CMR Client Partner User Guide

- Chapter 1: Introduction and Scope
  - The CMR Concept and Design
    - Security
    - CMR Capability And Functionality
    - CMR as a Spatially Enabled Metadata Search and Retrieval System
    - CMR Benefits to Client Partners
  - Client Partner Skills
  - Client Partner Tasks
  - CMR System Environments
- Chapter 2: Getting Started
  - Creating and Managing User Accounts and Access
    - User Accounts
    - Creating and Managing CMR Sessions
      - Creating a Token
      - Deleting the Token
- Chapter 3: Searching for metadata
  - CMR Environment URLs
  - Headers
    - Content Type
      - Authorization: Bearer
    - Accept Headers
    - Client-Id
  - Results
  - Searching
- API calls and parameters GET method
  - API calls and parameters POST method
  - Alternative Query Language (AQL)
- Chapter 4: Retrieving Metadata
- Chapter 5: Accessing data
- Acronyms
- Best Practices for CMR Client Operations
  - Enhancing the Speed of Queries
  - Increasing Efficiency of Spatial Queries
  - Optimizing HTTP CMR Requests

Chapter 1: Introduction and Scope

The NASA-developed Earth Observing System (EOS) Common Metadata Repository (CMR) is a spatial and temporal metadata registry that stores metadata from a variety of science disciplines and domains —including Climate Variability and Change, Carbon Cycle and Ecosystems, Earth Surface and Interior, Atmospheric Composition, Weather, and Water and Energy Cycles. The CMR is intended to enable broader use of NASA's EOS data by providing a more uniform view of NASA's substantial and diverse data holdings. Its two primary objectives are to: 1) help science communities in need of data from multiple organizations and disciplines more efficiently use search functions and access data and services; and 2) increase the potential for interoperability with new tools and services. As the potential to exchange and inter-operate increases, the value of these resources increases comparably.

The CMR was designed to accomplish these goals by providing a system with an Application Programming Interface (API). While the API facilitates the discovery, online access, and delivery of a Data Partner's data holdings; it is the responsibility of the CMR Data Partners to add new metadata, remove old metadata, and modify and control access to existing metadata. As such, Data Partners retain complete control over what metadata are represented in the CMR at any given time. Client Partners develop client applications that access the CMR API and take advantage of the services made available. These clients, such as Earthdata Search (https://search.earthdata.nasa.gov), CMR open search (https://cmr.earthdata.nasa.gov /opensearch), etc. allow end users to discover data which has been registered in the CMR's holdings; and can be custom made to meet the needs of a general user audience or a specific science application.

It should be noted that the CMR is a continuously evolving metadata system that merges all existing capabilities and metadata from Data Partners and the Global Change Master Directory (GCMD) systems. The CMR elements include all system components, which consist of: CMR itself (formerly the [ECHO]), GCMD, International Data Network (IDN), Earth Science Data and Information System (ESDIS) Metrics System (EMS), all related tools (internal and external), and all Metadata – including the Unified Metadata Model (UMM) concepts, the GCMD Keywords Controlled Vocabulary, and other controlled vocabularies. Thus, this is a living document. As the CMR matures, updated instructions will be incorporated in later revisions.

NASA's Earth science data has already proven essential to understanding Earth as an integrated system, and other organizations are also providing their Earth science metadata to the CMR for users to search and access. By simplifying discoverability and accessibility to the CMR's Earth Science holdings, and fostering interoperability with new tools and services, the user community will enlarge and the pace of scientific discovery and application will accelerate. For examples of how NASA's Earth science data is helping scientists understand the complexities of our Earth, visit Sensing our Planet and Other Featured Research Articles at https://earthdata.nasa.gov/.

The CMR Concept and Design

NASA's Earth Science Data and Information System (ESDIS) built the CMR based on Extensible Markup Language (XML) and Web Service technologies. The CMR interfaces with clients and users through various APIs. The CMR is an open system with published APIs available to the CMR Development and User community.

Internally, the CMR specifies APIs and provides middleware components in a layered architecture - including data and service search and access functions. The figure below depicts the CMR system context in relation to its public APIs.

? Unknown Attachment

CMR System Concept

Key features of the CMR architecture are:

- *Ease of Partner Participation* – Designed to be low-cost and minimally intrusive, the CMR offers a set of standard ways for Partners to interface with the system through provided web UIs and a metadata exchange approach that accommodates existing partners and technology.
- *Open System / Published APIs* – To accommodate independent CMR clients, CMR uses an open system approach and publishes domain APIs. These APIs are independent of the underlying transport protocols used. CMR communicates using WS-I Basic Profile v1.0 compliant web services for RESTful web services for CMR ingest, search, and metadata management.
- *Evolutionary Development* – The CMR system is being developed incrementally to allow for insight and feedback during the development cycle. Industry trends are followed and the use of commercial, off-the-shelf (COTS) products is optimized.

Security

The CMR system requires Secure Sockets Layer (SSL)-based communication from Client Applications to the CMR API, but does not require, secure communication from CMR to a Client Partner's service. Internally, the CMR system is protected through a layer of software and hardware control mechanisms to preserve the integrity of CMR's holdings. When configuring data access fulfillment, Client Partners are strongly encouraged to utilize SSL communications.

CMR Capability And Functionality

CMR provides an infrastructure that allows various communities to share tools, services, and metadata. It supports many data access paradigms - such as navigation and discovery, facilitates data access through appropriate Data Partners, decentralizes end user functionality, and supports interoperability of distributed functions.

Although this Guide focuses on the needs of Client Partners, support is provided for all of the following nonexclusive Partner types:

- *Data Partners* – Organizations that supply metadata representing their data holdings to the CMR system
- *Client Partners* – Organizations that participate by developing software applications to access the Earth science metadata in the CMR system
- *Service Partners* – Organizations that participate by advertising their Earth science-related services to the user community via the CMR, which maintains service descriptions in a Service Catalog (either special services, or services that are available as an option on a selected set of granules/collections) and support the user in accessing those services.

The CMR enables Client Partners to use the Client Partner APIs for the purpose of creating their own custom tailored software systems. These APIs allow clients to search and retrieve metadata, including but not limited to: collections, granules, and services. All CMR metadata is stored as received by the data partners and can be retrieved by the native specification or another specification as requested by the client.

The CMR approach allows users to build their own user interfaces to the CMR, rather than being limited to the data search and access system provided by NASA. Furthermore, the CMR addresses science user needs through a set of well-defined and open interfaces upon which the user community can build its own client applications. In this way, the CMR supports extendable, flexible user interfaces, allowing industry and the science community to drive the progress of available earth science applications. For Data Partners, the system offloads the burden of providing the system resources required for searching and gives users the flexibility to support community-specific services and functionality. The CMR's interoperability features allow all participants to benefit from the distributed development of functions, again reducing dependence on NASA resources.

CMR as a Spatially Enabled Metadata Search and Retrieval System

The CMR allows Data Partners to define the spatial extent of a granule or a collection with different spatial constructs (for example: point and polygon). These spatial extents may be in either the Geodetic or Cartesian coordinate systems. Orbital data may also be provided to describe a collection or granules spatial extents. A Client Partner can then construct a search using a point, a line, or a polygon (or multiple polygon) spatial type, and the CMR responds with data whose spatial region intersects the described region.

The CMR provides services for interacting with its catalog of metadata. Queries can be performed in a number of ways; result contents can be specified, and the resulting data sets can be incrementally accessed so that large return sets can be handled gracefully. The system also supports constructing, submitting, and tracking access requests for the data that the metadata represents. The CMR supports both an embedding of a Uniform Resource Locator (URL) within the metadata for accessing the data (which the client simply accesses via Hypertext Transfer Protocol [HTTP] or File Transfer Protocol (FTP)), and a more complicated data access process in which other options are accommodated.

CMR Benefits to Client Partners

The CMR's open system provides earth science data and services to a large, diverse pool of users, enabling scientific community interaction and collaboration. The Client Partners are benefited in the following ways:

- Ease of Participation - The primary goal of the CMR is to enable organizations to participate in making their resources and capabilities available to the Earth Science community. To facilitate participation by these organizations, the CMR has:

    - Minimized the number of requirements that a partner must meet to participate.
    - Involved partners in the system's development cycle and requirements definition.
    - Selected metadata insert and update mechanisms, that are based on current standard industry practice (for example, XML), and that most databases can generate automatically.
    - Provided mapping capabilities to convert from one XML representation into another.
- Cost to Field - The CMR minimizes the Cost to Field by continually evaluating performance and functionality against costs — such as licensing of Commercial Off-the-Shelf (COTS) applications, amount of custom code required, hardware platform requirements, and complexity of networking and installation.
- Cost to Operate - Once fielded, CMR Operations costs are minimized through enhanced efficiency and extensive automation, thereby reducing the need for support from operations staff.

Client Partner Skills

Since the CMR uses platform-independent web service definitions for its API, there are no requirements for a client programming language. All examples in this document use curl; however, the code samples provided could be translated to any web service capable language.

As a CMR Client Partner, you need to be familiar with basic software development and Service Oriented Architecture (SOA) concepts such as:

- XML and XML Schema (XSD)
- Client/Server-based programming (client stubs, remote endpoints, etc.)
- RESTful client and service communication programming
- Service-based Application Programmer's Interface (API)

Client Partner Tasks

As a Client Partner who is beginning to integrate with CMR, you should expect to perform the following tasks, which are detailed in later sections:

- Creating and managing user accounts and user access
- Searching for data
- Retrieving data
- Accessing data

CMR System Environments

Three CMR systems are accessible by Client Partners: CMR Operations, CMR UAT and CMR SIT. Each of these systems is briefly described below. For additional information, click on the associated link.

- **CMR Operations** - The CMR Operations system environment is a publicly accessible server that houses the production environment. The Data Holdings within this system include Earth Science data that has been made available to the Earth Science Community by the CMR Data Partners. This environment is monitored 24/7, updated with enhancements and fixes on a monthly cycle, and experiences virtually no down time.
- **CMR UAT** (User Acceptance Test) - The UAT environment provides a stable test system to serve the needs of the CMR Data, Client, and Service Partners.The Data Holdings within this system consist of whatever the CMR's Data Partners have made available for their own testing purposes. Any enhancements and fixes that are planned for the Operations Environment are installed in this environment two weeks prior to operations delivery. CMR Partners are encouraged to verify the capabilities when a new release is installed.
- **CMR SIT** (System Integration Test) The SIT system was established in order to facilitate an exchange of ideas and provide an initial testing ground for upcoming capabilities. There is often very little metadata available in this test environment, but it is fully functional. The next operational version is released into this system approximately 1 month before its schedule Operational release date. CMR SIT is a closed, private environment only available to CMR Developers.

Chapter 2: Getting Started

Creating and Managing User Accounts and Access

This chapter will discuss:

- Creating and managing user accounts
- CMR session management - creating, using, and deleting tokens to provide authorization

*Note: If searching and retrieving publicly available data is the only desired operation, proceed straight to Chapter 3.

User Accounts

User accounts are employed to enable access to restricted data, manage privileges, and/or to interact with other services and tools provided by the CMR. User accounts for the CMR system are created and managed by the Earthdata Login (URS) system. If you need to create an account, click on Earthdata Login and follow the instructions provided. Once created, you can always return to Earthdata Login to manage your account. If you are part of a Data Provider group or other team, the team administrator can set up permissions for you to access their restricted data. Special privilege requests can be made by contacting the CMR operational team at support@earthdata.nasa.gov.

Creating and Managing CMR Sessions

Tokens are used by the CMR to validate both the requester and their privileges for each request message (i.e., the http call to CMR) submitted. For most searches, a token is not needed because the metadata records are unrestricted and accessible by anyone. However, when specific metadata records are restricted, privileged users require a token to see and access those records.

The same token can be used for multiple requests before being deleted. A series of requests that use the same token is referred to as a "session." All tokens expire at the end of a predefined time period - which is currently fixed at 30 days. Because the token is used to track your session, it must be protected by client applications with the same level of security used for your login name and password.

A normal session is conducted via the following steps:

- Create a token
- Perform one or both of the following tasks in any order:
    - Search for records
    - Retrieve records
- Delete the token

Creating a Token

Information on creating EDL tokens can be found at https://urs.earthdata.nasa.gov/documentation/for_users/user_token

Information for requesting a Launchpad Token can be found here. More information on getting access to Launchpad tokens can be found on this guide .

Once a token is created, you can search and retrieve records, as well as conduct other functionality (described in later chapters) through the CMR APIs. When the token is no longer needed, it can be deleted.

Deleting the Token

Information on deleting EDL tokens can be found at https://urs.earthdata.nasa.gov/documentation/for_users/user_token

A Launchpad token will automatically expire after one hour.

Chapter 3: Searching for metadata

Currently clients can search the CMR for collection and granule metadata. In the future, clients will also be able to search for metadata describing services, visualizations, parameters (variables), and documents.

CMR Environment URLs

The CMR system has three environments:

The **Systems Integration Test (SIT) Environment** provides the CMR development team with a safe place to test new functionality. The newest, untested software and upgrades are are first uploaded here; thus, this environment is the least stable. Once the CMR software has been sufficiently tested, it gets deployed to the **User Acceptance Test (UAT) Environment**. Note that the CMR SIT environment is closed to the public and is intended for internal CMR development only.

The **UAT Environment** provides a stable environment for testing system software. The UAT environment is where Open Source developers and Data Partners can deploy new code and/or new metadata for testing before it becomes visible in Operations. Following a few weeks of successfully testing, the software is deployed to the **Operational Environment**.

**Note: Client Partners can test their software in the UAT environment before moving that software to Operations.*

The **Operational Environment** is the live system available to users around the world. The Operational Environment is also known as Production, Operations, or OPS.

**Note: All of the examples provided in the rest of the document are using the User Acceptance Test environment. To run the commands in the Operational Environment, replace all instances of cmr.uat.earthdata.nasa.gov with cmr.earthdata.nasa.gov.*

CMR Environment	Base API URL
Systems Integration Test (SIT)	https://cmr.sit.earthdata.nasa.gov/

User Acceptance Test (UAT)	https://cmr.uat.earthdata.nasa.gov/
Operations/Production (OPS)	https://cmr.earthdata.nasa.gov/

CMR Environments

Headers

Headers are a part of HTTP requests and for the CMR they provide information such as message content (content type), the format of the data that gets returned (accept), and tokens to allow increased privileges (EDL Token), etc.

Content Type

Content-Type is a standard HTTP header that specifies the content type of the body of the request for POST method messages. Search and retrieval requests support the following Content-Types.

**Note: If the Content-Type is not specified, XML is assumed.*

Body format	Content-Type
XML	application/xml
JSON	application/json

Content-Type headers

Authorization: Bearer

The Authorization: Bearer header allows a user to specify an EDL token so that CMR knows who is making a request. A token must first be generated as described in the **Creating a Token** section. Once the requester has the token, the token can be placed into the http header for the necessary API calls.

Authorization:

The Authorization: header allows a user to specify a Launchpad token so that CMR knows who is making a request. A token must first be generated as described in the **To Create a Token** section. Once the requester has the token, the token can be placed into the http header for the necessary API calls.

Accept Headers

Accept Headers are used in cases where the caller wants to control the format and/or specification by which the data gets returned. The following table lists the valid values. If this header or an alternative method is not used, XML results will be returned by default.

Type Received	Accept HeaderValue	Comments
xml	application/xml	returns a reference list of results using the XML format
json	application/json	returns a subset of metadata data list of results using the JSON format
echo10	application/echo10+xml	returns a full metadata record list of results in the echo 10 specification using the XML format
iso	application/iso19115+xml	returns a full metadata record list of results in the ISO 19115-2 (MENDS) specification using the XML format
iso19115	application/iso19115+xml	returns a full metadata record list of results in the ISO 19115-2 (MENDS) specification using the XML format
dif10	application/dif10+xml	supported for collections only and returns a full metadata record list of results in the DIF 10 specification using the XML format
csv	text/csv	supported for granules only and returns a subset of metadata list of results in a comma separated value format
atom	application/atom+xml	returns a subset of metadata list of results in the ATOM specification using the XML format
opendata	application/opendata+json	supported for collections only and returns a full metadata record list of results in the open data specification using the JSON format
kml	application/vnd.google-earth.kml+xml	returns a subset of spatial metadata list of results using the KML specification in the XML format

native	application/metadata+xml	returns a full metadata record list of results in their individual native specification using the XML format

Accept Headers

For more information about the accepted header values, please see the CMR API documentation.

Client-Id

Client-Id is an additional header that allows the client to specify their name. Client Partners are **strongly** encouraged to use this header for the following reasons:

- Helps the CMR operations team monitor query performance per client
- Aids the CMR operations team in identifying clients who are attempting to contact them for assistance with a request.
- Facilitates NASA in collecting information on how much traffic flows through a client provider and what kind of data interests their users.

Below are some examples depicting how to use headers. The purple part of the example will be explained in this section, while the rest will be addressed later in the document.

Example

The following curl command issues a search request with the search parameters contained in a file called searchterms in the current directory. The Content-Type - the specification and format of the searchterms file - is using the json format to specify the search parameters. The accept header states that we want the results as a reference list using the XML format. The Authorization: Bearer header allows the CMR to know who is making the request for authorization purposes. The Client-Id header allows the operations team and NASA to monitor and track statistics.

curl -v -XPOST -H "Content-Type: application/json" -H "Authorization: Bearer XXXX" -H "Accept: application/xml" -H "Client-Id: Client Partner Name" -d @searchterms -i https://cmr.uat.earthdata.nasa.gov/search/collections

Example

A user is issuing a search request using publicly available data and is returned a full metadata record list of results. Notice that only the Accept header is needed, but the Client-Id is encouraged.

curl -v -H "Accept: application/metadata+xml" -H "Client-Id: Client Partner Name" -i https://cmr.uat.earthdata.nasa.gov/search/collections

Example

A user is issuing a search request using publicly available data and the default result reference list. Notice that no headers are needed, but again the Client-Id is encouraged.

curl -v -H "Client-Id: Client Partner Name" -i https://cmr.uat.earthdata.nasa.gov/search/collections

As an alternate to using the Accept Header, the client can use the Type Received name in the query to get the same results. Instead of using "Accept: application/opendata+json", "opendata" can be used at the end of the main query before parameters are specified.

Example

curl -v -H "Client-Id: Client Partner Name" -i "https://cmr.uat.earthdata.nasa.gov/search/collections.opendata"

Other examples use the DIF 10 specification and the ISO specification respectively.

Example

curl -v -H "Client-Id: Client Partner Name" -i "https://cmr.uat.earthdata.nasa.gov/search/collections.dif10"

curl -v -H "Client-Id: Client Partner Name" -i "https://cmr.uat.earthdata.nasa.gov/search/collections.iso"

To change the type of request, simply replace the header with the desired information from the table above.

Results

Listed below are the 3 ways results can be returned, as well as the formats supported by each option:

- A reference list of results - XML
- A list of results with partial metadata records - XML, JSON, ATOM, KLM, and CSV (**Note: CSV is supported for granules only*)
- A list of results with full metadata records - XML and opendata

The following is an example of a reference list of results

```
<results>
   <hits>2215</hits>
   <took>16</took>
   <references>
      <reference>
         <name>100m Digital Elevation Model Data V001</name>
         <id>C1000000803-DEV08</id>
         <location>https://cmr.uat.earthdata.nasa.gov:443/search/concepts/C1000000803-DEV08>
         <revision-id>8</revision-id>
      </reference>
      <reference>
         <name>100m Digital Elevation Model Data V001</name>
         <id>C1000000719-EDF_OPS</id>
         <location>https://cmr.uat.earthdata.nasa.gov:443/search/concepts/C1000000719-EDF_OPS>
         <revision-id>8</revision-id>
      </reference>
      ...
   </references>
</results>
```

The results specify:

- The number of metadata records found by the "hits" tag
- The duration of the query in milliseconds
- A list of metadata record results specified by the "reference" tag

Within each reference tag, a limited amount of information about the metadata is provided including:

- The metadata name
- The CMR profile or concept ID - a CMR generated unique ID. The ID is encoded by a letter of the profile or concept (C for collection, G for granule, S for service), followed by a CMR generated number, followed by a "-" and then followed by the ID of the metadata provider. `<letter> <unique-number> "-" <provider-id>`
- The exact CMR location from which the metadata can be downloaded
- The latest revision number of the metadata record.

The following is an example of a full metadata record list of results in the ECHO 10 specification using the XML format.

```
<results>
   <hits>2215</hits>
   <took>53</took>
   <result concept-id="C1000000803-DEV08"
      format="application/echo10+xml" revision-id="8">
```

```
      <Collection>
         <ShortName>DEM_100M</ShortName>
         <VersionId>1</VersionId>
         <InsertTime>2002-04-27T15:27:55.293Z</InsertTime>
         <LastUpdate>2013-10-04T08:49:26.783Z</LastUpdate>
         <LongName>100m Digital Elevation Model Data</LongName>
...
</Collection>
   </result>
   <result concept-id="C1000000719-EDF_OPS"
      format="application/echo10+xml" revision-id="8">
      <Collection>
         <ShortName>DEM_100M</ShortName>
...
</Collection>
...
  </result>
</results>
```

The results specify:

- The number of metadata records found by the "hits" tag
- The duration of the query in milliseconds
- A list of metadata record results specified by the "results" tag

Within each result tag, prior to the full metadata record, the following three attributes are displayed:

- The CMR concept id (profile ID)
- The specification and format of the metadata
- The revision number of the shown metadata record.

For detailed information about the result specifications, format options, and examples - please see the CMR API documentation.

Searching

There are several ways to search the CMR system using the RESTful principles, which include employing:

- The API calls and parameters with the GET method
- The API calls and parameters with the POST method
- A JSON query language with a POST method
- The Alternative Query Language (AQL)

The most popular and preferred method is to use the API calls and parameters with the GET or POST methods. For more detailed information, see the API documentation located at https://cmr.earthdata.nasa.gov/search/site/search_api_docs.html.

**Note: The CMR URL character limit is currently set to take roughly 500k characters. Clients using the GET Search API with query parameters should be careful not to exceed this limit or an HTTP response of 413 FULL HEAD will be returned. Clients who expect that the query URL could exceed 500k characters should use the POST method as opposed to the GET method for searching.*

API calls and parameters GET method

The following examples demonstrates the basic search command:

Example

```
curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/collections"
```

The query returns the first 10 publicly available collection results in a reference list using XML format. *(Note: As described in the header section above, in order to see restricted data, you must have the correct privileges and will need to use a token).*

The search parameters listed below can be applied to provide more functionality:

Page_size - The number of results per page. The default is set at the minimum value =10.0 and the maximum value =2000.

page_size=100 shows 100 result records per page if that many results exist.

Page_num - The results page number to return.

page_num=1 is the first page of results; page_num=2 is the second page of results; page_num=10 is the 10th page of results.

Sort_key - Indicates one or more elements to use for sorting

sort_key[]=platform (the brackets "[" and "]" may need to be escaped by using the \ character)

Pretty- If set to "True" - returns formatted, readable results

pretty = true; *(*Note: This flag is used for all the returned examples in this document)*

Token

Specifies the client token. This is an alternative to using the Authorization: Bearer header.

Echo-compatible

Used by systems requiring ECHO results. To get the best use out of CMR, Client Partners should **NOT** use this parameter.

The following search parameters are for collection requests only:

- include_has_granules - Includes a has-granules tag or attribute in the response so the client knows if the collection encompasses any granules. E.g. include_has_granules=true
- include_granule_counts - Includes a granule-counts tag or attribute in the response with the number of granules represented by the collection. E.g. include_granule_counts=true
- include_facets - Includes a list of facets and their counts at the end of the results. This is mainly used for collection search displays. E.g. include_facets=true
- include_facets with hierarchical_facets - Includes a list of facets preserving the hierarchical order. This is mainly used for collection search displays. E.g include_facets=true&hierarchical_facets=true

The next box depicts a set of examples conducting a basic search using the search parameters described below.

- The first line requests 50 metadata references per page.
- The second line requests 50 metadata references per page using the formatted print.
- The third line requests page 2 of 20 results per page showing full records in the echo 10 specification using formatted print.
- The fourth line issues a request including token and client id headers and does a basic search sorting the results using the platform element.

The request returns page 2 results of 20 results per page showing full records in the ISO 19115 specification using the XML format in a formatted fashion.

Example

curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/collections?page_size=50"

curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/collections?page_size=50&pretty=true"

curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/collections.echo10?page_num=2&page_size=20&pretty=true"

curl -v -i -H "Authorization: Bearer XXXX" -H "Client-Id: Test_Team" "https://cmr.uat.earthdata.nasa.gov/search/collections.iso?sort_key\[\]=platform&page_num=2&page_size=20&pretty=true"

**Note: The "?" character separates the URL from the search parameters and the search parameters are separated from each other by the "&" character. The parameters can be in any order.*

The previous examples all demonstrate collections searches. The same search parameters apply to granules, unless it is explicitly stated that a parameter applies only to collections. To conduct a granule search, simply replace collections with granules in the URL. In the box below are the same four search requests that are listed in the box above, only granules has been substituted for collections.

Example

curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/granules?page_size=50"

curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/granules?page_size=50&pretty=true"

curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/granules.echo10?page_num=2&page_size=20&pretty=true"

curl -v -i -H "Authorization: Bearer XXXX" -H "Client-Id: Test_Team" "https://cmr.earthdata.nasa.gov/search/granules.iso?sort_key\[\]=platform&page_num=2&page_size=20&pretty=true"

The search can be refined using the set of search parameters documented in the table below. These parameters support collection searches. Most of these parameters have the brackets next to them and may need to be escaped (\[\]) depending on the language used or the method by which the query is being sent. All of CMR time search parameters (temporal, updated_since, revision_date, and equator_crossing_date) formats are specified as yyyy-MM-ddTHH:mm:ss.SSSZ format; Where yyyy is year; MM is month; dd is day; T is the date time separator character; HH is the hour; mm is the minute; ss is the second; SSS is the milliseconds (the .SSS can be omitted); and Z specifies Zulu time.

Example

January 2, 2000 at 4 minutes and 5 seconds past 3 o'clock in the morning Zulu time is represented as 2000-01-02T03:04:05Z.

Documented in the table below are collection supported search parameters:

**Note: To get the complete and most up to date set of parameters, visit* https://cmr.earthdata.nasa.gov/search/site/search_api_docs.html.

Search Parameters	Example	Notes	Supports Pattern Option	Supports Case Insensitivity Option	Supports AND Option (ALL values have to be present within an element)	Supports OR Option (ANY value has to be present within an element)
concept_id	concept_id\[\]=C123456-LPDAAC_ECS		NO	NO	NO	NO
echo_collection_id	echo_collection_id\[\]=C1000000001-CMR_PROV2	uses concept_id	NO	NO	NO	NO
entry_title	entry_title\[\]=this is a title		YES	YES	NO	NO
dataset_id	dataset_id\[\]=this is a title	uses entry_title	N/A	N/A	NO	NO
entry_id	entry_id\[\]=SHORT_V5		NO	NO	NO	NO
dif_entry_id	dif_entry_id\[\]=SHORT_V5	matches either entry_id or associated difs	NO	NO	NO	NO
archive_center	archive_center\[\]=SEDAC		YES	YES	NO	NO
temporal	temporal\[\]=2000-01-01T10:00:00Z,2010-03-10T12:00:00Z,30,60 or temporal\[\]=2000-01-01T10:00:00Z/P10Y2M10DT2H,30,60	format is: begin datetime, end datetime, period, duration or: begin datetime/ISO 8601 time interval One can leave out the begin time or end time or both the period and duration ranges are inclusive unless otherwise specified	N/A	N/A	NO	NO
project	project\[\]=ESI		YES	YES	YES	NO
campaign	campaign\[\]=ESI	uses project	YES	YES	YES	NO
updated_since	updated_since=2000-01-01T01:00:00Z	The time is inclusive.	NO	N/A	NO	NO
revision_date	revision_date\[\]=2000-01-01T01:00:00Z,2010-01-01T12:34:56Z revision_date\[\]=2000-01-01T01:00:00Z,	The beginning or ending date time can be left off, but comma must remain. Inclusive boundary search	NO	N/A	YES	NO

processing_level_id	processing_level_id\[\]=1B		YES	YES	NO	NO
platform	platform\[\]=AQUA	platform short name	YES	YES	YES	NO
instrument	instrument\[\]=CERES	instrument short name	YES	YES	YES	NO
sensor	sensor\[\]=CCD	sensor short name	YES	YES	YES	NO
spatial_keyword	spatial_keyword\[\]=VA		YES	YES	YES	NO
science_keywords	science_keywords\[0\]\[category\]=EARTH SCIENCE&science_keywords\[0\]\[topic\]=BIOLOGICAL CLASSIFICATION&science_keywords\[0\]\[term\] =ANIMALS/VERTEBRATES&science_keywords\[0\]\ [variable-level-1\]=MAMMALS&science_keywords\[0\]\ [variable-level-2\]=CARNIVORES&science_keywords\[0\]\ [variable-level-3\]=BEARS	There is a hierarchy for science keywords. These can be ANDed together which is the default or ORed.	NO	NO	YES	YES
two_d_coordinate_system_name	two_d_coordinate_system_name\[\]=Alpha		YES	NO	NO	NO
two_d_coordinate_system [name]	two_d_coordinate_system\[name\]=Alpha	alias of two_d_coordinate_system_name but does not support pattern	NO	NO	NO	NO
collection_data_type	collection_data_type\[\]=NEAR_REAL_TIME	valid values for near real time: "NEAR_REAL_TIME": "near_real_time", "nrt", "NRT", "near real time", "near-real time", "near-real-time", "near real-time" ALSO uses OTHER, SCIENCE QUALITY	NO	YES	NO	NO
provider	provider=ASF		YES	YES	YES	NO
short_name	short_name=MINIMAL		YES	YES	YES	NO
version	version=1	used together with short_name	YES	YES	YES	NO
polygon	polygon=10,10,30,10,30,20,10,20,10,10	Polygon points are provided in counter-clockwise order. The last point should match the first point to close the polygon. The values are listed comma separated in longitude latitude order, i.e. lon1, lat1, lon2, lat2, lon3, lat3, and so on.	N/A	N/A	NO	NO
bounding_box	bounding_box=-10,-5,10,5	Bounding boxes define an area on the earth aligned with longitude and latitude. The Bounding box parameters must be 4 comma-separated numbers: lower left longitude, lower left latitude, upper right longitude, upper right latitude.	N/A	N/A	NO	NO
point	point=100,20	Search using a point involves using a pair of values representing the point coordinates as parameters. The first value is the longitude and second value is the latitude.	N/A	N/A	NO	NO
line	line=-0.37,-14.07,4.75,1.25,25.13,-15.51	Lines are provided as a list of comma separated values representing coordinates of points along the line. The coordinates are listed in the format lon1, lat1, lon2, lat2, lon3, lat3, and so on.	N/A	N/A	NO	NO
keyword	keyword=alpha	By default keyword searches are case insensitive and support wild cards ? and *. The following elements are searched by a keyword search: • Concept ID • Provider ID • Entry Title • data type • short name and long name • abstract • version and version description • processing level id • science keywords • archive centers • additional attribute (names, data types, values, and descriptions) • spatial keywords • temporal keywords • associated DIFs • project short and long names • platform short and long names • instrument short & long names and techniques • sensor short & long names and techniques • characteristic names and descriptions • two d coordinate system names	NO	NO	NO	NO

online_only	online_only=true	valid values: true, false	NO	NO	NO	NO
downloadable	downloadable=true	valid values: true, false	NO	NO	NO	NO
browse_only	browse_only=false	valid values: true, false	NO	NO	NO	NO
browsable	browsable=true	valid values: true, false	NO	NO	NO	NO

Collection Search Parameters

Documented in the table below are granule supported search parameters.

Search Parameters	Example	Notes	Supports Pattern Option	Supports Case Insensitivity Option	Supports AND Option (ALL values have to be present within an element)	Supports OR Option (ANY value has to be present within an element)
granule_ur	granule_ur\[\]=SC:AST_L1B.003:2082836137		NO	NO	NO	NO
producer_granule_id	producer_granule_id\[\]=AST_L1B_00304092000162008_20110111183559_9769.hdf		NO	NO	NO	NO
readable_granule_name	readable_granule_name\[\]=SC:AST_L1B.003:2082836137	matches either granule ur or producer granule id	NO	NO	NO	NO
online_only	online_only=true	valid values: true, false	NO	NO	NO	NO
downloadable	downloadable=true	valid values: true, false	NO	NO	NO	NO
attribute	attribute\[\]=UpperLeftQuadCloudCoverage attribute\[\]=float,UpperLeftQuadCloudCoverage,25.5,30 attribute\[\]=float,UpperLeftQuadCloudCoverage,25.5, attribute\[\]=float,UpperLeftQuadCloudCoverage,,30 attribute\[\]=int,UpperLeftQuadCloudCoverage,4 attribute\[\]=float,UpperLeftQuadCloudCoverage,25.5,30&options\[attribute\]\[or\]=true attribute\[\]=float,UpperLeftQuadCloudCoverage,25.5,30&options\[attribute\]\[exclude_boundry\]=true attribute\[\]=float,UpperLeftQuadCloudCoverage,25.5,30&options\[attribute\]\[exclude_collection\]=true	full syntax:name - attribute name only full syntax: value type, attribute name, min value, max value - range search, can leave off beginning or ending of range, but comma is still needed. Ranges are inclusive. If this is not desired set to true the exclude_boundry option. full syntax:value type, attribute name, value - single value attribute. These searches include the granule collection - if this is not desired set to true the option exclude_collection	NO	NO	YES	YES
polygon	polygon=10,10,30,10,30,20,10,20,10,10	Polygon points are provided in counter-clockwise order. The last point should match the first point to close the polygon. The values are listed comma separated in longitude latitude order, i.e. lon1, lat1, lon2, lat2, lon3, lat3, and so on.	N/A	N/A	NO	NO
bounding_box	bounding_box=-10,-5,10,5	Bounding boxes define an area on the earth aligned with longitude and latitude. The Bounding box parameters must be 4 comma-separated numbers: lower left longitude, lower left latitude, upper right longitude, upper right latitude.	N/A	N/A	NO	NO
point	point=100,20	Search using a point involves using a pair of values representing the point coordinates as parameters. The first value is the longitude and second value is the latitude.	N/A	N/A	NO	NO
line	line=-0.37,-14.07,4.75,1.25,25.13,-15.51	Lines are provided as a list of comma separated values representing coordinates of points along the line. The coordinates are listed in the format lon1, lat1, lon2, lat2, lon3, lat3, and so on.	N/A	N/A	NO	NO

orbit_number	orbit_number=10 orbit_number=0.5,1.5	value or range	NO	NO	NO	NO
equator_crossing_longitude	equator_crossing_longitude=90 equator_crossing_longitude=170,-170	value or range	NO	NO	NO	NO
equator_crossing_date	equator_crossing_date=2000-01-01T10:00:00Z,2010-03-10T12:00:00Z	date range searches can be expressed using ISO 8601	NO	NO	NO	NO
updated_since	updated_since=2015-01-01T13:12:11Z		NO	N/A	NO	NO
revision_date	revision_date\[\]=2015-03-04T16:15:14Z,2015-04-04T17:18:19Z revision_date\[\]=2015-03-04T16:15:14Z,	The beginning or ending date time can be left off, but comma must remain. Inclusive boundary search	NO	N/A	YES	NO
cloud_cover	cloud_cover=-70.0,120.0	The beginning or ending range can be left off, but comma must remain. Inclusive boundary search	NO	N/A	NO	NO
platform	platform\[\]=AQUA	platform short name	YES	YES	YES	NO
instrument	instrument\[\]=CERES	instrument short name	YES	YES	YES	NO
sensor	sensor\[\]=CCD	sensor short name	YES	YES	YES	NO
project	project\[\]=ESI		YES	YES	YES	NO
campaign	campaign\[\]=ESI	uses project	YES	YES	YES	NO
concept_id	concept_id\[\]=G123456-LPDAAC_ECS concept_id\[\]=C123456-LPDAAC_ECS	This finds either the granule or the collection parent record - the difference is in the ID (C vs G)	NO	NO	NO	NO
echo_granule_id	echo_granule_id\[\]=G1000000001-CMR_PROV2	uses concept_id	NO	NO	NO	NO
collection_concept_id	collection_concept_id\[\]=C123456-LPDAAC_ECS		NO	NO	NO	NO
echo_collection_id	echo_collection_id\[\]=C123456-LPDAAC_ECS		NO	NO	NO	NO
day_night_flag	day_night_flag=day	valid values are day, night, unspecified	YES	YES	NO	NO
day_night	day_night=unspecified	valid values are day, night, unspecified - uses the day-night-flag element.	YES	YES	NO	NO
two_d_coordinate_system	two_d_coordinate_system\[\]=wrs-1:5,10:8-10,0-10:8,12	see API docs for description	NO	NO	NO	NO
grid	grid\[\]=wrs-1:5,10:8-10,0-10:8,12	uses two_d_coordinate_system element	NO	NO	NO	NO
provider	provider=ASF		YES	YES	YES	NO
short_name	short_name=MINIMAL		YES	YES	YES	NO
version	version=1	used together with short_name	YES	YES	YES	NO
entry_title	entry_title\[\]=this is a title		YES	YES	NO	NO
temporal	temporal\[\]=2000-01-01T10:00:00Z,2010-03-10T12:00:00Z,30,60 or temporal\[\]=2000-01-01T10:00:00Z/P10Y2M10DT2H,30,60	format is: begin datetime, end datetime, period, duration or: begin datetime/ISO 8601 time interval One can leave out the begin time or end time or both the period and duration ranges are inclusive unless otherwise specified	N/A	N/A	NO	NO
exclude	exclude\[echo_granule_id\]\[\]=G100000006-CMR_PROV exclude\[concept_id\]\[\]=G100000006-CMR_PROV exclude\[concept_id\]\[\]=C100000006-CMR_PROV	exclude metadata records by echo_granule_id, concept id, or parent concept id.	NO	NO	NO	NO

The following example demonstrates a request to find collection metadata records that contain an AQUA platform granule. It further requests that only a formatted reference list of results that contains 20 references be displayed.

Example

curl -v -i -H "Authorization: Bearer XXXX" -H "Client-Id: Test_Team" "https://cmr.uat.earthdata.nasa.gov/search/collections?platform\[\]=AQUA&page_size=20&pretty=true"

The following example demonstrates a request to find collection metadata records that contain an AQUA or an AURA platform granule. It further requests that only a formatted reference list of results that contains 20 references be displayed.

Example

curl -v -i -H "Authorization: Bearer XXXX" -H "Client-Id: Test_Team" "https://cmr.uat.earthdata.nasa.gov/search/collections?platform\[\]=AQUA&platform\[\]=AURA&page_size=20&pretty=true"

Certain search parameters have additional options to aid the user. The syntax to employ them is: options[parameter name][option_key]=value.

- Parameter name is the search parameter to be affected (ex: platform).
- Value is set to "true" or "false".
- Option_key is one of the following:

Option name	Description
ignore_case	If "ignore_case" is set to true the search will be case insensitive and if set to false the search will be case sensitive. The default value is true. E.g. ignore_case=true - the search will match on both AQUA and aqua
pattern	This is the wildcard capability. If "pattern" is set to true the CMR will treat '*' as matches zero or more characters and '?' matches any single character. For example: platform[]=AQUA will match only on the value 'AQUA'. if platform[]=A?U*&options[platform][pattern]=true platforms containing A followed by any alphanumeric character followed by U followed by any number of alphanumeric characters will be found. So AQUA, ASUBB, ADUSD34H, AUU, etc. will all be found. The pattern option defaults to false.
and	If "and" is set to true and if multiple values are listed for the parameter, the metadata records must contain ALL of these values in order to match. The default is false meaning metadata records that match ANY of the values will match.
or	This option only applies to granule attributes or science-keyword searches. If "or" is set to true, the search will find records that match any of the attributes. The default for this option is false.

Extra Options

The following example demonstrates the use of additional options in conjunction with the platform search parameter. A user would make the request in the box below to find any platform that begins with "A" followed by any number of alphanumeric characters and ends with "A". AQUA and AURA would both be located and returned with the results.

Example

curl -v -i -H "Authorization: Bearer XXXX" -H "Client-Id: Test_Team" "https://cmr.uat.earthdata.nasa.gov/search/collections.echo10?platform\[\]=A*A&options\[platform\]\[pattern\]=true&pretty=true"

This example in the box below depicts a request entered to search for records exclusively contains an instrument matching "HELLO" in uppercase letters.

Example

curl -v -i -H "Authorization: Bearer XXXX" -H "Client-Id: Test_Team" "https://cmr.uat.earthdata.nasa.gov/search/collections.echo10?instrument\[\]=HELLO&options\[instrument\]\[ignore-case\]=false&pretty=true"

Besides science_keywords, if any of the parameters that are searched are repeated, the metadata records that have ANY of the values will match. Th e following example demonstrates that the CMR system will match any metadata record containing either concept id.

Example

curl -v -i -H "Authorization: Bearer XXXX" -H "Client-Id: Test_Team" "https://cmr.uat.earthdata.nasa.gov/search/collections.iso?concept_id\[\]=C123456-LPDAAC_ECS&concept_id\[\]=C123457-LPDAAC_ECS&pretty=true"

For a complete set of examples using all of the search parameters, visit the API documentation link at: https://cmr.earthdata.nasa.gov/search/site/search_api_docs.html

API calls and parameters POST method

The API using the POST method is the same as with the GET method, with the exception of: 1) The method name used is POST instead of GET; and 2) the parameters are in the body of the message without a length constraint as opposed to existing in the URL string. The following example uses the curl command and shows the query.xml file that contains the query request. The subsequent example depicts the curl search request. As long as syntax remains the same in the query.xml file, the parameters can be one long set or formatted to be more easily read.

**Note: Notice the brackets ([ ]) were not escaped in this example.*

query.xml:

Example

pretty=true&

page_size=1&

page_num=3&

sort_key[]=platform&platform[]=AQUA&platform[]=AURA&revision_date[]=2015-07-01T01:00:00Z,2016-01-01T01:00:00Z&revision_date[]=2014-01-01T01:00:00Z,2014-06-01T01:00:00Z&temporal[]=2000-01-01T10:00:00Z/2010-03-10T12:00:00Z&include_has_granules=true&include_granule_counts=true&include_facets=true&hierarchical_facets=true

Example

curl -v -XPOST -i -d @query.xml "https://cmr.uat.earthdata.nasa.gov/search/collections.echo10"

**Note: In this specific instance, the Content-Type header is not used. This is deliberate as it will cause an error.*

JSON query language with a POST method

The CMR provides a JSON RESTful interface. The elements that can be searched are the same as described above, but this interface is applicable only to collection searches. This search method provides additional functionality in conducting a search by using conditions (AND, OR, NOT) against the elements. See the JSON schema https://cmr.uat.earthdata.nasa.gov/search/site/JSONQueryLanguage.json for more details. The example provided below demonstrates a query consisting of conditions and several elements.

Example

```
curl -XPOST -H "Content-Type: application/json" -H "Client-Id: GCMD" https://cmr.uat.earthdata.nasa.gov
/search/collections
-d '{"condition": { "and": [{ "not": { "or": [{ "provider": "TEST" },
{ "and": [{ "project": "test-project",
"platform": "mars-satellite" }]}]}},
{ "bounding_box": [-45,15,0,25],
"science_keywords": { "category": "EARTH SCIENCE" }}]}}'
```

Alternative Query Language (AQL)

The CMR supports the ECHO Alternative Query Language (AQL) and is available to clients. However, while the AQL is supported, it will not mature or be integrated with new CMR features. For a detailed explanation and examples of AQL, refer to the ECHO AQL documentation.

Chapter 4: Retrieving Metadata

There are several ways of retrieving metadata:

- Retrieve a result list consisting of full metadata records. While this has been demonstrated above, a example has been replicated here for convenience.

  Example

  curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/collections.native?pretty=true"

  curl -v -i "https://cmr.uat.earthdata.nasa.gov/search/granules.native?pretty=true"

- Use the concept id to retrieve a record: The syntax is https://cmr.uat.earthdata.nasa.gov/search/concepts/<concept id>.
  The concept id/revision number if a specific revision of a metadata record is wanted: The syntax is https://cmr.uat.earthdata.nasa.gov/search/concepts/<concept id>/<revision number>.

  Example

  curl -v "https://cmr.uat.earthdata.nasa.gov/search/concepts/C1000000803-DEV08"

  curl -v "https://cmr.uat.earthdata.nasa.gov/search/concepts/C1000000803-DEV08/7"

  curl -v "https://cmr.uat.earthdata.nasa.gov/search/concepts/G23447-ASF/8

- The CMR supports retrieving metadata records using different specifications and formats, which are listed in the table below:

Type Received	Accept HeaderValue	Supports Revision	Supports Granules	Comments
xml	application/xml	YES	YES	returns a reference list of results using the XML format
json	application/json	NO	YES	returns a subset of metadata data list of results using the JSON format
echo10	application /echo10+xml	YES	YES	returns a full metadata record list of results in the echo 10 specification using the XML format
iso	application /iso19115+xml	YES	YES	returns a full metadata record list of results in the ISO 19115-2 (MENDS) specification using the XML format
iso19115	application /iso19115+xml	YES	YES	returns a full metadata record list of results in the ISO 19115-2 (MENDS) specification using the XML format
dif10	application /dif10+xml	YES	NO	supported for collections only and returns a full metadata record list of results in the DIF 10 specification using the XML format
atom	application /atom+xml	NO	YES	returns a subset of metadata list of results in the ATOM specification using the XML format
native	application /metadata+xml	YES	YES	returns a full metadata record list of results in their individual native specification using the XML format

Supported Standards

Below are several examples using the supported standards mime types:

- Ex 1: Retrieves a granule metadata record in the JSON format.
- Ex 2: Retrieves a granule metadata record with a revision of 8 in the ISO specification.
- Ex 3: Retrieves a collection metadata record with a revision of 7 in the DIF 10 specification. (*Note: If the record was sent and stored using the ECHO10 specification, that record will be translated to the DIF 10 specification and returned to the caller.)*
- Ex 4: Lists a granule record using the native option with the pretty print option turned on. The native option lists the metadata in the specification it was sent and stored within the CMR.

  Example

  curl -v "https://cmr.uat.earthdata.nasa.gov/search/concepts/G23447-ASF.json"

  curl -v "https://cmr.uat.earthdata.nasa.gov/search/concepts/G23447-ASF/8.iso"

  curl -v "https://cmr.uat.earthdata.nasa.gov/search/concepts/C1000000803-DEV08/7.dif10"

  curl -v "https://cmr.uat.earthdata.nasa.gov/search/concepts/G23447-ASF.native?pretty=true"

Chapter 5: Accessing data

There are a variety of ways to access the data of interest from the response of a search query:

1. Access the data provider's site via the landing page in the collection or granule and follow their instructions for data retrieval. Note: *This information is not required and may not be present.*

2. Order the data through the Earthdata Search system, found at https://search.earthdata.nasa.gov.

Acronyms

Acronyms used throughout this document are contained in the table below.

API	Application Programming Interface
AQL	Alternative Query Language
ASF	Alaska Satellite Facility DAAC
COTS	Commercial Off The Shelf
DAAC	Distributed Active Archive Center
ECS	EOSDIS Core System
EOS	Earth Observing System
EOSDIS	EOS Data and Information System
ESDIS	Earth Science Data and Information System
FTP	File Transfer Protocol
GCMD	Global Change Master Directory
GMT	Greenwich Mean Time
NASA	National Aeronautics and Space Administration
SSL	Secure Sockets Layer
URL	Uniform Resource Locator
UTC	Universal Time, Coordinated (also called GMT/UTC)
WRS	Worldwide Reference System
XML	eXtensible Markup Language

Best Practices for CMR Client Operations

Below are some tips and recommended practices to increase overall performance of client interactions with the CMR services.

Enhancing the Speed of Queries

- Limit the end user choices — This will promote efficiency by displaying only choices applicable to user needs.
- Search for collections first and limit the collection search spatially, temporally, and/or by data center — Limiting the collection will result in a narrower search and a smaller, more focused result set.
- Request only what the user would see in the first few pages. For example, if the client only supports displaying 10 pages of 10 items, using a page size of 100 items will allow the client to pre-fetch the next page of results while the user is examining the first page.
- Use the value element — As a general rule, it will be more efficient than range element.

Increasing Efficiency of Spatial Queries

- If querying a single Data Partner — name the Data Partner in the query.
- If querying a single collection — include the name of the collection in the query.
- Note: Queries for smaller spatial regions return faster than queries for broader regions.
- Note: Queries for spatial regions with fewer points return faster results than queries with more points

Optimizing HTTP CMR Requests

- *Reuse HTTP connections* — This keeps the connection open between requests to ingest an item. This isn't a CMR feature itself but a feature of HTTP and the libraries. (Note these are usually not thread safe so you need one per thread)
- *Use multiple threads* — Between 2 and 5 is a reasonable number.
- *Use HTTP compression* — This will allow you to send the metadata efficiently. This isn't required but can potentially speed up the requests.
- *Reuse the authentication token for multiple requests* — The CMR will recognize when a token is reused and can efficiently authenticate and authorize permission for ingesting.