



## **PDS APIs**

*Release B12.1*

**NASA Planetary Data System**

**Jul 26, 2022**



# CONTENTS

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Overview</b>                                 | <b>3</b>  |
| <b>2</b> | <b>Search API User Guide</b>                    | <b>5</b>  |
| 2.1      | Quickstart                                      | 5         |
| 2.1.1    | Search With curl                                | 6         |
| 2.1.2    | Search with Python                              | 6         |
| 2.2      | Query Syntax                                    | 6         |
| 2.2.1    | Endpoints                                       | 7         |
| 2.2.2    | Search Products                                 | 7         |
| 2.2.3    | Resolve A Product Identifier                    | 10        |
| 2.2.4    | Crawl a Data Set Hierarchy                      | 11        |
| 2.3      | Response Formats                                | 12        |
| 2.3.1    | Content Negotiation                             | 12        |
| 2.3.2    | Examples  | 12        |
| 2.3.3    | ops Namespace                                   | 19        |
| 2.3.4    | Missing values                                  | 20        |
| 2.4      | Tutorials/Cookbooks                             | 20        |
| 2.4.1    | Sample JupyterLab Notebooks (Python)            | 20        |
| 2.4.2    | Web Search Interface Tutorial (HTML/Javascript) | 21        |
| 2.4.3    | Search Examples                                 | 21        |
| <b>3</b> | <b>Specifications</b>                           | <b>25</b> |
| <b>4</b> | <b>General API Conventions</b>                  | <b>27</b> |
| 4.1      | Reference Documents                             | 27        |
| 4.2      | General Applicable Open API Conventions         | 27        |
| 4.2.1    | Specification Standard                          | 27        |
| 4.3      | Restful Principles                              | 28        |
| 4.3.1    | Resources                                       | 28        |
| 4.3.2    | Verbs   | 28        |
| 4.3.3    | Resource Representation                         | 28        |
| 4.4      | Other Conventions                               | 28        |
| 4.4.1    | URL Resource Naming: Case                       | 29        |
| 4.4.2    | URL Resource Naming: Plural vs Singular         | 29        |
| 4.4.3    | URL Resource Naming: API Versioning             | 29        |
| 4.5      | Pagination/Sort                                 | 29        |
| <b>5</b> | <b>Tools and Services</b>                       | <b>31</b> |
| 5.1      | Search  | 31        |
| 5.2      | DOI   | 31        |

|          |  |           |
|----------|--|-----------|
| <b>6</b> | <b>Support</b>                         | <b>33</b> |
| 6.1      | Contact Us . . . . .                   | 33        |
| 6.2      | Contribute . . . . .                   | 33        |
| 6.3      | Request a New Tool / Service . . . . . | 33        |
| 6.4      | Report a Bug . . . . .                 | 33        |
| <b>7</b> | <b>Acknowledgements</b>                | <b>35</b> |
| 7.1      | PDS API Working Group . . . . .        | 35        |

The Planetary Data System (PDS) is a federated system of nodes that archive planetary science data.

The PDS Application Programming Interface (API) provides a consistent way for planetary science community to discover and share archival data across PDS. This API is one of the cornerstone applications for providing an integrated worldwide data services platform that enables the efficient discovery, dissemination, use and analysis of internationally sponsored planetary science archives.

PDS is willing to develop ReST-ful web APIs for different applications (so far, registry, dois).

These pages document how to access these APIs.

This web site is also available as a [PDS document](#)



## OVERVIEW

The PDS API base urls are provided under the following pattern:

```
https://pds.nasa.gov/api/{service}/{version}/{service_path+params}
```

where:

- {service}: the service such as 'search' (i.e. registry), 'doi', etc.. This component can have an optional node identifier (e.g. 'search-geo'). Absence of a node implies EN.
- {version}: the version of the service.
- {service\_path+params}: the ReST path for the service, including any query parameters - this is essentially the remaining portion of the URI after the version.

API entries currently available are:

| service | version | scope                   | specification               | user's guide                 |
|---------|---------|-------------------------|-----------------------------|------------------------------|
| search  | 1.0     | search PDS data archive | <a href="#">search_spec</a> | <a href="#">search_guide</a> |
| doi     | 0.2     | manage PDS DOIs         | <a href="#">doi_spec</a>    |                              |

So for example:

```
https://pds.nasa.gov/api/search-geo/1.0/products?limit=10
```

intends to obtain 10 product entries from the 0.4.2 version of the GEO node's search (registry).

The API specifications design is driven by the *PDS API general conventions*





## SEARCH API USER GUIDE

---

**Note:** The current guide is based on the PDS Search API version 1.0

---

**Warning:** Since our servers are not fully populated with all PDS data sets, the examples presented in this user guide may return empty results or 404 (Not Found) errors. If there is a data set you would like added, please contact the [PDS Help Desk](#) for assistance.

The PDS Search API provides endpoints:

- to **search** for bundles, collections and any PDS products with advanced search queries.
- to **browse** the archive hierarchically downward (e.g. collection's products) or upward (e.g. bundles containing products),
- to **resolve** an identifier (lid or lidvid) and retrieve the product label and data where ever it is in the Planetary Data System.

These pages provide a user guide for the PDS Search API.

### 2.1 Quickstart

The following section provides a quickstart guide to try out the PDS Search API.

**Warning:** Since our servers are not fully populated with all PDS data sets, the examples presented in this user guide may return empty results or 404 (Not Found) errors. If there is a data set you would like added, please contact the [PDS Help Desk](#) for assistance.

---

**Note:** curl command line tool is used to request the API in this documentation. curl is available in many operating systems by default. If not, you can get curl from <https://curl.se/> or using a package management tool specific to your operating system (brew, apt, ...).

---

### 2.1.1 Search With curl

1. Open a Terminal window (or your favorite command-line application).
2. Get 5 products' metadata from the API in JSON format:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products?limit=5' \
--header 'Accept: application/json'
```

3. Get 5 products' metadata from the API in XML format:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products?limit=5' \
--header 'Accept: application/xml'
```

4. To view this in a more readable way, you can pipe the output to a file, or pretty print (on Mac/Unix):

```
# Output JSON to a File
curl ... > my_first_query.json

# Pretty print JSON
curl ... | json_pp > my_first_query.json

# Output XML to a File
curl ... > my_first_query.xml
```

More details on how to use the API can be found in the *endpoints* .

### 2.1.2 Search with Python

Alternatively, it is possible to use other tools such as Postman and programming languages such as Python to call the PDS Search API.

To use the **PDS API Python Client**, you can read this other [Quickstart](#)

## 2.2 Query Syntax

**Warning:** Since our servers are not fully populated with all PDS data sets, the examples presented in this user guide may return empty results or 404 (Not Found) errors. If there is a data set you would like added, please contact the [PDS Help Desk](#) for assistance.

**Note:** curl command line tool is used to request the API in this documentation. curl is available in many operating systems by default. If not, you can get curl from <https://curl.se/> or using a package management tool specific to your operating system (brew, apt, ...).

## 2.2.1 Endpoints

The URLs for performing GET requests for searching PDS data are as follows.

The **base URL** of the PDS Search API, for search across all the PDS nodes, is:

```
https://pds.nasa.gov/api/search/1.0/
```

For specific discipline node search, there are node-specific endpoints available giving access to products of one node, for example:

```
https://pds.nasa.gov/api/search-geo/1.0/
```

Where geo is the **Node ID**

The **Node IDs** are:

| Node ID | Node Name                            |
|---------|--------------------------------------|
| atm     | Atmospheres                          |
| en      | Engineering                          |
| geo     | Geosciences                          |
| img     | Imaging                              |
| naif    | Navigation and Ancillary Information |
| ppi     | Planetary Plasma Interactions        |
| rms     | Ring-Moon Systems                    |
| sbnumd  | Small Bodies, Comets                 |
| sbnpsi  | Small Bodies, Asteroids/Dust         |

The main use cases, to search, crawl products or resolve a product identifier are given in the following sections.

## 2.2.2 Search Products

### Request Example

Search for the 10 latest collections which processing level is “Raw”:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/collections' \
  --data-urlencode 'limit=10' \
  --data-urlencode 'q=(pds:Primary_Result_Summary.pds:processing_level eq "Raw")'
```

### Request Template

The requests template is a follow:

```
GET https://pds.nasa.gov/api/search/1.0/{product_class}[?[{query-parameter}={query-parameter-value}
→]*]
```

Where *product\_class* is one of:

- **products**: search among all classes of products (observational products, collections, bundles...)
- **collections**: search among products which class is product\_collection
- **bundles**: search among products which class is product\_bundle

The concept of product class is derived from the [PDS4 standard](#).

## Query Detailed Syntax

### Query Parameters

The query parameters are:

| Query Parameter | Description  | Example  |
|-----------------|--|--|
| q               | (Optional, string) Query string you wish to parse and use for search. See <a href="#">query string syntax</a>  | q=target_name eq "Mars"                                |
| key-word        | (Optional, string) String used for text search on title and description of the PDS4 labels   | keyword=insight  |
| fields          | (Optional, array of strings) Array of fields you wish to return.   | fields=lid,Time_Coordinates.start_date_time            |
| start           | (Optional, integer, default=0) The search result to start with in the returned records. For instance, start=10 will return records 10-19. Useful for pagination of the results.  | start=100  |
| limit           | (Optional, integer, default=100) The number of records/results to return.  | limit=100  |
| sort            | (Optional, string, default=LIDVID) Field to sort on and whether it should be sorted ascending (ASC) or descending (DESC). <i>fieldName asc</i> or <i>fieldName desc</i> . There can be several sort parameters (order is important). | sort=lidvid asc, Time_Coordinates.start_date_time desc |
| summary-only    | (Optional, boolean, default=False) When true, only the summary of the results is returned, not the individual results  | true   |

*q* and *fields* use PDS4 [Fields Dot Notation](#)

### Query String Syntax

An example of query syntax (*q* query parameter) is:

For example:

```
((pds:Primary_Result_Summary.pds:processing_level eq "Raw") and not (ops:Data_File_Info.ops:file_size ge 8942))
```

The query syntax follows the rules:

```
{query} = {comparison}|{group}
{comparison} = {field} {comparison operator} {literal value}
{group} = [not] ({comparison} [[and|or] {group}])
```

- **{field}** follows the [Fields Dot Notation](#). The available fields can be found in responses *summary* object, *properties* attribute.
- **{comparison operator}** are eq, ne, gt, lt, ge, le
- **{literal value}** is either a string between " (double quotes) or a numerical value (float or integer). Wildcards (\*, ?) are supported in strings.

| Operator                    | Description           | Example   |
|-----------------------------|-----------------------|---|
| <i>Comparison Operators</i> |                       |   |
| eq                          | Equal                 | target_name <b>eq</b> "Mars"  |
| ne                          | Not equal             | target_name <b>ne</b> "Saturn"  |
| gt                          | Greater than          | Time_Coordinates.start_date_time <b>gt</b> 2001-05-10T00:00:00Z   |
| ge                          | Greater than or equal | Time_Coordinates.start_date_time <b>ge</b> 2001-05-10T00:00:00Z   |
| lt                          | Less than             | Time_Coordinates.start_date_time <b>lt</b> 2001-06-01T00:00:00Z   |
| le                          | Less than or equal    | Time_Coordinates.start_date_time <b>le</b> 2001-06-01T00:00:00Z   |
| <i>Logical Operators</i>    |                       |   |
| and                         | Logical and           | target_name <b>eq</b> "Mars" <b>and</b> instrument_name <b>eq</b> "hirise"  |
| or                          | Logical or            | target_name <b>eq</b> "Mars" <b>or</b> target_name <b>eq</b> "Phobos"   |
| not                         | Logical negation      | <b>not</b> target_name <b>eq</b> "Mars"   |
| <i>Grouping Operators</i>   |                       |   |
| ()                          | Precedence grouping   | ((target_name <b>eq</b> "Mars" <b>or</b> target_name <b>eq</b> "Phobos") <b>and</b> ( instrument_name <b>eq</b> "hirise" )) |

## Fields Dot Notation

### General Case

The syntax of the field names use a combination of the PDS4 Information Model and [dot notation](#) representations of an XML XPath.

Query parameters will use a combination of an attribute with its parent class in *all lowercase*:

```
{namespace:parent_class}.{namespace:attribute}
```

For example:

```
pds:Science_Facets.pds:discipline_name
pds:Investigation_Area.pds:type
```

The classes and attributes are defined in the [PDS4 Data Dictionaries](#).

The PDS4 data dictionaries are augmented with a specific [ops Namespace](#) which contains attributes managed by the [PDS Registry](#) in addition to the PDS4 labels attributes.

### NOT IMPLEMENTED

In the event that the {parent\_class}.{attribute} combination does sufficiently guarantee uniqueness or sufficiency of search when a class is inherited by multiple classes, additional ancestor classes should be prepended to the query parameter until sufficient uniqueness is attained:

```
{ns:ancestor_class}.{ns:parent_class}.{ns:attribute}
```

If the query parameter grows beyond 3 ancestor classes, a :ref:custom query parameter <Custom Query Parameters> should be considered.

In the event that multiple attributes are to be grouped together for search, the parent class should be used as the query parameter:

```
{ancestor_class}.{parent_class}
```

## Custom Query Parameters

### NOT IMPLEMENTED

There are several cases where custom query parameters are preferred over the Dot Notation, but should only be avoided wherever possible in order to minimize confusion amongst developers attempting to use the API. These are also subject to approval by Search Integration Working Group representative for each node. That member is responsible for providing those updates to Engineering Node.

Some reasons for custom query parameters:

- Combination of multiple attribute values into one
- Special cases where XQuery needs to be used for finding specific values (e.g. instrument/spacecraft described in Observing\_System\_Component class)
- Custom search fields on non-PDS4 metadata (e.g. image tags, operations note, etc.)
- Support common search or PDS4 terminology (e.g. target\_name, lidvid)

## 2.2.3 Resolve A Product Identifier

### Default Resolution

If you know the lid (for example *urn:nasa:pds:insight\_rad*) or lidvid (for example *urn:nasa:pds:insight\_rad::2.1*) identifier of a product, you can retrieve its description, wherever it is managed in the PDS system, with the following request:

```
https://pds.nasa.gov/api/search/1.0/products/{identifier}
```

For example

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:insight_rad::2.1' \
--header 'Accept: application/json'
```

### Search for Latest vs. All Versions

#### Latest Version

By default, when the identifier is a lid (without a version, for example *urn:nasa:pds:insight\_rad*) only the latest description of the product is returned.

The request:

```
https://pds.nasa.gov/api/search/1.0/products/{lid}
```

is equivalent to:

```
https://pds.nasa.gov/api/search/1.0/products/{lid}/latest
```

## All Versions

If you want to retrieve **all** the versions of a product, the request is:

```
https://pds.nasa.gov/api/search/1.0/products/{lid}/all
```

The *all* and *latest* suffixes apply also to all the crawling end-points which description follows.

## 2.2.4 Crawl a Data Set Hierarchy

For a given product with identifier *lidvid1*, you can browse its parent products or children.

### If the Product ‘lidvid1’ Is a Bundle

Get its **collections**:

```
https://pds.nasa.gov/api/search/1.0/bundles/lidvid1/collections/[all|latest]
```

For example, run:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/bundles/urn:nasa:pds:insight_rad::2.1/collections' \
  --header 'Accept: application/json'
```

Get its **observational products**:

```
https://pds.nasa.gov/api/search/1.0/bundles/lidvid1/products/[all|latest]
```

### If the Product ‘lidvid1’ Is a Collection

Get its **bundle**:

```
https://pds.nasa.gov/api/search/1.0/collections/lidvid1/bundles/[all|latest]
```

Get its **observational products**:

```
https://pds.nasa.gov/api/search/1.0/collections/lidvid1/products/[all|latest]
```

### If the Product ‘lidvid1’ Is an Observational Product

Get its **bundle**:

```
https://pds.nasa.gov/api/search/1.0/products/lidvid1/bundles/[all|latest]
```

Get its **collection**:

```
https://pds.nasa.gov/api/search/1.0/products/lidvid1/collections/[all|latest]
```

## 2.3 Response Formats

**Note:** curl command line tool is used to request the API in this documentation. curl is available in many operating systems by default. If not, you can get curl from <https://curl.se/> or using a package management tool specific to your operating system (brew, apt, ...).

### 2.3.1 Content Negotiation

A simple style of [content negotiation](#) is used to match the format requested by the client and the capability of the server.

The client can specify the desired response format by including the HTTP header *Accept*. If no *Accept* header is present in the request, or if the requested content type is not available, the server will provide the response in JSON format by default.

The following table provides a list of the supported HTTP Accept header types:

| Accept Header                      | Format | Note   |
|------------------------------------|--------|--|
| application/json                   | JSON   | Simplified JSON view of the PDS4 metadata label. Contains “flattened” PDS4 properties extracted from the metadata label  |
| application/xml                    | XML    | Same as application/json, but in an XML  |
| application/vnd.nasa.pds.pds4+json | JSON   | JSON response containing the full PDS4 metadata translated to JSON, along with some additional supplemental  |
| application/vnd.nasa.pds.pds4+xml  | XML    | Same as application/vnd.nasa.pds.pds4+json, but in an XML format. This response format contains the original PDS4 labels.  |
| application/kvp+json               | JSON   | JSON response containing key-value-pairs for the applicable metadata.  |
| text/csv                           | CSV    | Returns a CSV table containing values for the parameters in the request. If no parameters were specified in the request, a default set is returned. The first row of the CSV is a header that describes the values in each column. |
| text/html                          | HTML   | JSON response embedded in an HTML body. This format is provided for requests coming from the browsers (e.g. Google Chrome) URL bar.  |

*application/vnd.nasa.pds.pds4+json* and *application/vnd.nasa.pds.pds4+xml* have been chosen to comply with [RFC6838](#)

### 2.3.2 Examples

#### application/json

The request:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:insight_rad:2.1' \
--header 'Accept: application/json'
```

Returns



```

{
  "id": "urn:nasa:pds:insight_rad::2.1",
  "type": "Product_Bundle",
  "title": "Mars InSight Lander Radiometer Data Archive",
  "start_date_time": "2018-05-05T00:00:00Z",
  "stop_date_time": "3000-01-01T00:00:00.000Z",
  "investigations": [
    {
      "id": "urn:nasa:pds:context:investigation:mission.insight",
      "href": "http://localhost:8080/products/urn:nasa:pds:context:investigation:mission.insight"
    }
  ],
  "observing_system_components": [
    {
      "id": "urn:nasa:pds:context:instrument_host:spacecraft.insight",
      "href": "http://localhost:8080/products/urn:nasa:pds:context:instrument_host:spacecraft.insight"
    },
    {
      "id": "urn:nasa:pds:context:instrument:radiometer.insight",
      "href": "http://localhost:8080/products/urn:nasa:pds:context:instrument:radiometer.insight"
    }
  ],
  "targets": [
    {
      "id": "urn:nasa:pds:context:target:planet.mars",
      "href": "http://localhost:8080/products/urn:nasa:pds:context:target:planet.mars"
    }
  ],
  "metadata": {
    "label_url": "/data/bundle_insight_rad.xml",
    "update_date_time": "2018-02-01T00:00:00Z",
    "version": "2.1"
  },
  "properties": {
    "pds:Stream_Text.pds:name": [
      "Introduction to the Radiometer Data Bundle"
    ],
    "pds:Modification_Detail.pds:description": [
      "Pre-peer review version",
      "First release",
      "The collections urn:nasa:pds:insight_rad:data_calibrated and urn:nasa:pds:insight_rad:data_
↳derived were added to this bundle with InSight Release 1b.",
      "Changed Observing_System_Component name in this label from RAD to RADIOMETER to_
↳match context product name. Expanded Citation_Information description."
    ],
    "...",
    "pds:Investigation_Area.pds:type": [
      "Mission"
    ]
  }
}

```

Properties follow the *Fields Dot Notation*.

**application/xml**

The request:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:insight_rad::2.1' \
--header 'Accept: application/xml'
```

Returns:

```
<PdsProduct xmlns="http://pds.nasa.gov/api">
  <id>urn:nasa:pds:insight_rad::2.1</id>
  <type>Product_Bundle</type>
  <title>Mars InSight Lander Radiometer Data Archive</title>
  <description/>
  <start_date_time>2018-05-05T00:00:00Z</start_date_time>
  <stop_date_time>3000-01-01T00:00:00.000Z</stop_date_time>
  <investigations>
    <investigations>
      <title/>
      <id>urn:nasa:pds:context:investigation:mission.insight</id>
      <href>http://localhost:8080/products/urn:nasa:pds:context:investigation:mission.insight</href>
      <type/>
      <description/>
    </investigations>
  </investigations>
  <observing_system_components>
    <observing_system_components>
      <title/>
      <id>urn:nasa:pds:context:instrument_host:spacecraft.insight</id>
      <href>http://localhost:8080/products/urn:nasa:pds:context:instrument_host:spacecraft.insight
→</href>
      <type/>
      <description/>
    </observing_system_components>
    <observing_system_components>
      <title/>
      <id>urn:nasa:pds:context:instrument:radiometer.insight</id>
      <href>http://localhost:8080/products/urn:nasa:pds:context:instrument:radiometer.insight</
→href>
      <type/>
      <description/>
    </observing_system_components>
  </observing_system_components>
  <targets>
    <targets>
      <title/>
      <id>urn:nasa:pds:context:target:planet.mars</id>
      <href>http://localhost:8080/products/urn:nasa:pds:context:target:planet.mars</href>
      <type/>
      <description/>
    </targets>
  </targets>
  <metadata xmlns="">
    <archive_status xmlns="http://pds.nasa.gov/api"/>
```

(continues on next page)

(continued from previous page)

```

    <creation_date_time xmlns="http://pds.nasa.gov/api"/>
    <label_url xmlns="http://pds.nasa.gov/api">/data/bundle_insight_rad.xml</label_url>
    <update_date_time xmlns="http://pds.nasa.gov/api">2018-02-01T00:00:00Z</update_date_
↪time>
    <version xmlns="http://pds.nasa.gov/api">2.1</version>
  </metadata>
  <properties>
    <pds:Stream_Text.pds:name>Introduction to the Radiometer Data Bundle</pds:Stream_Text.
↪pds:name>
    <pds:Modification_Detail.pds:description>Pre-peer review version</pds:Modification_Detail.
↪pds:description>
    ...
    <pds:Investigation_Area.pds:type>Mission</pds:Investigation_Area.pds:type>
  </properties>
</PdsProduct>

```

Tag names under *properties* follow the *Fields Dot Notation*.

### application/vnd.nasa.pds.pds4+json

The request:

```

curl --get 'https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:insight_rad::2.1' \
--header 'Accept: application/vnd.nasa.pds.pds4+json'

```

Returns:

```

{
  "id": "urn:nasa:pds:insight_rad::2.1",
  "meta": {
    "node_name": "PDS_ENG",
    "ops:Label_File_Info": {
      "ops:file_name": "bundle_insight_rad.xml",
      "ops:file_ref": "/data/bundle_insight_rad.xml",
      "ops:creation_date": "2020-01-15T17:40:30Z",
      "ops:file_size": "6805",
      "ops:md5_checksum": "adfd86bbf2573c37d862e27e08f332db"
    },
    "ops>Data_Files": [
      {
        "ops:file_name": "readme.txt",
        "ops:file_ref": "/data/readme.txt",
        "ops:creation_date": "2020-01-03T17:58:09Z",
        "ops:file_size": "1114",
        "ops:md5_checksum": "192de32c12437c180a9e14d60fe4b89a",
        "ops:mime_type": "text/plain"
      }
    ],
    "ops:Tracking_Meta": [
      {
        "ops:archive_status": "archived"
      }
    ]
  }
}

```

(continues on next page)

(continued from previous page)

```

    ]
  },
  "pds4": {
    "Product_Bundle": {
      "Identification_Area": {
        "product_class": "Product_Bundle",
        "Modification_History": {
          "Modification_Detail": [
            {
              "modification_date": "2018-02-01",
              "description": "Pre-peer review version",
              "version_id": 0.1
            },
            {
              "modification_date": "2019-04-22",
              "description": "First release",
              "version_id": 1
            },
            "... "
          ]
        },
        "information_model_version": "1.11.0.0",
        "logical_identifier": "urn:nasa:pds:insight_rad",
        "version_id": 2.1,
        "Citation_Information": {
          "publication_year": 2018,
          "description": "The InSight Radiometer data bundle consists of data in three collections:\r\n
            ↪ data_raw, data_calibrated, and data_derived.\r\n
            ↪ The bundle also\r\n
            ↪ includes the HP3/RAD Software Interface Specification in \r\n
            ↪ the HP3/RAD document\r\n
            ↪ collection.",
          "author_list": "InSight RAD Science Team",
          "doi": "10.17189/1517568"
        },
        "title": "Mars InSight Lander Radiometer Data Archive"
      },
      "... "
    }
  }
}

```

*pds4* property contains a translation in JSON of the PDS4 XML Label.

In addition a *meta* object contains fields related to the managed of the record in the [PDS Registry](#), see *ops Namespace* for details.

**application/vnd.nasa.pds.pds4+xml**

The request:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:insight_rad::2.1' \
--header 'Accept: application/vnd.nasa.pds.pds4+xml'
```

Returns:

```
<pds_api:product xmlns:pds_api="http://pds.nasa.gov/api" xmlns:ops="https://pds.nasa.gov/pds4/
→ops/v1">
  <pds_api:id>urn:nasa:pds:insight_rad::2.1</pds_api:id>
  <pds_api:meta>
    <node_name>PDS_ENG</node_name>
    <ops:Label_File_Info>
      <ops:file_name>bundle_insight_rad.xml</ops:file_name>
      <ops:file_ref>/data/bundle_insight_rad.xml</ops:file_ref>
      <ops:creation_date>2020-01-15T17:40:30Z</ops:creation_date>
      <ops:file_size>6805</ops:file_size>
      <ops:md5_checksum>adfd86bbf2573c37d862e27e08f332db</ops:md5_checksum>
    </ops:Label_File_Info>
    <ops>Data_Files>
      <ops>Data_Files>
        <ops:file_name>readme.txt</ops:file_name>
        <ops:file_ref>/data/readme.txt</ops:file_ref>
        <ops:creation_date>2020-01-03T17:58:09Z</ops:creation_date>
        <ops:file_size>1114</ops:file_size>
        <ops:md5_checksum>192de32c12437c180a9e14d60fe4b89a</ops:md5_checksum>
        <ops:mime_type>text/plain</ops:mime_type>
      </ops>Data_Files>
    </ops>Data_Files>
    <ops:Tracking_Meta>
      <ops:Tracking_Meta>
        <ops:archive_status>archived</ops:archive_status>
      </ops:Tracking_Meta>
    </ops:Tracking_Meta>
  </pds_api:meta>
  <pds_api:pds4>
    <Product_Bundle
xmlns="http://pds.nasa.gov/pds4/pds/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://pds.nasa.gov/pds4/pds/v1 https://pds.nasa.gov/pds4/pds/v1/PDS4_
→PDS_1B00.xsd">
      <Identification_Area>
        <logical_identifier>urn:nasa:pds:insight_rad</logical_identifier>
        <version_id>2.1</version_id>
        <title>Mars InSight Lander Radiometer Data Archive</title>
        <information_model_version>1.11.0.0</information_model_version>
        <product_class>Product_Bundle</product_class>
        <Citation_Information>
          <author_list>InSight RAD Science Team</author_list>
          <publication_year>2018</publication_year>
          <doi>10.17189/1517568</doi>
          <description>
```

(continues on next page)

(continued from previous page)

```

    The InSight Radiometer data bundle consists of data in three collections:
    data_raw, data_calibrated, and data_derived.
    The bundle also includes the HP3/RAD Software Interface Specification in
    the HP3/RAD document collection.
    </description>
    </Citation_Information>
    ...
    </Identification_Area>
    ...
    </Product_Bundle>
  </pds_api:pds4>
</pds_api:product>

```

The tag *pds\_api:pds4* contains the XML PDS4 label.

In addition a *meta* object contains fields related to the managed of the record in the [PDS Registry](#), see *ops Namespace* for details.

### application/kvp+xml

This format is useful when one only need a few fields from the metadata.

The request:

```

curl --get 'https://pds.nasa.gov/api/search/1.0/products?limit=10&fields=lidvid&fields=title' \
--header 'Accept: application/kvp+json'

```

Returns:

```

{
  "summary": {
    "q": "",
    "hits": 17,
    "took": 55,
    "start": 0,
    "limit": 10,
    "sort": [],
    "properties": [
      "lidvid",
      "title"
    ]
  },
  "data": [
    {
      "lidvid": "urn:nasa:pds:insight_rad:data_derived::7.0",
      "title": "InSight RAD Derived Data Collection"
    },
    {
      "lidvid": "urn:nasa:pds:insight_rad:data_raw::8.0",
      "title": "InSight RAD Raw Data Collection"
    },
    ...
  ]
}

```

(continues on next page)

(continued from previous page)

```

}
}

```

Properties follow the *Fields Dot Notation* when they are coming from the PDS4 standard or the *ops Namespace*.

### text/csv

This format is useful when one only need a few fields from the metadata.

The request:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products?limit=10&fields=lidvid&fields=title' \
--header 'Accept: text/csv'
```

Returns:

```
lidvid,title
"urn:nasa:pds:insight_rad:data_derived::7.0","InSight RAD Derived Data Collection"
"urn:nasa:pds:insight_rad:data_raw::8.0","InSight RAD Raw Data Collection"
"urn:nasa:pds:insight_rad:data_derived:hp3_rad_der_00014_20181211_073042::1.0","InSight HP3_
↪Radiometer Experiment Derived Product:hp3_rad_der_00014_20181211_073042"
...
```

### Open Data

NOT IMPLEMENTED

See <<https://project-open-data.cio.gov/>>\_ and example of application at <<https://cmr.earthdata.nasa.gov/search/site/docs/search/api.html#data>>\_

### 2.3.3 ops Namespace

The response content, in addition to the information found in the PDS4 label contains some attributes related to the management of the datasets in the registry.

A dedicated namespace *ops* is used to manage these attributes in the API, for example: *ops:Label\_File\_Info.ops:file\_name* in *Fields Dot Notation* used in JSON or in XML tag *<ops:Label\_File\_Info><ops:file\_name>*.

The list of *ops* attributes is given in the following table:

| Class           | Attributes         | Description | Example                          |
|-----------------|--------------------|-------------|----------------------------------|
| Data_File_Info  | creation_date_time |             | 2021-09-10T15:58:03Z             |
|                 | file_name          |             | collection_document_hp3rad.csv   |
|                 | file_ref           |             | <a href="#">link</a>             |
|                 | file_size          | (in bites)  | 137                              |
|                 | md5_checksum       |             | cd24cbc46c45ed023f039b3e2beb6606 |
|                 | mime_type          |             | text/plain                       |
| Label_File_Info | creation_date_time |             | 2021-09-10T15:58:03Z             |
|                 | file_name          |             | collection_document_hp3rad.xml   |
|                 | file_ref           |             | <a href="#">link</a>             |
|                 | file_size          | (in bites)  | 8655                             |
|                 | md5_checksum       |             | aa584be2cd34d1899f19d39c23ccba1  |
| Harvest_Info    | harvest_date_time  |             | 2021-11-16T06:03:30.952311900Z   |
|                 | node_name          |             | PDS_GEO                          |
| Tracking_Meta   | archive_status     |             | archived                         |

### 2.3.4 Missing values

Properties with empty or null values should be dropped from the JSON response unless the user asked specifically for the field (through *field* API parameter). In this case the value must be **null**, without quotes.

#### Rationale

If a property is optional or has an empty or null value, consider dropping the property from the JSON, unless there's a strong semantic reason for its existence (taken from this [discussion](#))

Following interactions with OGC/EDR specification group: see [ticket](#)

We choose **null** without quotes for missing values of fields explicitly requested by the user.

We conform to EDR specification for this aspect, see [EDR parameter response](#)

This should not be mistaken for an actual PDS4 value since missing values in PDS4 labels. are detailed with a nil:reason attribute.

## 2.4 Tutorials/Cookbooks

### 2.4.1 Sample JupyterLab Notebooks (Python)

The following Git repository contains example JupyterLab notebooks for the application programmer's interface (API) of the Planetary Data System, that can be used as a tutorial to work with the PDS Search API.

<https://github.com/NASA-PDS/pds-api-notebook/>



## 2.4.2 Web Search Interface Tutorial (HTML/Javascript)

TO BE IMPLEMENTED

## 2.4.3 Search Examples

Here are some examples of Search API Recipes:

### Search API Cookbook

Recipes for various search scenarios using the PDS Search API.

---

**Note:** curl command line tool is used to request the API in this documentation. curl is available in many operating systems by default. If not, you can get curl from <https://curl.se/> or using a package management tool specific to your operating system (brew, apt, ...).

---

### Search For Product Versions

Recipes for searching for the latest version of a product, or all versions of a product, including superseded versions.

**Warning:** Since our servers are not fully populated with all PDS data sets, the examples presented in this user guide may return empty results or 404 (Not Found) errors. If there is a data set you would like added, please contact the [PDS Help Desk](#) for assistance.

### Search the Latest Version of a Product

To retrieve the **latest** versions of product urn:nasa:pds:mars2020.spice, the request is:

The request:

`https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:mars2020.spice`

which is equivalent to:

`https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:mars2020.spice/latest`

### Search for All Versions of a Product

To retrieve **all** the versions of product urn:nasa:pds:mars2020.spice, the request is:

`https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:mars2020.spice/all`

---

### Search By Specific Metadata

The following recipes describe some example queries of the Search API using the *q=* query parameter showing some more complex use cases for querying PDS data.

**Warning:** Since our servers are not fully populated with all PDS data sets, the examples presented in this user guide may return empty results or 404 (Not Found) errors. If there is a data set you would like added, please contact the [PDS Help Desk](#) for assistance.

### Search by Processing Level

Search for the 10 latest collections which processing level is “Raw”:

**Query:** (pds:Primary\_Result\_Summary.pds:processing\_level eq "Raw")

Listing 1: curl command

```
curl --get 'https://pds.nasa.gov/api/search/1.0/collections' \
  --data-urlencode 'limit=10' \
  --data-urlencode 'q=(pds:Primary_Result_Summary.pds:processing_level eq "Raw")'
```

### Search by Target

Search for all Observational Products targeting Bennu:

**Query:** (ref\_lid\_target eq "urn:nasa:pds:context:target:asteroid.101955\_bennu")

Listing 2: curl command

```
curl --get 'https://pds.nasa.gov/api/search/1.0/collections' \
  --data-urlencode 'q=(ref_lid_target eq "urn:nasa:pds:context:target:asteroid.101955_bennu")'
```

---

### Search for DOIs

[Digital Object Identifiers](#) are useful to cite the data you are using in your research. DOIs for PDS data are minted for PDS4 Bundles, PDS4 Collections, PDS4 Documents, and PDS3 Data Sets. The level at which the DOI is minted differs from data set to data set.

The following recipes describe how to find a DOI for a particular product or data set in the Search API metadata.

See the [DOI Search](<https://pds.nasa.gov/tools/doi/>) for an online interface for searching this information.

See the documentation on [Citing PDS Data](#) for more information on how to use a DOI to cite your data.

**Warning:** Since our servers are not fully populated with all PDS data sets, the examples presented in this user guide may return empty results or 404 (Not Found) errors. If there is a data set you would like added, please contact the [PDS Help Desk](#) for assistance.

## How to Find the DOI associated with an Observational Product

We assume you know the identifier of the product you are working with. In our example it is: `urn:nasa:pds:insight_rad:data_derived:hp3_rad_der_00014_20181211_073042::1.0`

### Search for a Products Collection DOI

Run the following request to get the DOI associated with the product's collection:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:insight_rad:data_derived:hp3_
↪rad_der_00014_20181211_073042::1.0/collections' \
--data-urlencode 'fields=ops:Identifiers/ops:doi' \
--header 'Accept: application/kvp+json'
```

You will get the following result:

```
{
  "summary": {
    "q": "",
    "hits": 1,
    "took": 172,
    "start": 0,
    "limit": 100,
    "sort": [],
    "properties": []
  },
  "data": [
    {
      "ops:Identifiers/ops:doi": "10.17189/1517568"
    }
  ]
}
```

Note that you might not find any DOI at the collection level, in this case you can try to get the DOI from the bundle.

### Search for a Product's Bundle DOI

To get the DOI associated with the bundle the observational product `urn:nasa:pds:insight_rad:data_derived:hp3_rad_der_00014_20181211_073042::1.0` belongs to:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products/urn:nasa:pds:insight_rad:data_derived:hp3_
↪rad_der_00014_20181211_073042::1.0/bundles' \
--data-urlencode 'fields=pds:Citation_Information/pds:doi,ops:Identifiers/ops:doi' \
--header 'Accept: application/kvp+json'
```

You will get the same response as for a *collection request*

## How to Find the PDS Product Associated with a DOI

To get the PDS product metadata associated with a the DOI *10.17189/1517568*:

```
curl --get 'https://pds.nasa.gov/api/search/1.0/products' \
  --data-urlencode 'q=(ops:Identifiers/ops:doi eq "10.17189/1517568")' \
  --header 'Accept: application/json'
```

You will get a JSON response of the PDS products (any class of product, for example collections or bundles) which have referenced the given DOI.

You can get the result in different format using content negotiation with the Accept header parameter.

**Looking for more recipes? Or have some useful recipes of your own?** Checkout the [PDS API Discussion Board](#) or contact the [PDS Help Desk](#)

**Looking for more recipes? Or have some useful recipes of your own?** Checkout the [PDS API Discussion Board](#) or contact the [PDS Help Desk](#)

## SPECIFICATIONS

Each published version of NASA PDS APIs is documented here:

- [Search API v1.0.0](#)
- [DOI API v0.2](#)

More details and rationale for the design can be found in the *general conventions* and the *search API user's guide*.



## GENERAL API CONVENTIONS

### 4.1 Reference Documents

Several websites and documents were used as references for designing this API and the accompanying guidelines, including:

1. [Open API Initiative](#)
2. [Open APIs Specification](#)
3. [Microsoft API Guidelines](#)
4. [Microsoft API Design Best Practices](#)
5. [NASA Earth Data APIs](#)
6. [Google Custom Search REST API](#)
7. [EPN-TAP](#)
8. [Earth Data Common Metadata Repository \(CMR\)](#)
9. [Swagger for Developing API Spec](#)
10. [Open Search](#)
11. [Library of Congress Search/Retrieval by URL](#)
12. [PDS OPUS API](#)
13. [PDS Imaging Atlas API](#)
14. [OGC Environmental Data Retrieval](#)

### 4.2 General Applicable Open API Conventions

#### 4.2.1 Specification Standard

The API complies with Open API 3.0.

## 4.3 Restful Principles

### 4.3.1 Resources

Resources are coded as URI (e.g. <http://domain/api/pets>). Resources should be nouns (verbs are bad)

### 4.3.2 Verbs

Users interact with resources through [HTTP request verbs](#). The PDS API uses GET and POST:

- GET is relevant to get resource representation from the API when the extraction criteria is simple.
- POST, in a read-only context, is relevant to provide the API with complex request criteria.

Future iterations of the API will transform it to be an [idempotent REST API](#), utilizing GET, PUT, DELETE, HEAD, OPTIONS and TRACE HTTP methods.

### 4.3.3 Resource Representation

When a HTTP request verb (e.g. GET, POST, etc.) is applied to a resource (e.g. <http://domain/api/pets>) he/she gets a resource representation.

Many flavors of representations are possible to be returned from a single resource. For example: subsets of a whole, formats, versions, etc...

The resource representation should be self-described as much as possible.

They should be wrapped in envelopes which prevent from vulnerabilities linked to the direct access to json arrays in javascript code (see <https://haacked.com/archive/2008/11/20/anatomy-of-a-subtle-json-vulnerability.aspx/>). A response with this format is fine:

```
{
  "summary": {"..."},
  "data": ["..."]
}
```

## 4.4 Other Conventions

Beyond the OpenAPI standard, there are multiple options regarding general design of an API. We primarily use the following source which is very complete and not too dogmatic: <https://www.moesif.com/blog/api-guide/api-design-guidelines/>

Some peer web API specifications are also considered as references for the design for the PDS API specification:

- [ESDIS Common Metadata Repository API](#)
- [OGC environment data retrieval](#)



### 4.4.1 URL Resource Naming: Case

**Kebab-case** (lower case and hyphens '-' used to fill the spaces in) is used for url resource naming.

For example:

- `discipline-nodes` in <http://pds.nasa.gov/api/references/0.1/discipline-nodes>

(for the rationale see this [stackoverflow](#) discussion)

### 4.4.2 URL Resource Naming: Plural vs Singular

Resources are named plural or singular depending on the use case.

Plural are used when the resources is a collection the user will subset from, for example `/pets/scooby-doo` or `/planets/mars` or `/users/user-id` or `collections?q=...`

Singular are used when the resource is accessed as one. For example `/profile` to access the profile of the current user.

See this [post](#) for more details.

### 4.4.3 URL Resource Naming: API Versioning

The API will have versions and the deployed versions are likely to be heterogeneous in the PDS system.

Two options have been considered to manage versions (see <https://restfulapi.net/versioning/> :

- Version in the URL, e.g. `pds.nasa.gov/api/search/1.0/`
- Content negotiation headers (e.g. `Accept: application/vnd.example+json;version=1.0`)

To keep things as simple as possible, content negotiation will not be used for version management. A server API implementation will implement a single version of the API definition.

However:

- We advise to use the version in the URL of the API when it is deployed, although it is not part of the API definition.
- The version is mandatory in the resource representations (result of a request)

## 4.5 Pagination/Sort

The query parameters for pagination are:

| Parameter | Description                                     |
|-----------|---|
| start     | Index of first item returned in the response    |
| limit     | Maximum number of item expected in the response |

See <https://www.moesif.com/blog/technical/api-design/REST-API-Design-Filtering-Sorting-and-Pagination/>



## TOOLS AND SERVICES

### 5.1 Search

- Server: <https://github.com/NASA-PDS/registry-api>
- Client: <https://github.com/NASA-PDS/pds-api-client>

### 5.2 DOI

- Server: <https://github.com/NASA-PDS/doi-service/>
- Client (DOI Editor Web App): <https://github.com/NASA-PDS/doi-ui/>
- Client (DOI Search Web App): <https://github.com/NASA-PDS/wds-react>



**SUPPORT**

## 6.1 Contact Us

Feel free to contact us the [PDS Operator](#) for any additional questions, comments or concerns.

## 6.2 Contribute

We are always looking for community input into our software. Feel free to contribute to the PDS repos:

- Contribute to this documentation: <https://github.com/NASA-PDS/NASA-PDS.github.io>
- Contribute to a specific software project: <https://github.com/NASA-PDS>

## 6.3 Request a New Tool / Service

Feel free to propose a new idea to our [Engineering Node Operations repo](#) and we will triage it appropriately.

## 6.4 Report a Bug

To report a bug found in either a software tool or services, report in the individual repo for the project, or in our [Engineering Node Operations repo](#) and we will triage it appropriately.



## ACKNOWLEDGEMENTS

The PDS Search API is designed by a working group involving all the nodes of the Planetary Data System.

### 7.1 PDS API Working Group

- Mcclanahan, Timothy (PDS PO)
- Lynn Neakrase (ATM)
- Ed Guinness (GEO)
- Tom Stein (GEO)
- Dan Scholes (GEO)
- Mark Bentley (ESA)
- Myche McAuley (IMG)
- In Sook Moon (PPI)
- Rob French (RMS)
- Matt Tiscareno (RMS)
- Conor Kingston (SBN)
- David Chang (SBN)
- Daniel Darg (SBN)
- Emily Law (EN)
- Vivian Tang (EN)
- Al Niessner (EN)
- Ramesh Maddegoda (EN)
- Yevgen Karpenko (EN)
- Thomas Loubrieu (EN)
- Jordan Padams (EN)
- Boris Semenov (NAIF)