ADAMS Public Search Application Programming Interface (API) – Developer's Guide

Version 1.0

Table of Contents

1. In This Guide

This guide describes how to create a RESTful query to request data from the ADAMS Public Search via the Application Programming Interface (API). The ADAMS Public Search API provides programmatic access to publicly available NRC documents stored in ADAMS. At the end of the guide are appendices listing specific query parameters, operators, and values allowed for the APIs, along with a sample JSON result set to show the format of the data that the API will return when it receives a request.

This guide describes how to:

- ✓ Register for API access through the NRC's API Management (APIM) portal.
- ✓ Subscribe to the ADAMS Public Search API.
- ✓ Use the REST endpoints with JSON input/output to integrate document search and retrieval into applications.

What You Should Know About https://adams-search.nrc.gov Data

Adams.nrc.gov contains millions of full-text documents that the NRC has released since November 1, 1999, and several hundred new documents are added each day.

The following document libraries at https://adams-search.nrc.gov are available to the APIs:

•**Main Public Library.** Contains all image and text documents that the NRC has made public since November 1, 1999— (virtual containers of related documents). This collection includes publicly available regulatory guides, NUREG-series reports, inspection reports, Commission documents, correspondence, and other regulatory and technical documents written by NRC staff, contractors, and licensees.

• **Legacy Public Library.** Contains bibliographic records (some with abstracts and full text) that the NRC made public before November 1999. The Legacy Public Library is available only through the Advanced Search API.

Data will change over the course of each day as the NRC issues new documents. For this reason, counts may vary and may age quickly, depending on site activity and time.

The most current properties available for search are published via the ADAMS Public Search API Developer Portal for each of the endpoints discussed below in section 3.

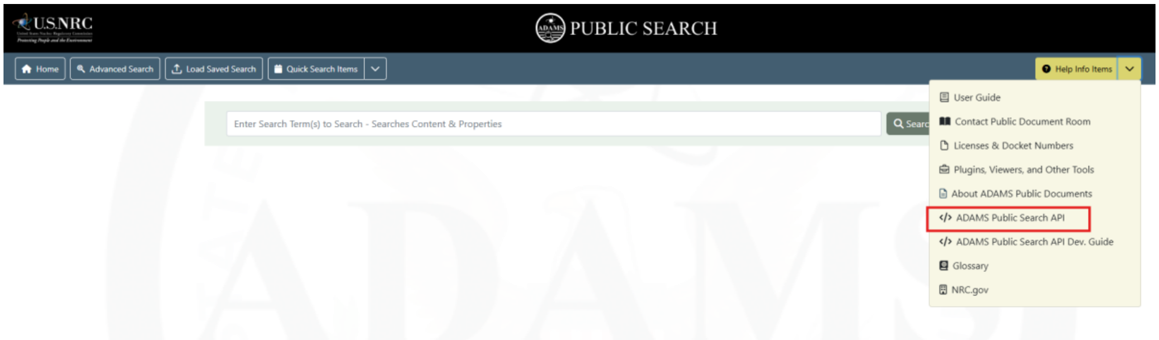
2. The ADAMS Application Programming Interface

The U.S. Nuclear Regulatory Commission (NRC) ADAMS Public Search Application Programming Interface (API) is a web service for accessing data from adams.nrc.gov, which contains a copy of the NRC's official repository of publicly available documents. The Agency-wide Documents Access and Management System (ADAMS) is the NRC's official record system for documents.

By sending RESTful HTTP requests, you can programmatically request lists of documents and associated metadata. The data is output in JSON format, which can be used for data analyses, building custom reports, integrations, and other programmatic access to NRC data. **Section 3** below describes the REST endpoints available and how to make a request.

2.1 Accessing the API Portal

The easiest way to access the **ADAMS Public Search API Developer Portal** is via the Help menu on the ADAMS Public Search website located at: https://adams-search.nrc.gov/ .

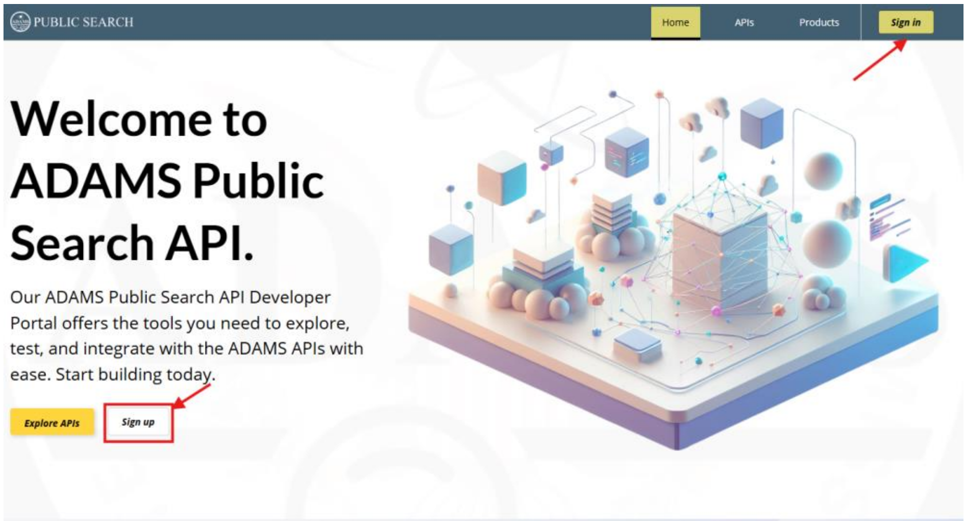

U.S.NRC
PUBLIC SEARCH
Home
Advanced Search
Load Saved Search
Quick Search Items
Help Info Items
Enter Search Term(s) to Search - Searches Content & Properties
Searc
User Guide
Contact Public Document Room
Licenses & Docket Numbers
Plugins, Viewers, and Other Tools
About ADAMS Public Documents
ADAMS Public Search API
ADAMS Public Search API Dev. Guide
Glossary
NRC.gov

The API also has a **Developer Portal** is published at: https://adams-api-developer.nrc.gov/ for more direct access. This is the location where you can register for the API, get a subscription key, and find up to date details on the APIs available from the NRC.

Steps to register to gain access to the API are below.

2.2 Registration

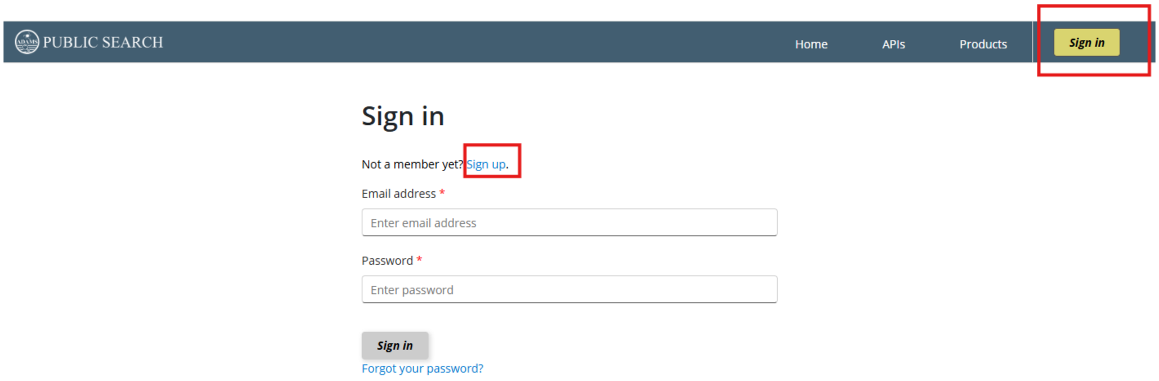
You will need to register to gain access to the API. After visiting https://adams-api-developer.nrc.gov/ , you can find the sign up button in a couple places. The first is on the home page in the hero section which has a button "Sign Up" that will take you to the sign up page.



Welcome to ADAMS Public Search API.

Our ADAMS Public Search API Developer Portal offers the tools you need to explore, test, and integrate with the ADAMS APIs with ease. Start building today.

The other location is via the sign in screen:



Sign in

Not a member yet? Sign up.

Email address *

Enter email address

Password *

Enter password

Sign in

Forgot your password?

Once you complete the sign up form, you will get an email to confirm your email address. You should click the link to confirm your email and sign in. You will now have access to the API and can explore the API endpoints and the request/response objects required to fetch data.

Summary of Steps to Register

1. Navigate to the API Portal.
2. Select Sign up and create an account with your email address.
3. Verify your email and log in.

2.3 Subscription

Before making a REST HTTP call to either of the ADAMS Public Search API endpoints you will need to request a subscription key. This subscription key will provide access to the endpoints and provides a level of security for the endpoints.

Steps to get Subscription Key

1. After logging in, browse the Products section, via the Top Menu.
2. Select **ADAMS Public Search API (ADAMS APS API)**.
3. You will see a section labeled "Your Subscriptions", which will be empty initially.
4. Simply enter a name for your subscription, then click the **Subscribe** button to generate the subscription key.
5. Use this key in the Ocp-Apim-Subscription-Key header when calling the API.
    a. **NOTE:** You can have more than one subscription, one for each of your apps for tracking purposes.

You are now ready to make calls to the ADAMS APS API.

3. API Overview

The ADAMS Public Search API exposes two endpoints described below. All endpoints:

- ✓ Are REST-based HTTP requests
- ✓ Accept and return application/json payloads
- ✓ Require a subscription key

ADAMS APS API Endpoints

1. **Get Document** – Retrieve a document and all associated metadata by Accession Number.
2. **Search Document Library** – Perform boolean and filtered searches across the ADAMS Public Library.

3.1 Endpoints

Below we will describe how to use each endpoint in detail. More examples can be found in the appendix below. You can always test your queries in the APS API Developer Portal.

3.1.1 Get Document

The Get Document endpoint allows you to retrieve a document's metadata properties as well as the indexed content (in plain text) for the document identified with the accessionNumber parameter.

This endpoint uses a HTTP GET verb along with a URI parameter to make the request.

Endpoint: **GET** https://adams-api.nrc.gov/aps/api/search/{accessionNumber}

Description: Returns a single document object by accession number, including metadata and content.

Parameters:

- **accessionNumber** (location: *path, required*): Unique NRC accession number (e.g., ML12345A678).

Sample Get Document Request:

```
GET /aps/api/search/ML12345A678
Host: adams-api.nrc.gov
Ocp-Apim-Subscription-Key: {your_api_subscription_key}
Accept: application/json
```

Sample Response:

```
{
 "document": {
  "Id": "12345",
  "Name": "Inspection Report",
  "AccessionNumber": "ML12345A678",
  "DocumentTitle": "Reactor Inspection Summary",
  "AuthorName": ["NRC Staff"],
  "DocumentDate": "2025-09-01",
  "DocumentType": ["Inspection Report"],
  "Url": "https://...",
  "content": "Base64 or inline text content",
  "EstimatedPageCount": "42"
 }
}
```

3.1.2 Search Document Library

The Search Document Library endpoint allows you to execute Boolean style searches with and/or operators, property filters and library filters (Main or Legacy).

This endpoint uses a HTTP POST verb along with a JSON body in the request to make the query.

Endpoint: **POST** https://adams-api.nrc.gov/aps/api/search

Description: Executes a Boolean search across the ADAMS Public Library.

Request Body Parameters:

- **q (string)** – Search query text
- **filters (array)** – Property filters for AND operations, ***all*** the filters here MUST be true.
- **anyFilters (array)** – Property filters for OR operations, ***any*** the filters here CAN be true.
- **legacyLibFilter (boolean)** – Include legacy library (pre-1999)
- **mainLibFilter (boolean)** – Include main library
- **sort (string)** – Field to sort by (e.g., DocumentDate)
- **sortDirection (integer)** – 0 = Ascending, 1 = Descending
- **skip (integer)** – [Default =0] Number of items to skip for this query. Used primarily to page through results for larger data result sets.

Sample Search Document Library Request:

```
POST /aps/api/search
Host: adams-api.nrc.gov
Ocp-Apim-Subscription-Key: {your_api_subscription_key}
Content-Type: application/json

{

"q":"Safety valve",

"filters":[

                {"field":"DocumentType","value":"Inspection Report","operator":"equals"},

                {"field":"DocumentDate","value":"(DocumentDate ge '2024-01-01')"}

        ],

"anyFilters":[],

"legacyLibFilter":true,

"mainLibFilter":true,

"sort":"",

"sortDirection":1,

"skip":0

}
```

3.2 Best Practices

- Always provide your subscription key in the request header.
- Use filters to narrow down searches and improve performance.
- Expect paged or truncated results for very large queries.

4. Supporting JSON Objects and Options

4.1 Filter Object

The filter object is used in the Search Document Library endpoint as a part of the search request to filter items. The object can be used in either of the boolean filters (filters or anyFilters). Each object item should be added to the filter array for the boolean operator for which the query should apply.

- Filters array for query AND criteria
- anyFilters for OR criteria

The shape of the object is slightly different depending on the property type we are using. There are two types of properties for the filters, text properties and date properties. The shape of the filters are described below.

4.1.1 Text Filters

Most document properties (except those called out below) are considered text filters.

To query these items the filter is made up of an object that contains a field property, a value property, and an operator.

```
{

        "field":"DocumentType",

        "value":"Inspection Report",

        "operator":"equals"

}
```

The **Field** property requires the property name of the metadata field to be filtered. Case matters for the names of the metadata properties. The Developer Portal will have the latest list of available properties.

The **Value** property is the value that should be used to filter the documents. This is similar to the search terms used in the "q" property of the query, but it will be matched against the metadata field supplied in the field property.

The Operator property is the operation that should be used to do the match for the field and value provided. There are 6 operators in ADAMS Public Search, please refer to the APS user guide on the https://adams-search.nrc.gov site for details on these operators. The quoted text for each operator is what should be passed in the operator property:

- Contains ("*contains*") – The field contains the search term
- Does Not Contain ("*notcontains*") – The field does not contain the search term
- Starts With ("*starts*")
- Does Not Start With ("*notstarts*")
- Equals ("*equals*") – exact match of the value for that field
- Does Not Equal ("notequals") – does not match the value exactly

4.1.2 Date Filters

There are currently only 2 Date properties in the ADAMS Public Search index:

- DateAddedTimestamp
- DocumentDate

To query these items the filter is made up of an object that contains a field property and a value property:

```
{

        "field":"DocumentDate",

        "value":"(DocumentDate ge '2024-01-01')"

}
```

The **Field** property requires the property name of the metadata field to be filtered. In the case of dates there are only two properties that should be used:

- **DateAddedTimestamp**
- **DocumentDate**

The **Value** property should contain the filter statement for a date filter there are 3 operators to filter dates:

- **On or Before** => “le” **ex:** “*(DocumentDate le '2024-01-01')*”
- **On or After** => “ge” **ex:** “*(DocumentDate ge '2024-01-01')*”
- **Equals** => “eq” **ex:** “*(DateAddedTimestamp eq '2024-01-01')*”
- **Between** => this is a combination operator that uses the “le” and “ge” operator to set the boundaries for the dates needed.
  ex: “*(DocumentDate ge '2024-01-01') and “(DocumentDate le '2024-12-31')*”

ALL Date values are formatted in a “**YYYY-MM-DD**” format. Any dates provided outside of this format will not match.

5. Support

Developer Portal: https://adams-api-developer.nrc.gov/ Technical documentation and schema examples are available in the portal.

For issues, please email, APSSupport.Resource@nrc.gov and describe the issue you might have.

Appendices

Appendix A: Document Properties

The following properties may be returned in a document object or used as filters in search queries:

- AccessionNumber: Unique identifier for each document
- DocumentTitle: Title of the document
- AuthorName: One or more authors of the document
- AuthorAffiliation: Organization of the author
- AddresseeName: Recipient name if available
- AddresseeAffiliation: Recipient organization
- DocumentDate: Date of the document
- DocumentType: Type of the document (e.g., Inspection Report, Letter)
- Keyword: Keywords associated with the document
- DocketNumber: NRC-assigned docket number (if applicable)
- DateAddedTimestamp: Date the document was added to ADAMS Public Library with timestamp
- EstimatedPageCount: Estimated length of the document
- Url: Direct URI to the document resource

Appendix B: Example Search Queries

Below are two examples of HTTP requests that use the API to search the library with a filter (example 1) and then use an Accession Number to retrieve a particular document.

Send a Boolean Search with Date Filter

Example 1: Find all inspection reports mentioning 'safety valve' published after January 1, 2024.

Below is the REST Request to query:

```
POST https://adams-api.nrc.gov/aps/api/search HTTP/1.1

Content-Type: application/json

Cache-Control: no-cache

Ocp-Apim-Subscription-Key: <YOU-SUBSCRIPTION-KEY>

{

 "q": "safety valve",

  "content": true,

 "filters": [

        {"field": "DocumentType", "operator": "equals", "value": "Inspection Report"},

        {"field": "DocumentDate", "value": "(DocumentDate ge '2024-01-01')"}

 ],

"anyFilters": [ ],

 "mainLibFilter": true,

 "legacyLibFilter": true,

 "sort": "DocumentDate",

 "sortDirection": 1

}
```

Find a Document By AccessionNumber

Example 2: Retrieve a specific document by accession number.

GET https://adams-api.nrc.gov/aps/api/search/***ML2024012345*** HTTP/1.1

Cache-Control: no-cache

Ocp-Apim-Subscription-Key: <YOU-SUBSCRIPTION-KEY>

Accept: application/json

Find Part 21 Reports

Example 3: Part 21 Component Defect Reports can be found using the property search features of the ADAMS Public Search API, which will then return a list of Part 21 reports. In this example we use 1 filter:

- **Document Type** *Equals*: “Part 21 Correspondence”

Below is the REST Request to query:

POST https://adams-api.nrc.gov/aps/api/search HTTP/1.1

Content-Type: application/json

Cache-Control: no-cache

Ocp-Apim-Subscription-Key: <YOU-SUBSCRIPTION-KEY>

```
{

  "q":"",

  "filters":[

                        {

                        "field":"DocumentType",
                        "value":"Part 21 Correspondence",
                        "operator":"equals"

                        }
              ],

 "legacyLibFilter":true,
 "mainLibFilter":true,
 "anyFilters":[],
 "sort":"",
 "sortDirection":1

}
```

Find Operating Reactor Inspection Reports

Example 4: Operating Reactor Inspection Reports can be found using the property search features of the ADAMS Public Search API, which will then return a list of these reports. In this query we use two filters.

- **Document Type** *Starts*: “Inspection Report”
- **Docket Number** *Starts*: “05000”

Below is the REST Request to query:

POST https://adams-api.nrc.gov/aps/api/search HTTP/1.1

Content-Type: application/json

Cache-Control: no-cache

```
Ocp-Apim-Subscription-Key: <YOU-SUBSCRIPTION-KEY>
{
"q":"",
"filters":[

                {

                "field":"DocumentType",

                "value":"inspection report",

                "operator":"starts"

                },

                {

                "field":"DocketNumber",

                "value":"05000",

                "operator":"starts"

                }

        ],

"legacyLibFilter":true,
"mainLibFilter":true,
"anyFilters":[],
"sort":"",
"sortDirection":1
}
```

Appendix C: Sample JSON Result Set

Below is a representative JSON response from a search query:

```
{
  "count": 2,
  "results": [{
    "score": 12.175547,
    "highlights": null,
    "semanticSearch": {
      "rerankerScore": null,
      "captions": null
    },
    "document": {
      "IdT": "ADC8BC2E-D7F9-4089-9D8C-743E50E3B8FE",
      "Id": "{ADC8BC2E-D7F9-4089-9D8C-743E50E3B8FE}",
      "Name": "LaSalle County Station - Biennial Problem Identification and Resolution Inspection Report 05000373/2023012 and 05000374/2023012",
      "AccessionNumber": "ML24017A120",
      "Availability": "Publicly Available",
      "AuthorName": ["Ruiz R"],
      "AuthorAffiliation": ["NRC/RGN-III/DORS/RPB1"],
      "AddresseeName": ["Rhoades D P"],
      "AddresseeAffiliation": ["Constellation Energy Generation, LLC", "Constellation Nuclear"],
      "CaseReferenceNumber": [],
      "Comment": "<! SBW1 !><BR>",
      "ContactPerson": null,
      "DocumentTitle": "LaSalle County Station - Biennial Problem Identification and Resolution Inspection Report 05000373/2023012 and 05000374/2023012",
```

"DocumentTitleEquals": "LaSalle County Station - Biennial Problem Identification and Resolution Inspection Report 05000373/2023012 and 05000374/2023012",

"DocumentTitleWord": "LaSalle County Station - Biennial Problem Identification and Resolution Inspection Report 05000373/2023012 and 05000374/2023012",

"DateAdded": "2024-01-23",

"DateAddedTimestamp": "2024-01-23 08:34",

"DocumentDate": "2024-01-18",

"DocumentType": ["Inspection Report", "Letter"],

"DocumentReportNumber": ["IR 2023012"],

"DocketNumber": ["05000373", "05000374"],

"DistributionListCodes": null,

"DocumentsFiledInPackage": [],

"LicenseNumber": ["NPF-011", "NPF-018"],

"Keyword": ["NRC-002", "gps1", "dnd", "utsPARS"],

"PackageNumber": null,

"PackagesFiledIn": [],

"Url": "https://www.nrc.gov/docs/ML2401/ML24017A120.pdf",

"content": "\nDavid P. Rhoades\nSenior Vice President\nConstellation Energy Generation, LLC\nPresident and Chief Nuclear Officer (CNO)\nConstellation Nuclear\n4300 Winfield Road\nWarrenville, IL 60555\n\nSUBJECT: LASALLE COUNTY STATION - BIENNIAL PROBLEM IDENTIFICATION AND \nRESOLUTION INSPECTION REPORT 05000373/2023012 AND \n05000374/2023012\n\nDear David P. Rhoades:\n\nOn December 15, 2023, the U.S. Nuclear Regulatory Com",

"IsPackage": "No",

"IsLegacy": "No",

"MicroformAddresses": null,

"PhysicalFileLocation": "ADAMS",

"EstimatedPageCount": null,

"AccessionNumberLower": "ml24017a120",

```
            "ItemType": "doc"
        }
    }, {
        "score": 10.164975,
        "highlights": null,
        "semanticSearch": {
            "rerankerScore": null,
            "captions": null
        },
        "document": {
            "AccessionNumber": "ML21234A129",
            "Availability": "Publicly Available",
            "AuthorName": ["Doe B"],
            "AuthorAffiliation": ["NRC/RGN-II/DRS"],
            "AddresseeName": ["Case D G"],
            "AddresseeAffiliation": ["Electric & Power Co"],
            "CaseReferenceNumber": [],
            "Comment": "jrs16<BR>",
            "ContactPerson": null,
            "DocumentTitle": "Power Station - Age Related Degradation Inspection Report",
            "DateAdded": "2024-01-22",
            "DateAddedTimestamp": "2024-01-22 08:45",
            "DocumentDate": "2024-01-18",
            "DocumentType": ["Inspection Report", "Letter"],
            "DocumentReportNumber": ["IR 2023234"],
            "DocketNumber": ["050004521", "05000123"],
            "DistributionListCodes": null,

            "DocumentsFiledInPackage": [],

            "LicenseNumber": ["NRC-052", "NRC-055"],

            "Keyword": ["SUNSI Review Complete", "ODS", "jef1", "utsPARS"],

            "PackageNumber": null,

            "PackagesFiledIn": [],

            "Url": "https://www.nrc.gov/docs/ML2401/ML24123A129.pdf",

            "content": "Inspection report",

            "IsPackage": "No",

            "IsLegacy": "No",

            "MicroformAddresses": null,

            "PhysicalFileLocation": "ADAMS",

            "EstimatedPageCount": "12",

            "AccessionNumberLower": "ml24018a129",

            "ItemType": "doc"

        }

    }],

    "pageNumber": 1

}
```

Appendix E: Support Resources

For the latest API updates, schema definitions, and troubleshooting guides, visit the API portal: https://adams-api-developer.nrc.gov/

For general information about ADAMS and NRC documents, see the NRC Reading Room: http://www.nrc.gov/reading-rm/adams.html