

PDAC sample queries and test cases

- "Aliveness" checks:
 - Is DAX running?
 - Is dbserv running?
 - Is the Qserv "czar" server running?
 - Are the Qserv workers running?
- Basic parameters of queries needed for PDAC:
- First data set: SDSS stripe 82
 - DB Name:
 - "sdss_stripe82_01"
 - Recommended RA and Dec: 9.469944999999996 -1.1521076379165134
- Second data set: WISE
 - Catalogs:
- (1) Catalog (both objects and forced sources) and Image Searches (using dbServ):
 - Search methods:
- (1.1) Use the qserv functions for catalog search:
 - Catalog Cone Search (Find the sources in the given cone):
 - Catalog Box Search (Find the sources in the given box):
 - Catalog Ellipse Search (Find the sources in the give ellipse):
 - Catalog Polygon Search (Find the sources in the give polygon):
- (1.2) Use the scisql functions for image search:
 - Image Search (Find the images which contain the given target):
 - Image Search (Find Images which cover the entire search region as a box):
 - Image Search (Find Images which are entirely enclosed by the search region as a box):
- (1.3) Do not use the scisql functions for catalog search: they generate full table scan and are very slow. Please use qserv funtions in (1.1) above.
 - Catalog Box Search (Find the sources in the given box):
 - Catalog Ellipse Search (Find the sources in the give ellipse):
 - Catalog Polygon Search (Find the sources in the given polygon):
- (1.4) ForcedSource searches (RunDeepForcedSource table)
- (2) ImgServ (pixel data retrieval), see ImageServ API - v1[Deprecated]:
 - "Aliveness" checks:
 - Is imgserv running?
 - Calibrated single-epoch image (and cutout) retrieval
 - Coadded image (and cutout) retrieval
- (3) Basic light curve / time series viewer query on RunDeepForcedSource
- For returning the magnitude:
- For returning the flux:
- For WISE Light Curve Query:

"Aliveness" checks:

Is DAX running?

- curl <http://lsst-qserv-dax01.ncsa.illinois.edu:5000/>

Is dbserv running?

- curl <http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/>

Is the Qserv "czar" server running?

- curl -d 'query=SHOW+DATABASES' <http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>
- curl -d 'query=SHOW+TABLES' <http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync> (show default DB tables)
- curl -d 'query=SHOW TABLES IN wise_2band_00;' <http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Are the Qserv workers running?

- curl -d 'query=SELECT+COUNT(*)+FROM+RunDeepSource' <http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Basic parameters of queries needed for PDAC:

Host info: <http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Test dbserv (Kotlin version): <http://lsst-qserv-dax01:8080/sync> (Logical database name is a required part of database name. For example, sdss_stripe82_01 should appear as W13_sdss_v2.sdss_stripe82_01 in the queries. "W13_sdss_v2" is logical database name.)

First data set: SDSS stripe 82

DB Name:

"sdss_stripe82_01"

DB Tables:

Catalogs:

Object (Measurements on coadds): table name = "RunDeepSource"

Forced photometry based on i-band coadds: table name = "RunDeepForcedSource"

Images:

Single-band coadded image metadata: table name = "Science_Ccd_Exposure"

Calibrated single-epoch image metadata: table name = "DeepCoadd"

Recommended RA and Dec: 9.469944999999996 -1.1521076379165134

Second data set: WISE

The summary of the data loaded is capture in this ticket:

[DM-12910 - Getting issue details...](#)

STATUS

Catalogs:

1st DB:

wise_00

Tables:

allwise_p3as_psd (source catalog)

allwise_p3as_mep (forced source catalog)

2nd DB:

wise_ext_00

Tables:

allwise_p3as_psr (rejected source catalog)

3rd DB:

neowiser_yr1_00

Tables:

[neowiser_yr1_p1bs_psd \(NEOWISE-R Year 1 Single Exposure \(L1b\) Source Table\)](#)

Images:

DB:

wise_00

Tables:

allwise_p3am_cdd (Atlas coadd images)

allwise_p3as_cdd (Atlas meta table)

allsky_4band_p1bm_frm (single exposure images)

allsky_3band_p1bm_frm (single expsoure images)

allsky_2band_p1bm_frm (single exposure images)

See detail and updated information about WISE data in PDAC in [PDAC v2 data list](#).

Recommended RA and Dec to search for WISE: any

(1) Catalog (both objects and forced sources) and Image Searches (using dbServ):

Search methods:

cone, box, ellipse, polygon

(1.1) Use the qserv functions for catalog search:

Catalog Cone Search (Find the sources in the given cone):

Inputs: RA, DEC, radius (all in degree)

URL:

```
curl -o <output file> \
-d 'query=SELECT+<column names>+FROM+<database name>.<table name>+WHERE+qserv_areaspec_circle(<RA>,<DEC>,<radius>);'\
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Examples:

```
curl -o runDeepSourceConeSdss.json \
-d 'query=SELECT+*+FROM+sdss_stripe82_01.RunDeepSource+WHERE+qserv_areaspec_circle(9.469,-1.152,0.01);'\
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

```
curl -o forcedSourceConeWise.json \
-d 'query=SELECT+*+FROM+wise_00.allwise_p3as_mep+WHERE+qserv_areaspec_circle(9.469,-1.152,0.01);'\
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Catalog Box Search (Find the sources in the given box):

Inputs: RA, DEC, minRA, minDec, maxRA, maxDec (all in degree)

URL:

```
curl -o <output file> \
-d 'query=SELECT+<column names>+FROM+<database name>.<table name>+WHERE+qserv_areaspec_box(minRA,minDEC,maxRA,maxDec);'\
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Examples:

```
curl -o catalogBox.json \
-d 'query=SELECT+*+FROM+sdss_stripe82_01.RunDeepSource+WHERE+qserv_areaspec_box(9.5,-1.23,9.6,-1.22);'\
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Catalog Ellipse Search (Find the sources in the give ellipse):

Inputs: RA (degree), DEC (degree), semi-majorAxis (arcsec), semi-minorAxis(arcsec), positionAngle (degree)

Note: The semi-majorAxis must be longer than the semi-minorAxis

URL:

```
curl -o <output file> \
```

```
-d 'query=SELECT+<column names>+FROM+<database name>.<table name>+WHERE+qserv_areaspec_ellipse(<RA>,<DEC>,<semi-majorAxis,semi-minorAxis,positionAngle>);'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Examples:

```
curl -o catalogEllipse.json \
```

```
-d 'query=SELECT+*+FROM+sdss_stripe82_01.RunDeepSource+WHERE+qserv_areaspec_ellipse(9.469,-1.152,58,36,0);'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Catalog Polygon Search (Find the sources in the give polygon):

Inputs: Polygon vertexes coordinates: v1_RA, v1_DEC; v2_RA, v2_DEC; v3_RA, v3_DEC; ... (all in degree)

Note: The order of the vertex coordinates matters, either clockwise or anticlockwise.

URL:

```
curl -o <output file> \
```

```
-d 'query=SELECT+<column names>+FROM+<database name>.<table name>+WHERE+qserv_areaspec_poly(v1_RA, v1_DEC, v2_RA, v2_DEC, ...);'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Examples:

```
curl -o catalogPolygon.json \
```

```
-d 'query=SELECT+*+FROM+sdss_stripe82_01.RunDeepForcedSource+WHERE+qserv_areaspec_poly(9.4, -1.2, 9.6, -1.2, 9.4, -1.1);'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

(1.2)Use the scisql functions for image search:

Note: The image search commands below can be used for either Science_Ccd_Exposure or DeepCoadd tables.

Image Search (Find the images which contain the given target):

Use the function scisql_s2PtInCPoly to search the images which four corner polygon contains the given target RA and DEC.

Inputs: RA, DEC, corner1Ra, corner1Decl, corner2Ra, corner2Decl, corner3Ra, corner3Decl, corner4Ra, corner4Decl

Example:

The target: (RA, DEC)

The command:

```
curl -o imagesContainTarget.json -d 'query=SELECT+*+FROM+<database name>.<table name>+WHERE+\'
```

```
scisql_s2PtInCPoly(RA, DEC, <corner1Ra>, <corner1Decl>, <corner2Ra>, <corner2Decl>, <corner3Ra>, <corner3Decl>, <corner4Ra>, <corner4Decl>)=1;'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Example for SDSS images:

```
curl -o imagesContainTarget.json -d 'query=SELECT+*+FROM+sdss_stripe82_01.DeepCoadd+WHERE+\'
```

```
scisql_s2PtInCPoly(9.462, -1.152, corner1Ra, corner1Decl, corner2Ra, corner2Decl, corner3Ra, corner3Decl, corner4Ra, corner4Decl)=1;'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Example for WISE images:

```
curl -o imagesContainTarget.json -d 'query=SELECT+*+FROM+wise_00.allwise_p3am_cdd+WHERE+\'
```

```
scisql_s2PtInCPoly(9.462, -1.152, ra1, dec1, ra2, dec2, ra3, dec3, ra4, dec4)=1;'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Image Search (Find Images which cover the entire search region as a box):

Call the function `scisql_s2PtInCPoly` four times to search the images which contain all the four vortexes of the given search box.

Inputs of each call: `RA_one_corner_of_box`, `DEC_of_one_corner_of_box`, `corner1Ra`, `corner1Decl`, `corner2Ra`, `corner2Decl`, `corner3Ra`, `corner3Decl`, `corner4Ra`, `corner4Decl`

The search box four corners:

(9.5, -1.23), (9.5, -1.22), (9.6, -1.23), and (9.6, -1.22)

Example for SDSS images:

```
curl -o imagesContainTheRegion.json -d 'query=SELECT+*+FROM+sdss_stripe82_01.Science_Ccd_Exposure+WHERE+\
(scisql_s2PtInCPoly(9.5, -1.23, corner1Ra, corner1Decl, corner2Ra, corner2Decl, corner3Ra, corner3Decl, corner4Ra, corner4Decl)
=1)+AND+\
(scisql_s2PtInCPoly(9.5, -1.22, corner1Ra, corner1Decl, corner2Ra, corner2Decl, corner3Ra, corner3Decl, corner4Ra, corner4Decl)
=1)+AND+\
(scisql_s2PtInCPoly(9.6, -1.23, corner1Ra, corner1Decl, corner2Ra, corner2Decl, corner3Ra, corner3Decl, corner4Ra, corner4Decl)
=1)+AND+\
(scisql_s2PtInCPoly(9.6, -1.22, corner1Ra, corner1Decl, corner2Ra, corner2Decl, corner3Ra, corner3Decl, corner4Ra, corner4Decl)
=1);'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Example for WISE images:

```
curl -o allsky_2band_p1bm_frmContainTheRegion.json -d 'query=SELECT+*+FROM+wise_00.allsky_2band_p1bm_frm+WHERE+\
(scisql_s2PtInCPoly(9.5, -1.23, ra1, dec1, ra2, dec2, ra3, dec3, ra4, dec4)=1)+AND+\
(scisql_s2PtInCPoly(9.5, -1.22, ra1, dec1, ra2, dec2, ra3, dec3, ra4, dec4)=1)+AND+\
(scisql_s2PtInCPoly(9.6, -1.23, ra1, dec1, ra2, dec2, ra3, dec3, ra4, dec4)=1)+AND+\
(scisql_s2PtInCPoly(9.6, -1.22, ra1, dec1, ra2, dec2, ra3, dec3, ra4, dec4)=1);'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Image Search (Find Images which are entirely enclosed by the search region as a box):

Call the function `scisql_s2PtInBox` four times to search the images which four corners are entirely contained by the given search box.

Inputs of each call: `corner1Ra`, `corner1Decl` (or `corner2Ra`, `corner2Decl`, or `corner3Ra`, `corner3Decl`, or `corner4Ra`, `corner4Decl`), `box_minRA`, `box_minDEC`, `box_maxRA`, `box_maxDEC`

The search box:

`minRA=0, minDEC=-1, maxRA=5, maxDEC=1`

Example for SDSS images:

```
curl -o imagesContainedByTheRegionBox.json -d 'query=SELECT+*+FROM+sdss_stripe82_01.Science_Ccd_Exposure+WHERE+\
(scisql_s2PtInBox(corner1Ra, corner1Decl, 0, -1, 5, 1)=1)+AND+\
(scisql_s2PtInBox(corner2Ra, corner2Decl, 0, -1, 5, 1)=1)+AND+\
(scisql_s2PtInBox(corner3Ra, corner3Decl, 0, -1, 5, 1)=1)+AND+\
(scisql_s2PtInBox(corner4Ra, corner4Decl, 0, -1, 5, 1)=1);'\
```

<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Example for WISE images:

```
curl -o allwise_p3am_cddContainedByTheRegionBox.json -d 'query=SELECT+*+FROM+wise_00.allwise_p3am_cdd+WHERE+\
(scisql_s2PtInBox(ra1, dec1, 0, -1, 5, 1)=1)+AND+\
(scisql_s2PtInBox(ra2, dec2, 0, -1, 5, 1)=1)+AND+\
(scisql_s2PtInBox(ra3, dec3, 0, -1, 5, 1)=1)+AND+\
(scisql_s2PtInBox(ra4, dec4, 0, -1, 5, 1)=1);' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

(1.3) Do not use the scisql functions for catalog search: they generate full table scan and are very slow. Please use qserv funtions in (1.1) above.

Inputs: RA, DEC, radius (all in degree)

URL:

```
curl -o <output file> \
-d 'query=SELECT+<column names>+FROM+<database name>.<table name>+WHERE+scisql_s2PtInCircle(<ra column name>,<dec column
name>,<RA>,<DEC>,<radius>)=1;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Example:

```
curl -o catalogCone.json \
-d 'query=SELECT+*+FROM+sdss_stripe82_01.RunDeepSource+WHERE+scisql_s2PtInCircle(coord_ra,coord_dec,9.469,-1.152,0.01)=1;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Catalog Box Search (Find the sources in the given box):

Inputs: RA, DEC, minRA, minDec, maxRA, maxDec (all in degree)

URL:

```
curl -o <output file> \
-d 'query=SELECT+<column names>+FROM+<database name>.<table name>+WHERE+scisql_s2PtInBox(<ra column name>,<dec column
name>,minRA,minDEC,maxRA,maxDec)=1;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Examples:

```
curl -o catalogBox.json \
-d 'query=SELECT+*+FROM+sdss_stripe82_01.RunDeepSource+WHERE+scisql_s2PtInBox(coord_ra,coord_dec,9.5,-1.23,9.6,-1.22)=1;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Catalog Ellipse Search (Find the sources in the give ellipse):

Inputs: RA (degree), DEC (degree), semi-majorAxis (arcsec), semi-minorAxis(arcsec), positionAngle (degree)

Note: The semi-majorAxis must be longer than the semi-minorAxis

URL:

```
curl -o <output file> \
-d 'query=SELECT+<column names>+FROM+<database name>.<table name>+WHERE+scisql_s2PtInEllipse(<ra column name>,<dec column
name>,<RA>,<DEC>,<semi-majorAxis>,semi-minorAxis,positionAngle)=1;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Examples:

```
curl -o catalogEllipse.json \
-d 'query=SELECT+*+FROM+sdss_stripe82_01.RunDeepSource+WHERE+scisql_s2PtInEllipse(coord_ra,coord_decl,9.469,-1.152,58,36,0)=1;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Catalog Polygon Search (Find the sources in the given polygon):

Inputs: Polygon vertexes coordinates: v1_RA, v1_DEC; v2_RA, v2_DEC; v3_RA, v3_DEC; ... (all in degree)

Note: The order of the vertex coordinates matters, either clockwise or anticlockwise.

URL:

```
curl -o <output file> \
-d 'query=SELECT+<column names>+FROM+<database name>.<table name>+WHERE+scisql_s2PtInCPoly(<ra column name>,<dec column name>, v1_RA, v1_DEC, v2_RA, v2_DEC, ...)=1;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Examples:

```
curl -o catalogPolygon.json \
-d 'query=SELECT+*+FROM+sdss_stripe82_01.RunDeepForcedSource+WHERE+scisql_s2PtInCPoly(coord_ra,coord_decl,9.4, -1.2, 9.6, -1.2, 9.4, -1.1)=1;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

(1.4) ForcedSource searches (RunDeepForcedSource table)

One reference query here is to start with an Object ID and retrieve all the ForcedSource measurements associated with that object. That is a direct foreign-key relationship between the two tables and is trivially expressed in SQL. **This query must be highly performant** - the goal should be for this to take only a few seconds. (The formal LSST requirement is <10s, but we should aspire to be quite a bit faster than that.)

Input: OBJ_ID

URL:

```
curl -o forcedSource.json \
-d 'query=SELECT+<column names>+FROM+sdss_stripe82_01.RunDeepForcedSource+WHERE+objectId=OBJ_ID;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

Example:

```
curl -o forcedSource.json \
-d 'query=SELECT+count(*)+FROM+sdss_stripe82_01.RunDeepForcedSource+WHERE+objectId=3448068867358968;' \
http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

(2) Imgserv (pixel data retrieval), see [ImageServ API - v1\[Deprecated\]](#):

"Aliveness" checks:

Is imgserv running?

- v0: `curl http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/`
- v1: `curl http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/`

Calibrated single-epoch image (and cutout) retrieval

#l_6: Retrieve calexp image closest to ra, dec.

v0: `curl -o calexpRaDecFilter.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp?filter=r&ra=37.644598&dec=0.104625"`

v1: curl -o calexpRaDecFilter.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Strip82?ds=calexp&ra=37.644598&dec=0.104625&filter=r"

#l_7: Retrieve calexp image by Butler Datald.

v0: curl -o calexpBrokenId.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/ids?run=5646&camcol=4&field=694&filter=g>"

v1: curl -o calexpBrokenId.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Strip82?ds=calexp&run=5646&camcol=4&field=694&filter=g"

#l_8: Retrieve calexp image by Butler Datald. (use id = 5646240694)

v0: curl -o calexpId.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/id?id=5646240694>"

v1: curl -o calexpId.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Strip82?ds=calexp&sid=3325410171"

#l_9: Retrieve calexp image cutout by Datald (Butler) and dimensions in 'arcsecond'.

v0: curl -o calexpRaDecFilterW300H450.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/cutout?ra=37.644598&dec=0.104625&filter=r&width=300.0&height=450.0>"

v0: curl -o calexpRaDecFilterW450H300.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/cutout?ra=37.644598&dec=0.104625&filter=r&width=450.0&height=300.0>"

v1: curl -o calexpRaDecFilterW450H300.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Strip82?ds=calexp&ra=37.644598&dec=0.104625&filter=r&width=450.0&height=300.0&unit=arcsec"

#l_10: Retrieve calexp image cutout by Butler Datald and dimensions in 'pixel'.

v0: curl -o calexpRaDecFilterPixW300H450.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/cutoutPixel?ra=37.644598&dec=0.104625&filter=r&width=300.0&height=450.0>"

v0: curl -o calexpRaDecFilterPixW450H300.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/cutoutPixel?ra=37.644598&dec=0.104625&filter=r&width=450.0&height=300.0>"

v1: curl -o calexpRaDecFilterPixW450H300.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Strip82?ds=calexp&ra=37.644598&dec=0.104625&filter=r&width=450&height=300&unit=pixel"

#l_11: Retrieve calexp image cutout by id and dimensions in 'arcsecond' centered on ra, dec.

v0: curl -o calexpIdW300H450.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/5646240694/cutout?ra=37.6292&dec=0.104625&widthAng=300.0&heightAng=450.0>"

v0: curl -o calexpIdW450H300.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/5646240694/cutout?ra=37.6292&dec=0.104625&widthAng=450.0&heightAng=300.0>"

v1: curl -o calexpIdW450H300.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Strip82?ds=calexp&sid=5646240694&ra=37.6292&dec=0.104625&width=300.0&height=450.0&unit=arcsec"

#l_12: Retrieve calexp image cutout by id and dimensions in 'pixel' centered on ra, dec.

v0: curl -o calexpIdPixW100H100.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/5646240694/cutout?ra=37.6292&dec=0.104625&widthPix=100&heightPix=100>"

v0: curl -o calexpIdPixW200H100.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/5646240694/cutout?ra=37.6292&dec=0.104625&widthPix=200&heightPix=100>"

v0: curl -o calexpIdPixW100H200.fits "<http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/calexp/5646240694/cutout?ra=37.6292&dec=0.104625&widthPix=100&heightPix=200>"

v1: curl -o calexpIdPixW100H200.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Strip82?ds=calexp&sid=5646240694&ra=37.6292&dec=0.104625&width=100&height=200&unit=pixel"

(note that single-epoch image metadata is available in the Science_Ccd_Exposure table - see above)

Is image missing?

curl -d 'query=SELECT scienceCcdExposureId FROM Science_Ccd_Exposure_NoFile where scienceCcdExposureId=1755110401' <http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync>

Coadded image (and cutout) retrieval

#l_13: Retrieve deepCoadd image by ra and dec and filter

v0: curl -o coaddOldRaDecFilter.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd?ra=19.36995&dec=-0.3146&filter=r"

v0: curl -o coaddNewRaDecFilter.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd?ra=37.6292&dec=0.104625&filter=r"

v1: curl -o coaddNewRaDecFilter.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Stripe82?ds=deepcoadd&ra=19.37995&dec=-0.3146&filter=r"

#l_14: Retrieve deepCoadd image by Butler DataId.

v0: curl -o coaddBrokenId.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd/ids?tract=0&patch=225,1&filter=i"

v1: curl -o coaddBrokenId.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Stripe82?ds=deepcoadd&tract=0&patch=225,1&filter=i"

#l_15: Retrieve deepCoadd image by id.

v0: curl -o coaddId.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd/id?id=23986176"

v1: curl -o coaddId.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Stripe82?ds=deepcoadd&sid=23986176"

#l_16: Retrieve deepCoadd cutout image nearest to ra, dec with dimensions in 'arcsecond'.

v0: curl -o coaddRaDecFilterOldRaDecW115H235.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd/cutout?ra=19.36995&dec=-0.3146&filter=r&width=115&height=235"

v0: curl -o coaddRaDecFilterOldRaDecW30H45.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd/cutout?ra=19.36995&dec=-0.3146&filter=r&width=30&height=45"

v0: curl -o coaddRaDecFilterNewRaDecW115H235.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd/cutout?ra=37.6292&dec=0.104625&filter=r&width=115&height=235"

v0: curl -o coaddRaDecFilterNewRaDecW300H450.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd/cutout?ra=37.6292&dec=0.104625&filter=r&width=300&height=450"

v1: curl -o coaddRaDecFilterNewRaDecW300H450.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Stripe82?ds=deepcoadd&ra=19.36995&dec=-0.3147&filter=r&width=300&height=400&unit=arcsec"

#l_17: Retrieve deepCoadd cutout image nearest to ra, dec with dimensions in 'Pixel'.

v0: curl -o coaddRaDecFilterPixW115H235.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd/cutoutPixel?ra=19.36995&dec=-0.3146&filter=r&width=115&height=235"

v0: curl -o coaddRaDecFilterPixW235H115.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v0/deepCoadd/cutoutPixel?ra=19.36995&dec=-0.3146&filter=r&width=235&height=115"

v1: curl -o coaddRaDecFilterPixW235H115.fits "http://lsst-qserv-dax01.ncsa.illinois.edu:5000/image/v1/DC_W13_Stripe82?ds=deepcoadd&ra=19.36995&dec=-0.3147&filter=r&width=115&height=235&unit=pixel"

(note that coadded patch image metadata is available in the DeepCoadd table - see above)

(3) Basic light curve / time series viewer query on RunDeepForcedSource

For returning the magnitude:

Computing a calibrated magnitude requires correcting the observed flux for the photometric zero point. This requires a join between the RunDeepForcedSource table, for the information about the individual observations, and the Science_Ccd_Exposure table (for a per-visit-image photometric zero point).

Input: an objectId:

Example:

Tested (Gregory Dubois-Felsmann 28 Nov 2016):

```
curl -d 'query=SELECT objectId, id, fsrc.exposure_id, fsrc.exposure_time_mid, exp.run, scisql_dnToAbMag(fsrc.flux_psf,exp.fluxMag0) AS g, scisql_dnToAbMagSigma(fsrc.flux_psf, fsrc.flux_psf_err, exp.fluxMag0, exp.fluxMag0Sigma) AS gErr \n\nFROM sdss_stripe82_01.RunDeepForcedSource AS fsrc, sdss_stripe82_01.Science_Ccd_Exposure AS exp WHERE exp.scienceCcdExposureId = fsrc.exposure_id AND fsrc.exposure_filter_id=1 AND objectId=3448068867358968 \n\nORDER BY exposure_time_mid' http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

With **flag bits not as yet corrected to their Summer 2013 / PDAC form**:

```
SELECT\n  objectId,\n  id,\n  fsrc.exposure_id,\n  fsrc.exposure_time_mid,\n  exp.run,\n  scisql_dnToAbMag(fsrc.flux_psf, exp.fluxMag0) AS g,\n  scisql_dnToAbMagSigma(fsrc.flux_psf, fsrc.flux_psf_err, exp.fluxMag0, exp.fluxMag0Sigma) AS gErr\nFROM\n\n  sdss_stripe82_01.RunDeepForcedSource AS fsrc,\n\n  sdss_stripe82_01.Science_Ccd_Exposure AS exp\n\nWHERE\n  exp.scienceCcdExposureId = fsrc.exposure_id\n  AND fsrc.exposure_filter_id = 1\n  AND NOT (fsrc.flagPixEdge | fsrc.flagPixSaturAny |\n           fsrc.flagPixSaturCen | fsrc.flagBadApFlux |\n           fsrc.flagBadPsfFlux)\n  AND objectId = 3448068867358968\nORDER BY\n  exposure_time_mid
```

(See <https://dev.lsstcorp.org/trac/wiki/DC/Winter2013> for more information. I have updated the RunDeepForcedSource table name, all the column names except for the flag bits, added the exposure ID to the 'SELECT' clause, and updated the objectId to the one in the other reference queries on the present page, compared to the query given on Trac. Note that we'll need the single-epoch exposure ID in the results table in order to set up the link to the image cutouts for the light curve viewer. I've also updated the objectId to the one in the other reference queries on the present page.)

- Gregory Dubois-Felsmann ~~Fix the flag bits~~

For returning the flux:

Computing a calibrated flux (**call functions `scisql_dnToFlux` and `scisql_dnToFluxSigma`**) requires correcting the observed flux for the photometric zero point. This requires a join between the RunDeepForcedSource table (for the information about the individual observations) and the Science_Ccd_Exposure table (for a per-visit-image photometric zero point).

Input: an objectId:

Example:

Tested (Yi Mei 23 Jan 2017):

```
curl -d 'query=SELECT objectId, id, fsrc.exposure_id, fsrc.exposure_time_mid, exp.run, scisql_dnToFlux(fsrc.flux_psf,exp.fluxMag0) AS tsv_flux, scisql_dnToFluxSigma(fsrc.flux_psf, fsrc.flux_psf_err, exp.fluxMag0, exp.fluxMag0Sigma) AS tsv_fluxErr \n\nFROM\n\n  sdss_stripe82_01.RunDeepForcedSource AS fsrc,\n\n  sdss_stripe82_01.Science_Ccd_Exposure AS exp\n\nWHERE exp.scienceCcdExposureId = fsrc.exposure_id AND fsrc.exposure_filter_id=1 AND\nobjectId=3448068867358968 \n\nORDER BY exposure_time_mid' http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync
```

With **flag bits not as yet corrected to their Summer 2013 / PDAC form**:

```

SELECT
    objectId,
    id,
    fsrc.exposure_id,
    fsrc.exposure_time_mid,
    exp.run,
    scisql_dnToFlux(fsrc.flux_psf, exp.fluxMag0) AS tsv_flux,
    scisql_dnToFluxSigma(fsrc.flux_psf, fsrc.flux_psf_err, exp.fluxMag0, exp.fluxMag0Sigma) AS tsv_fluxErr
FROM

    sdss_stripe82_01.RunDeepForcedSource AS fsrc,

    sdss_stripe82_01.Science_Ccd_Exposure AS exp

WHERE
    exp.scienceCcdExposureId = fsrc.exposure_id
    AND fsrc.exposure_filter_id = 1
    AND NOT (fsrc.flagPixEdge | fsrc.flagPixSaturAny |
             fsrc.flagPixSaturCen | fsrc.flagBadApFlux |
             fsrc.flagBadPsfFlux)
    AND objectId = 3448068867358968
ORDER BY
    exposure_time_mid

```

For WISE Light Curve Query:

Step 1. Get the object id (the column cntr) from the object table: target RA = 10.68479, Dec = 41.26906, radius = 0.02777777777777776 deg

```

curl -o wiseObjectsCone.json\

-d 'query=SELECT+*+FROM+wise_00.allwise_p3as_psd+WHERE+qserv_areaspec_circle(10.68479,41.26906,0.02777777777777777);\
' http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync

```

Step 2. Pick one object id from Step 1 result to query the forced sources (the column cntr_mf):

```

curl -o wiseForcedSourceConeObjId.json\

-d 'query=SELECT+*+FROM+wise_00.allwise_p3as_mep+WHERE+qserv_areaspec_circle(10.68479, 41.26906,0.027777777777777776)
+AND+ cntr_mf = 98140801351043831;\
' http://lsst-qserv-dax01.ncsa.illinois.edu:5000/db/v0/tap/sync

```

Note: The `qserv_areaspec_circle` can be removed from the second step (search the forced sources) after Qserv improves the WISE data table indexing.

(See <https://dev.lsstcorp.org/trac/wiki/DC/Winter2013> for more information. I have updated the RunDeepForcedSource table name, all the column names except for the flag bits, added the exposure ID to the 'SELECT' clause, and updated the objectId to the one in the other reference queries on the present page, compared to the query given on Trac. Note that we'll need the single-epoch exposure ID in the results table in order to set up the link to the image cutouts for the light curve viewer. I've also updated the objectId to the one in the other reference queries on the present page.)

- [Gregory Dubois-Felsmann](#) ~~Fix the flag bits~~