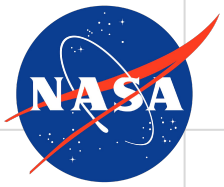
HAPI - A Standard Time Series Data Access API for Heliophysics and Planetary Data

WEIGEL, Bob (George Mason University), FADEN Jeremy (Cottage Systems) VANDEGRIFF Jon (JHU/APL) LINDHOLM Doug (LASP) CANDEY Robert (NASA/GSFC) HARRIS Bernie (NASA/GSFC) ROBERTS Aaron (NASA/GSFC) KING Todd (IGPP/UCLA)



Presented at the European Space Weather Week Conference, November 20-24. Work supported by NASA Grant 80NSSC23K0767

Abstract

Inter- and Intra-operability between Heliophysics and Planetary datasets are needed to address new problems in space weather and planetary science. Although there are a few standard file formats commonly in these communities, no standard has been developed for an Application Programmer's Interface (API) for time series data. HAPI (Heliophysics API) is a specification that captures the lowest common denominator method for accessing time-series data in Heliophysics and Planetary science. HAPI has been recognized as a standard by the Committee on Space Research (COSPAR) and has gained adoption at multiple institutions in the US and Europe, including Goddard Space Flight Center's Coordinated Data Analysis Web (CDAWeb) and Satellite Situation Center Web (SSCWeb); the Planetary Data System Planetary Plasma Interactions Node (PDS/PPI); European Space Agency ViRES/Swarm mission data server, International Real-time Magnetic Observatory Network (INTERMAGNET), and the Laboratory for Atmospheric and Space Physics Interactive Solar Irradiance Data Center (LiSIRD). Several additional plasma data centers, such as the French Plasma Physics Data Centre (CDPP) and the European Space Astronomy Centre (ESAC), are also adopting HAPI. In this presentation, we provide an overview of the HAPI specification and the many software tools developed for accessing data from a HAPI-enabled server.

Motivation for Specification

In the Heliophysics community, the methods for how data providers expose data include:

A. A FTP or HTTP directory of files (usually one day of data per file);
B. A HTTP request that returns a web page with link to a file when processing is complete;
C. A HTTP request that returns a web page with link to an archive of files (zip or tgz) when processing complete; and
D. An API that returns a data stream.

Also note variations in implementation in each category, e.g., for A., providers may have different directory structures, file types, and file naming conventions.

`https://github.com/hapi-server/data-specification`

Motivation for Specification

Some data providers and the methods available include

1. **AMDA** [http://amda.irap.omp.eu/service/hapi] - B, D
2. *CARISMA* [http://www.carisma.ca] - C
3. **CCMC/iSWA** [https://iswa.gsfc.nasa.gov/] - D
4. **CDAWeb** [https://cdaweb.gsfc.nasa.gov/] - A, B, D
5. *CAIO* [https://csa.esac.esa.int/csa/aio/] - C, D
6. **Das2** [http://das2.org/] - D
7. IMAGE [http://space.fmi.fi/image] - C
8. **INTERMAGNET** [http://intermagnet.org] - A, C
9. **LiSIRD** [http://lasp.colorado.edu/lisird/] - A, D
10. *MADRIGAL* [https://openmadrigal.org/] - A, D
11. **OMNIWeb** [https://omniweb.gsfc.nasa.gov/] - A, D
12. *PDS* [https://pds-ppi.igpp.ucla.edu/hapi]
13. **SSCWeb** [https://sscweb.gsfc.nasa.gov/] - D
14. *SuperMAG* [http://supermag.jhuapl.edu/] - D
15. **VirES-for-Swarm** [https://vires.services/hapi] - D

Bold = HAPI available

Italic = HAPI under development

A. A FTP or HTTP directory of files (usually one day of data per file);
B. A HTTP request that returns a web page with link
C. A HTTP request that returns a web page with link to zip or tgz
D. An API that returns a data stream.

A single specification for method (streaming API) can be used to describe and serve data from all of these providers

Primary design considerations

1. Should be simple to write a basic HAPI server and client
2. Metadata should be just enough to create a plot with sensible scientific labels.

   More detailed science-interpretation-level metadata (e.g., SPASE or provider web page) is pointed to in HAPI metadata.

Facilitating Adoption

To facilitate adoption, in parallel to the development of the specification, we have developed

1. Clients for Autoplot, IDL, Java, Javascript, MATLAB, Python, R;
2. A web interface for selecting data from any HAPI server + plotting + script builder (https://hapi-server.org/servers/);
3. A server validator; and
4. A general-use servers (In Java and Javascript; only need to provide metadata + command line program that returns data); less generalized code for Python server exists.

Spec and full list of software at https://github.com/hapi-server

Other Adoption

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.

We are in communication with IVOA (International Virtual Observatory Alliance) about incorporating HAPI into their data access standards.

API - Endpoints


```
http://server/hapi/about
http://server/hapi/capabilities
http://server/hapi/catalog
http://server/hapi/info
http://server/hapi/data
```

API - http://server/hapi/info

Example: `http://amda.irap.omp.eu/service/hapi/info?id=uks-orb-all`


```
{
    "startDate": "1984-08-18T23:55:00Z",
    "stopDate": "1985-01-15T08:00:00Z",
    "cadence": "PT300S",
    "description": "ephemeris<br/> Sampling: 300S",
    "resourceID": "spase://CNES/NumericalData/CDPP-AMDA/UKS/ORBIT/uks-orb-all",
    "parameters": [
        {
            "name": "Time",
            "type": "isotime",
            "length": 24,
            "units": "UTC",
            "fill": null
        },
        {
            "name": "uks_orb_xyz",
            "type": "double",
            "size": [
                3
            ],
            "units": "Re",
            "fill": "-1e31",
            "description": "Spacecraft position in GSE Cartesian coordinates"
        },
        {
            "name": "uks_orb_gsm",
            "type": "double",
            "size": [
                3
            ],
            "units": "Re",
            "fill": "-1e31",
            "description": "Spacecraft position in GSM Cartesian coordinates"
        },
```

API - http://server/hapi/data

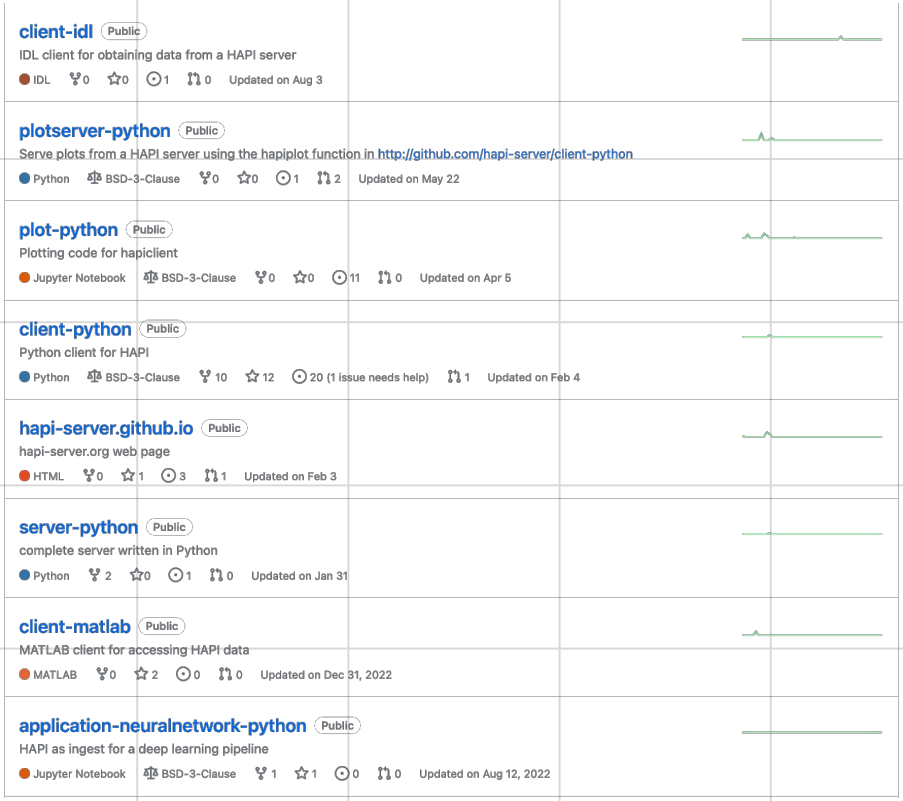

```
https://server/hapi/data
    ?dataset=DATASET
    &parameters=P1,P2,...          If not given, all parameters served
    &start=ISO8601                 Constrained ISO8601 timestamp
    &stop=ISO8601                  Constrained ISO8601 timestamp
    [&format={csv,json,binary}]    Server only needs to support csv
    [&include=header]              Include /info response JSON as header
```

Default output is a CSV table for parameters P1, P2, ... (parameters can be multidimensional - client uses metadata to reshape associated columns).

```
2001-01-01T00:01:33.00Z,1.1,2.1
2001-01-01T00:01:34.00Z,1.2,2.2
...
```

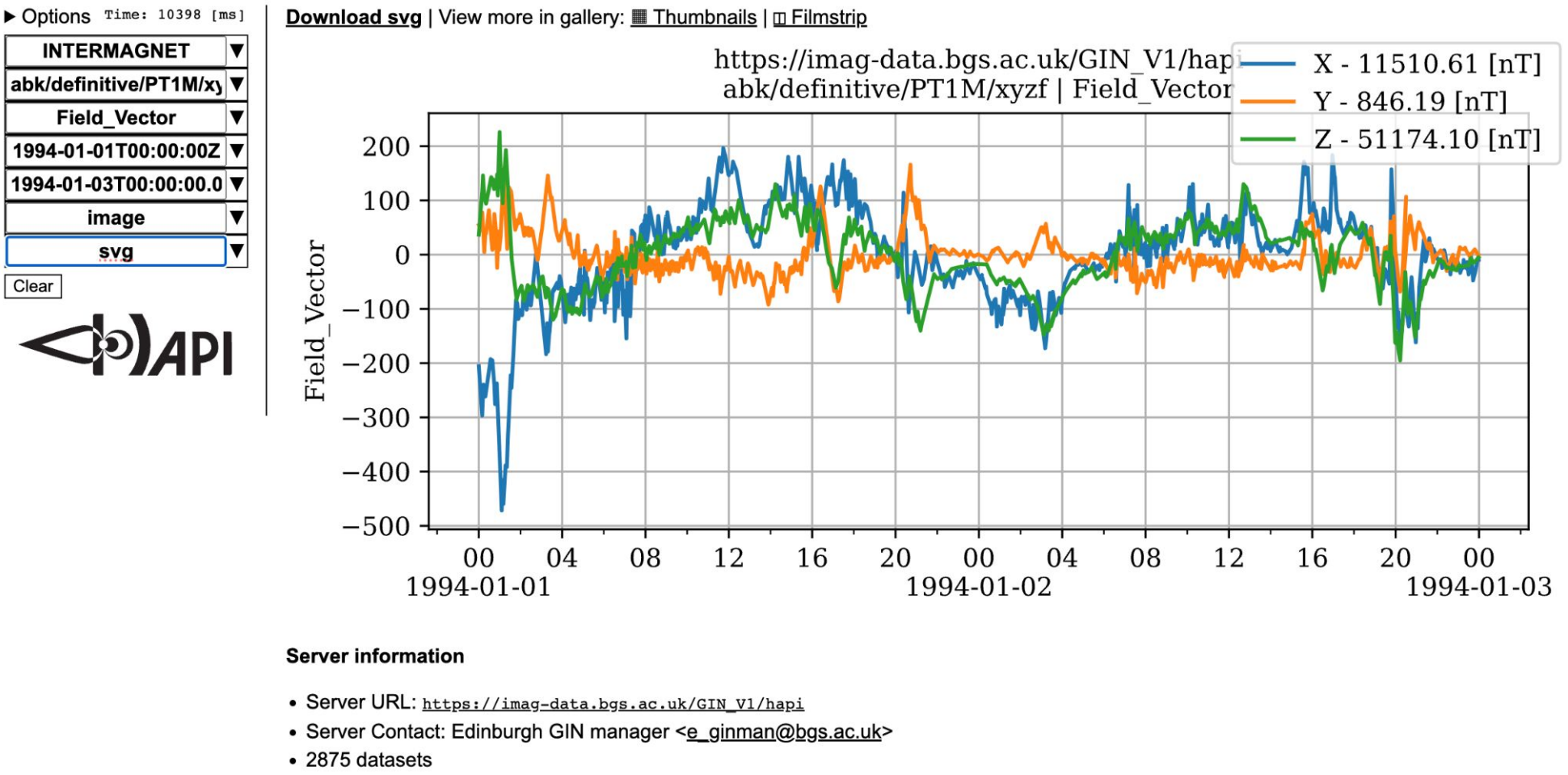
Tools - GitHub

`https://github.com/hapi-server` - GitHub project with ~20 repositories for code related to HAPI specification



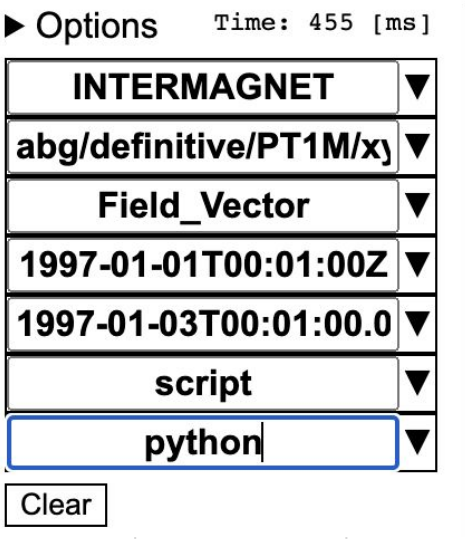
Tools - Preview and Download

https://hapi-server.org/servers/#server=INTERMAGNET



Tools - Script Builder

https://hapi-server.org/servers/#server=INTERMAGNET



Download script

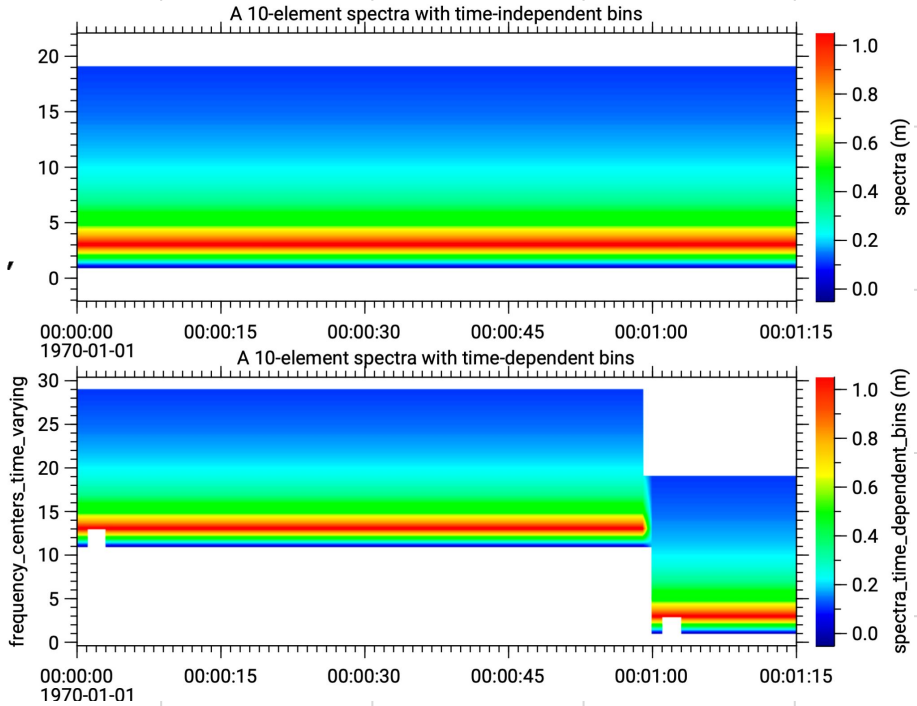
```
from hapiclient import hapi

server     = 'https://imag-data.bgs.ac.uk/GIN_V1/hapi'
dataset    = 'abg/definitive/PT1M/xyzf'
# Notes:
# 1. Use parameters='' to request all parameters from abg/definitive/PT1M/xyzf.
# 2. Multiple parameters can be requested using a comma-separated
#    list, e.g., parameters='Field_Vector,Field_Magnitude'
parameters = 'Field_Vector'
start      = '1997-01-01T00:01:00Z' # min 1997-01-01T00:01:00Z
stop       = '1997-01-03T00:01:00.000Z'  # max 2022-12-31T23:59:00Z

data, meta = hapi(server, dataset, parameters, start, stop)
```

`data` is a NumPy ndarray with named fields and `meta` is a dict of metadata.

Recent Additions to Specification


```
{
    "name": "spectra_time_dependent_bins",
    "type": "double",
    "units": "m",
    "fill": "-1e31",
    "size": [
        10
    ],
    "description": "A 10-element spectra with time-dependent bins",
    "bins": [
        {
            "name": "frequency",
            "units": "Hz",
            "centers": "frequency_centers_time_varying",
            "ranges": "frequency_ranges_time_varying"
        }
    ]
},
{
    "name": "frequency_centers_time_varying",
    "type": "double",
    "units": "Hz",
    "fill": "-1e31",
    "size": [
        10
    ],
    "description": "Bin centers for spectra frequencies"
},
{
    "name": "frequency_ranges_time_varying",
    "type": "double",
    "units": "Hz",
    "fill": "-1e31",
    "size": [
        10,
        2
    ],
    "description": "Bin ranges for spectra frequencies"
}
```

Instead of being fixed values, point is given to a parameter.



Top: Bin centers and ranges are time independent.
Bottom: Bin centers and ranges vary with time.

Other Additions/Developments

- Schema for units strings can be given. This will allow validator to verify that given unit strings are valid. Current options are `astropy3` and `cdfunits`.
- Specification for cache (client and server). This work is in development. Caching is commonly requested, used, and developed ad-hoc. We seek to have a specification that will allow clients and servers to share cache.
- Parameters of `type=string` can be a URI. This will allow, for example,
  - a sequence of URLs to images
  - a listing of files used to fulfill a data request.