

Data Access Hangout 2015-01-26

Date

26 Jan 2015

Attendees

- IPAC: [Tatiana Goldina](#), [Loi Ly](#), [Trey Roby](#), [Unknown User \(xiuqin\)](#), [Gregory Dubois-Felsmann](#), [Unknown User \(zhang\)](#), [Unknown User \(jar\)](#)
- SLAC: [Kian-Tat Lim](#), [Fritz Mueller](#), [Fabrice Jammes](#), [John Gates](#), [Andy Salnikov](#), [Jacek Becla](#)

Related page

- Web Data Access Service [API](#)

Discussion items

API versions - should we have "current" version?

- difficult to maintain
- based on best practices from others - they don't do that
- so, "no". We can easily add it later if we decide to
- version will change only when there is a change breaking the API
- extending API can be handled without breaking the API
 - unless we have to roll back a new change, and new clients have already been deployed
- API won't change much, so no need to deal with major, minor etc, single number should suffice

paging - should we break results into pages?

- we don't really need "paging", it is driven by the fact we want to minimize damage if someone accidentally requests huge amount of data (e.g. all images ever produced by Isst), that will be easy to do through the API
- don't call it "pageSize", call it "maxResult" / "firstResult"
- set by default to something relatively large
- tricky for dynamic data (e.g., for continuously changing L1)
- it will be OK if it works the same way as MySQL paging, e.g., we don't have to try super hard to keep results stable as we page through them.
 - sorting results might help keep the results stable (e.g sorting that ensures new data is added at the end)
- definitely document all this
- add to API indicator that will allow users to determine if result is stable or not
- potentially http chunked responses might help to lessen server load. But this is harder for json/xml

image type - should image types be part of URI?

- we are talking about coadds, raw, calpex, etc here, not formats (eg not fits, jpeg)
- don't call it image "type". Proposed name: call it image "kind"
- note that image kind is part of primary key, together with the id.
 - e.g., two different kinds of images might end up having the same id, but they will have nothing in common

multiple images

- Isst pipelines will always produce images with image plane, mask and variance, all 3 together in one physical fits file
- some will argue "never give data without mask" etc, but we should optimize performance, network traffic etc, and deliver only what user really wants
- decision: by default, deliver entire image with all 3 planes, but allow selecting individual planes
- use commonly used rest notation ";", as Gregory documented in comments of



DM-1694 - Jira project doesn't exist or you don't have permission to view it.

(so our M12 from API "GET /meta/v0/image

/coadd;mask/12345" would become "GET /meta/v0/image/coadd/12345;plane=data;mask)

image ids: "/image/123" vs "/image?id=123"

- the former

cutout - is it separate resource or not?

- depends. Two cases here:
 - a. retrieving an existing image or part of such image (and the image already has an id etc), if we are using original image coordinate system - there is no need to create a new resource

b. cutout that involves complex operations (stitching etc), or rotating, or transforming coordinates etc. Here we need to produce a new resource

- the first case will be rare, for internal debugging etc. Selection criteria will be very limited, a simple "ra, dec + height/length in arcsec" should cover most cases
- note that raw images will be in random rotations, so in most cases we will want at minimum rotate them, which already puts us in "case 2"
- note, I6 from API page needs to be rewritten
- to limit (full vs cutout), one can use ":" notation:
 - GET /image/v0/coadd/12345;full
 - GET /image/v0/coadd/12345;cutout

metadata for columns returned by "GET /db/v0/query"

- things like units, types, null/not null etc
- in most cases, we want to avoid extra call, so we should deliver metadata with query results
- so, send with data. How fancy we get is format dependent, eg
 - in csv, typically people specify column names in first line as comment
 - in json, key-value pairs, specify everything: names and types etc
- btw, we will need to implement a dedicated call to get just metadata too
- and we will need something like "GET /db/v0/query/explain" to get estimated query time, # chunks involved etc.

result format for db queries?

- for now use IPAC table format
 - ipac team will provide the spec

Topics not covered are being moved to next week hangout, see [Data Access Hangout 2015-02-02](#)