NCI GDC API

Version 1.0.0
Date: 2016-01-25

Introduction

The GDC Application Programming Interface (API) provides developers with a programmatic interface to query and download GDC data as well as to submit data to GDC

The GDC API is the external facing REST interface for the National Cancer Institute (NCI) Genomic Data Commons (GDC). The GDC API drives the GDC Data Portal and is made accessible to external developers for programmatic access to the same functionality found through the GDC Data Portal. This includes searching for and downloading subsets of data files, metadata, and annotations based on specific parameters. GDC API also allows programatic access for slicing BAM files and for submission of dbGaP registered projects (studies) and its associated data files, metadata, and annotations to GDC.

GDC API Key Features are:

- **Query GDC data** Using a wide set of parameters, GDC API can be queried to return results of a search or details about a specific entity. Helper features are available to assist the user in building their query and understand available fields
- **Download files** GDC API can be used to download data files, if restricted data is requested, a token must be provided by the user along with his API call. This token can be downloaded directly from GDC Data Portal or GDC Submission Portal upon login.
- **BAM File Slcing** GDC API also supports BAM slicing to retrieve only a portion of a file.
- **Submit data to GDC** GDC API can be used to validate and submit data to the GDC by performing create, update, retrieve actions on entities.

Authentication

Each GDC API request must include “X-Auth-Token” custom header.

```
# With shell, you can just pass the correct header with each request
export TOKEN='your_GDC_API_token'
curl -H 'X-Auth-Token:$TOKEN' -k -O https://gdc-api.nci.nih.gov/data/64cfcd9d-8a23-4f88-95b6-49c05c7cbde4

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 53.3M    0 53.3M    0     0   513k      0 --:--:--  0:01:46 --:--:-- 1208k
```

```
import requests

token = open("gdc-user-token.2016-01-20T22_22_14-05_00.txt").read()
data_endpt = "https://gdc-api.nci.nih.gov/data"
item_uuid = "64cfcd9d-8a23-4f88-95b6-49c05c7cbde4"
headers = {"X-Auth-Token":token}

response = requests.get("{url}/{uuid}".format(url=data_endpt,uuid=item_uuid),headers=headers)

outf = open(item_uuid+".cel","wb")
outf.write(response.content)
outf.close()
```

Make sure to replace `your_GDC_API_token` with your GDC token.

Open data and metadata in GDC can be downloaded via GDC API without authentication. However, in order to obtain access to controlled data available in GDC, researchers must first obtain an NIH eRA Commons account and authorization to access the data through the NIH database of Genotypes and Phenotypes (dbGaP).

How can I access controlled data in GDC?

Once a researcher has obtained an eRA Commons account and dbGaP access, they can log into the GDC Data Portal and acquire an authentication token from the portal that will allow them to download the controlled data sets for which they have access via the GDC API.

To obtain the authentication token:

1. Log in to the GDC Data Portal by clicking the Login button at the top right of the page. This will redirect to the eRA Commons login page.
2. Once logged in to the eRA Commons, the GDC Data Portal will recognize the user as logged in and display the username in place of the login button.
3. Clicking the username will trigger a drop-down menu. From that menu, the option ‘Download Token’ may be selected, which will initiate an HTTPS download of a token file.
4. This file will contain the authentication token to be used with any secure data access associated with the GDC.

Remember — The authentication token should be kept in a secure location, as it allows

access to all data accessible by the associated user.

Token Expiration A token lasts for 1 month. Once it expires you will get a 401 HTTP error code.

Unauthorized access to data (for example invalid token) will produce the following error message: ' HTTP/1.1 403 FORBIDDEN { “error”: “You don’t have access to the data” } '

Remember — Invalid credentials will result in a server error even if the data is open access.

Search and Retrieval

Query Parameters

The following query parameters can be used with all methods and resources in the GDC API. The use of any particular parameter is optional except where noted.

Parameter	Default	Description
facets	false	Provides a list document counts for each included facet
fields	false	Query option to specify which fields to include in the response
filters	false	Query option filters specify criteria for the returned response
from	false	allows to specify the first record to return from the set resulting of a query
size	false	determines the number of results to return
sort	false	specifies a field to sort the returned results by sort order: + use asc for ascending order + use des for descending order
pretty	false	Returns response with indentations and line breaks in a human-readable format
format	false	Returns response in XML or TSV format, JSON is default

FACETS

```
$ curl  'https://gdc-api.nci.nih.gov/projects?facets=program.name&from=1&size=0&sort=program.name:asc&pretty=true'
```

```
import requests
import json

projects_endpt = 'https://gdc-api.nci.nih.gov/projects'
params = {'facets':'program.name',
          'from':1, 'size':0,
          'sort':'program.name:asc'}
response = requests.get(projects_endpt, params = params)
print json.dumps(response.json(), indent=2)
```

The above command returns JSON structured like this:


```
{
  "data": {
    "pagination": {
      "count": 0,
      "sort": "program.name:asc",
      "from": 1,
      "pages": 46,
      "total": 46,
      "page": 1,
      "size": 0
    },
    "hits": [],
    "aggregations": {
      "program.name": {
        "buckets": [
          {
            "key": "TCGA",
            "doc_count": 37
          },
          {
            "key": "TARGET",
            "doc_count": 9
          }
        ]
      }
    }
  },
  "warnings": {}
}
```


The ***facets*** query parameter provides an aggregated data based on a search query. In the simplest case, a terms facet can return facet counts for various facet values for a specific field.

For Example, to get a count of projects in each program, *facets=program.name* can be passed to the *projects* endpoint.

A list of all valid *field* names that can be used as facets is available in Appendix A .

FIELDS

To get back only the file names for each file, **fields=file_name** can be passed to the files endpoint.

```
$ curl 'https://gdc-api.nci.nih.gov/files?fields=file_name&pretty=true'
```

```
import requests
import json

files_endpt = 'https://gdc-api.nci.nih.gov/files'
params = {'fields':'file_name'}
response = requests.get(files_endpt, params = params)
print json.dumps(response.json(), indent=2)
```

The above command returns JSON structured like this:

```
{
  "data": {
    "hits": [
      {
        "file_name": "Collagen_VI-R-V_GBL1112757.tif"
      },
      {
        "file_name": "DUNGS_p_TCGA_b84_115_SNP_N_GenomeWideSNP_6_C09_771624.birdseed.data.txt"
      },
      {
        "file_name": "unc.edu.ba350477-3c49-4258-8409-f39447687497.2145690.junction_quantification.txt"
      },
      {
        "file_name": "SWEDE_p_TCGAb322_23_24_25_26NSP_GenomeWideSNP_6_C06_1364960.hg18.seg.txt"
      },
      {
        "file_name": "unc.edu.a696be4d-8576-43a3-8137-30778dff9b89.2445604.junction_quantification.txt"
      },
      {
        "file_name": "MSK_252152921979_S01_CGH-v4_10_27Aug08__GCN_V3_A1__CBS_out.txt"
      },
      {
        "file_name": "9721366028_R05C01_Red.idat"
      },
      {
```

```
            "file_name": "28bdce76404c4a5a9a766c6e93f390ed.bam"
          },
          {
            "file_name": "PKC-delta_pS664-R-V_GBL9016761.tif"
          },
          {
            "file_name": "jhu-usc.edu_LAML.HumanMethylation450.2.lvl-2.TCGA-AB-3011-03A-01D-0742-05.txt"
          }
        ],
        "pagination": {
          "count": 10,
          "sort": "",
          "from": 1,
          "pages": 54777,
          "total": 547761,
          "page": 1,
          "size": 10
        }
      },
      "warnings": {}
    }
```


This query option specifies which fields to include in the response. Using fields can help improve performance. There are about 2,000 fields currently available in the data model. A complete listing of all valid fields for each endpoint type are available in Appendix A .

FILTERS

Using the query option ***filters*** lets users specify criteria for the returned response. See Filter Available Operations (op).

The *filters* syntax is a JSON object that contains the query filters that are translatable to ElasticSearch JSON-based queries by the GDC middleware

Users can get a list of available values for a specific field in the filter by making a call to the appropriate API endpoint using the ‘facets’ parameters.

> To get a list of available values for the clinical.gender field, users can query the cases endpoint using the 'facets’ parameter.

```
$ curl 'https://gdc-api.nci.nih.gov/cases?facets=clinical.gender&from=1&size=0&sort=clinical.gender:asc&pretty=true'
```

```
import requests
import json

cases_endpt = 'https://gdc-api.nci.nih.gov/cases'
```

```
params = {'facets':'clinical.gender',
          'from':1, 'size':0,
          'sort':'clinical.gender:asc'}
response = requests.get(cases_endpt, params = params)
print json.dumps(response.json(), indent=2)
```

The above command returns JSON structured like this:


```
{
  "data": {
    "pagination": {
      "count": 0,
      "sort": "clinical.gender:asc",
      "from": 1,
      "pages": 14052,
      "total": 14052,
      "page": 1,
      "size": 0
    },
    "hits": [],
    "aggregations": {
      "clinical.gender": {
        "buckets": [
          {
            "key": "female",
            "doc_count": 6598
          },
          {
            "key": "male",
            "doc_count": 6303
          },
          {
            "key": "_missing",
            "doc_count": 1151
          }
        ]
      }
    }
  },
  "warnings": {}
}
```


Filters support complex nested operations as well as simple queries on a single field. There are different types of operations available for many uses. For more examples see 7 - Examples.

It is possible to obtain multiple values from multiple fields in one single query, by listing fields `(facets=field1,field2)`.

Filter cases keeping only 'male'. The JSON to be passed to the filter parameter looks like.

```
    {"op": "=",
          "content": {
              "field": "cases.clinical.gender",
              "value": ["male"]
          }
    }
```

```
filt = {"op":"=",
        "content":{
            "field": "cases.clinical.gender",
            "value": ["male"]
        }
}
```

The above JSON is URL encoded and passed to the filters parameter for the API call below.

```
 curl  'https://gdc-api.nci.nih.gov/cases?filters=%7b%22op%22%3a+%22%3d%22%2c%0d%0a++++++%22content%22%3a+%7b%0d%0a++++++++++%22field%22%3a+%22cases.clinical.gender%22%2c%0d%0a++++++++++%22value%22%3a+%5b%22male%22%5d%0d%0a+++++%7d%0d%0a%7d'
```

```
import requests
import json
cases_endpt = 'https://gdc-api.nci.nih.gov/cases'
filt = {"op":"=",
        "content":{
            "field": "cases.clinical.gender",
            "value": ["male"]
        }
}
params = {'filters':json.dumps(filt), 'sort':'clinical.gender:asc'}
# requests encodes automatically
response = requests.get(cases_endpt, params = params)
print json.dumps(response.json(), indent=2)
```

More in depth examples of various filter types supported in GDC are available in the Appendix A

AVAILABLE FILTERING OPERATIONS

Operators allow users to define query conditions. These can be used to restrict facet values and then to connect these in logical statements.

Operators can relate an operation to one field (Single Field Operators, e.g. A = B) or multiple fields (e.g.

[and (a,b,c,d)].

Operators (**op** in the examples in Section 6.2) can take different values depending of the context and type of data.

Type	Possible Values
Single field	=, != , <, <=, =, >, >=, in, is, not, range, exclude
Multiple fields	and, or

When using multiple fields, operator content requires nested data containing additional operators.

FROM

To get 5 file results starting from the 100th result from a set of 500 file results.

```
curl 'https://gdc-api.nci.nih.gov/files?fields=file_name&from=101&size=5&pretty=true'
```

```
import requests
import json

files_endpt = 'https://gdc-api.nci.nih.gov/files'
params = {'fields':'file_name',
          'from':101, 'size':5}
response = requests.get(files_endpt, params = params)
print json.dumps(response.json(), indent=2)
```

The above command returns JSON structured like this:

```
{
  "data": {
    "hits": [
      {
        "file_name": "unc.edu.956590e9-4962-497b-a59f-81ee0a1c0caf.1379536.junction_quantification.txt"
      },
      {
        "file_name": "MATZO_p_TCGAb40_SNP_1N_GenomeWideSNP_6_G09_667760.ismpolish.data.txt"
      },
      {
        "file_name": "GIRTH_p_TCGA_b108_137_SNP_N_GenomeWideSNP_6_D06_787864.hg18.seg.txt"
      },
      {
        "file_name": "PLENA_p_TCGAb63and64_SNP_N_GenomeWideSNP_6_B12_697382.CEL"
      },
```

```
        {
          "file_name": "TCGA-HU-8604-01A-11R-2402-13.isoform.quantification.txt"
        }
      ],
      "pagination": {
        "count": 5,
        "sort": "",
        "from": 100,
        "pages": 109553,
        "total": 547761,
        "page": 21,
        "size": 5
      }
    },
    "warnings": {}
  }
```

The GDC API uses pagination, the **from** query parameter specifies the first record to return out of the set of results. For example, if there are 20 cases returned from the case endpoint, **from** can be set to 10 and results 10-20 will be returned. The **from** parameter can be used in conjunction with the **size** parameter to return a specific subset of results. For more information see 7.1.2 Filters examples.

SIZE

This example returned the first 2 file names

```
$ curl 'https://gdc-api.nci.nih.gov/files?fields=file_name&from=0&size=2&pretty=true'
```

```
import requests
import json

files_endpt = 'https://gdc-api.nci.nih.gov/files'
params = {'fields':'file_name',
          'from':0, 'size':2}
response = requests.get(files_endpt, params = params)
print json.dumps(response.json(), indent=2)
```

The above command returns JSON structured like this:

```
{
  "data": {
    "hits": [
      {
        "file_name": "unc.edu.276a1e00-cf3a-4463-a97b-d544381219ea.2363081.rsem.isoforms.normalized_results"
      },
      {
```

```
        "file_name": "nationwidechildrens.org_clinical.TCGA-EY-A5W2.xml"
      }
    ],
    "pagination": {
      "count": 2,
      "sort": "",
      "from": 1,
      "pages": 300936,
      "total": 601872,
      "page": 1,
      "size": 2
    }
  },
  "warnings": {}
}
```

The **size** query parameter specifies the number of results to return. When size is not specified the default is 10.

SORT

Sort returned cases by ascending order by submitter ID

```
$ curl  'https://gdc-api.nci.nih.gov/cases?fields=submitter_id&sort=submitter_id:asc&pretty=true'
```

```
import requests
import json

cases_endpt = 'https://gdc-api.nci.nih.gov/cases'
params = {'fields':'submitter_id',
          'sort':'submitter_id:asc'}
response = requests.get(cases_endpt, params = params)
print json.dumps(response.json(), indent=2)
```

The above command returns JSON structured like this:

```
{
  "data": {
    "hits": [
      {
        "submitter_id": "TARGET-20-PABGKN"
      },
      {
        "submitter_id": "TARGET-20-PABHET"
      },
      {
```

```
        "submitter_id": "TARGET-20-PABHKY"
      },
      {
        "submitter_id": "TARGET-20-PABLDZ"
      },
      {
        "submitter_id": "TARGET-20-PACDZR"
      },
      {
        "submitter_id": "TARGET-20-PACEGD"
      },
      {
        "submitter_id": "TARGET-20-PADDXZ"
      },
      {
        "submitter_id": "TARGET-20-PADYIR"
      },
      {
        "submitter_id": "TARGET-20-PADZCG"
      },
      {
        "submitter_id": "TARGET-20-PADZKD"
      }
    ],
    "pagination": {
      "count": 10,
      "sort": "submitter_id.raw:asc",
      "from": 1,
      "pages": 1406,
      "total": 14052,
      "page": 1,
      "size": 10
    }
  },
  "warnings": {}
}
```


The **sort** query parameter specifies a single field to sort the returned results by sort order, `sort=field:asc` for ascending order use `sort=field:dsc` for descending order.

PRETTY

Returns response with indentations and line breaks in a human-readable format.

Returned results without pretty formatting

```
curl  'https://gdc-api.nci.nih.gov/cases?fields=submitter_id&sort=submitter_id:asc&size=5'
```


```
{"data": {"hits": [{"submitter_id": "TARGET-20-PABGKN"}, {"submitter_id": "TARGET-20-PABHET"}, {"submitter_id": "T
ARGET-20-PABHKY"}, {"submitter_id": "TARGET-20-PABLDZ"}, {"submitter_id": "TARGET-20-PACDZR"}], "pagination": {"co
unt": 5, "sort": "submitter_id.raw:asc", "from": 1, "pages": 2811, "total": 14052, "page": 1, "size": 5}}, "warnin
```

```
gs": {}}
```

Same query as above with pretty=true returns a more human-readable format

```
  $ curl  'https://gdc-api.nci.nih.gov/cases?fields=submitter_id&sort=submitter_id:asc&size=5&pretty=true'

{
  "data": {
    "hits": [
      {
        "submitter_id": "TARGET-20-PABGKN"
      },
      {
        "submitter_id": "TARGET-20-PABHET"
      },
      {
        "submitter_id": "TARGET-20-PABHKY"
      },
      {
        "submitter_id": "TARGET-20-PABLDZ"
      },
      {
        "submitter_id": "TARGET-20-PACDZR"
      }
    ],
    "pagination": {
      "count": 5,
      "sort": "submitter_id.raw:asc",
      "from": 1,
      "pages": 2811,
      "total": 14052,
      "page": 1,
      "size": 5
    }
  },
  "warnings": {}
}
```

FORMAT

A Query with format=TSV returns a results in a Tab Separated Values (TSV) format

```
curl  'https://gdc-api.nci.nih.gov/cases?fields=submitter_id&size=5&format=TSV&pretty=true'
submitter_id
TCGA-RC-A6M6
TCGA-B6-A0RV
TCGA-MB-A5Y8
TCGA-BQ-5876
TCGA-Z6-A9VB
```

```
import requests

cases_endpt = 'https://gdc-api.nci.nih.gov/cases'
params = {'fields':'submitter_id',
          'format':'TSV'}
response = requests.get(cases_endpt, params = params)
print response.content
```

Same query as above with format=XML returns results in an XML format

```
   curl  'https://gdc-api.nci.nih.gov/cases?fields=submitter_id&size=5&format=XML&pretty=true'
<?xml version="1.0" ?>
<response>
    <data>
        <hits>
            <item>
                <submitter_id>TCGA-MQ-A4LV</submitter_id>
            </item>
            <item>
                <submitter_id>TCGA-N9-A4Q1</submitter_id>
            </item>
            <item>
                <submitter_id>TCGA-78-7154</submitter_id>
            </item>
            <item>
                <submitter_id>TCGA-S7-A7WX</submitter_id>
            </item>
            <item>
                <submitter_id>TCGA-XF-AAML</submitter_id>
            </item>
        </hits>
        <pagination>
            <count>5</count>
            <sort/>
            <from>1</from>
            <pages>2811</pages>
            <total>14052</total>
            <page>1</page>
            <size>5</size>
        </pagination>
    </data>
    <warnings/>
</response>
```

```
import requests

cases_endpt = 'https://gdc-api.nci.nih.gov/cases'
```

```
params = {'fields':'submitter_id',
          'format':'XML',
          'pretty':'true'}
response = requests.get(cases_endpt, params = params)
print response.content
```

Returns the response data in a format other than JSON.

The following options are available:

- TSV
- XML

Query Endpoints

The following search and retrieval endpoints are available on the GDC API.

Endpoints	Type	Description
projects	Search & Retrieval	Search all data generated by a project
files	Search & Retrieval	Find all files with specific characteristics such as file_name, md5sum, data_format and others.
cases	Search & Retrieval	Find all files related to a specific case, or sample donor.
annotations	Search & Retrieval	Search annotations added to data after curation

The GDC API supports a wide range of query string operators. Users can use a query string operation by placing the query terms and a `?` after the endpoint in the URL `https://gdc-api.nci.nih.gov/projects?`. This allows users to select, filter and order the desired data as described in Query Parameters.

PROJECT ENDPOINT

This example is a query for Projects contained in GDC. It returns only the first two projects sorted by project id.

```
$  curl 'https://gdc-api.nci.nih.gov/projects?from=1&size=2&sort=project.project_id:asc&pretty=true'
{
  "data": {
```

```
    "hits": [
      {
        "state": "legacy",
        "project_id": "TCGA-ACC",
        "primary_site": "Adrenal Gland",
        "disease_type": "Adrenocortical Carcinoma",
        "name": "Adrenocortical Carcinoma"
      },
      {
        "dbgap_accession_number": "phs000464",
        "disease_type": "Acute Lymphoblastic Leukemia",
        "state": "legacy",
        "primary_site": "Blood",
        "project_id": "TARGET-ALL-P2",
        "name": "Acute Lymphoblastic Leukemia - Phase II"
      }
    ],
    "pagination": {
      "count": 2,
      "sort": "project.project_id:asc",
      "from": 1,
      "pages": 22,
      "total": 44,
      "page": 1,
      "size": 2
    }
  },
  "warnings": {}
}
```


The GDC Project Endpoint `https://gdc-api.nci.nih.gov/projects` provides overall access to all the data served by GDC organized by Project such project(study) name, program,disease, primary site and state.

FILES ENDPOINT

This example is a query for files contained in GDC. It returns only the first two files, sorted by file size, from smallest to largest.

```
$ curl 'https://gdc-api.nci.nih.gov/files?from=1&size=2&sort=file_size:asc&pretty=true'
{
  "data": {
    "hits": [
      {
        "origin": "migrated",
        "data_type": "Raw microarray data",
        "platform": "HG-U133_Plus_2",
        "file_name": "TCGA-AB-2842-03A-01R-0757-21.CEL.README",
        "md5sum": "56f9a6d58b450bf7e9f6431a86220b9d",
        "data_format": "CEL",
        "acl": "open",
```

```
        "access": "open",
        "uploaded_datetime": 1425340539,
        "state": "live",
        "data_subtype": "Raw intensities",
        "file_id": "ca13321c-02aa-4141-bdb6-84d31e3c5711",
        "file_size": 43,
        "experimental_strategy": "Gene expression array"
      },
      {
        "origin": "migrated",
        "data_type": "Raw microarray data",
        "platform": "HG-U133_Plus_2",
        "file_name": "TCGA-AB-2809-03A-01R-0757-21.CEL.README",
        "md5sum": "56f9a6d58b450bf7e9f6431a86220b9d",
        "data_format": "CEL",
        "acl": "open",
        "access": "open",
        "uploaded_datetime": 1425340539,
        "state": "live",
        "data_subtype": "Raw intensities",
        "file_id": "299d500b-49e2-4c62-9111-c0691592dce1",
        "file_size": 43,
        "experimental_strategy": "Gene expression array"
      }
    ],
    "pagination": {
      "count": 2,
      "sort": "file_size:asc",
      "from": 1,
      "pages": 300936,
      "total": 601872,
      "page": 1,
      "size": 2
    }
  },
  "warnings": {}
}
```


The GDC Files Endpoint `'https://gdc-api.nci.nih.gov/files` allows search and retrieval of all files with specific characteristics such as file_name, md5sum, data_format and others.

CASES ENDPOINT

This example is a query for files contained in GDC. It returns case where submitter id is 'TCGA-BH-A0EA', using the filters {"op":"and","content":[{"op":"in","content" :{"field":"submitter_id","value":["TCGA-BH-A0EA"]}}]} .

```
$ curl 'https://gdc-api.nci.nih.gov/cases?filters=%7B%22op%22%3A%22and%22%2C%22content%22%3A%5B%7B%22op%22%3A%22in%22%2C%22content%22%3A%7B%22field%22%3A%22submitter_id%22%2C%22value%22%3A%5B%22TCGA-BH-A0EA%22%5D%7D%7D%5D%7D%0A%0A%0A&pretty=true'
```

```
{
  "data": {
    "hits": [
      {
        "sample_ids": "7f791228-dd77-4ab0-8227-d784a4c7fea1",
        "portion_ids": "8629bf5a-cdaf-4f6a-90bb-27dd4a7565c5",
        "submitter_portion_ids": "TCGA-BH-A0EA-01A-21-A13C-20",
        "submitter_aliquot_ids": "TCGA-BH-A0EA-01A-11D-A10Y-09",
        "days_to_index": 0,
        "submitter_analyte_ids": "TCGA-BH-A0EA-01A-11D",
        "analyte_ids": "66ed0f86-5ca5-4dec-ba76-7ee4dcf31831",
        "submitter_id": "TCGA-BH-A0EA",
        "case_id": "1f601832-eee3-48fb-acf5-80c4a454f26e",
        "slide_ids": "90154ea1-6b76-4445-870e-d531d6fa1239",
        "submitter_sample_ids": "TCGA-BH-A0EA-01A",
        "aliquot_ids": "561b8777-801a-49ed-a306-e7dafeb044b6",
        "submitter_slide_ids": "TCGA-BH-A0EA-01A-01-TSA"
      }
    ],
    "pagination": {
      "count": 1,
      "sort": "",
      "from": 1,
      "pages": 1,
      "total": 1,
      "page": 1,
      "size": 10
    }
  },
  "warnings": {}
}
```


The GDC Cases Endpoint `'https://gdc-api.nci.nih.gov/cases` allows search and retrieval of related to a specific case, or sample donor.

ANNOTATIONS ENDPOINT

This example is a query for Annotations contained in GDC. It returns only the first two annotations.

```
$ curl 'https://gdc-api.nci.nih.gov/annotations?from=1&size=2&pretty=true'
```


```
{
  "data": {
    "hits": [
      {
        "category": "Item flagged DNU",
        "status": "Approved",
        "entity_id": "2b61b856-b988-43ca-8dc5-9f97600118ec",
        "classification": "CenterNotification",
```

```
        "entity_type": "aliquot",
        "created_datetime": 1294525038,
        "annotation_id": "7d01080f-e82d-5e58-98a6-910c041ee2b3",
        "notes": "SDRF in broad.mit.edu_READ.Genome_Wide_SNP_6.mage-tab.1.1003.0 flagged aliquot to be excluded fo
r analysis based on file 'SCENA_p_TCGAb29and30_SNP_N_GenomeWideSNP_6_C04_569122.ismpolish.data.txt'.",
        "creator": "DCC",
        "submitter_id": "1099",
        "case_id": "e7503a51-6647-4cc2-80dd-645d0df4db43",
        "entity_submitter_id": "TCGA-AG-A008-10A-01D-A003-01"
      },
      {
        "category": "Item flagged DNU",
        "status": "Approved",
        "entity_id": "d1f35d46-c6c9-4cff-ad95-e86d88b38b51",
        "classification": "CenterNotification",
        "entity_type": "aliquot",
        "created_datetime": 1414794925,
        "annotation_id": "c6a9e076-bb56-5dd9-89e7-c340594fa8f7",
        "notes": "SDRF in broad.mit.edu_COAD.Genome_Wide_SNP_6.mage-tab.1.2010.0 flagged aliquot to be excluded fo
r analysis based on file 'SNORT_p_TCGA_b89_SNP_N_GenomeWideSNP_6_E05_777376.birdseed.data.txt'.",
        "creator": "DCC",
        "submitter_id": "23507",
        "case_id": "57b0f89f-1b75-453e-922c-01cd4d44ca49",
        "entity_submitter_id": "TCGA-CK-5914-10A-01D-1649-01"
      }
    ],
    "pagination": {
      "count": 2,
      "sort": "",
      "from": 1,
      "pages": 12296,
      "total": 24592,
      "page": 1,
      "size": 2
    }
  },
  "warnings": {}
}
```


The GDC Annotation Endpoint `'https://gdc-api.nci.nih.gov/annotations` allows search and retrieval of annotations added to data after curation |

Download

Submission

BAM Slicing

Data Transfer Tool

Errors

This error section is stored in a separate file in `includes/_errors.md`. Slate allows you to optionally separate out your docs into many files…just save them to the `includes` folder and add them to the top of your `index.md`'s frontmatter. Files are included in the order listed.

The Kittn API uses the following error codes:

Error Code	Meaning
400	Bad Request – Your request sucks
401	Unauthorized – Your API key is wrong
403	Forbidden – The kitten requested is hidden for administrators only
404	Not Found – The specified kitten could not be found
405	Method Not Allowed – You tried to access a kitten with an invalid method
406	Not Acceptable – You requested a format that isn't json

410	Gone – The kitten requested has been removed from our servers
418	I'm a teapot
429	Too Many Requests – You're requesting too many kittens! Slow down!
500	Internal Server Error – We had a problem with our server. Try again later.
503	Service Unavailable – We're temporarially offline for maintanance. Please try again later.