.url"`.

limit

integer

Maximum number of rows that should be returned, starting from offset.

offset

integer <int32> (The Offset Schema)

Number of rows to skip.

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs >

object

Container of query evaluation attributes

count

integer <int32> (The Count Schema)

Number of rows returned.

hasMore

boolean (The Hasmore Schema)

If set to true there were more objs that were not returned.

POST /api/1/subsequence/fetch

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
```

```
- "objs": {
```



```
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

AminoAcidLookup

AminoAcidLookup contains the lookup table to map the [IUPAC Amino Acid Codes](#) to their full names and abbreviations.

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
id	String	The IUPAC Code.
name	String	Full name of the Amino Acid.
abbreviation	String	The Three-letter abbreviation of the Amino Acid.

Fetch

Example (Click on the arrow to expand)

The following example shows how to use this API.

► Fetch all AminoAcids

HEADER PARAMETERS

Content-Type required	string Set this to application/json.
---------------------------------	---

Accept required	string Set this to application/json.
---------------------------	---

REQUEST BODY SCHEMA: application/json

spec ▾	object Container of query evaluation attributes
--------	--

filter	string Filter expression for which Obj instances to return. For example: <code>"filter": 'id == "Afghanistan" && age == 45'</code> . Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, and most non-timeseries functions supported by the expression engine."
--------	--

include	string Specifies which fields to bring back values for in the returned objects. A list of fields. For example: <code>"include": "productType, description, origin, links.url"</code> .
---------	---

limit	integer Maximum number of rows that should be returned, starting from offset.
-------	--

offset	integer <int32> (The Offset Schema) Number of rows to skip.
--------	--

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs >	object Container of query evaluation attributes
count	integer <int32> (The Count Schema) Number of rows returned.
hasMore	boolean (The Hasmore Schema) If set to true there were more objs that were not returned.

POST /api/1/aminoacidlookup/fetch

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
  - "objs": {
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

NucleotideLookup

NucleotideLookup contains the lookup table to map the [IUPAC Nucleotide Codes](#) to their full names and abbreviations.

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
id	String	The IUPAC Code.
name	String	Full name of the Nucleotide.
abbreviation	String	The Three-letter abbreviation of the Nucleotide.

Fetch

Example (Click on the arrow to expand)

The following example shows how to use this API.

► Fetch all Nucleotides

HEADER PARAMETERS

Content-Type
required

string
Set this to application/json.

Accept
required

string
Set this to application/json.

REQUEST BODY SCHEMA: application/json

spec ▼

object
Container of query evaluation attributes

filter

string
Filter expression for which Obj instances to return. For example: `"filter": "id == 'Afghanistan' && age == 45"`. Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, || and most non-timeseries functions supported by the expression engine."

include

string
Specifies which fields to bring back values for in the returned objects. A list of fields. For example: `"include": "productType, description, origin, links.url"`.

limit	integer Maximum number of rows that should be returned, starting from offset.
offset	integer <int32> (The Offset Schema) Number of rows to skip.

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs >	object Container of query evaluation attributes
count	integer <int32> (The Count Schema) Number of rows returned.
hasMore	boolean (The Hasmore Schema) If set to true there were more objs that were not returned.

POST /api/1/nucleotidelookup/fetch

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
```

```
- "spec": {
```

```
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  - "objs": {
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

BiblioEntry stores the metadata about the journal articles in the CORD-19 Dataset.

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
id	String	Journal article ID.
sha	String	ID that links the BiblioEntry to a JSON file containing the journal article's full text.
source	String	The database from which the journal article is sourced. Allowed values are: <code>Elsevier</code> , <code>biorxiv</code> , <code>CZI</code> , <code>medrxiv</code> , <code>PMC</code> , <code>WHO</code> .
title	String	Title of the journal article.
doi	Datetime	Journal article's BioRxiv/MedRxiv ID.
pmcid	String	Journal article's PMC ID.
pubmedId	String	Journal article's PubMed ID.
license	String	Journal article's license. Allowed values are: <code>biorxiv</code> , <code>els-covid</code> , <code>medrxiv</code> , <code>cc-by</code> , <code>cc-by-nc</code> , <code>cc-by-nc-nd</code> , <code>cc-by-nc-sa</code> , <code>cc-by-nd</code> , <code>cc-by-sa</code> , <code>cc0</code> , <code>pd</code> .
abstractText	String	Journal article's abstract text.
publishTime	Datetime	Date the journal article was published.
authors	String	List of journal article's authors.
journal	String	Journal which published the article.
url	String	The URL of the article.
idMsftPaper	String	Journal article's Microsoft Academic Paper ID.
whoCovidence	String	Journal article's WHO ID.

Field	Data type	Description
hasFullText	Boolean	Is the JournalArticle's full text available in the dataset? Allowed values are: <code>true</code> , <code>false</code> .
fullTextFile	String	The type of the journal article. Allowed values are: <code>custom_license</code> , <code>comm_use_subset</code> , <code>biorxiv_medrxiv</code> , <code>noncomm_use_subset</code> .

Fetch

Examples (Click on the arrows to expand)

The following examples show how to use this API.


- ▶ Fetch metadata for the first two thousand (2000) BiblioEntry journal articles (request only)
- ▶ Fetch metadata for the first two thousand (2000) full text BiblioEntry journal articles
- ▶ Fetch metadata for the first two thousand (2000) BiblioEntry journal articles approved for commercial use
- ▶ Fetch metadata for the first two thousand (2000) full text PMC journal articles with full text cc-by license (request only)

HEADER PARAMETERS

`Content-Type`
required string
Set this to application/json.

`Accept`
required string
Set this to application/json.

REQUEST BODY SCHEMA: application/json

`spec`  object
Container of query evaluation attributes

`filter` string

Filter expression for which Obj instances to return. For example: `"filter": 'id == "Afghanistan" && age == 45'`. Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, || and most non-timeseries functions supported by the expression engine."

include	string Specifies which fields to bring back values for in the returned objects. A list of fields. For example: <code>"include": "productType, description, origin, links.url"</code> .
limit	integer Maximum number of rows that should be returned, starting from offset.
offset	integer <int32> (The Offset Schema) Number of rows to skip.

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs >	object Container of query evaluation attributes
count	integer <int32> (The Count Schema) Number of rows returned.
hasMore	boolean (The Hasmore Schema) If set to true there were more objs that were not returned.

POST /api/1/biblioentry/fetch

Request samples

Payload

Content type

application/json

Copy Expand all Collapse all

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples**200**

Content type

application/json

Copy Expand all Collapse all

```
{
  - "objs": {
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

GetArticleMetadata

Example (Click on the arrow to expand)

The following example shows how to use this API.

- Get the full-text, in JSON-format, of the journal article from the CORD-19 dataset

HEADER PARAMETERS

Content-Type required	string Set this to application/json.
--------------------------	---

Accept required	string Set this to application/json.
--------------------	---

REQUEST BODY SCHEMA: application/json

this ▾	object Container of time series evaluation attributes
--------	--

id	string Biblio reference ID.
----	--------------------------------

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

value	object (The Result Schema)
-------	----------------------------

POST /api/1/biblioentry/getarticlemetadata

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
  - "this": {
      "id": "string"
    }
}
```

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
  "value": { }
}
```

TherapeuticAsset

TherapeuticAsset stores details about the research and development (R&D) of coronavirus therapies, for example, vaccines, diagnostics, and antibodies.

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
id	String	Therapy ID.
productType	String	Therapy's drug or platform class.
links	ExternalLink	C3.ai Type ExternalLink.
description	String	Description of the therapy.
clinicalTrialsOtherDiseases	String	Other diseases or pathogens for which the therapy has undergone or is undergoing clinical development.
developer	String	Organization that developed the therapy.
stageOfDevelopment	String	The therapy's current phase of clinical development. Allowed values: <code>Clinical</code> , <code>Pre-Clinical</code> , <code>Compassionate Use</code> , <code>Phase 1/2 (not yet recruiting)</code> , <code>Phase I</code> , <code>Expanded access</code> .
fundingSources	String	The organization funding the therapy's R&D.
nextSteps	String	Anticipated next steps for the therapy's clinical development.

Field	Data type	Description
therapyType	String	The type of therapy. Allowed values: <code>Vaccine</code> , <code>Antibodies</code> , <code>Antivirals</code> , <code>Cell-based therapies</code> , <code>RNA-based therapies</code> , <code>Scanning compounds to repurpose</code> , <code>Dormant Discontinued</code> , <code>Other</code> .
origin	String	The source of the data containing the therapy's R&D details. Allowed values: <code>WHO</code> , <code>Milken</code> .
target	String	The virus the therapy targets or treats. Allowed values: <code>COVID-19</code> , <code>MERS</code> , <code>SARS</code> .

Fetch

Examples (Click on the arrows to expand)

The following examples show how to use this API.

- ▶ Fetch all therapies
- ▶ Fetch therapies tracked by Milken Institute (request only)
- ▶ Fetch therapies tracked by WHO (request only)
- ▶ Fetch vaccine therapies
- ▶ Fetch pre-clinical DNA therapies, targeting COVID-19, tracked by WHO

HEADER PARAMETERS

Content-Type
required string
Set this to application/json.

Accept
required string
Set this to application/json.

REQUEST BODY SCHEMA: application/json

spec ▾

object

Container of query evaluation attributes

filter

string

Filter expression for which Obj instances to return. For example: `"filter": 'id == "Afghanistan" && age == 45'`. Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, || and most non-timeseries functions supported by the expression engine."

include

string

Specifies which fields to bring back values for in the returned objects. A list of fields. For example: `"include": "productType, description, origin, links.url"`.

limit

integer

Maximum number of rows that should be returned, starting from offset.

offset

integer <int32> (The Offset Schema)

Number of rows to skip.

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: application/json

objs >

object

Container of query evaluation attributes

count

integer <int32> (The Count Schema)

Number of rows returned.

hasMore

boolean (The Hasmore Schema)

If set to true there were more objs that were not returned.

POST /api/1/therapeuticasset/fetch

Request samples

Payload

Content type

application/json

Copy Expand all Collapse all

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  - "objs": {
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

```
    "urlType": "Clinical Trial",  
  }  
}
```

ExternalLink

ExternalLink stores website URLs cited in the data sources containing the therapies stored in the TherapeuticAssets C3.ai Type.

Fields:

NOTE: If the field is not present in a returned object, then that field will not be returned.

Field	Data type	Description
id	String	External link ID.
urlType	String	The type of information available at the URL. Allowed values: <code>News</code> (for example: news article, press release), <code>Clinical Trial</code> , <code>Published Results</code> .
url	String	The URL of the website.
therapeuticAsset	TherapeuticAsset	C3.ai Type TherapeuticAsset.
origin	String	The source of the data containing the therapy's R&D details. Allowed values: <code>WHO</code> , <code>Milken</code> .

Fetch

Examples (Click on the arrows to expand)

The following examples show how to use this API.

- ▶ Fetch URLs for all therapies (request only)
- ▶ Fetch URLs for therapies tracked by Milken Institute (request only)
- ▶ Fetch URLs for clinical trials of therapies (request only)
- ▶ Fetch URLs for a particular therapy

HEADER PARAMETERS

Content-Type required	string Set this to application/json.
---------------------------------	---

Accept required	string Set this to application/json.
---------------------------	---

REQUEST BODY SCHEMA: application/json

spec ▾	object Container of query evaluation attributes
--------	--

filter	string Filter expression for which Obj instances to return. For example: <code>"filter": "id == 'Afghanistan' && age == 45"</code> . Filter expressions must evaluate to a value type of Boolean. They support basic comparison operators (e.g. ==, <, <=, >, >=, !=), arithmetic operators (e.g. +, -, /), &&, and most non-timeseries functions supported by the expression engine."
--------	--

include	string Specifies which fields to bring back values for in the returned objects. A list of fields. For example: <code>"include": "productType, description, origin, links.url"</code> .
---------	---

limit	integer Maximum number of rows that should be returned, starting from offset.
-------	--

offset	integer <int32> (The Offset Schema) Number of rows to skip.
--------	--

Responses

✓ **200 OK.** The request has succeeded.

RESPONSE SCHEMA: `application/json`

objs >	object Container of query evaluation attributes
count	integer <int32> (The Count Schema) Number of rows returned.
hasMore	boolean (The Hasmore Schema) If set to true there were more objs that were not returned.

POST `/api/1/externallink/fetch`

Request samples

Payload

Content type
`application/json`

Copy Expand all Collapse all

```
{
  - "spec": {
    "filter": "string",
    "include": "string",
    "limit": 0,
    "offset": 0
  }
}
```

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  - "objs": {
    "type": "string",
    "location": { },
    "name": "string",
    "id": "string",
    "version": "string",
    "meta": { },
    "typeIdent": "string"
  },
  "count": 0,
  "hasMore": true
}
```

[News](#)[Contact](#)[Careers](#)[Privacy Policy](#)[Terms of Use](#)[Site Map](#)[Data Use Conditions](#)

© 2020 C3.ai, Inc. All Rights Reserved.