

Observatoire  
de Paris

PSL



OSCARS

Astro-CC

This work has received funding from the European Commission's Horizon Europe Research and Innovation programme, through the ASTRO-CC (Astronomy Competence Center) cascading grant of the Open Science Clusters' Action for Research and Society (OSCARS) European project under grant agreement No. 101129751. It is also supported by PADC (Paris Astronomical Data Centre), Observatoire de Paris-PSL and CNES (Centre National d'Études Spatiales)

# HAPI Tutorial

## A simple heliophysics time-series access protocol

*Shamelessly inspired from Vandegriff et al. (2025), DASH conference talk. [doi:10.5281/zenodo.17387480](https://doi.org/10.5281/zenodo.17387480)*

**Cecconi, B. — ASTRO-CC Data Provider Forum — Heidelberg, 25-27 March 2026**

# What is HAPI?

## Heliophysics API

- HAPI is a data access specification for time-series data



- <https://github.com/hapi-server/data-specification>

# What is HAPI?

## Heliophysics API

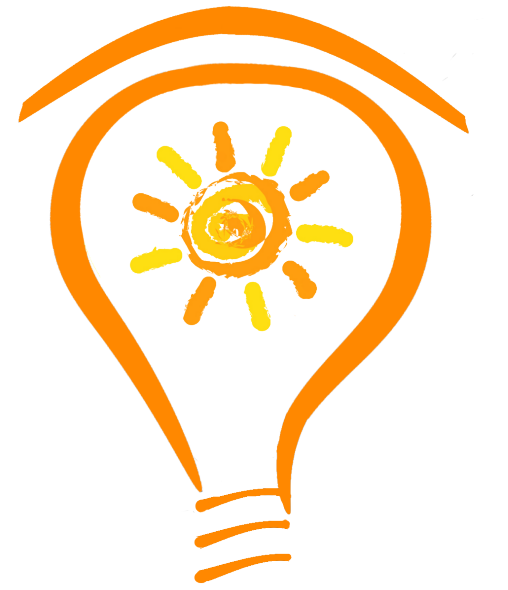
- HAPI is a way to hide data storage details behind a simple access API



- hide complexity, be interoperable

# What is HAPI?

## Heliophysics API



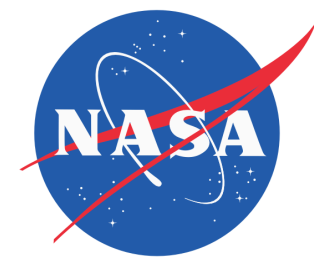
<https://ihdea.net>

- HAPI is a **widely adopted standard in Heliophysics:**
  - *ground-based measurements (magnetometers, radars, aurora)*
  - *upper atmosphere*
  - *ionosphere*
  - *magnetosphere*
  - *the Sun*
  - *heliosphere*
  - *planetary plasma environments*
  - *space weather*
  - *modeling of any of the above*

# What is HAPI?

## Heliophysics API

- HAPI is an official COSPAR standard for space weather data
- HAPI is in use at multiple Heliophysics institutions



### COSPAR Panel on Space Weather Resolution on Data Access

Accepted at COSPAR PSW Business Meeting on 18 July 2018 (updated 15 October 2021).

Taking into account that:

1. It is in the general interest of the international heliophysics and space weather community that data be made as widely accessible as possible,
2. The open exchange of data benefits from well-defined and standardized methods of access,
3. The ILWS-COSPAR Roadmap has recommended to standardize metadata and harmonize access to data and model archives, and
4. The Heliophysics Application Programmer's Interface (HAPI) specification has demonstrated that it is comprehensive and can meet the needs of the community,

The COSPAR PSW resolves that there is a need for at least one common data access API to facilitate and enhance international access to data.

Therefore, it is recommended that:

1. HAPI (<https://doi.org/10.5281/zenodo.4757597>) be the common data access API for space science and space weather data.
2. Funding agencies provide encouragement and adequate support to enable data produced by projects to be accessed by using HAPI compliant services.

# Further details

- Main website: <https://hapi-server.org/>
- Weigel, R. S., Vandegriff, J., Faden, J., King, T., Roberts, D. A., Harris, B., et al. (2021). *HAPI: An API standard for accessing Heliophysics time series data*. *Journal of Geophysical Research: Space Physics*, 126, e2021JA029534. <https://doi.org/10.1029/2021JA029534>
- All HAPI data accessible via: <http://hapi-server.org/servers/>
- Tools and clients available at: <http://github.com/hapi-server/>
- Contact: [hapi-help@groups.com](mailto:hapi-help@groups.com)

# HAPI Interface specification

<https://github.com/hapi-server/data-specification>

- Current version is 3.3.1
- A HAPI server must implement the following endpoints

<b>/hapi/capabilities</b>	lists the output formats the server can stream (csv, binary, or json).
<b>/hapi/about</b>	lists the server id and title, contact information, and a brief description of the datasets served.
<b>/hapi/catalog</b>	lists the catalog of available datasets; each dataset is associated with a unique id and may optionally have a title.
<b>/hapi/info</b>	lists information about a dataset with a given id; a primary component of the description is the list of parameters in the dataset.
<b>/hapi/data</b>	streams data for a dataset of a given id and over a given time range; a subset of parameters in a dataset may be requested (default is all parameters).

# HAPI Interface specification

**/hapi**

- Request parameters: None
- Response: The response is in HTML format with a mime type of **text/html**. There is no specification for the content but should provide an overview that is useful for science users.
- Optional

# HAPI Interface specification

## /hapi/about

- Request parameters: None
- Response: The server's response to this endpoint must be in JSON, and the response must indicate a mime type of **application/json**.

Server attributes are described using keyword-value pairs, with the required and optional keywords described in the following table.

- details: <https://github.com/hapi-server/data-specification/blob/master/hapi-3.3.1/HAPI-data-access-spec-3.3.1.md#33-about>

```
{
  "HAPI": "3.3",
  "status": {"code": 1200, "message": "OK"},
  "id": "TestData3.3",
  "title": "HAPI 3.3 Test Data and Metadata",
  "contact": "example1@example.org",
  "dataTest": {
    "name": "Ping Test",
    "query": {
      "dataset": "dataset1",
      "start": "2023-01-01T12:00:00Z",
      "stop": "2023-01-01T14:00:01Z",
      "parameters": "parameter1,parameter2"
    }
  }
}
```

# HAPI Interface specification

## /hapi/capabilities

```
{
  "HAPI": "3.3",
  "status": {"code": 1200, "message": "OK"},
  "outputFormats": ["csv", "binary", "json", "x_hdf"]
}
```

- Request parameters: None
- Response: The server's response to this endpoint must be in JSON, and the response must indicate a mime type of **application/json**.

Server capabilities are described using keyword-value pairs, with outputFormats being the only keyword currently in use.

- details: <https://github.com/hapi-server/data-specification/blob/master/hapi-3.3.1/HAPI-data-access-spec-3.3.1.md#34-capabilities>

# HAPI Interface specification

## /hapi/catalog

```
{
  "HAPI" : "3.3",
  "status": {"code": 1200, "message": "OK"},
  "catalog":
    [
      {"id": "ACE_MAG", title:"ACE Magnetometer data"},
      {"id": "data/IBEX/ENA/AVG5MIN"},
      {"id": "data/CRUISE/PLS"},
      {"id": "any_identifier_here"}
    ]
}
```

- Request parameters: optional parameters
- Response: The server's response to this endpoint must be in JSON, and the response must indicate a mime type of **application/json**.

The catalog is a simple listing of identifiers for the datasets available from the server.

- details: <https://github.com/hapi-server/data-specification/blob/master/hapi-3.3.1/HAPI-data-access-spec-3.3.1.md#35-catalog>

# HAPI Interface specification

## /hapi/catalog?depth=all

- Default is to get the list of datasets.
- The **depth** optional request parameter allows to retrieve the **/info** content for each dataset.
- The **/capabilities** response indicates that the **/catalog** endpoint allows the **depth=all** option

```
{
  "HAPI": "3.3",
  "status": {"code": 1200, "message": "OK"},
  "outputFormats": ["csv", "binary", "json"],
  "catalogDepthOptions": ["dataset", "all"]
}
```

/capabilities

# HAPI Interface specification

## /hapi/info

- Request parameters: **dataset** identifier (required), **parameters** (optional)
- This endpoint provides a data header for a given dataset. The header is expressed in JSON and has a MIME type of **application/json**.
- The specification for the header is that it provides a minimal amount of metadata that allows for the automated reading by a client of the data content streamed via the **/data** endpoint.
- details: <https://github.com/hapi-server/data-specification/blob/master/hapi-3.3.1/HAPI-data-access-spec-3.3.1.md#36-info>

# HAPI Interface specification

## /hapi/info

- Response object: required attributes

Dataset Attribute	Type	Description
HAPI	string	<b>Required</b> The version number of the HAPI specification with which this description complies.
status	object	<b>Required</b> Server response status for this request; see <a href="#">HAPI Status Codes</a> .
format	string	<b>Required</b> (when the header is prefixed to data stream) Format of the data as <code>csv</code> or <code>binary</code> or <code>json</code> .
parameters	array of Parameter	<b>Required</b> Description of the parameters in the data.
startDate	string	<b>Required</b> <a href="#">Restricted ISO 8601</a> date/time of first record of data in the entire dataset.
stopDate	string	<b>Required</b> <a href="#">Restricted ISO 8601</a> date/time of the last record of data in the entire dataset. For actively growing datasets, the end date can be approximate, but it is the server's job to report an accurate end date.

- more details: <https://github.com/hapi-server/data-specification/blob/master/hapi-3.3.1/HAPI-data-access-spec-3.3.1.md#362-info-response-object>

# HAPI Interface specification

## /hapi/info

- Example response:

```
{
  "HAPI": "3.3",
  "status": { "code": 1200, "message": "OK"},
  "startDate": "1998-001Z",
  "stopDate": "2017-100Z",
  "coordinateSystemSchema": { "schemaName": "SPASE", "schemaURI": "TBD"},
  "parameters": [
    { "name": "Time",
      "type": "isotime",
      "units": "UTC",
      "fill": null,
      "length": 24 },
    { "name": "radial_position",
      "type": "double",
      "units": "km",
      "fill": null,
      "description": "radial position of the spacecraft",
      "label": "R Position"},
    { "name": "quality_flag",
      "type": "integer",
      "units": "none",
      "fill": null,
      "description": "0=OK and 1=bad"},
    { "name": "mag_GSE",
      "type": "double",
      "units": "nT",
      "fill": "-1e31",
      "size": [3],
      "coordinateSystemName": "GSE",
      "description": "hourly average Cartesian magnetic field in nT in GSE",
      "label": "B field in GSE"}
  ]
}
```

# HAPI Interface specification

## unitsSchema

- Optional attribute for specifying which units schema is used

Convention Name	Current URL	Description (context help if link is broken)
<code>udunits2</code>	<a href="https://www.unidata.ucar.edu/software/udunits">https://www.unidata.ucar.edu/software/udunits</a>	Unidata from UCAR; a C library for units of physical quantities
<code>astropy3</code>	<a href="https://docs.astropy.org/en/stable/units/">https://docs.astropy.org/en/stable/units/</a>	package inside <code>astropy</code> that handles defining, converting between, and performing arithmetic with physical quantities, such as meters, seconds, Hz, etc
<code>cdf-cluster</code>	<a href="https://caa.esac.esa.int/documents/DS-QMW-TN-0010.pdf">https://caa.esac.esa.int/documents/DS-QMW-TN-0010.pdf</a> which is referenced on this page: <a href="https://www.cosmos.esa.int/web/csa/documentation">https://www.cosmos.esa.int/web/csa/documentation</a>	conventions created and used by ESA's Cluster mission
<code>vounits1.1</code>	<a href="https://www.ivoa.net/documents/VOUnits/">https://www.ivoa.net/documents/VOUnits/</a>	IVOA Recommendation for Units in the VO

# HAPI Interface specification

## /hapi/data

- Request parameters: **dataset**, **start**, **stop** (required), **parameters**, **format** (optional)
- Response: The data endpoint response is in one of three formats:
  - CSV format as defined with a mime type of **text/csv**;
  - binary format where floating points number are in IEEE 754 format and byte order is LSB and a mime type of **application/octet-stream**;
  - JSON format with the structure as described in the specification and a mime type of **application/json**.

*The default data format is CSV.*
- details: <https://github.com/hapi-server/data-specification/blob/master/hapi-3.3.1/HAPI-data-access-spec-3.3.1.md#37-data>

# HAPI Interface specification

## /hapi/data

- Example with header:

```
#{
#  "HAPI": "3.3",
#  "status": { "code": 1200, "message": "OK"},
#  "format": "csv",
#  "startDate": "1998-001Z",
#  "stopDate" : "2017-001Z",
#  "parameters": [
#    { "name": "Time",
#      "type": "isotime",
#      "units": "UTC",
#      "fill": null,
#      "length": 24
#    },
#    { "name": "radial_position",
#      "type": "double",
#      "units": "km",
#      "fill": null,
#      "description": "radial position of the spacecraft"
#    },
#    { "name": "quality flag",
#      "type": "integer",
#      "units": null,
#      "fill": null,
#      "description": "0=OK and 1=bad "
#    },
#    { "name": "mag_GSE",
#      "type": "double",
#      "units": "nT",
#      "fill": "-1e31",
#      "size" : [3],
#      "description": "hourly average Cartesian magnetic field in nT in GSE"
#    }
#  ]
#}
2016-01-01T00:00:00.000Z,6.848351,0,0.05,0.08,-50.98
2016-01-01T01:00:00.000Z,6.890149,0,0.04,0.07,-45.26
...
...
2016-01-01T02:00:00.000Z,8.142253,0,2.74,0.17,-28.62
```

# HAPI Interface specification

## /hapi/data

- Example JSON output

```
{  "HAPI": "3.3",
  "status": { "code": 1200, "message": "OK"},
  "startDate": "2005-01-21T12:05:00.000Z",
  "stopDate" : "2010-10-18T00:00:00Z",
  "parameters": [
    { "name": "Time", "type": "isotime", "units": "UTC", "fill": null, "length": 24 },
    { "name": "quality_flag", "type": "integer", "description": "0=ok; 1=bad", "fill": null },
    { "name": "mag_GSE", "type": "double", "units": "nT", "fill": "-1e31", "size" : [3],
      "description": "hourly average Cartesian magnetic field in nT in GSE" },
    { "name": "region", "type": "string", "length": 20, "fill": "???", "units" : null}
  ],
  "format": "json",
  "data" : [
    ["2010-001T12:01:00Z", 0, [0.44302, 0.398, -8.49], "sheath"],
    ["2010-001T12:02:00Z", 0, [0.44177, 0.393, -9.45], "sheath"],
    ["2010-001T12:03:00Z", 0, [0.44003, 0.397, -9.38], "sheath"],
    ["2010-001T12:04:00Z", 1, [0.43904, 0.399, -9.16], "sheath"]
  ]
}
```

# HAPI Interface specification

## HAPI Status Code & Error Handling

- Status codes: <https://github.com/hapi-server/data-specification/blob/master/hapi-3.3.1/HAPI-data-access-spec-3.3.1.md#42-status-error-codes>
- Error handling: <https://github.com/hapi-server/data-specification/blob/master/hapi-3.3.1/HAPI-data-access-spec-3.3.1.md#43-client-error-handling>
- Example:

```
{  
  "HAPI": "3.3",  
  "status": { "code": 1406, "message": "Bad request – unknown dataset id"}  
}
```

# Accessing HAPI services

- direct access to HAPI service using a web browser  
For example: <https://amda.irap.omp.eu/service/hapi>
- Using <https://hapi-server.org/servers> (demo)
- Using python (notebook demo using EOSC EU Node)  
<https://github.com/hapi-server/client-python-notebooks.git>
- Using Autoplot (demo)
- Using TOPCAT (demo)

# Publishing a HAPI service

- hands-on tests using <https://github.com/hapi-server/server-python>

# Current developments

- Relations between streams, using RDF formalism  
*(Collaboration with OPAL/OntoPortal-Astro team)*
- Registering HAPI services in IVOA Registry