

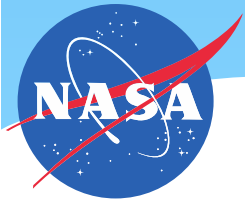
Web Services for Earth Science Data DAS-TAG presentation

Edward Armstrong, Thomas Huang, Charles Thompson, Nga
Quach, Richard Kim, Zhangfan Xing

14th GHR SST Science Team Meeting

Woods Hole, MA, USA

18 June 2013



National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

Web Services and Tools

- * New and emerging PO.DAAC web services
 - * Concerned with search, discovery, data subsetting, data and metadata extraction, imaging services
 - * Integrated.....output of one service an input to another
 - * Available in 6 months (some components are available now)
 - * Work through a use case
- * Other Tools and Services
 - * HiTIDE improvements
 - * PO.DAAC GUI subsetter
 - * Webification
 - * Data subsetting and virtualization

PO.DAAC Portal



- * PO.DAAC Web Portal for Human Consumption
- * Find Datasets
- * Learn Datasets
- * Download Datasets
- * Through Graphical User Interface (GUI)

Use Case

- * I know what I am looking for: e.g., WindSat level2 SST granules.
- * I want to get the latest granules every day, subset them and create browse images.
- * I want to write/run a script or program and schedule to run it everyday through PO.DAAC Web Services.

Available Web Services

- * Metadata for Dataset/Granule Web Service
- * Search Dataset/Granule Web Service
- * Image Granule Web Service
- * Extract Granule Web Service

PO.DAAC Web Services

- * PO.DAAC Web Services can be invoked from any programming languages or scripts over HTTP.
- * Java, Python, Perl, IDL, Matlab, etc.
- * Each Web Service has API Documentation.
- * API Doc contains
 - * Description
 - * Endpoint
 - * Input parameters
 - * Output
 - * Example
 - * Sample Run

Demo

- * Find ASCAT level 2 dataset
- * Generate ISO Metadata for the dataset
- * Find Granule on startTime 2009-02-02 to 2009-02-03
- * Run Image Service
- * Run Extract Service

1. Find ASCAT Level 2 Dataset

- * From the browser, go to <http://podaac-test.jpl.nasa.gov/ws/search/dataset>
- * Generate the Web Service call
 - * in the sample run section on the API Doc, type “ascat” as keyword and “2” as the process level and click “run”
 - * The follow URL will be generated
<http://podaac-test.jpl.nasa.gov/ws/search/dataset/?keyword=ascat&processLevel=2&pretty=true&format=html>
- * From the html result page, copy the Entry ID of the second result which is ASCAT L2 25km.

PODAAC-ASOP2-25X01 ← this Dataset persistent ID is needed to call other services.

2. Generate ISO Metadata

- * From the browser, go to

<http://podaac-test.jpl.nasa.gov/ws/metadata/dataset>

- * From the API Doc, in the Sample Run section, paste the selected dataset id from the previous slide “PODAAC-ASOP2-25X01” and click run

- * Following URL will be generated

<http://podaac-test.jpl.nasa.gov/ws/metadata/dataset/?format=gcmd&datasetId=PODAAC-ASOP2-25X01>

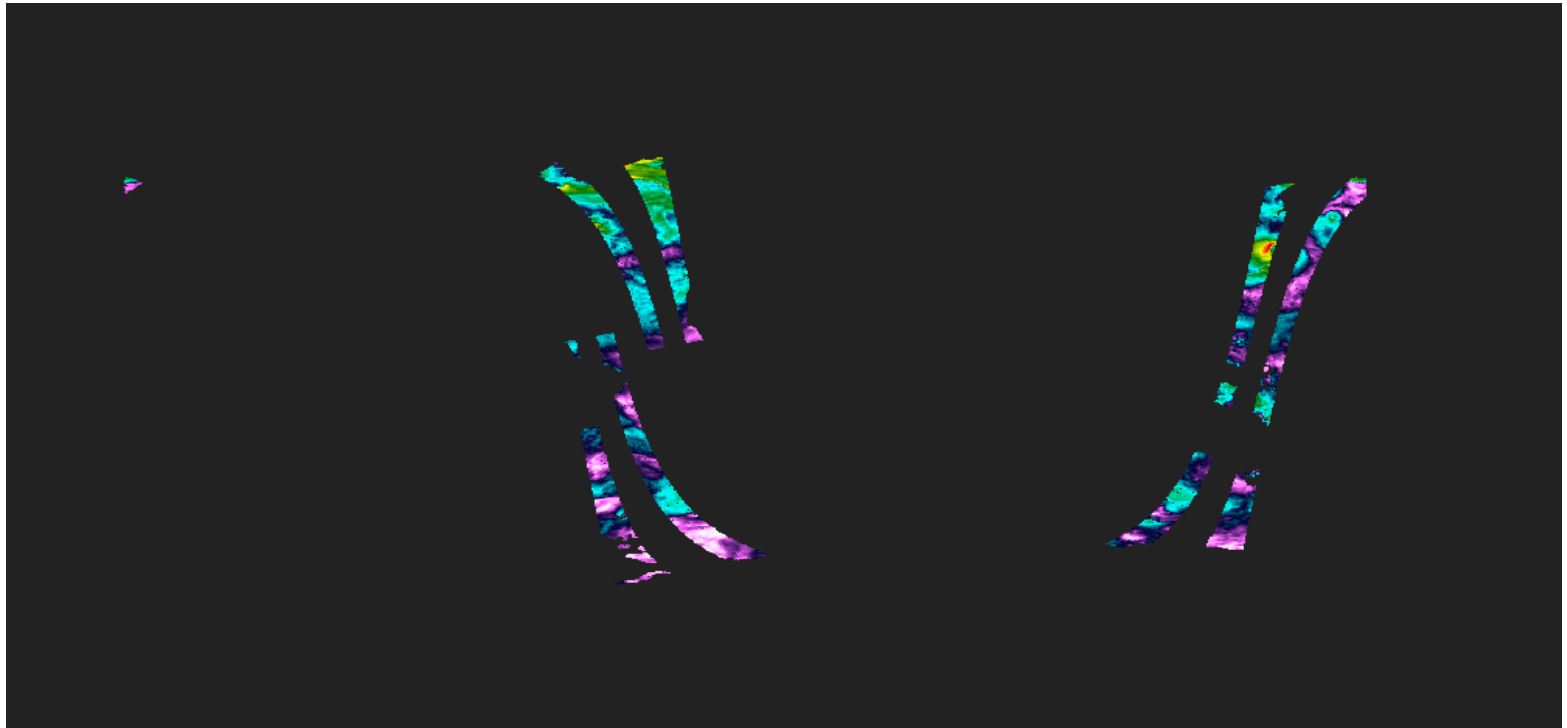
3 Find Granule

- * Go to : <http://podaac-test.jpl.nasa.gov/ws/search/granule>
- * From the sample run, paste “PODAAC-ASOP2-25X01” to datasetId and 2009-02-02 as startTime and 2009-02-03 as endTime and click run
- * From the result, copy the granule name (Entry ID)
ascat_20090203_071800_metopa_11900_eps_o_250_1015_ovw.l2.nc

4. Run image service

- * Go to <http://podaac-test.jpl.nasa.gov/ws/image/granule>
- * `://podaac-tools.jpl.nasa.gov/ws/image/granule/?datasetId=PODAAC-ASOP2-25X01&granuleName=ascat_20130122_224800_metop_a_32495_eps_o_250_2101_ovw.l2.nc&request=GetMap&version=1.3.0&format=image%2Fpng&bbox=-180%2C-90%2C180%2C90&height=400&width=800&Transparent=true&layers=&styles=&srs=`

Image result

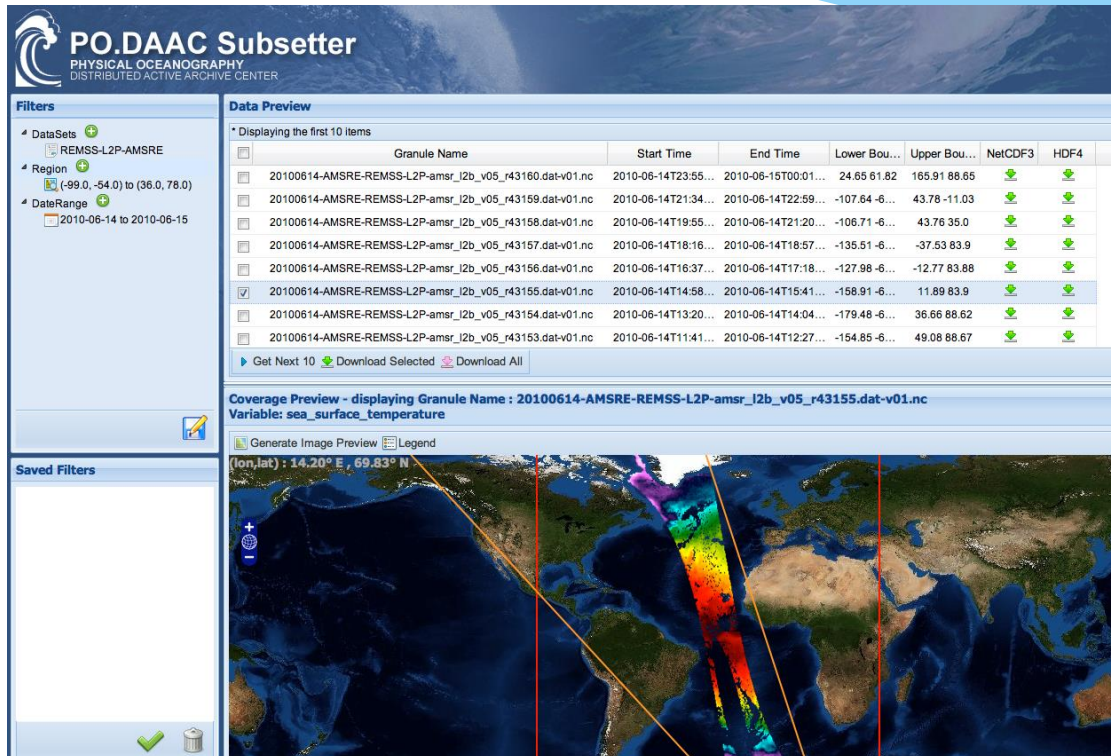


5. Run Extract service

- * Go to <http://podaac-test.jpl.nasa.gov/ws/extract/granule>
- * http://podaac-tools.jpl.nasa.gov/ws/extract/granule/?datasetId=PODAAC-ASOP2-25X01&granuleName=ascats_20130122_224800_metopa_32495_eps_o_250_2101_ovw.l2.nc&bbox=-180%2C-90%2C180%2C90&format=netcdf

.

HITIDE



The **High-level Tool for Interactive Data Extraction (HITIDE)** is a web-based interface facilitating the search, imaging, and extraction of select Level 2 "swath" data products from PO.DAAC's archive.

HITIDE Status

- * HITIDE
 - * Hardened tools, services
 - * Better swath coverage display
 - * Paged results
 - * Multi-dataset search
 - * Multi-granule selection for swath polygon display, extraction, and imaging
 - * Faster spatial & temporal search

Ongoing & Future Work

- * **Features**

- * **Planned for 2013**

- * Adding More Datasets (SST, ASCAT 12.5km & Coastal, Aquarius), ongoing.
 - * Colocation searching
 - * Spatially and temporally related data
 - * Non-truncated swath scans in extraction

- * **Other possible features**

- * Search on events
 - * Re-enable statistical search

Webification

The goal: make data easy to use in the “web” way

The idea: Inner components of an arbitrary data store, such as attributes, labels, image bands, and data arrays, are directly **addressable** and **accessible** by **well-defined** and **meaningful** URLs.

Example:

[http://data/earth/climatology.nc/SST\[0:2,45:55,85:95\]?output=nc](http://data/earth/climatology.nc/SST[0:2,45:55,85:95]?output=nc)

Another Example:

[http://data/saturn/N1506378043_1.IMG/o/image\[\(10,20\)300x200\]?output=gif](http://data/saturn/N1506378043_1.IMG/o/image[(10,20)300x200]?output=gif)

Webification is an enabling technology. Abbreviated as w10n.

Webification

Open specification:

<http://w10n.org>

Summary:

- Resource is viewed as a tree of nodes and leaves.
- They have semantic URLs, accessible through HTTP.
- Meta info exchange format is JSON, by default.
- Full ReSTful style request/response. Read/Write.

Disciplines:

Earth science (NetCDF, HDF 4/5, GRIB)

Planetary Science (VICAR/PDS)

Astronomy (FITS) and more

Extended URL Syntax - Identifier

Identifier	What is identified
/	Meta information about the store.
/node/	Meta info of a particular node in the store.
/node/leaf/	Meta info of a leaf under a node.
/node/leaf[]	Entire byte array for that leaf.
/node/leaf[indexer]	A subset of that leaf indicated by an indexer.
more can be defined.	

Extended URL Syntax - Indexer

Indexer	Type	Example
[start:end:step,start:end:step,...]	range	[0:100,100:200:2]
[n0,n1,n2,...]	list	[2,4,6,8,10]
[(x,y)width*height]	tile	[(10,20)300*200]
More can be introduced.		

URL Syntax - Parameters defined for `query_string`

Parameter	Value	Type	Meaning
output	json/html/raw/nc/...	String	Format of output
ignoreCache	false/true	Boolean	If cached output is used
flatten	false/true	Boolean	If array is flattened
traverse	false/true	Boolean	If traverse
More can be introduced.			

`query_string` is a string of parameter-value pairs concatenated by '&'.

W10N HTTP Request

Whether meta or data info, the HTTP request is unambiguous by the URL used.

GET Request – Read API

`http://host:port/path/identifier?query_string`

Message body is absent.

PUT Request – Write API

`http://host:port/path/identifier?query_string`

Message body contains data.

Use Case – Quality filtering the SST observations

- * Subset a L2P granule
 - * [http://host:port/path/2013/123/20130503-MODIS_T-JPL-L2P-T2013123065500.L2_LAC_GHRSSST_N-v01.nc.bz2/sea_surface_temperature\[-130<lon<-120,35<lat<45\]?output=format](http://host:port/path/2013/123/20130503-MODIS_T-JPL-L2P-T2013123065500.L2_LAC_GHRSSST_N-v01.nc.bz2/sea_surface_temperature[-130<lon<-120,35<lat<45]?output=format)
- * Apply quality filter
 - * [http://host:port/path/2013/123/20130503-MODIS_T-JPL-L2P-T2013123065500.L2_LAC_GHRSSST_N-v01.nc.bz2/sea_surface_temperature\[quality_flag>=4\]?output=format](http://host:port/path/2013/123/20130503-MODIS_T-JPL-L2P-T2013123065500.L2_LAC_GHRSSST_N-v01.nc.bz2/sea_surface_temperature[quality_flag>=4]?output=format)
- * Quality filter, wind screen, subset all in one step !
 - * [http://host:port/path/2013/123/20130503-MODIS_T-JPL-L2P-T2013123065500.L2_LAC_GHRSSST_N-v01.nc.bz2/sea_surface_temperature\[quality_flag>=4,wind_speed>6,-130<lon<-120,35<lat<45\]?output=format](http://host:port/path/2013/123/20130503-MODIS_T-JPL-L2P-T2013123065500.L2_LAC_GHRSSST_N-v01.nc.bz2/sea_surface_temperature[quality_flag>=4,wind_speed>6,-130<lon<-120,35<lat<45]?output=format)