

GDC API User's Guide

NCI Genomic Data Commons (GDC)

Contents

1	Getting Started	10
	Getting Started	10
	The GDC Application Programming Interface (API): An Overview	10
	Tools for communicating with the GDC API	10
	API Endpoints	11
	GDC Legacy Archive	11
	Entity UUIDs	11
	Sample Request	11
	Authentication	12
	Using Authentication Tokens	12
2	Search and Retrieval	14
	Search and Retrieval	14
	Introducing Search and Retrieval Requests	14
	Components of a Request	14
	POST Example	14
	Request	15
	Payload	15
	GET Example	15
	Endpoints	15
	Project Endpoint	16
	Example	16
	Retrieval of project metadata using project_id	17
	Files Endpoint	18
	Example	18
	Retrieval of file metadata using individual UUIDs:	19
	Example of retrieving file version information:	20
	Cases Endpoint	22
	Example	22
	Retrieval of case metadata using individual UUIDs:	24
	Annotations Endpoint	26

Example	27
History Endpoint	28
Example	28
_mapping Endpoint	29
Example	29
Request Parameters	30
Filters: Specifying the Query	30
Query Format	30
Example: HTTP GET Request	31
Example: HTTP POST Request	44
Format	46
Examples	46
Pretty	47
Example	47
Fields	48
Example	48
Expand	50
Example	50
Size and From	51
Example	52
Sort	53
Example	53
Facets	55
Example	55
Example	56
Alternative Request Format	58
Using Wildcards	58
Quicksearch Endpoint	59
Additional Examples	60
3 Downloading Files	61
Downloading Files	61
Data endpoint	61
Downloading a Single File using GET	61
Related Files	61
Downloading Multiple Files using GET	62
Downloading an Uncompressed Group of Files	62
Downloading Multiple Files using POST	62
POST request with form data payload	62

POST request with JSON payload	63
Downloading Controlled-access Files	64
Manifest endpoint	64
Using the manifest endpoint	64
Using return_type=manifest	64
4 Data Analysis	65
Data Analysis	65
Endpoints	65
Genes Endpoint Examples	67
Simple Somatic Mutation Endpoint Examples	68
Simple Somatic Mutation Occurrence Endpoint Examples	71
Copy Number Variation Endpoint Examples	73
Copy Number Variation Occurrence Enpoint Examples	75
Analysis Endpoints	78
Survival Analysis Endpoint	90
5 BAM Slicing	98
BAM Slicing	98
About the slicing endpoint	98
Query Parameters	98
JSON Schema	99
Examples: Specifying a region	99
Examples: Specifying a gene	100
Errors	100
JSON Error Responses	100
HTTP error codes	100
Transfer Errors	101
6 Submission	102
Submission	102
Overview	102
Submission endpoint	102
Constructing the endpoint URL	102
Example	102
Submission Formats	103
Metadata Formats	103
JSON and TSV	103
BCR XML	103
Data File Formats	103

GDC Data Model	103
UUIDs	104
Submitter IDs	104
GDC Data Dictionary Endpoints	104
Submission Templates	104
Entity JSON Schemas	105
Making Requests to the Submission API	105
Submission Transactions	105
Dry Run Transactions	105
Dry Run Commit	106
Dry Run Close	107
Asynchronous Transactions	107
Example	107
Transaction Status	108
Creating and Updating Entities	108
Example: Creating and Updating Case Entities (JSON)	108
Example: Creating an Aliquot Entity (JSON)	111
Request 1: Creating Links Using submitter_id	111
Request 2: Creating Links Using UUID	112
Example: Creating Two Samples (TSV)	114
Example: Updating a Sample Entity (JSON)	115
Updating a sample	115
Updating a sample using a BCR XML	119
Retrieving Entities	139
Entities Endpoint	139
Export Endpoint	139
GraphQL	142
Deleting Entities	142
Working With Files	143
Uploading Data Files	143
Upload Manifest	143
Uploading New Versions of Data Files	144
Downloading Files	146
Deleting Files	146
Querying Submitted Data Using GraphQL	146
GraphQL Overview	146
GraphQL IDE	146
GraphQL Endpoint	147
Constructing a Query	147

Bare GraphQL query	147
Passing GraphQL queries to GDC API directly	148
Additional Examples	150
Example: File UUID	150
Example: Case Without Diagnosis	150
Example: Number of Cases Without Diagnosis	151
Example: Aliquot State	151
Example: Aliases	151
Example: Biospecimen Tree	152
7 Python Examples	154
Using Python to Query the GDC API	154
Querying Metadata	154
A Basic Query	154
A Filtered Query	155
Complex Filters	155
Downloading Files	157
A Simple Download Request	157
Passing a Token to Download a Controlled-Access File	157
Post Request to Download Multiple Files	158
Downloading a Set of Files Based on a Filter	158
BAM Slicing	160
Basic Troubleshooting	161
8 GraphQL Examples	162
Introduction to GDC GraphQL	162
Using GDC GraphQL vs GDC REST API	162
GDC GraphQL Overview	162
GDC GraphQL Endpoints	162
GDC GraphQL Schema	162
Basic GraphQL queries in GDC	163
Anatomy of a typical GDC GraphQL Query	164
GDC GraphQL Examples	164
Nodes And Edges Example	164
Query Case File Counts	165
Query Simple Static Mutations Based on Gene IDs	165

9	System Information	167
	System Information	167
	Overview	167
	GDC Notifications Endpoint	167
	Sample Request	167
	API Status Endpoint	168
	Sample Request	168
10	Additional Examples	169
	Additional Examples	169
	Data Search and Retrieval	169
	Endpoint Examples	169
	Project Endpoint Example	169
	Files Endpoint Example	170
	Cases Endpoint Example	171
	Annotations Endpoint Example	178
	Filters Examples	179
	Example: Basic syntax	179
	Example: Filter cases keeping only ‘male’	180
	Example: Filter using a range	180
	Example: Multiple fields	181
11	Appendix A: Available Fields	182
	Appendix A: Available Fields	182
	Field Listing by Endpoint	182
	Project Fields	182
	Case Fields	183
	File Fields	196
	Annotation Fields	210
	Field Group Listing by Endpoint	210
	Projects Field Groups	210
	Cases Field Groups	210
	Files Field Groups	212
	Annotations Field Groups	213
	GDC Conventions Supplemental	213
12	Appendix B: Key Terms	215
	Appendix B: Key Terms	215

13 Appendix C: Format of Submission Queries and Responses	216
Appendix C: Format of Submission Requests and Responses	216
Format of Submission Request	216
Format of API Response to a Submission Request	216
Error Types	218
Status Messages	218
14 Release Notes	219
API Release Notes	219
v1.19.0	219
New Features and Changes	220
Bugs Fixed Since Last Release	220
Known Issues and Workarounds	220
v1.18.0	220
New Features and Changes	220
Bugs Fixed Since Last Release	220
Known Issues and Workarounds	220
v1.17.0	220
New Features and Changes	221
Bugs Fixed Since Last Release	221
Known Issues and Workarounds	221
v1.16.0	221
New Features and Changes	221
Bugs Fixed Since Last Release	221
Known Issues and Workarounds	221
v1.15.0	221
New Features and Changes	222
Bugs Fixed Since Last Release	222
Known Issues and Workarounds	222
v1.14.1	222
New Features and Changes	222
Bugs Fixed Since Last Release	222
Known Issues and Workarounds	222
v1.14.0	222
New Features and Changes	223
Bugs Fixed Since Last Release	223
Known Issues and Workarounds	223
v1.13.0	223
New Features and Changes	223

Bugs Fixed Since Last Release	223
Known Issues and Workarounds	223
v1.12.0	223
New Features and Changes	223
Bugs Fixed Since Last Release	223
Known Issues and Workarounds	224
v1.11.0	224
New Features and Changes	224
Bugs Fixed Since Last Release	224
Known Issues and Workarounds	224
v1.10.0	224
New Features and Changes	224
Known Issues and Workarounds	225
v1.9.0	225
New Features and Changes	225
Known Issues and Workarounds	225
v1.8.0	225
New Features and Changes	225
Known Issues and Workarounds	226
v1.7.1	226
New Features and Changes	226
Bugs Fixed Since Last Release	226
Known Issues and Workarounds	226
v1.5.0	227
New Features and Changes	227
Bugs Fixed Since Last Release	227
Known Issues and Workarounds	227
v1.4.0	227
New Features and Changes	227
Bugs Fixed Since Last Release	227
Known Issues and Workarounds	228
v1.3.1	228
New Features and Changes	228
Bugs Fixed Since Last Release	228
Known Issues and Workarounds	228
v1.2.0	228
New Features and Changes	229
Bugs Fixed Since Last Release	229
Known Issues and Workarounds	229

v1.1.0 229
 New Features and Changes 229
 Bugs Fixed Since Last Release 229
 Bugs Fixed Since Last Release 229
 Known Issues and Workarounds 230
v1.0.1 230
 New Features and Changes 230
 Bugs Fixed Since Last Release 230
 Known Issues and Workarounds 230

Chapter 1

Getting Started

Getting Started

The GDC Application Programming Interface (API): An Overview

The GDC API drives the GDC Data and Submission Portals and provides programmatic access to GDC functionality. This includes searching for, downloading, and submitting data and metadata. The GDC API uses JSON as its communication format, and standard HTTP methods like `GET`, `PUT`, `POST` and `DELETE`.

This guide explains how to construct and execute API requests and interpret API responses.

Tools for communicating with the GDC API

Many third-party tools can be used for communicating with the GDC API and for preparing and visualizing API calls.

Examples of tools for communicating with the GDC API:

Tool	Type
Curl	Command line tool
HTTPIe	Command line tool
Postman REST Client	App for Google Chrome and OS X
DHC REST Client	Google Chrome extension
Google Chrome	Google Chrome web browser

Examples of tools that can help build GDC API calls:

Tool	Description
JSONLint	Validate JSON
JSON Formatter	Format, validate, and convert JSON to other formats
Percent-(URL)-encoding tool	Tool for percent-encoding strings

API Endpoints

Communicating with the GDC API involves making calls to API endpoints. Each GDC API endpoint represents specific API functionality, as summarized in the following table:

Endpoint	Type	Description
status	Status	Get the API status and version information
projects	Search & Retrieval	Search all data generated by a project
cases	Search & Retrieval	Find all files related to a specific case, or sample donor.
files	Search & Retrieval	Find all files with specific characteristics such as file_name, md5sum, data_format and others.
annotations	Search & Retrieval	Search annotations added to data after curation
data	Download	Used to download GDC data
manifest	Download	Generates manifests for use with GDC Data Transfer Tool
slicing	BAM Slicing	Allows remote slicing of BAM format objects
submission	Submission	Returns the available resources at the top level above programs i.e., registered programs

The HTTP URL that corresponds to the latest version of a GDC API endpoint is `https://api.gdc.cancer.gov/<endpoint>`, where `<endpoint>` is the name of the endpoint.

The HTTP URL of an endpoint corresponding to a specific major version of the GDC API is `https://api.gdc.cancer.gov/<version>/<endpoint>` where `<endpoint>` is the name of the endpoint and `<version>` is the GDC API version.

For example, the address of the latest version of the `status` endpoint is `https://api.gdc.cancer.gov/status`, whereas the address of the `status` endpoint corresponding to version 0 of GDC API is `https://api.gdc.cancer.gov/v0/status`.

GDC Legacy Archive

To interact with data in the GDC Legacy Archive, add `legacy` to the endpoint URL before the `<endpoint>`:

```
1 https://api.gdc.cancer.gov/legacy/<endpoint>
```

NOTE: The version can also be applied to a Legacy Archive search by placing the `<version>` before `"/legacy/"`

Entity UUIDs

All objects (*entities*) in the GDC are assigned a unique identifier in the form of a [version 4 universally unique identifier \(UUID\)](#). The UUID uniquely identifies the entity in the GDC, and is stored in the entity's `id` property.

UUIDs are frequently used in GDC API requests and responses to identify specific entities like files, cases, and samples.

See GDC Data Model for details.

Sample Request

The following is an example of a request to the `files` endpoint, which retrieves information about a BAM file stored in the GDC.

```
1 curl https://api.gdc.cancer.gov/files/d853e541-f16a-4345-9f00-88e03c2dc0bc?pretty=true
```

```
“ python import requests import json
```

```
file_endpt = 'https://api.gdc.cancer.gov/files/' file_uuid = 'd853e541-f16a-4345-9f00-88e03c2dc0bc' response = re-  
quests.get(file_endpt + file_uuid) print json.dumps(response.json(), indent=2)
```

```
1 {  
2   "data": {  
3     "data_type": "Aligned Reads",  
4     "updated_datetime": "2016-05-26T17:06:40.003624-05:00",  
5     "created_datetime": "2016-05-26T17:06:40.003624-05:00",  
6     "file_name": "0017ba4c33a07ba807b29140b0662cb1_gdc_realn.bam",  
7     "md5sum": "a08304b120c5df76b6532da0e9a35ced",  
8     "data_format": "BAM",  
9     "acl": [  
10      "phs000178"  
11    ],  
12     "access": "controlled",  
13     "platform": "Illumina",  
14     "state": "submitted",  
15     "file_id": "d853e541-f16a-4345-9f00-88e03c2dc0bc",  
16     "data_category": "Raw Sequencing Data",  
17     "file_size": 23650901931,  
18     "submitter_id": "c30188d7-be1a-4b43-9a17-e19ccd71792e",  
19     "type": "aligned_reads",  
20     "file_state": "processed",  
21     "experimental_strategy": "WXS"  
22   },  
23   "warnings": {}  
24 }
```

Authentication

Authentication is required for downloading controlled-access data, and for all data submission functionality. The GDC API uses tokens for authentication.

Users can obtain authentication tokens from the [GDC Data Portal](#) and the [GDC Data Submission Portal](#). See the [GDC Data Portal User's Guide](#) and the [GDC Data Submission Portal User's Guide](#) for instructions.

Using Authentication Tokens

All API requests that require authentication must include a token as an `X-Auth-Token` custom HTTP header.

In the following example, an authentication token is saved as an environment variable and passed to `curl` to download a controlled-access file:

```
1 token=$(<gdc-token-text-file.txt)  
2  
3 curl -O -J -H "X-Auth-Token: $token"  
   'https://api.gdc.cancer.gov/data/a1c1b23b-cc41-4e85-b1b7-62a42873c5af'  
  
1  % Total    % Received % Xferd  Average Speed   Time    Time       Time  Current  
2                                Dload  Upload    Total   Spent    Left   Speed  
3 100 31.4M  100 31.4M    0     0   290k      0  0:01:50  0:01:50  ---:---:-- 172k  
4 curl: Saved to filename  
   'ACOLD_p_TCGA_Batch17_SNP_N_GenomeWideSNP_6_A03_466078.tangent.copynumber.data.txt'
```

For more information about authentication tokens, including token expiration and rotation, see [Data Security](#).

NOTE: The authentication token should be kept in a secure location, as it allows access to all data accessible by the associated user account.

Chapter 2

Search and Retrieval

Search and Retrieval

Introducing Search and Retrieval Requests

The GDC API provides endpoints that search and retrieve information stored in the GDC according to the GDC Data Model. The general format of requests to search & retrieval endpoints is described below.

Note: Queries described in this section work for datasets that have been released to the GDC Data Portal. Unreleased data that is in the process of being submitted to GDC cannot be queried using these methods. See [Submission](#) to learn how to query unreleased data using GraphQL.

Components of a Request

A typical search and retrieval API request specifies the following parameters:

- a `filters` parameter, that specifies the search terms for the query
- several parameters that specify the API response, such as:
 - `format` — specifies response format (JSON, TSV, XML)
 - `fields` — specifies the which data elements should be returned in the response, if available
 - `size` — specifies the the maximum number of results to include in the response
 - other parameters are described below.

Requests can be executed using HTTP GET or HTTP POST. GET requests are limited by maximum URL length, so the POST method is recommended for large queries.

Note: Requests for information stored in the GDC Legacy Archive must be directed to `legacy/` endpoints. See [Getting Started](#) for details.

POST Example

The following is an example of an HTTP POST request to the `files` endpoint of the GDC API. It looks for Gene Expression Quantification files associated with specific TCGA cases (represented by TCGA barcodes) and retrieves the associated biospecimen metadata in TSV format.

Request

```
1 curl --request POST --header "Content-Type: application/json" --data @Payload  
  'https://api.gdc.cancer.gov/files' > response.tsv
```

Payload

```
1 {  
2   "filters":{  
3     "op":"and",  
4     "content":[  
5       {  
6         "op":"in",  
7         "content":{  
8           "field":"cases.submitter_id",  
9           "value":[  
10            "TCGA-CK-4948",  
11            "TCGA-D1-A17N",  
12            "TCGA-4V-A9QX",  
13            "TCGA-4V-A9QM"  
14          ]  
15        }  
16      },  
17      {  
18        "op":"=",  
19        "content":{  
20          "field":"files.data_type",  
21          "value":"Gene Expression Quantification"  
22        }  
23      }  
24    ]  
25  },  
26  "format":"tsv",  
27  "fields":"file_id,file_name,cases.submitter_id,cases.case_id,data_category,data_type,cases.samples.tumor_des",  
28  "size":"1000"  
29 }
```

Each component of the request is explained below.

GET Example

The above request can be executed as an HTTP GET:

```
1 https://api.gdc.cancer.gov/files?filters=%7B%22op%22%3A%22and%22%2C%22content%22%3A%5B%7B%22op%22%3A%22in%22%2C%
```

Each component of the request is explained below.

Endpoints

The following search and retrieval endpoints are available in the GDC API:

Endpoints	Description
files	Information about files stored in the GDC
cases	Information related to cases, or sample donors
history	Information related to file version history

Endpoints	Description
projects	Information about projects
annotations	Information about annotations to GDC data
_mapping	Information about elements that can be used to query other endpoints

The choice of endpoint determines what is listed in the search results. The `files` endpoint will generate a list of files, whereas the `cases` endpoint will generate a list of cases. Each of the above endpoints, other than `_mapping`, can query and return any of the related fields in the GDC Data Model. So the `cases` endpoint can be queried for file fields (e.g. to look for cases that have certain types of experimental data), and the `files` endpoint can be queried for clinical metadata associated with a case (e.g. to look for files from cases diagnosed with a specific cancer type).

Project Endpoint

The `projects` endpoint provides access to project records, the highest level of data organization in the GDC.

Example

This example is a query for projects contained in the GDC. It uses the `from`, `size`, `sort`, and `pretty` parameters, and returns the first two projects sorted by project id.

```
1 curl 'https://api.gdc.cancer.gov/projects?from=0&size=2&sort=project.project_id:asc&pretty=true'
```

```
1 {
2   "data": {
3     "hits": [
4       {
5         "dbgap_accession_number": null,
6         "disease_type": [
7           "Brain Lower Grade Glioma"
8         ],
9         "released": true,
10        "state": "legacy",
11        "primary_site": [
12          "Brain"
13        ],
14        "project_id": "TCGA-LGG",
15        "id": "TCGA-LGG",
16        "name": "Brain Lower Grade Glioma"
17      },
18      {
19        "dbgap_accession_number": null,
20        "disease_type": [
21          "Thyroid Carcinoma"
22        ],
23        "released": true,
24        "state": "legacy",
25        "primary_site": [
26          "Thyroid"
27        ],
28        "project_id": "TCGA-THCA",
29        "id": "TCGA-THCA",
30        "name": "Thyroid Carcinoma"
31      }
32    ]
33  }
34 }
```

```

31     }
32 ],
33 "pagination": {
34     "count": 2,
35     "sort": "project.project_id:asc",
36     "from": 0,
37     "page": 1,
38     "total": 39,
39     "pages": 20,
40     "size": 2
41 }
42 },
43 "warnings": {}
44 }

```

Retrieval of project metadata using project_id

The project endpoint supports a simple query format that retrieves the metadata of a single project using its `project_id`:

```

1 curl
   'https://api.gdc.cancer.gov/projects/TARGET-NBL?expand=summary,summary.experimental_strategies,summary.data_

```

```

1 {
2   "data": {
3     "dbgap_accession_number": "phs000467",
4     "disease_type": [
5       "Neuroblastoma"
6     ],
7     "summary": {
8       "data_categories": [
9         {
10          "case_count": 151,
11          "file_count": 471,
12          "data_category": "Transcriptome Profiling"
13        },
14        {
15          "case_count": 1127,
16          "file_count": 3,
17          "data_category": "Biospecimen"
18        },
19        {
20          "case_count": 216,
21          "file_count": 1732,
22          "data_category": "Simple Nucleotide Variation"
23        },
24        {
25          "case_count": 7,
26          "file_count": 1,
27          "data_category": "Clinical"
28        },
29        {
30          "case_count": 270,
31          "file_count": 599,
32          "data_category": "Raw Sequencing Data"
33        }
34      ],
35     "case_count": 1127,

```

```

36     "file_count": 2806,
37     "experimental_strategies": [
38       {
39         "case_count": 221,
40         "file_count": 2174,
41         "experimental_strategy": "WXS"
42       },
43       {
44         "case_count": 151,
45         "file_count": 628,
46         "experimental_strategy": "RNA-Seq"
47       }
48     ],
49     "file_size": 8157614402888
50   },
51   "released": true,
52   "state": "legacy",
53   "primary_site": [
54     "Nervous System"
55   ],
56   "project_id": "TARGET-NBL",
57   "name": "Neuroblastoma"
58 },
59 "warnings": {}
60 }

```

Files Endpoint

The GDC Files Endpoint <https://api.gdc.cancer.gov/files> enables search and retrieval of information relating to files stored in the GDC, including file properties such as `file_name`, `md5sum`, `data_format`, and others.

Example

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

```
1 curl 'https://api.gdc.cancer.gov/files?from=0&size=2&sort=file_size:asc&pretty=true'
```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "data_release": "13.0",
6         "data_type": "Raw Simple Somatic Mutation",
7         "updated_datetime": "2018-07-20T22:27:55.342974+00:00",
8         "file_name": "333193d5-ca9a-4262-81f5-e9f3b44358fe.vcf.gz",
9         "submitter_id": "AD19_SimpleSomaticMutation",
10        "file_id": "333193d5-ca9a-4262-81f5-e9f3b44358fe",
11        "file_size": 866,
12        "id": "333193d5-ca9a-4262-81f5-e9f3b44358fe",
13        "created_datetime": "2017-09-10T19:16:02.549312-05:00",
14        "md5sum": "e33e95edb778fe67643162ef0ae3297e",
15        "data_format": "VCF",
16        "acl": [
17          "phs001179"
18        ],

```

```

19     "access": "controlled",
20     "state": "released",
21     "version": "1",
22     "data_category": "Simple Nucleotide Variation",
23     "type": "simple_somatic_mutation",
24     "experimental_strategy": "Targeted Sequencing"
25 },
26 {
27     "data_release": "13.0",
28     "data_type": "Raw Simple Somatic Mutation",
29     "updated_datetime": "2018-07-20T22:27:55.342974+00:00",
30     "file_name": "d9114e23-0f62-4979-aefc-0dd4d5eb891b.vcf.gz",
31     "submitter_id": "AD116_SimpleSomaticMutation",
32     "file_id": "d9114e23-0f62-4979-aefc-0dd4d5eb891b",
33     "file_size": 866,
34     "id": "d9114e23-0f62-4979-aefc-0dd4d5eb891b",
35     "created_datetime": "2017-09-10T21:53:02.376246-05:00",
36     "md5sum": "95bbfd0586d3c284e9f88edf3bf26065",
37     "data_format": "VCF",
38     "acl": [
39         "phs001179"
40     ],
41     "access": "controlled",
42     "state": "released",
43     "version": "1",
44     "data_category": "Simple Nucleotide Variation",
45     "type": "simple_somatic_mutation",
46     "experimental_strategy": "Targeted Sequencing"
47 }
48 ],
49 "pagination": {
50     "count": 2,
51     "sort": "file_size:asc",
52     "from": 0,
53     "page": 1,
54     "total": 356381,
55     "pages": 178191,
56     "size": 2
57 }
58 },
59 "warnings": {}
60 }

```

Retrieval of file metadata using individual UUIDs:

The /files endpoint supports a simple query format that retrieves the metadata of a single file using its UUID. Note that the /files endpoint is inactive when querying for earlier file versions. In that case, the /history or /files/versions endpoints should be used instead.

```
1 curl 'https://api.gdc.cancer.gov/files/874e71e0-83dd-4d3e-8014-10141b49f12c?pretty=true'
```

```

1 {
2   "data": {
3     "data_release": "13.0",
4     "data_type": "Raw Simple Somatic Mutation",
5     "updated_datetime": "2018-07-20T22:27:55.342974+00:00",
6     "created_datetime": "2016-06-03T17:03:06.608739-05:00",

```

```

7   "file_name": "874e71e0-83dd-4d3e-8014-10141b49f12c.vcf.gz",
8   "md5sum": "acf2929b1b825bcd1377023e8b8767ec",
9   "data_format": "VCF",
10  "acl": [
11    "phs000178"
12  ],
13  "access": "controlled",
14  "state": "live",
15  "version": "1",
16  "file_id": "874e71e0-83dd-4d3e-8014-10141b49f12c",
17  "data_category": "Simple Nucleotide Variation",
18  "file_size": 122293,
19  "submitter_id": "TCGA-V4-A9EZ-01A-11D-A39W-08_TCGA-V4-A9EZ-10A-01D-A39Z-08_mutect",
20  "type": "simple_somatic_mutation",
21  "experimental_strategy": "WXS"
22 },
23 "warnings": {}
24 }

```

Note: The `file_size` field associated with each file is reported in bytes.

Example of retrieving file version information:

The `https://api.gdc.cancer.gov/files/versions` endpoint enables search and retrieval of version information about a file. A file may be versioned if a file is updated by the GDC (e.g. using a new alignment algorithm or fixing a file that contained an error). Version refers to the instance of a particular file. Inputs can either be a list of UUIDs as shown in example 1 or a download manifest as shown in example 2. Output includes information about the current and latest version for any given file. While `/files` also returns information about a file version this endpoint will only work for the most recent version of a file whereas `/files/versions` will work for all previous and current versions of a file. In both examples below the output format can be modified by adding the `format=tsv` parameter.

```

1 curl
   'https://api.gdc.cancer.gov/files/versions/1dd28069-5777-4ff9-bd2b-d1ba68e88b06,2a03abac-f1a2-49a9-a57c-7543739dd862'

```

```

1 [
2   {
3     "latest_size": 332092,
4     "latest_id": "1dd28069-5777-4ff9-bd2b-d1ba68e88b06",
5     "latest_version": "1",
6     "filename": "1dd28069-5777-4ff9-bd2b-d1ba68e88b06.vcf.gz",
7     "state": "validated",
8     "version": "1",
9     "latest_filename": "1dd28069-5777-4ff9-bd2b-d1ba68e88b06.vcf.gz",
10    "latest_release": [
11      "13.0"
12    ],
13    "latest_state": "validated",
14    "release": "13.0",
15    "latest_md5": "c2f9b196e154906a70c7ec46492a859d",
16    "size": 332092,
17    "id": "1dd28069-5777-4ff9-bd2b-d1ba68e88b06",
18    "md5": "c2f9b196e154906a70c7ec46492a859d"
19  },
20  {
21    "latest_size": 6653119038,
22    "latest_id": "2a03abac-f1a2-49a9-a57c-7543739dd862",
23    "latest_version": "1",

```

```

24 "filename": "a5d86cde-32ca-4ed6-b1a5-5a47575f2ac6_gdc_realn_rehead.bam",
25 "state": "validated",
26 "version": "1",
27 "latest_filename": "a5d86cde-32ca-4ed6-b1a5-5a47575f2ac6_gdc_realn_rehead.bam",
28 "latest_release": [
29   "13.0"
30 ],
31 "latest_state": "validated",
32 "release": "13.0",
33 "latest_md5": "48686fcd84ac713d44261ca9e26b89fb",
34 "size": 6653119038,
35 "id": "2a03abac-f1a2-49a9-a57c-7543739dd862",
36 "md5": "48686fcd84ac713d44261ca9e26b89fb"
37 }
38 ]

```

```

1 curl --request POST --header "Content-Type: text/tsv"
  https://api.gdc.cancer.gov/files/versions/manifest?pretty=true --data-binary
  @gdc_manifest_20180809_154816.txt

```

```

1 [{
2   "latest_size": 44857,
3   "state": "validated",
4   "latest_version": "1",
5   "filename": "nationwidechildrens.org_clinical.TCGA-13-1500.xml",
6   "latest_id": "0b20e27c-9a09-4f15-923f-d5b4f185dc22",
7   "version": "1",
8   "latest_filename": "nationwidechildrens.org_clinical.TCGA-13-1500.xml",
9   "latest_release": [
10    "12.0"
11  ],
12  "latest_state": "validated",
13  "release": "12.0",
14  "latest_md5": "597aa4df24c4d544b6c25cbd8b25a33e",
15  "md5": "597aa4df24c4d544b6c25cbd8b25a33e",
16  "id": "0b20e27c-9a09-4f15-923f-d5b4f185dc22",
17  "size": 44857
18 },{
19  "latest_size": 27620,
20  "state": "validated",
21  "latest_version": "1",
22  "filename": "BUCKS_p_TCGA_272_273_N_GenomeWideSNP_6_G05_1320676.grch38.seg.v2.txt",
23  "latest_id": "3edc7084-013c-4493-8507-c00b0e9962d8",
24  "version": "1",
25  "latest_filename": "BUCKS_p_TCGA_272_273_N_GenomeWideSNP_6_G05_1320676.grch38.seg.v2.txt",
26  "latest_release": [
27    "12.0"
28  ],
29  "latest_state": "validated",
30  "release": "12.0",
31  "latest_md5": "35a18d990a05eedfaf96e753bee0b96d",
32  "md5": "35a18d990a05eedfaf96e753bee0b96d",
33  "id": "3edc7084-013c-4493-8507-c00b0e9962d8",
34  "size": 27620
35 },{
36  "latest_size": 2346,
37  "state": "validated",
38  "latest_version": "1",

```

```

39 "filename": "a22f5e32-b16e-458f-a412-7e438056ece6.vcf.gz",
40 "latest_id": "a22f5e32-b16e-458f-a412-7e438056ece6",
41 "version": "1",
42 "latest_filename": "a22f5e32-b16e-458f-a412-7e438056ece6.vcf.gz",
43 "latest_release": [
44   "12.0"
45 ],
46 "latest_state": "validated",
47 "release": "12.0",
48 "latest_md5": "68b2433b31679bbbc6681919a1b81762",
49 "md5": "68b2433b31679bbbc6681919a1b81762",
50 "id": "a22f5e32-b16e-458f-a412-7e438056ece6",
51 "size": 2346
52 },{
53   "latest_size": 35411,
54   "state": "validated",
55   "latest_version": "1",
56   "filename": "CYANS_p_TCGAb_422_423_424_NSP_GenomeWideSNP_6_G12_1513758.nocnv_grch38.seg.v2.txt",
57   "latest_id": "ac7d2078-bd6b-446e-b30a-d889da5624b6",
58   "version": "1",
59   "latest_filename": "CYANS_p_TCGAb_422_423_424_NSP_GenomeWideSNP_6_G12_1513758.nocnv_grch38.seg.v2.txt",
60   "latest_release": [
61     "12.0"
62   ],
63   "latest_state": "validated",
64   "release": "12.0",
65   "latest_md5": "6338826b620773062232830fad51ae64",
66   "md5": "6338826b620773062232830fad51ae64",
67   "id": "ac7d2078-bd6b-446e-b30a-d889da5624b6",
68   "size": 35411
69 }]

```

Cases Endpoint

The GDC Cases Endpoint <https://api.gdc.cancer.gov/cases> enables search and retrieval of information related to a specific case.

The `cases` endpoint is designed to retrieve the metadata associated with one or more cases, including all nested biospecimen entities. Filters can be applied to retrieve information for entire cases, but not for lower-level biospecimen entities. For example, a sample within a case cannot be used to query for aliquots that are associated only with that sample. All aliquots associated with the case would be retrieved.

Example

This example is a query for files contained in GDC. It returns case where submitter id is TCGA-BH-AOEA, using the `pretty` and `filters` parameters and the following filtering operators:

```
1 {"op":"and","content":[{"op":"in","content":{"field":"submitter_id","value":["TCGA-BH-AOEA"]}]}
```

Command:

```

1 curl
2   'https://api.gdc.cancer.gov/cases?filters=%7B%22op%22%3A%22and%22%2C%22content%22%3A%5B%7B%22op%22%3A%22in%2
3 {
4   {
5     "data": {

```

```

4     "hits": [
5         {
6             "updated_datetime": "2017-03-04T16:39:19.244769-06:00",
7             "submitter_analyte_ids": [
8                 "TCGA-BH-AOEA-01A-11R",
9                 "TCGA-BH-AOEA-10A-01W",
10                "TCGA-BH-AOEA-01A-11W",
11                "TCGA-BH-AOEA-01A-11D",
12                "TCGA-BH-AOEA-10A-01D"
13            ],
14            "analyte_ids": [
15                "fe678556-acf4-4bde-a95e-860bb0150a95",
16                "66ed0f86-5ca5-4dec-ba76-7ee4dcf31831",
17                "f19f408a-815f-43d9-8032-e9482b796371",
18                "69ddc092-88a0-4839-a2bb-9f1c9e760409",
19                "30cb470f-66d4-4085-8c30-83a42e8453d4"
20            ],
21            "submitter_id": "TCGA-BH-AOEA",
22            "case_id": "1f601832-eee3-48fb-acf5-80c4a454f26e",
23            "id": "1f601832-eee3-48fb-acf5-80c4a454f26e",
24            "disease_type": "Breast Invasive Carcinoma",
25            "sample_ids": [
26                "9a6c71a6-82cd-42b1-a93f-f569370848d6",
27                "7f791228-dd77-4ab0-8227-d784a4c7fea1"
28            ],
29            "portion_ids": [
30                "cb6086d1-3416-4310-b109-e8fa6e8b72d4",
31                "8629bf5a-cdaf-4f6a-90bb-27dd4a7565c5",
32                "ae4f5816-f97a-4605-9b05-9ab820467dee"
33            ],
34            "submitter_portion_ids": [
35                "TCGA-BH-AOEA-01A-21-A13C-20",
36                "TCGA-BH-AOEA-01A-11",
37                "TCGA-BH-AOEA-10A-01"
38            ],
39            "created_datetime": null,
40            "slide_ids": [
41                "90154ea1-6b76-4445-870e-d531d6fa1239",
42                "a0826f0d-986a-491b-8c6f-b34f8929f3ee"
43            ],
44            "state": "live",
45            "aliquot_ids": [
46                "eef9dce1-6ba6-432b-bbe2-53c7dbe64fe7",
47                "cde982b7-3b0a-49eb-8710-a599cb0e44c1",
48                "b1a3739d-d554-4202-b96f-f25a444e2042",
49                "97c64d6a-7dce-4d0f-9cb3-b3e4eb4719c5",
50                "561b8777-801a-49ed-a306-e7dafeb044b6",
51                "42d050e4-e8ee-4442-b9c0-0ee14706b138",
52                "ca71ca96-cbb7-4eab-9487-251dda34e107",
53                "cfbd5476-e83a-401d-9f9a-639c73a0e35b",
54                "edad5bd3-efe0-4c5f-b05c-2c0c2951c45a",
55                "262715e1-835c-4f16-8ee7-6900e26f7cf5",
56                "2beb34c4-d493-4a73-b21e-de77d43251ff",
57                "bc7fc6d-60a0-48b7-aa81-14c0dda72d76"
58            ],
59            "primary_site": "Breast",
60            "submitter_aliquot_ids": [
61                "TCGA-BH-AOEA-10A-01D-A113-01",

```



```

62         "TCGA-BH-AOEA-01A-11R-A115-07",
63         "TCGA-BH-AOEA-01A-11D-A10Y-09",
64         "TCGA-BH-AOEA-01A-11D-A314-09",
65         "TCGA-BH-AOEA-01A-11R-A114-13",
66         "TCGA-BH-AOEA-01A-11D-A111-01",
67         "TCGA-BH-AOEA-01A-11D-A112-05",
68         "TCGA-BH-AOEA-01A-11D-A10X-02",
69         "TCGA-BH-AOEA-10A-01D-A110-09",
70         "TCGA-BH-AOEA-10A-01W-A12U-09",
71         "TCGA-BH-AOEA-10A-01D-A10Z-02",
72         "TCGA-BH-AOEA-01A-11W-A12T-09"
73     ],
74     "submitter_sample_ids": [
75         "TCGA-BH-AOEA-10A",
76         "TCGA-BH-AOEA-01A"
77     ],
78     "submitter_slide_ids": [
79         "TCGA-BH-AOEA-01A-01-MSA",
80         "TCGA-BH-AOEA-01A-01-TSA"
81     ]
82 }
83 ],
84 "pagination": {
85     "count": 1,
86     "sort": "",
87     "from": 0,
88     "page": 1,
89     "total": 1,
90     "pages": 1,
91     "size": 10
92 }
93 },
94 "warnings": {}
95 }

```

Retrieval of case metadata using individual UUIDs:

The `cases` endpoint supports a simple query format that retrieves the metadata of a single case using its UUID:

```
1 curl 'https://api.gdc.cancer.gov/cases/1f601832-eee3-48fb-acf5-80c4a454f26e?pretty=true&expand=diagnoses'
```

```

1 {
2   "data": {
3     "diagnoses": [
4       {
5         "classification_of_tumor": "not reported",
6         "last_known_disease_status": "not reported",
7         "updated_datetime": "2016-05-16T10:59:16.740358-05:00",
8         "primary_diagnosis": "c50.9",
9         "submitter_id": "TCGA-BH-AOEA_diagnosis",
10        "tumor_stage": "stage iia",
11        "age_at_diagnosis": 26548.0,
12        "vital_status": "dead",
13        "morphology": "8500/3",
14        "days_to_death": 991.0,
15        "days_to_last_known_disease_status": null,
16        "days_to_last_follow_up": null,

```

```

17     "state": null,
18     "days_to_recurrence": null,
19     "diagnosis_id": "84654ad5-2a2c-5c3b-8340-ecac6a5550fe",
20     "tumor_grade": "not reported",
21     "tissue_or_organ_of_origin": "c50.9",
22     "days_to_birth": -26548.0,
23     "progression_or_recurrence": "not reported",
24     "prior_malignancy": "not reported",
25     "site_of_resection_or_biopsy": "c50.9",
26     "created_datetime": null
27 }
28 ],
29 "sample_ids": [
30     "7f791228-dd77-4ab0-8227-d784a4c7fea1",
31     "9a6c71a6-82cd-42b1-a93f-f569370848d6"
32 ],
33 "portion_ids": [
34     "cb6086d1-3416-4310-b109-e8fa6e8b72d4",
35     "8629bf5a-cdaf-4f6a-90bb-27dd4a7565c5",
36     "ae4f5816-f97a-4605-9b05-9ab820467dee"
37 ],
38 "submitter_portion_ids": [
39     "TCGA-BH-AOEA-01A-11",
40     "TCGA-BH-AOEA-01A-21-A13C-20",
41     "TCGA-BH-AOEA-10A-01"
42 ],
43 "created_datetime": null,
44 "submitter_aliquot_ids": [
45     "TCGA-BH-AOEA-01A-11R-A114-13",
46     "TCGA-BH-AOEA-01A-11D-A111-01",
47     "TCGA-BH-AOEA-01A-11W-A12T-09",
48     "TCGA-BH-AOEA-01A-11R-A114-13",
49     "TCGA-BH-AOEA-01A-11R-A115-07",
50     "TCGA-BH-AOEA-01A-11D-A111-01",
51     "TCGA-BH-AOEA-01A-11D-A314-09",
52     "TCGA-BH-AOEA-01A-11D-A112-05",
53     "TCGA-BH-AOEA-01A-11D-A10Y-09",
54     "TCGA-BH-AOEA-01A-11D-A10X-02",
55     "TCGA-BH-AOEA-01A-11W-A12T-09",
56     "TCGA-BH-AOEA-01A-11D-A10X-02",
57     "TCGA-BH-AOEA-01A-11D-A10Y-09",
58     "TCGA-BH-AOEA-01A-11D-A314-09",
59     "TCGA-BH-AOEA-01A-11R-A115-07",
60     "TCGA-BH-AOEA-01A-11D-A112-05",
61     "TCGA-BH-AOEA-10A-01D-A110-09",
62     "TCGA-BH-AOEA-10A-01D-A113-01",
63     "TCGA-BH-AOEA-10A-01W-A12U-09",
64     "TCGA-BH-AOEA-10A-01D-A10Z-02",
65     "TCGA-BH-AOEA-10A-01D-A113-01",
66     "TCGA-BH-AOEA-10A-01D-A110-09",
67     "TCGA-BH-AOEA-10A-01W-A12U-09",
68     "TCGA-BH-AOEA-10A-01D-A10Z-02"
69 ],
70 "updated_datetime": "2016-05-02T14:37:43.619198-05:00",
71 "submitter_analyte_ids": [
72     "TCGA-BH-AOEA-01A-11R",
73     "TCGA-BH-AOEA-01A-11D",
74     "TCGA-BH-AOEA-01A-11W",

```

```

75     "TCGA-BH-AOEA-10A-01W",
76     "TCGA-BH-AOEA-10A-01D"
77 ],
78 "analyte_ids": [
79     "30cb470f-66d4-4085-8c30-83a42e8453d4",
80     "66ed0f86-5ca5-4dec-ba76-7ee4dcf31831",
81     "f19f408a-815f-43d9-8032-e9482b796371",
82     "69ddc092-88a0-4839-a2bb-9f1c9e760409",
83     "fe678556-acf4-4bde-a95e-860bb0150a95"
84 ],
85 "submitter_id": "TCGA-BH-AOEA",
86 "case_id": "1f601832-eee3-48fb-acf5-80c4a454f26e",
87 "state": null,
88 "aliquot_ids": [
89     "bc7fc6d-60a0-48b7-aa81-14c0dda72d76",
90     "97c64d6a-7dce-4d0f-9cb3-b3e4eb4719c5",
91     "edad5bd3-efe0-4c5f-b05c-2c0c2951c45a",
92     "bc7fc6d-60a0-48b7-aa81-14c0dda72d76",
93     "ca71ca96-cbb7-4eab-9487-251dda34e107",
94     "97c64d6a-7dce-4d0f-9cb3-b3e4eb4719c5",
95     "eef9dce1-6ba6-432b-bbe2-53c7dbe64fe7",
96     "42d050e4-e8ee-4442-b9c0-0ee14706b138",
97     "561b8777-801a-49ed-a306-e7dafeb044b6",
98     "262715e1-835c-4f16-8ee7-6900e26f7cf5",
99     "edad5bd3-efe0-4c5f-b05c-2c0c2951c45a",
100    "262715e1-835c-4f16-8ee7-6900e26f7cf5",
101    "561b8777-801a-49ed-a306-e7dafeb044b6",
102    "eef9dce1-6ba6-432b-bbe2-53c7dbe64fe7",
103    "ca71ca96-cbb7-4eab-9487-251dda34e107",
104    "42d050e4-e8ee-4442-b9c0-0ee14706b138",
105    "cfbd5476-e83a-401d-9f9a-639c73a0e35b",
106    "2beb34c4-d493-4a73-b21e-de77d43251ff",
107    "b1a3739d-d554-4202-b96f-f25a444e2042",
108    "cde982b7-3b0a-49eb-8710-a599cb0e44c1",
109    "2beb34c4-d493-4a73-b21e-de77d43251ff",
110    "cfbd5476-e83a-401d-9f9a-639c73a0e35b",
111    "b1a3739d-d554-4202-b96f-f25a444e2042",
112    "cde982b7-3b0a-49eb-8710-a599cb0e44c1"
113 ],
114 "slide_ids": [
115     "90154ea1-6b76-4445-870e-d531d6fa1239",
116     "a0826f0d-986a-491b-8c6f-b34f8929f3ee"
117 ],
118 "submitter_sample_ids": [
119     "TCGA-BH-AOEA-01A",
120     "TCGA-BH-AOEA-10A"
121 ]
122 },
123 "warnings": {}
124 }

```

Annotations Endpoint

The GDC Annotation Endpoint <https://api.gdc.cancer.gov/annotations> enables search and retrieval of annotations stored in the GDC.

Example

This example is a query for any annotations **directly** associated with the following GDC entities:

- the case with UUID e0d36cc0-652c-4224-bb10-09d15c7bd8f1
- the sample with UUID 25ebc29a-7598-4ae4-ba7f-618d448882cc
- the aliquot with UUID fe660d7c-2746-4b50-ab93-b2ed99960553

The query uses the filters parameter to specify entity UUIDs. Code samples below include the bare and percent-encoded filter JSON.

```
1 {
2   "op": "in",
3   "content": {
4     "field": "entity_id",
5     "value": [
6       "e0d36cc0-652c-4224-bb10-09d15c7bd8f1",
7       "25ebc29a-7598-4ae4-ba7f-618d448882cc",
8       "fe660d7c-2746-4b50-ab93-b2ed99960553"
9     ]
10  }
11 }
```

```
1 %7B%22op%22%3A%22in%22%2C%22content%22%3A%7B%22field%22%3A%22entity_id%22%2C%22value%22%3A%5B%22e0d36cc0-652c-4224-bb10-09d15c7bd8f1%22%2C%2225ebc29a-7598-4ae4-ba7f-618d448882cc%22%2C%22fe660d7c-2746-4b50-ab93-b2ed99960553%22%5D%22%7D%22%7D
```

```
1 curl
   'https://api.gdc.cancer.gov/annotations?filters=%7B%22op%22%3A%22in%22%2C%22content%22%3A%7B%22field%22%3A%22entity_id%22%2C%22value%22%3A%5B%22e0d36cc0-652c-4224-bb10-09d15c7bd8f1%22%2C%2225ebc29a-7598-4ae4-ba7f-618d448882cc%22%2C%22fe660d7c-2746-4b50-ab93-b2ed99960553%22%5D%22%7D%22%7D'
```

```
1 {
2   "data": {
3     "hits": [
4       {
5         "category": "Item flagged DNU",
6         "status": "Approved",
7         "entity_id": "fe660d7c-2746-4b50-ab93-b2ed99960553",
8         "classification": "CenterNotification",
9         "entity_type": "aliquot",
10        "created_datetime": "2015-09-28T00:00:00",
11        "annotation_id": "5ddadefe-8b57-5ce2-b8b2-918d63d99a59",
12        "notes": "The aliquot failed Broad pipeline QC and not all files are suitable for use. Consult
13          the SDRF file to determine which files are usable.",
14        "updated_datetime": "2017-03-09T13:20:38.962182-06:00",
15        "submitter_id": "29087",
16        "state": "submitted",
17        "case_id": "41b59716-116f-4942-8b63-409870a87e26",
18        "case_submitter_id": "TCGA-DK-A3IM",
19        "entity_submitter_id": "TCGA-DK-A3IM-10A-01D-A20B-01",
20        "id": "5ddadefe-8b57-5ce2-b8b2-918d63d99a59"
21      },
22      {
23        "category": "Item is noncanonical",
24        "status": "Approved",
25        "entity_id": "25ebc29a-7598-4ae4-ba7f-618d448882cc",
26        "classification": "Notification",
27        "entity_type": "sample",
28        "created_datetime": "2012-07-12T00:00:00",
29        "annotation_id": "d6500f94-618f-5334-a810-ade76b887ec9",
30        "notes": "No Matching Normal",
31      }
32    ]
33  }
34 }
```

```

30     "updated_datetime": "2017-03-09T13:47:18.182075-06:00",
31     "submitter_id": "8009",
32     "state": "submitted",
33     "case_id": "bd114e05-5a97-41e2-a0d5-5d39a1e9d461",
34     "case_submitter_id": "TCGA-08-0514",
35     "entity_submitter_id": "TCGA-08-0514-01A",
36     "id": "d6500f94-618f-5334-a810-ade76b887ec9"
37 },
38 {
39     "category": "Prior malignancy",
40     "status": "Approved",
41     "entity_id": "e0d36cc0-652c-4224-bb10-09d15c7bd8f1",
42     "classification": "Notification",
43     "entity_type": "case",
44     "created_datetime": "2013-03-12T00:00:00",
45     "annotation_id": "33336cdf-2cf0-5af2-bb52-fecd3427f180",
46     "notes": "Patient had a prior lymphoma. Unknown radiation or systemic chemotherapy.",
47     "updated_datetime": "2017-03-09T12:11:31.786013-06:00",
48     "submitter_id": "15630",
49     "state": "submitted",
50     "case_id": "e0d36cc0-652c-4224-bb10-09d15c7bd8f1",
51     "case_submitter_id": "TCGA-FS-A1ZF",
52     "entity_submitter_id": "TCGA-FS-A1ZF",
53     "id": "33336cdf-2cf0-5af2-bb52-fecd3427f180"
54 }
55 ],
56 "pagination": {
57     "count": 3,
58     "sort": "",
59     "from": 0,
60     "page": 1,
61     "total": 3,
62     "pages": 1,
63     "size": 10
64 }
65 },
66 "warnings": {}
67 }

```

History Endpoint

The GDC History Endpoint <https://api.gdc.cancer.gov/history> enables search and retrieval of version and release information about a file. This endpoint will return the entire provenance of all versions of a file. A file may be versioned if a file is updated by the GDC (e.g. using a new alignment algorithm or fixing a file that contained an error). **Version** refers to the instance of a particular file. **Release** refers to which data release a file was part of. A file may be a part of many different data releases with no change in version number or content.

Example

This example is a query for versioning information associated with the follow with file 1dd28069-5777-4ff9-bd2b-d1ba68e88b06.

```
1 curl 'https://api.gdc.cancer.gov/history/1dd28069-5777-4ff9-bd2b-d1ba68e88b06'
```

```
1 [{"release_date": "2018-07-23", "version": "1", "uuid": "1dd28069-5777-4ff9-bd2b-d1ba68e88b06",
   "file_change": "released", "data_release": "13.0"}]
```

_mapping Endpoint

Each search and retrieval endpoint is equipped with a `_mapping` endpoint that provides information about available fields. For example, `files/_mapping` endpoint provides information about fields and field groups available at the `files` endpoint: https://api.gdc.cancer.gov/files/_mapping.

The high-level structure of a response to a `_mapping` query is as follows:

```
1 "_mapping": {}
2 , "defaults": []
3 , "expand": []
4 , "fields": []
5 , "multi": []
6 , "nested": []
```

Each part of the response is described below:

Part	Description
<code>_mapping</code>	All available fields and their descriptions. The endpoint-agnostic field names provided here are compatible with the <code>filters</code> parameter but are not always compatible with the <code>fields</code> parameter
<code>defaults</code>	The default set of fields included in the API response when the <code>fields</code> parameter is not used in the request
<code>expand</code>	Field group names for use with the <code>expand</code> parameter
<code>fields</code>	All available fields in an endpoint-specific format that is compatible with both the <code>filters</code> and <code>fields</code> parameters
<code>multi</code>	GDC internal use
<code>nested</code>	Nested fields

Example

```
1 curl 'https://api.gdc.cancer.gov/projects/_mapping'
```

```
1 {
2   ...
3
4   "_mapping": {
5     "projects.disease_type": {
6       "doc_type": "projects",
7       "field": "disease_type",
8       "type": "id"
9     },
10    "projects.name": {
11      "doc_type": "projects",
12      "field": "name",
13      "type": "id"
14    }
15  }
16
17  ...
18
19 }
```

Similar information can be obtained using the `fields` parameter; `fields` queries provide additional information in the response, such as the name of the Elastic Search document (`doc_type`), the field name and the type of value. A list of supported types (such as `string`, `long`, `float`, ...) can be obtained from [Elastic Search Documentation](#).

Request Parameters

The GDC API supports the following search & retrieval request parameters:

Parameter	Default	Description
filters	null	Specifies search parameters
format	JSON	Specifies the API response format: JSON, XML, or TSV
pretty	false	Returns response with indentations and line breaks in a human-readable format
fields	null	Specifies which fields to include in the response
expand	null	Returns multiple related fields
size	10	Specifies the number of results to return
from	0	Specifies the first record to return from a set of search results
sort	null	Specifies sorting for the search results
facets	null	Provides all existing values for a given field and the number of records having this value.

Filters: Specifying the Query

The `filters` parameter enables passing of complex search queries to the GDC API. The parameter carries a query in the form of a JSON object.

Query Format

A `filters` query consists of an operator (or a nested set of operators) with a set of `field` and `value` operands.

The following `filters` query operators are supported by the GDC API:

Operator	Description	Number of Operands	Logic example
=	equals (string or number)	one	gender = "female"
!=	does not equal (string or number)	one	project_id != "TARGET-AML"
<	less than (number)	one	age at diagnosis < 90y
<=	less than or equal (number)	one	age at diagnosis <= 17
>	greater than (number)	one	age at diagnosis > 50
>=	greater than or equal (number)	one	age at diagnosis >= 18
=			
is	is (missing)	one	gender is missing
not	not (missing)	one	race not missing
in	matches a string or number in (a list)	multiple	primary_site in [Brain, Lung]
exclude	does not match any strings or values in (a list)	multiple	experimental_strategy exclude [WXS, WGS, "Genotyping array"]
and	(operation1) and (operation2)	multiple	{primary_site in [Brain, Lung]} and {gender = "female"}
or	(operation1) or (operation2)	multiple	{project_id != "TARGET-AML"} or {age at diagnosis < 90y}

The `field` operand specifies a field that corresponds to a property defined in the GDC Data Dictionary. A list of supported fields is provided in Appendix A; the list can also be accessed programmatically at the `__mapping` endpoint.

The `value` operand specifies the search terms. Users can get a list of available values for a specific property by making a call to the appropriate API endpoint using the `facets` parameter, e.g. <https://api.gdc.cancer.gov/v0/cases?facets=demographic.gender&size>. See [Facets](#) for details.

A simple query with a single operator looks like this:

```
1 {
2   "op": "=",
3   "content": {
4     "field": "cases.demographic.gender",
5     "value": [
6       "male"
7     ]
8   }
9 }
```

A more complex query with multiple operators looks like this:

```
1 {
2   "op": "and",
3   "content": [
4     {
5       "op": "in",
6       "content": {
7         "field": "cases.submitter_id",
8         "value": [
9           "TCGA-CK-4948",
10          "TCGA-D1-A17N",
11          "TCGA-4V-A9QX",
12          "TCGA-4V-A9QM"
13        ]
14      }
15    },
16    {
17      "op": "=",
18      "content": {
19        "field": "files.data_type",
20        "value": "Gene Expression Quantification"
21      }
22    }
23  ]
24 }
```

Example: HTTP GET Request

This example requests `male` cases using HTTP GET.

The JSON object to be passed to the GDC API looks like:

```
1 {
2   "op": "=",
3   "content": {
4     "field": "cases.demographic.gender",
5     "value": [
6       "male"
7     ]
8   }
9 }
```


URL-encoding the above JSON object using [Percent-\(URL\)-encoding tool](#) results in the following string:

```
1 %7B%0D%0A++++%22op%22%3A+%22%3D%22%2C%0D%0A++++%22content%22%3A+%7B%0D%0A++++%22field%22%3A+%22cases.demogra
```

The above string can now be passed to the GDC API using the `filters` parameter:

```
1 curl
   'https://api.gdc.cancer.gov/cases?filters=%7B%22op%22%3A+%22%3D%22%2C%0D%0A++++%22content%22%3A+%7B%0D%0A
```

```
1 import requests
2 import json
3 cases_endpt = 'https://api.gdc.cancer.gov/cases'
4 filt = {"op": "=",
5         "content": {
6             "field": "cases.demographic.gender",
7             "value": ["male"]
8         }
9     }
10 params = {'filters': json.dumps(filt), 'sort': 'demographic.gender:asc'}
11 ## requests URL-encodes automatically
12 response = requests.get(cases_endpt, params = params)
13 print json.dumps(response.json(), indent=2)
```

```
1 {
2   "data": {
3     "hits": [
4       {
5         "sample_ids": [
6           "1d014bf1-95ae-42e3-ae39-97ff4841d8ca",
7           "6b685bfc-651b-48d1-8e68-32c8096ea205"
8         ],
9         "portion_ids": [
10          "c061217a-266a-496d-8a96-3489191afa87",
11          "0d3a6a58-0e00-4889-bc73-5ddb5a387738",
12          "e858ee92-0438-48e9-a70d-80ef2c0ad539"
13        ],
14        "submitter_portion_ids": [
15          "TCGA-66-2770-01A-21-2193-20",
16          "TCGA-66-2770-01A-01",
17          "TCGA-66-2770-11A-01"
18        ],
19        "created_datetime": null,
20        "submitter_aliquot_ids": [
21          "TCGA-66-2770-01A-01D-1522-08",
22          "TCGA-66-2770-01A-01D-0848-05",
23          "TCGA-66-2770-01A-01W-0879-09",
24          "TCGA-66-2770-11A-01W-0878-08",
25          "TCGA-66-2770-01A-01R-0849-01",
26          "TCGA-66-2770-01A-01W-0877-08",
27          "TCGA-66-2770-01A-01D-0846-06",
28          "TCGA-66-2770-11A-01W-0880-09",
29          "TCGA-66-2770-01A-01D-0964-09",
30          "TCGA-66-2770-11A-01D-0846-06",
31          "TCGA-66-2770-01A-01D-0845-04",
32          "TCGA-66-2770-01A-01W-0881-10",
33          "TCGA-66-2770-11A-01D-0963-08",
34          "TCGA-66-2770-11A-01D-0844-01",
35          "TCGA-66-2770-01A-01R-0851-07",
36          "TCGA-66-2770-11A-01W-0882-10",
```

```

37     "TCGA-66-2770-11A-01D-1522-08",
38     "TCGA-66-2770-01A-01T-1557-13",
39     "TCGA-66-2770-01A-01D-0847-02",
40     "TCGA-66-2770-01A-01D-0844-01",
41     "TCGA-66-2770-11A-01D-0847-02",
42     "TCGA-66-2770-11A-01D-0964-09",
43     "TCGA-66-2770-01A-01D-0963-08",
44     "TCGA-66-2770-01A-01R-0850-03",
45     "TCGA-66-2770-11A-01D-0845-04",
46     "TCGA-66-2770-01A-01T-0852-07"
47 ],
48 "updated_datetime": "2016-05-02T15:57:03.730994-05:00",
49 "submitter_analyte_ids": [
50     "TCGA-66-2770-01A-01D",
51     "TCGA-66-2770-11A-01W",
52     "TCGA-66-2770-01A-01T",
53     "TCGA-66-2770-01A-01W",
54     "TCGA-66-2770-01A-01R",
55     "TCGA-66-2770-11A-01D"
56 ],
57 "analyte_ids": [
58     "385807d3-78de-4558-8d93-702d93fc835a",
59     "247acc7a-b4f5-47e9-86da-5ea9b04ad444",
60     "151b8cb9-6b0a-4db9-9b0e-62aa501b35d9",
61     "e549aebd-4dda-4ea8-8ccf-56c03bc8b2be",
62     "631ad4eb-845a-4e70-96ad-4b40157218a8",
63     "9a75640e-09d4-42b7-8cb4-75d62b39e98a"
64 ],
65 "submitter_id": "TCGA-66-2770",
66 "case_id": "f1b357e4-d67a-42c9-b0b7-12f69fa3da58",
67 "state": null,
68 "aliquot_ids": [
69     "a2d10f8e-6b27-4df0-bd25-ac24992d0bb4",
70     "8c1c733a-abed-468f-b4d0-d1ac34ba6d8b",
71     "cad8d384-3b7a-4f70-89c2-5584ae75c5eb",
72     "42e774cf-3c4a-4efd-9665-378cb6b4afac",
73     "3755168b-f5da-422d-847a-566cb112a8d7",
74     "cae4d249-ba67-4316-8761-7e71e3813182",
75     "aa6e700c-ce01-4cc9-87de-8bf615a8aa1a",
76     "ad5c4069-e616-4ab4-9b03-b196f9189b20",
77     "07c26ea4-0584-4cb0-8e5a-d057b8fe6c14",
78     "f95c2cb5-d20a-4f1f-8f2a-95a2d37fbdc4",
79     "817bf327-e583-4704-b294-c3645dcc4adf",
80     "2246cb75-38bd-491f-b6ee-99f4781f2564",
81     "a81b9090-626d-492d-9baf-7fa3ef70111c",
82     "5cd6f026-894e-45f6-bc59-d6f056e63846",
83     "e417903d-ab76-44f0-aae9-3a91fa9a8d3c",
84     "1d809a56-31ca-49d8-a57b-e773236b24de",
85     "df60a743-ef4b-43ea-bc5a-4d75e8befb8a",
86     "871350e2-958f-401c-ae86-6bc880a01942",
87     "3dc4207d-5671-4c3d-b75a-d39ef69b564c",
88     "69b77cc0-d00a-4ea3-9b39-3e3019d9e292",
89     "3d035ee8-9523-4771-8738-c8a5a2f91403",
90     "775e46bd-e56f-40fa-9891-aaedc1d49395",
91     "d1c60049-922a-42d4-bd7e-8cf4ace47f05",
92     "5220a53f-f3fc-476c-aa72-65a038eb2fd8",
93     "b7e44e6e-ccf9-4b75-a258-159912ab51ca",
94     "42750622-28d7-4d32-9262-b139fe77bc01"

```

```

95 ],
96 "slide_ids": [
97     "a10196d2-7a81-4e1e-a9a7-62d123c30875",
98     "72edc1ba-916d-42a2-9f22-6254c6e54c5c",
99     "ff15eeb9-550e-4c78-90cc-a6cce8ccc3df",
100    "71ccfb52-169d-4176-94d6-fff5b75f853d"
101 ],
102 "submitter_sample_ids": [
103     "TCGA-66-2770-11A",
104     "TCGA-66-2770-01A"
105 ]
106 },
107 {
108     "sample_ids": [
109         "06889714-2a40-4248-98ee-f690b301e36a",
110         "9f43a0c6-ea19-4021-b0ed-026f33ce1c33"
111     ],
112     "portion_ids": [
113         "3a001d28-7cf9-4c61-b155-73938aebaa25",
114         "79554cfd-e853-481e-8e37-1e296034094e"
115     ],
116     "submitter_portion_ids": [
117         "TCGA-02-0075-01A-01",
118         "TCGA-02-0075-10A-01"
119     ],
120     "created_datetime": null,
121     "submitter_aliquot_ids": [
122         "TCGA-02-0075-01A-01W-0204-02",
123         "TCGA-02-0075-01A-01R-0194-03",
124         "TCGA-02-0075-01A-01D-0198-02",
125         "TCGA-02-0075-01A-01R-0202-01",
126         "TCGA-02-0075-10A-01W-0207-09",
127         "TCGA-02-0075-01A-01R-0676-04",
128         "TCGA-02-0075-10A-01D-0198-02",
129         "TCGA-02-0075-10A-01D-0197-06",
130         "TCGA-02-0075-10A-01D-0193-01",
131         "TCGA-02-0075-01A-01W-0207-09",
132         "TCGA-02-0075-01A-01W-0206-08",
133         "TCGA-02-0075-01A-01D-0193-01",
134         "TCGA-02-0075-10A-01W-0205-10",
135         "TCGA-02-0075-01A-01R-0201-02",
136         "TCGA-02-0075-10A-01W-0204-02",
137         "TCGA-02-0075-01A-01D-0199-05",
138         "TCGA-02-0075-10A-01W-0206-08",
139         "TCGA-02-0075-01A-01D-0196-04",
140         "TCGA-02-0075-01A-01T-0195-07",
141         "TCGA-02-0075-10A-01D-0196-04",
142         "TCGA-02-0075-01A-01D-0197-06",
143         "TCGA-02-0075-01A-01D-0888-01",
144         "TCGA-02-0075-01A-01R-0195-07",
145         "TCGA-02-0075-01A-01W-0205-10"
146     ],
147     "updated_datetime": "2016-05-02T15:00:01.972331-05:00",
148     "submitter_analyte_ids": [
149         "TCGA-02-0075-01A-01R",
150         "TCGA-02-0075-10A-01D",
151         "TCGA-02-0075-01A-01W",
152         "TCGA-02-0075-01A-01T",

```

```

153     "TCGA-02-0075-01A-01D",
154     "TCGA-02-0075-10A-01W"
155 ],
156 "analyte_ids": [
157     "fec22de0-a2b9-45df-9854-1ebe76cee84e",
158     "b4d11c50-61f1-4d4a-815f-1c0413018d7f",
159     "c48673d0-a38d-44e1-8cfd-e91cb23ea2d5",
160     "24f1852c-999a-4ea8-917c-fcfd683e2aca",
161     "aa431260-a0fc-4924-80ce-61cab8b5e83e",
162     "11f21140-d761-44ca-a9b2-b24099df3b15"
163 ],
164 "submitter_id": "TCGA-02-0075",
165 "case_id": "b196f82b-ef3f-4e05-99f7-da5df65e691e",
166 "state": null,
167 "aliquot_ids": [
168     "75531fe0-101e-4220-bd47-98892c90ee70",
169     "e5ea38d4-f47c-4c8a-8bab-13631e0a9a7b",
170     "d48b7c2c-daac-4496-af8f-1f45ca43f627",
171     "bbba08fc-2514-4e15-afb7-41eecc7e876f",
172     "0685b37f-a47c-4222-a846-bf9f3c000de3",
173     "683986da-3cee-446d-9b7a-83bef25815c9",
174     "e6ffdb20-a1be-4664-bcd3-cc7a4de6f40b",
175     "5d1f25c0-9e1a-41ad-9735-134f39dbf70e",
176     "528b40b9-246f-4ba3-8209-777136638e62",
177     "33131479-5d69-4262-a549-ba8864320f3b",
178     "5c7822fc-cf4f-4f62-8482-7c0ce1b7ab9a",
179     "b95e7659-e3a4-4e96-b98c-f67d26b85322",
180     "30c84aca-f9db-4e07-ac34-1a92b1652ca1",
181     "d5e3b5cc-06e0-4294-9d3c-8f3b63acae3d",
182     "b14b3d09-3a7f-41a6-81df-2757efa67906",
183     "513040e2-dc29-4e2c-86fb-57371eede17a",
184     "21c3be1b-7c1e-4864-99d1-486cfe5d8f1d",
185     "5e28e5dc-6dfa-44a9-8793-9134cb4cdda5",
186     "b8c25892-4773-428f-a02c-f930931268e8",
187     "266d5260-08e4-4cec-87f3-ca415bd98575",
188     "8859a3ae-f85d-4ef2-830b-80f42f98d53e",
189     "ac018a8c-a6e2-4291-a4bf-a330ae9c441e",
190     "4b022f7f-7549-4d97-9d41-4e5f2e9ec74c",
191     "caad3dfa-74a9-4ecc-95c1-86f6fbfd4ab5"
192 ],
193 "slide_ids": [
194     "39f547cd-5dc3-4bf4-99ea-073bb161c23c",
195     "5f096267-0cc2-4cc5-a206-7357159633d7"
196 ],
197 "submitter_sample_ids": [
198     "TCGA-02-0075-10A",
199     "TCGA-02-0075-01A"
200 ]
201 },
202 {
203     "sample_ids": [
204         "ba08195b-31cf-4bb1-a470-23740225c99d",
205         "929889c4-e474-4104-b69b-fac7e414a59e"
206     ],
207     "portion_ids": [
208         "48a36eb4-79fb-45e7-8bb1-0fa1d5fcda2c",
209         "1de5e67a-ac3f-4c18-92c4-27ba1868c7ac",
210         "e09fc5e7-e8d2-4bf9-b12b-17b22e0387e4"

```

```

211 ],
212 "submitter_portion_ids": [
213     "TCGA-EJ-A8FU-10A-01",
214     "TCGA-EJ-A8FU-01A-21-A43L-20",
215     "TCGA-EJ-A8FU-01A-11"
216 ],
217 "created_datetime": null,
218 "submitter_aliquot_ids": [
219     "TCGA-EJ-A8FU-01A-11R-A36B-13",
220     "TCGA-EJ-A8FU-01A-11R-A36G-07",
221     "TCGA-EJ-A8FU-01A-11D-A363-01",
222     "TCGA-EJ-A8FU-10A-01D-A361-01",
223     "TCGA-EJ-A8FU-10A-01D-A362-08",
224     "TCGA-EJ-A8FU-01A-11W-A447-08",
225     "TCGA-EJ-A8FU-01A-11D-A365-05",
226     "TCGA-EJ-A8FU-01A-11D-A364-08",
227     "TCGA-EJ-A8FU-10A-01W-A446-08"
228 ],
229 "updated_datetime": "2016-05-02T15:57:04.948573-05:00",
230 "submitter_analyte_ids": [
231     "TCGA-EJ-A8FU-01A-11W",
232     "TCGA-EJ-A8FU-01A-11D",
233     "TCGA-EJ-A8FU-01A-11R",
234     "TCGA-EJ-A8FU-10A-01W",
235     "TCGA-EJ-A8FU-10A-01D"
236 ],
237 "analyte_ids": [
238     "2d4e4925-6ac8-498f-882b-4bbf319f6b7b",
239     "8d09b982-1256-4674-b383-d6ca4b4bb3c8",
240     "c74495d9-63bf-4ac0-b10e-04b3b06103c1",
241     "b9884d98-af57-4901-8b9d-4fdbf73d2c5a",
242     "2f16ac02-13bf-44fd-bbd7-658c1c384928"
243 ],
244 "submitter_id": "TCGA-EJ-A8FU",
245 "case_id": "23e56e08-e11d-4e83-88a8-1254675b3af8",
246 "state": null,
247 "aliquot_ids": [
248     "e77da017-5dc6-4e32-9568-755e4ee9b533",
249     "c9b286d1-d500-4bb3-bb3d-5bf40b1b1265",
250     "b7867d52-7987-46d4-a595-0ff5b5375a58",
251     "5586ad35-94b7-459e-8982-8e7fb25697a1",
252     "162a63f7-594f-4669-a06d-b4899c7fe86a",
253     "b8b1ab44-ee6e-4ac5-9efd-d5bd07e67b9c",
254     "7adcdf73-3ad3-4da7-ab27-2888f1d4f53a",
255     "eb498e52-3eae-402f-8cac-ec930f8d938d",
256     "293f781c-c2c7-479b-b1a6-5f951a2c5e5a"
257 ],
258 "slide_ids": [
259     "454a95d5-d084-4f36-b1f1-32c6c23ab46e"
260 ],
261 "submitter_sample_ids": [
262     "TCGA-EJ-A8FU-01A",
263     "TCGA-EJ-A8FU-10A"
264 ]
265 },
266 {
267     "sample_ids": [
268         "d43f0112-fe59-4842-9fda-1189e5fb7248",

```

```

269     "213cbb5-c382-47a1-b936-bf40c2c99091"
270 ],
271 "portion_ids": [
272     "26441aae-22e5-4e69-b3f5-34ccde356c93",
273     "60d7a93c-0634-438e-a72a-ce63630bb890",
274     "246a8f01-7ef2-4737-a984-49aa0b41c089"
275 ],
276 "submitter_portion_ids": [
277     "TCGA-F2-6879-10A-01",
278     "TCGA-F2-6879-01A-21-A39M-20",
279     "TCGA-F2-6879-01A-11"
280 ],
281 "created_datetime": "2016-05-02T16:23:44.347995-05:00",
282 "submitter_aliquot_ids": [
283     "TCGA-F2-6879-01A-11R-2155-13",
284     "TCGA-F2-6879-10A-01D-2153-01",
285     "TCGA-F2-6879-10A-01D-2152-26",
286     "TCGA-F2-6879-01A-11D-2157-05",
287     "TCGA-F2-6879-10A-01D-2154-08",
288     "TCGA-F2-6879-01A-11D-A45X-08",
289     "TCGA-F2-6879-01A-11D-2154-08",
290     "TCGA-F2-6879-01A-11W-2179-08",
291     "TCGA-F2-6879-01A-11D-2153-01",
292     "TCGA-F2-6879-01A-11R-2156-07",
293     "TCGA-F2-6879-01A-11D-2152-26",
294     "TCGA-F2-6879-10A-01D-A45X-08",
295     "TCGA-F2-6879-10A-01W-2179-08",
296     "TCGA-F2-6879-01A-01D-YYYY-23"
297 ],
298 "updated_datetime": "2016-05-02T16:23:44.347995-05:00",
299 "submitter_analyte_ids": [
300     "TCGA-F2-6879-10A-01D",
301     "TCGA-F2-6879-01A-11R",
302     "TCGA-F2-6879-10A-01W",
303     "TCGA-F2-6879-01A-11W",
304     "TCGA-F2-6879-01A-11D"
305 ],
306 "analyte_ids": [
307     "e87dde8d-3bf5-42d8-9a77-620d5c4943e0",
308     "30ade77d-996b-4031-93ab-6b341d49eb0a",
309     "1d94bd70-6621-4a94-8102-d673663e6665",
310     "ea65d92e-1597-410d-84d8-abb2a6235b3e",
311     "79697034-1cec-4d92-8195-8a35258ab477"
312 ],
313 "submitter_id": "TCGA-F2-6879",
314 "case_id": "8d9bd437-8b4b-4da5-87ba-6b5790f05022",
315 "state": null,
316 "aliquot_ids": [
317     "e7533585-b062-4d74-b511-05dc806a1357",
318     "e107952a-cc2b-4410-b0f9-62e7115430a0",
319     "61f1c8b1-986a-485a-9d96-4e4285b6425a",
320     "c043e276-fece-4cb9-a848-a0b16e6099b6",
321     "e5d110e1-63ad-49ce-b9b7-22bbd7ef8a88",
322     "7accb08d-acdb-46bc-bf7f-b9f678193115",
323     "a52cd04b-41d6-40db-b050-00ef3a143f7e",
324     "207fcf5e-c422-4333-9ec2-5dab38d240c7",
325     "5ddd3f83-28a8-4b7f-9aec-203a3c2efbe5",
326     "ccd4dd70-c0e4-42cf-870e-33d1013b201a",

```

```

327     "e12314fe-f16a-4d85-95b4-e712ede450f6",
328     "695461e3-283c-4b5b-9325-6b2588b67fd8",
329     "8481be1e-0993-487d-8d73-b0eb72b304ee",
330     "d7200791-4f1c-418f-8744-91b793486d9f"
331 ],
332 "slide_ids": [
333     "bcbcc947-cab1-4400-aebc-1d9e251a3ce8",
334     "cae8d0b9-3605-40af-bf99-7c23df8110a9"
335 ],
336 "submitter_sample_ids": [
337     "TCGA-F2-6879-10A",
338     "TCGA-F2-6879-01A"
339 ]
340 },
341 {
342     "sample_ids": [
343         "3a66b5bd-7037-463c-9f8d-2ba3de9d5571",
344         "84f603d6-9f71-48fb-b2e3-190424407452"
345     ],
346     "portion_ids": [
347         "fe90de9f-8ee3-4d55-834f-a90538958cb7",
348         "7a0042fd-07f0-4894-adb0-03cebce8aa02"
349     ],
350     "submitter_portion_ids": [
351         "TCGA-VQ-A922-01A-11",
352         "TCGA-VQ-A922-10A-01"
353     ],
354     "created_datetime": "2016-05-02T16:26:23.121974-05:00",
355     "submitter_aliquot_ids": [
356         "TCGA-VQ-A922-10A-01D-A412-01",
357         "TCGA-VQ-A922-01A-11D-A40Z-01",
358         "TCGA-VQ-A922-10A-01D-A413-08",
359         "TCGA-VQ-A922-01A-01D-YYYY-23",
360         "TCGA-VQ-A922-01A-11R-A414-31",
361         "TCGA-VQ-A922-01A-11D-A410-08",
362         "TCGA-VQ-A922-01A-11R-A415-13",
363         "TCGA-VQ-A922-01A-11D-A411-05"
364     ],
365     "updated_datetime": "2016-05-02T16:26:23.121974-05:00",
366     "submitter_analyte_ids": [
367         "TCGA-VQ-A922-01A-11R",
368         "TCGA-VQ-A922-10A-01D",
369         "TCGA-VQ-A922-01A-11D"
370     ],
371     "analyte_ids": [
372         "15bec495-04c7-412b-ad69-26b1f9274ccf",
373         "26a24673-04a1-4837-b888-702b0578aef2",
374         "2c0ecd67-b9ff-4e60-8d2f-7744c79a13aa"
375     ],
376     "submitter_id": "TCGA-VQ-A922",
377     "case_id": "8bd783a3-d6c9-4c87-a2a1-09f903b9c7ca",
378     "state": null,
379     "aliquot_ids": [
380         "58a121b4-265c-44ae-b6a9-79d087ee8b34",
381         "76fbba49-0123-4524-89aa-a1818c5507cb",
382         "0b0805bb-edaa-400f-ae9f-effed3dbb605",
383         "3370d626-d572-4d13-9cd3-1823a5df3d34",
384         "60934993-a9df-4389-b64d-da6844ef22df",

```

```

385     "243f24ba-bb0f-44e0-bcb1-69a97b395981",
386     "6cae9f2a-1c6c-4645-98b6-20719aec1413",
387     "44d020d1-c516-4a15-94e8-bcf0cb9c2683"
388 ],
389 "slide_ids": [
390     "0ff02899-57f8-419e-8872-c6ede53f4d3c"
391 ],
392 "submitter_sample_ids": [
393     "TCGA-VQ-A922-10A",
394     "TCGA-VQ-A922-01A"
395 ]
396 },
397 {
398     "sample_ids": [
399         "5bb5bd60-cf47-413b-88fa-f14977e24035",
400         "82f2cf670-1646-4a28-9578-f7e5b2f426e5",
401         "3b87fed0-cfbd-4ee3-b71d-ab595853e836"
402     ],
403     "portion_ids": [
404         "18bf160e-702a-464a-9920-f115024b5484",
405         "10a9c093-009d-4bc0-a344-2afd3f0f9b9f",
406         "8ebd06e1-5eda-47ec-8888-61965ecf005e"
407     ],
408     "submitter_portion_ids": [
409         "TCGA-HU-8243-11A-01",
410         "TCGA-HU-8243-01A-11",
411         "TCGA-HU-8243-10A-01"
412     ],
413     "created_datetime": "2016-05-02T16:17:09.754748-05:00",
414     "submitter_aliquot_ids": [
415         "TCGA-HU-8243-01A-01D-YYYY-23",
416         "TCGA-HU-8243-01A-11D-2340-08",
417         "TCGA-HU-8243-01A-11D-2338-01",
418         "TCGA-HU-8243-01A-11D-2342-05",
419         "TCGA-HU-8243-11A-01D-2338-01",
420         "TCGA-HU-8243-11A-01D-2340-08",
421         "TCGA-HU-8243-10A-01D-2339-01",
422         "TCGA-HU-8243-01A-11R-2343-13",
423         "TCGA-HU-8243-10A-01D-2341-08"
424     ],
425     "updated_datetime": "2016-05-02T16:17:09.754748-05:00",
426     "submitter_analyte_ids": [
427         "TCGA-HU-8243-11A-01D",
428         "TCGA-HU-8243-10A-01D",
429         "TCGA-HU-8243-01A-11R",
430         "TCGA-HU-8243-01A-11D"
431     ],
432     "analyte_ids": [
433         "89c9094d-5cf6-4c7d-ad24-41b7ad9427cc",
434         "2c413e60-0122-426b-afb3-ae94810e2513",
435         "57d41760-0fed-49d2-8606-48231cb244ea",
436         "37ed51fd-b540-408e-8bd6-4447ae4aa84a"
437     ],
438     "submitter_id": "TCGA-HU-8243",
439     "case_id": "77a8eab6-f6a1-4739-9031-75ead40d68cb",
440     "state": null,
441     "aliquot_ids": [
442         "ace3edd6-14a9-42cc-84f3-6127237f2913",

```



```

443     "a711abd1-f1c2-4e42-8b66-79b4514ac1c4",
444     "6af7ba34-58f7-4472-8c7e-89fc91ad5ac1",
445     "558ff67a-a584-46f8-9089-8f4a08015294",
446     "71c0a224-5953-4b59-a49c-b7aa1e959f1e",
447     "a460c222-bcac-4959-961f-4dbd73e1ce13",
448     "6e5789d7-4988-457a-86eb-e618c7ab06eb",
449     "ff31f56b-398c-45ee-b122-f10027774527",
450     "9635cfd4-3d26-4fc6-846c-fd74d5b60098"
451 ],
452 "slide_ids": [
453     "60b7c6b8-594a-40c3-9341-a0902e4e6938",
454     "e55e00a0-2048-404a-b83a-f34106468694"
455 ],
456 "submitter_sample_ids": [
457     "TCGA-HU-8243-10A",
458     "TCGA-HU-8243-01A",
459     "TCGA-HU-8243-11A"
460 ]
461 },
462 {
463     "sample_ids": [
464         "2f5cc9c9-31a9-5eb3-952a-b21e7cef50ca",
465         "4f3f4fc8-4465-5230-83ec-c0ef6aceb2ea"
466     ],
467     "updated_datetime": "2016-05-25T19:12:45.610324-05:00",
468     "submitter_aliquot_ids": [
469         "TARGET-30-PAUXFZ-01A-01D",
470         "TARGET-30-PAUXFZ-10A-01D"
471     ],
472     "submitter_id": "TARGET-30-PAUXFZ",
473     "case_id": "a7ccef7c-14c0-5232-b647-58b4a54fb343",
474     "aliquot_ids": [
475         "9e1e30a8-7607-5b7e-b33c-9a6c5828d5fb",
476         "c56898f9-c394-516a-bdbb-bf32a5af9d3f"
477     ],
478     "submitter_sample_ids": [
479         "TARGET-30-PAUXFZ-01A",
480         "TARGET-30-PAUXFZ-10A"
481     ]
482 },
483 {
484     "sample_ids": [
485         "c1bcb8d1-e13d-4af4-93f4-02d5f7f616a2",
486         "52fcf737-cdcc-43ea-b33c-4018039b42dd"
487     ],
488     "portion_ids": [
489         "e0e97a05-656a-468e-8418-0d08c38e76ab",
490         "3e2a0eab-7d89-4f3c-9c0e-8942e53d3c45"
491     ],
492     "submitter_portion_ids": [
493         "TCGA-KK-A8I9-01A-11",
494         "TCGA-KK-A8I9-11A-11"
495     ],
496     "created_datetime": null,
497     "submitter_aliquot_ids": [
498         "TCGA-KK-A8I9-11A-11D-A361-01",
499         "TCGA-KK-A8I9-11A-11D-A362-08",
500         "TCGA-KK-A8I9-11A-11W-A446-08",

```

```

501     "TCGA-KK-A8I9-01A-11R-A36G-07",
502     "TCGA-KK-A8I9-11A-11D-A40C-01",
503     "TCGA-KK-A8I9-01A-11D-A363-01",
504     "TCGA-KK-A8I9-01A-11W-A447-08",
505     "TCGA-KK-A8I9-01A-11D-A365-05",
506     "TCGA-KK-A8I9-01A-11D-A364-08",
507     "TCGA-KK-A8I9-01A-11R-A36B-13"
508 ],
509 "updated_datetime": "2016-05-02T15:57:29.451686-05:00",
510 "submitter_analyte_ids": [
511     "TCGA-KK-A8I9-11A-11W",
512     "TCGA-KK-A8I9-01A-11R",
513     "TCGA-KK-A8I9-11A-11D",
514     "TCGA-KK-A8I9-01A-11W",
515     "TCGA-KK-A8I9-01A-11D"
516 ],
517 "analyte_ids": [
518     "ddec19cb-5e4c-4151-8b6d-741044abff1e",
519     "96c5b539-8eb7-4156-81d0-7b7fec68900",
520     "ced38a45-7610-49d4-8bf9-d53a1fc2d489",
521     "476f5deb-1b3f-4a35-8a31-f27763ba8d8a",
522     "c284f2af-1e9b-40cc-8936-b61cfd251d62"
523 ],
524 "submitter_id": "TCGA-KK-A8I9",
525 "case_id": "261c3d74-706e-4751-bd15-8f3c1a402ff0",
526 "state": null,
527 "aliquot_ids": [
528     "4f76de2d-e07a-402b-9818-7f04d3704a43",
529     "96802a73-b1db-47d7-8f5f-4504f3ece5ad",
530     "f376fc45-370a-4d96-833b-9a1322e32a42",
531     "d3e88dd3-66d7-40d4-978a-4ddab868373a",
532     "06f1d087-75c9-4da8-8339-80aff3bfaa12",
533     "50b1e243-b45a-42a1-8692-b7ae5d51250f",
534     "0f1c00d3-f3dc-4d2b-bd8a-ecc31e4f4089",
535     "986a3ed6-ba56-4025-a2bd-9909648e703a",
536     "bebc84b6-9179-420b-8207-858b999e8c0c",
537     "239d5e7e-5fb5-4df3-ae6b-a5a06ee296ae"
538 ],
539 "slide_ids": [
540     "1e174ca5-9298-41b6-a705-728f111a3e7b",
541     "a3e31324-9e06-4799-85b4-4f6236848009"
542 ],
543 "submitter_sample_ids": [
544     "TCGA-KK-A8I9-11A",
545     "TCGA-KK-A8I9-01A"
546 ]
547 },
548 {
549     "sample_ids": [
550         "d43f727a-96d6-40b8-86ae-7a3e0aa46853",
551         "b8329a6d-a87b-47f4-ad00-9e979e62647b"
552     ],
553     "portion_ids": [
554         "8960ddcc-0950-4d6e-a557-8727b652c93b",
555         "e36bfd07-c911-4a98-8424-e58e5e9aaa68"
556     ],
557     "submitter_portion_ids": [
558         "TCGA-QR-A70H-10A-01",

```

```

559     "TCGA-QR-A70H-01A-12"
560 ],
561 "created_datetime": null,
562 "submitter_aliquot_ids": [
563     "TCGA-QR-A70H-01A-12R-A35K-07",
564     "TCGA-QR-A70H-01A-12R-A35M-13",
565     "TCGA-QR-A70H-01A-12D-A35E-05",
566     "TCGA-QR-A70H-10A-01D-A35A-01",
567     "TCGA-QR-A70H-01A-12D-A35C-01",
568     "TCGA-QR-A70H-01A-12W-A43Z-08",
569     "TCGA-QR-A70H-10A-01D-A35B-08",
570     "TCGA-QR-A70H-10A-01W-A441-08",
571     "TCGA-QR-A70H-01A-12D-A35D-08"
572 ],
573 "updated_datetime": "2016-05-02T15:37:31.996088-05:00",
574 "submitter_analyte_ids": [
575     "TCGA-QR-A70H-10A-01D",
576     "TCGA-QR-A70H-10A-01W",
577     "TCGA-QR-A70H-01A-12D",
578     "TCGA-QR-A70H-01A-12W",
579     "TCGA-QR-A70H-01A-12R"
580 ],
581 "analyte_ids": [
582     "c4a41555-dd45-4e10-a3be-50d49a1121a3",
583     "957e01f6-eb3f-446e-9f45-b50c66337e2d",
584     "1acde950-2e0c-4586-852b-b4ac4e1ea4a4",
585     "67c033c0-9fe8-4004-967e-c605e1890f4d",
586     "b0873010-5d60-4691-b700-e172950f1d7c"
587 ],
588 "submitter_id": "TCGA-QR-A70H",
589 "case_id": "13b41b15-a785-4ab7-b864-ffff6d35dd45",
590 "state": null,
591 "aliquot_ids": [
592     "d9120f00-7f10-49d5-ae84-6177e9424c7c",
593     "31c6fa50-200a-46c1-a546-61b52592fd8f",
594     "ab50f38c-2e7d-4d75-a216-27aeaa4d9305",
595     "382d5e31-6c66-4df3-a695-6b8c29cfc681",
596     "51d1fb14-c918-4439-b816-ef6cd3253c64",
597     "f586d8d5-d0c6-4979-aaa7-10217a88fa4c",
598     "2f9a60eb-602e-44bb-bc57-87e20d946f76",
599     "fbafc85e-defff-46cd-a40f-479b9dc92a60",
600     "cacbc8a6-0eb0-4277-931f-d0075c9b1de9"
601 ],
602 "slide_ids": [
603     "2310e34c-0ea5-4876-9f87-bad0b7a44513"
604 ],
605 "submitter_sample_ids": [
606     "TCGA-QR-A70H-01A",
607     "TCGA-QR-A70H-10A"
608 ]
609 },
610 {
611     "sample_ids": [
612         "19dee039-9c98-4d4a-8baf-eea1b6dda8eb",
613         "fdf1e501-f34f-450c-9a5c-611157079a86"
614     ],
615     "portion_ids": [
616         "10b6ccb4-3637-4769-8988-417c0306eaeef",

```

```

617     "92f8cd48-451d-4ed6-8e60-b15aa93d2c09",
618     "d0d55efa-c91d-45de-92bf-cf6f0d263b21"
619 ],
620 "submitter_portion_ids": [
621     "TCGA-BJ-A18Z-01A-21",
622     "TCGA-BJ-A18Z-01A-11-A21L-20",
623     "TCGA-BJ-A18Z-10A-01"
624 ],
625 "created_datetime": null,
626 "submitter_aliquot_ids": [
627     "TCGA-BJ-A18Z-01A-21D-A13U-02",
628     "TCGA-BJ-A18Z-10A-01D-A13V-01",
629     "TCGA-BJ-A18Z-01A-21R-A13Y-07",
630     "TCGA-BJ-A18Z-01A-21W-A14T-08",
631     "TCGA-BJ-A18Z-01A-21D-A13Z-05",
632     "TCGA-BJ-A18Z-01A-21D-A37T-08",
633     "TCGA-BJ-A18Z-10A-01D-A13W-08",
634     "TCGA-BJ-A18Z-01A-21R-A13X-13",
635     "TCGA-BJ-A18Z-01A-21D-A13W-08",
636     "TCGA-BJ-A18Z-10A-01D-A13U-02",
637     "TCGA-BJ-A18Z-10A-01W-A14T-08",
638     "TCGA-BJ-A18Z-01A-21D-A13V-01"
639 ],
640 "updated_datetime": "2016-05-02T16:18:19.199189-05:00",
641 "submitter_analyte_ids": [
642     "TCGA-BJ-A18Z-01A-21W",
643     "TCGA-BJ-A18Z-01A-21D",
644     "TCGA-BJ-A18Z-01A-21R",
645     "TCGA-BJ-A18Z-10A-01D",
646     "TCGA-BJ-A18Z-10A-01W"
647 ],
648 "analyte_ids": [
649     "119ebfa1-75b2-4f24-816a-4e9a5061f6b5",
650     "f86759fd-ecc5-4f42-b5fe-b9f079d23968",
651     "39691042-bd28-40ed-b66b-26414ecf1ba0",
652     "76ea5056-d7fa-49fb-94bf-11171ca7c100",
653     "71a822c9-b510-4a4c-8c30-18b8083acc2d"
654 ],
655 "submitter_id": "TCGA-BJ-A18Z",
656 "case_id": "0d497faf-2c1c-4173-a5fe-770cca73323c",
657 "state": null,
658 "aliquot_ids": [
659     "fa580596-e70f-4ed0-85a2-6fb594ca679a",
660     "776cb4b1-8efd-4ea2-b53f-9dff7dd94b10",
661     "85a7922f-0327-437c-bdf5-1bb67a1e932f",
662     "6d532180-0175-4610-8bfa-cca3a7c3697a",
663     "b5977e73-49d8-4e99-9e97-993cc44dad17",
664     "918793fa-b35e-4745-ac75-4d1c868089f8",
665     "ba9479a1-929f-4e4e-8bf5-e23cb280dfcf",
666     "e9776ff5-69b9-4669-ab33-e4bb030461ec",
667     "8ba98907-ab03-4c9e-a900-e31aa16ff810",
668     "35e18649-183e-4223-b2f6-d812bdd9becd",
669     "4aa17671-4420-4989-a6dd-379250f4aeda",
670     "815c53c3-8add-4612-b93c-3ed4bfa530aa"
671 ],
672 "slide_ids": [
673     "7c5b5c77-9fbc-4b48-81f5-48b5ede7c436"
674 ],

```

```

675     "submitter_sample_ids": [
676         "TCGA-BJ-A18Z-01A",
677         "TCGA-BJ-A18Z-10A"
678     ]
679 }
680 ],
681 "pagination": {
682     "count": 10,
683     "sort": "",
684     "from": 0,
685     "page": 1,
686     "total": 6340,
687     "pages": 634,
688     "size": 10
689 }
690 },
691 "warnings": {}
692 }

```

Example: HTTP POST Request

This example demonstrates how to obtain metadata in TSV format for a set of files using their UUIDs (e.g. UUIDs obtained from a [download manifest file generated by the GDC Data Portal](#)).

The first step is to construct a JSON query object, including **filters**, **fields**, **format**, and **size** parameters. The object is then submitted as HTTP POST payload to the GDC API using curl, in order to retrieve a TSV file with the requested metadata.

```

1 {
2     "filters":{
3         "op":"in",
4         "content":{
5             "field":"files.file_id",
6             "value":[
7                 "0001801b-54b0-4551-8d7a-d66fb59429bf",
8                 "002c67f2-ff52-4246-9d65-a3f69df6789e",
9                 "003143c8-bbbf-46b9-a96f-f58530f4bb82",
10                "0043d981-3c6b-463f-b512-ab1d076d3e62",
11                "004e2a2c-1acc-4873-9379-ef1aa12283b6",
12                "005239a8-2e63-4ff1-9cd4-714f81837a61",
13                "006b8839-31e5-4697-b912-8e3f4124dd15",
14                "006ce9a8-cf38-462e-bb99-7f08499244ab",
15                "007ce9b5-3268-441e-9ffd-b40d1127a319",
16                "0084a614-780b-42ec-b85f-7a1b83128cd3",
17                "00a5e471-a79f-4d56-8a4c-4847ac037400",
18                "00ab2b5a-b59e-4ec9-b297-76f74ffd3fb",
19                "00c5f14e-a398-4076-95d1-25f320ee3a37",
20                "00c74a8b-10aa-40cc-991e-3365ea1f3fce",
21                "00df5a50-bce3-4edf-a078-641e54800dcb"
22            ]
23        }
24    },
25    "format":"TSV",
26    "fields":"file_id,file_name,cases.submitter_id,cases.case_id,data_category,data_type,cases.samples.tumor_des
27    "size":"100"
28 }

```

```

1 curl --request POST --header "Content-Type: application/json" --data @Payload.txt
   'https://api.gdc.cancer.gov/files' > File_metadata.txt

```

1	cases_0_submitter_id	cases_0_case_id	data_type	cases_0_samples_0_sample_type	cases_0_samples_0_tissue_type	file_name	cases_0_samples_0_submitter_id	cases_0_samples_0_portions_0_analytes_0_aliquots_0_aliquot_id	cases_0_samples_0_sample_id	file_id	data_category	cases_0_samples_0_tumor_descriptor	cases_0_samples_0_portions_0_analytes_0_aliquots_0_submitter_id
2	TCGA-B0-5094	8aaa4e25-5c12-4ace-96dc-91aaa0c4457c	Aligned Reads	Solid Tissue Normal	C345.TCGA-B0-5094-11A-01D-1421-08.5_gdc_realn.bam	TCGA-B0-5094-11A	b4e4630a-b38c-4b62-b0e8-d73f0e3b4e47	7519d7a8-c3ee-417b-9cfc-111bc5ad0637	0001801b-54b0-4551-8d7a-d66fb59429bf	Raw Sequencing Data	TCGA-B0-5094-11A-01D-1421-08		
3	TCGA-B0-5117	ae55b2d3-62a1-419e-9f9a-5ddfac356db4	Aligned Reads	Solid Tissue Normal	C345.TCGA-B0-5117-11A-01D-1421-08.5_gdc_realn.bam	TCGA-B0-5117-11A	45c68b6b-0bed-424d-9a77-4f87bbaa3649	b1116541-bece-4df3-b3dd-cec50aeb277b	003143c8-bbbf-46b9-a96f-f58530f4bb82	Raw Sequencing Data	TCGA-B0-5117-11A-01D-1421-08		
4	TCGA-G7-6790	e7a1cbe2-793c-4747-8412-8be794f2382b	Aligned Reads	Blood Derived Normal	C489.TCGA-G7-6790-10A-01D-1962-08.2_gdc_realn.bam	TCGA-G7-6790-10A	66cbb40f-14b3-40c0-a332-e8a8e21bca11	4be83d0f-8b09-4e9e-8318-358371d34332	004e2a2c-1acc-4873-9379-ef1aa12283b6	Raw Sequencing Data	TCGA-G7-6790-10A-01D-1962-08		
5	TCGA-B9-A69E	a4225cb2-7b4b-4122-b6b9-629c26e3ea56	Aligned Reads	Blood Derived Normal	TCGA-B9-A69E-10A-01D-A31X-10_Illumina_gdc_realn.bam	TCGA-B9-A69E-10A	f4799bdc-b207-4053-9a4b-5a26ebf8ab91	5d6d6cd4-6a7b-499d-936a-1be9bf74b07f	0084a614-780b-42ec-b85f-7a1b83128cd3	Raw Sequencing Data	TCGA-B9-A69E-10A-01D-A31X-10		
6	TCGA-EE-A2GU	24faa36a-268d-4a13-b3ae-eacd431a2bcc	Aligned Reads	Blood Derived Normal	C828.TCGA-EE-A2GU-10A-01D-A198-08.2_gdc_realn.bam	TCGA-EE-A2GU-10A	c3feacc2-5a26-4bb2-a312-8b2ee53ccad1	cc4a5ed8-376a-4842-a25d-ffb07d8e1ca0	00c74a8b-10aa-40cc-991e-3365ea1f3fce	Raw Sequencing Data	TCGA-EE-A2GU-10A-01D-A198-08		
7	TCGA-CE-A484	e62a728d-390f-428a-bea1-fc8c9814fb11	Aligned Reads	Blood Derived Normal	C499.TCGA-CE-A484-10A-01D-A23U-08.3_gdc_realn.bam	TCGA-CE-A484-10A	641a0220-6eec-434a-b606-e256113b65da	27a8008e-044a-4966-b518-cc6905e292ca	00df5a50-bce3-4edf-a078-641e54800dcb	Raw Sequencing Data	TCGA-CE-A484-10A-01D-A23U-08		
8	TCGA-DA-A1IB	8fc9cc74-f388-49f0-b957-debb62638634	Aligned Reads	Blood Derived Normal	C828.TCGA-DA-A1IB-10A-01D-A198-08.2_gdc_realn.bam	TCGA-DA-A1IB-10A	30919a1a-df9f-4604-835e-f66ac7bcacdf	432952c5-6505-4220-a581-f65270a45281	00ab2b5a-b59e-4ec9-b297-76f74ff1d3fb	Raw Sequencing Data	TCGA-DA-A1IB-10A-01D-A198-08		
9	TCGA-AX-A2HG	7a2cf5ce-8317-4fff-946e-b9937afab815	Aligned Reads	Blood Derived Normal	6c2a8ea343da8d6cc0fd2043492f16df_gdc_realn.bam	TCGA-AX-A2HG-10A	8c34ffe2-9012-4b4a-b610-a42a9c6a9780	ef4b80ec-b453-48ec-8ad8-ccac83e1e4db	00c5f14e-a398-4076-95d1-25f320ee3a37	Raw Sequencing Data	TCGA-AX-A2HG-10A-01D-A17D-09		
10	TCGA-EC-A24G	b5c1e511-baf2-45b3-9919-110e8941e3c2	Aligned Reads	Blood Derived Normal	671333b193812fc2bd2744053b383459_gdc_realn.bam	TCGA-EC-A24G-10A	2a8cb8fe-b64f-453e-8139-7ede12f3fc51	61cf2e54-1b8d-40a0-9c73-a7449cbd570a	00a5e471-a79f-4d56-8a4c-4847ac037400	Raw Sequencing Data	TCGA-EC-A24G-10A-01D-A16D-09		
11	TCGA-B5-A0K0	29c8f468-5ac1-4d6c-8376-e36e6d246926	Aligned Reads	Blood Derived Normal	TCGA-B5-A0K0-10A-01W-A062-09_IlluminaGA-DNASeq_exome_gdc_realn.bam	TCGA-B5-A0K0-10A	02e65074-ffda-4795-b8f5-1bfd20bd1019	1df69e2e-f392-465f-8e61-4671ba2fcd35	007ce9b5-3268-441e-9ffd-b40d1127a319	Raw Sequencing Data	TCGA-B5-A0K0-10A-01W-A062-09		
12	TCGA-C8-A27B	f0d8a1fe-e313-44f1-99cc-b965cbeeff0e	Aligned Reads	Blood Derived Normal	3c99d98ea8eb6acb819e67fc77623d9_gdc_realn.bam	TCGA-C8-A27B-10A	922226ba-6244-4953-ad42-f4daa474c288	31139082-7978-45aa-9d8f-ac4789ac5cec	006b8839-31e5-4697-b912-8e3f4124dd15	Raw Sequencing Data	TCGA-C8-A27B-10A-01D-A167-09		
13	TCGA-E9-A295	fec0da58-1047-44d2-b6d1-c18cceed43dc	Aligned Reads	Blood Derived Normal	fd4421a6bbf3efd4e3d5c17fdd610314_gdc_realn.bam	TCGA-E9-A295-10A	cd761feb-9a20-4495-8943-c6243532a5cf	e74183e1-f0b4-412a-8dac-a62d404add78	002c67f2-ff52-4246-9d65-a3f69df6789e	Raw Sequencing Data	TCGA-E9-A295-10A-01D-A16D-09		
14	TCGA-EB-A440	c787c4da-c564-44f1-89eb-dd9da107acb1	Aligned Reads	Blood Derived Normal	C828.TCGA-EB-A440-10A-01D-A250-08.3_gdc_realn.bam	TCGA-EB-A440-10A	c723584a-c404-4c88-bfea-e40f5dbba542	5b738547-1825-4684-81bd-864bf2eb43ef	006ce9a8-cf38-462e-bb99-7f08499244ab	Raw Sequencing Data	TCGA-EB-A440-10A-01D-A250-08		
15	TCGA-A2-A3XX	53886143-c1c6-40e9-88e6-e4e5e0271fc8	Aligned Reads	Blood Derived Normal									

```

b40998d4778f18ed80d6dd8bff0eb761_gdc_realn.bam TCGA-A2-A3XX-10A
e96d5811-4736-40dd-966d-e0e172aeb0af c6eb6218-ad71-40a6-88b7-a4f1a015b816
0043d981-3c6b-463f-b512-ab1d076d3e62 Raw Sequencing Data TCGA-A2-A3XX-10A-01D-A23C-09
16 TCGA-EB-A3XB a9255dcb-b236-4777-ac43-555e3a5386c3 Aligned Reads Blood Derived Normal
C828.TCGA-EB-A3XB-10B-01D-A23B-08.1_gdc_realn.bam TCGA-EB-A3XB-10B
9f4ffc2f-d006-4d86-b3b1-b25020481893 0e1d4c7c-204d-4765-b090-68ed4cd83835
005239a8-2e63-4ff1-9cd4-714f81837a61 Raw Sequencing Data TCGA-EB-A3XB-10B-01D-A23B-08

```

Format

Specifies the format of the API response: JSON (default), TSV or XML.

Examples

```
1 curl 'https://api.gdc.cancer.gov/cases?fields=submitter_id&size=5&format=TSV'
```

```

1 import requests
2
3 cases_endpt = 'https://api.gdc.cancer.gov/cases'
4 params = {'fields': 'submitter_id',
5           'format': 'TSV'}
6 response = requests.get(cases_endpt, params = params)
7 print response.content

```

```

1 submitter_id
2 TCGA-RC-A6M6
3 TCGA-B6-AORV
4 TCGA-MB-A5Y8
5 TCGA-BQ-5876
6 TCGA-Z6-A9VB

```

```
1 curl 'https://api.gdc.cancer.gov/cases?fields=submitter_id&size=5&format=XML&pretty=true'
```

```

1 import requests
2
3 cases_endpt = 'https://api.gdc.cancer.gov/cases'
4 params = {'fields': 'submitter_id',
5           'format': 'XML',
6           'pretty': 'true'}
7 response = requests.get(cases_endpt, params = params)
8 print response.content

```

```

1 <?xml version="1.0" ?>
2 <response>
3   <data>
4     <hits>
5       <item>
6         <submitter_id>TCGA-MQ-A4LV</submitter_id>
7       </item>
8       <item>
9         <submitter_id>TCGA-N9-A4Q1</submitter_id>
10      </item>
11      <item>
12        <submitter_id>TCGA-78-7154</submitter_id>
13      </item>
14      <item>

```

```

15         <submitter_id>TCGA-S7-A7WX</submitter_id>
16     </item>
17     <item>
18         <submitter_id>TCGA-XF-AAML</submitter_id>
19     </item>
20 </hits>
21 <pagination>
22     <count>5</count>
23     <sort/>
24     <from>0</from>
25     <pages>2811</pages>
26     <total>14052</total>
27     <page>1</page>
28     <size>5</size>
29 </pagination>
30 </data>
31 <warnings/>
32 </response>

```

Pretty

Returns when the `pretty` parameter is set to `true`, the API response is formatted with additional whitespace to improve legibility.

Example

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

```
1 {"data": {"hits": [{"id": "f7af65fc-97e3-52ce-aa2c-b707650e747b", "submitter_id": "TARGET-00-NAAEMA"},
  {"id": "513d0a2a-3c94-5a36-97a4-24c3656fc66e", "submitter_id": "TARGET-00-NAAEMB"}, {"id":
  "b5f20676-727b-50b0-9b5a-582cd8572d6d", "submitter_id": "TARGET-00-NAAEMC"}, {"id":
  "0c0b183f-0d4a-5a9d-9888-0617cebcc462", "submitter_id": "TARGET-20-PABGKN"}, {"id":
  "0f5ed7a7-226d-57bc-a4ce-8a6b18560c55", "submitter_id": "TARGET-20-PABHET"}], "pagination": {"count":
  5, "sort": "submitter_id:asc", "from": 0, "page": 1, "total": 14551, "pages": 2911, "size": 5}},
  "warnings": {}}
```

```
1 curl 'https://api.gdc.cancer.gov/cases?fields=submitter_id&sort=submitter_id:asc&size=5&pretty=true'
```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "id": "f7af65fc-97e3-52ce-aa2c-b707650e747b",
6         "submitter_id": "TARGET-00-NAAEMA"
7       },
8       {
9         "id": "513d0a2a-3c94-5a36-97a4-24c3656fc66e",
10        "submitter_id": "TARGET-00-NAAEMB"
11      },
12      {
13        "id": "b5f20676-727b-50b0-9b5a-582cd8572d6d",
14        "submitter_id": "TARGET-00-NAAEMC"
15      },
16      {
17        "id": "0c0b183f-0d4a-5a9d-9888-0617cebcc462",
18        "submitter_id": "TARGET-20-PABGKN"
19      },
20      {

```



```

21     "id": "0f5ed7a7-226d-57bc-a4ce-8a6b18560c55",
22     "submitter_id": "TARGET-20-PABHET"
23   }
24 ],
25 "pagination": {
26   "count": 5,
27   "sort": "submitter_id:asc",
28   "from": 0,
29   "page": 1,
30   "total": 14551,
31   "pages": 2911,
32   "size": 5
33 }
34 },
35 "warnings": {}
36 }

```

Fields

This query parameter specifies which fields are to be included in the API response. The fields in the API response will be unordered. A listing of available fields for each endpoint is provided in Appendix A.

Example

The following example requests case submitter ID, file UUID, file name and file size from the `files` endpoint.

```
1 curl 'https://api.gdc.cancer.gov/files?fields=cases.submitter_id,file_id,file_name,file_size&pretty=true'
```

```

1 import requests
2 import json
3
4 files_endpt = 'https://api.gdc.cancer.gov/files'
5 params = {'fields': 'cases.submitter_id,file_id,file_name,file_size'}
6 response = requests.get(files_endpt, params = params)
7 print json.dumps(response.json(), indent=2)

```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "file_name": "NARKY_p_TCGAb69_SNP_N_GenomeWideSNP_6_H03_697832.grch38.seg.txt",
6         "cases": [
7           {
8             "submitter_id": "TCGA-BP-4989"
9           }
10        ],
11        "file_id": "3bd4d5dc-563a-481c-87a6-ec0017d0d58a",
12        "file_size": 54200
13      },
14      {
15        "file_name": "652ecf99-1af9-41fc-b0a5-d3e5c07a7b5d.FPKM.txt.gz",
16        "cases": [
17          {
18            "submitter_id": "TCGA-60-2709"
19          }
20        ],
21        "file_id": "b3286166-01f9-4149-81b5-a2ea5f27c50e",
22        "file_size": 530665

```

```

23 },
24 {
25   "file_name": "CUSKS_p_TCGAb47_SNP_1N_GenomeWideSNP_6_D05_628212.nocnv_grch38.seg.txt",
26   "cases": [
27     {
28       "submitter_id": "TCGA-A8-A07Z"
29     }
30   ],
31   "file_id": "282cc9d1-c5e9-49ff-b27b-e00c1e5529c6",
32   "file_size": 15806
33 },
34 {
35   "file_name": "REEDY_p_TCGAb65_SNP_N_GenomeWideSNP_6_F01_697686.nocnv_grch38.seg.txt",
36   "cases": [
37     {
38       "submitter_id": "TCGA-CJ-4871"
39     }
40   ],
41   "file_id": "fe44a644-eefc-42c5-aac7-a216bc1e88e1",
42   "file_size": 6179
43 },
44 {
45   "file_name": "84df7a8fee9fedb5e8e22849ec66d294_gdc_realn.bam",
46   "cases": [
47     {
48       "submitter_id": "TCGA-A2-A0CO"
49     }
50   ],
51   "file_id": "acd0ec73-c1fe-463e-912c-84e8416510e5",
52   "file_size": 15545555724
53 },
54 {
55   "file_name": "ed8c4bb6-891a-4cf2-80ba-42c5594760d0.vcf",
56   "cases": [
57     {
58       "submitter_id": "TCGA-BQ-7059"
59     }
60   ],
61   "file_id": "ed8c4bb6-891a-4cf2-80ba-42c5594760d0",
62   "file_size": 264694
63 },
64 {
65   "file_name": "nationwidechildrens.org_clinical.TCGA-IG-A6QS.xml",
66   "cases": [
67     {
68       "submitter_id": "TCGA-IG-A6QS"
69     }
70   ],
71   "file_id": "fe8cf009-f033-4536-95c7-836adcba5bf3",
72   "file_size": 36996
73 },
74 {
75   "file_name": "05f6f9f7-6fb7-4c95-b79c-fdfaba16539d.vep.reheader.vcf.gz",
76   "cases": [
77     {
78       "submitter_id": "TCGA-DK-A3IV"
79     }
80   ],

```

```

81     "file_id": "05f6f9f7-6fb7-4c95-b79c-fdfaba16539d",
82     "file_size": 415044
83   },
84   {
85     "file_name": "C484.TCGA-12-5301-01A-01D-1486-08.7_gdc_realn.bam",
86     "cases": [
87       {
88         "submitter_id": "TCGA-12-5301"
89       }
90     ],
91     "file_id": "3b0293c2-4a26-428c-b097-9489f23a2a2d",
92     "file_size": 23661175335
93   },
94   {
95     "file_name": "75a36e71-400d-46a5-93b0-7813cf0595ea.FPKM.txt.gz",
96     "cases": [
97       {
98         "submitter_id": "TCGA-BF-A5E0"
99       }
100    ],
101    "file_id": "28f763c7-8064-4151-ae0e-31e70cd9bfe8",
102    "file_size": 488422
103  }
104 ],
105 "pagination": {
106   "count": 10,
107   "sort": "",
108   "from": 0,
109   "page": 1,
110   "total": 216435,
111   "pages": 21644,
112   "size": 10
113 }
114 },
115 "warnings": {}
116 }

```

Expand

The `expand` parameter provides a shortcut to request multiple related fields (field groups) in the response. Instead of specifying each field using the `fields` parameter, users can specify a field group name using the `expand` parameter to request all fields in the group. Available field groups are listed in [Appendix A](#); the list can also be accessed programmatically at the `__mapping` endpoint. The `fields` and `expand` parameters can be used together to request custom combinations of field groups and individual fields.

Example

```

1 curl
   'https://api.gdc.cancer.gov/files/ac2ddebd-5e5e-4aea-a430-5a87c6d9c878?expand=cases.samples&pretty=true'

```

```

1 {
2   "data": {
3     "data_type": "Aligned Reads",
4     "updated_datetime": "2016-09-18T04:25:13.163601-05:00",
5     "created_datetime": "2016-05-26T18:55:53.506549-05:00",
6     "file_name": "000aa811c15656604161e8f0e3a0aae4_gdc_realn.bam",
7     "md5sum": "200475f5f6e42520204e5f6aadfe954f",

```

```

 8  "data_format": "BAM",
 9  "acl": [
10    "phs000178"
11  ],
12  "access": "controlled",
13  "platform": "Illumina",
14  "state": "submitted",
15  "file_id": "ac2ddebd-5e5e-4aea-a430-5a87c6d9c878",
16  "data_category": "Raw Sequencing Data",
17  "file_size": 12667634731,
18  "cases": [
19    {
20      "samples": [
21        {
22          "sample_type_id": "11",
23          "updated_datetime": "2016-09-08T11:00:45.021005-05:00",
24          "time_between_excision_and_freezing": null,
25          "oct_embedded": "false",
26          "tumor_code_id": null,
27          "submitter_id": "TCGA-QQ-A5VA-11A",
28          "intermediate_dimension": null,
29          "sample_id": "b4e7558d-898e-4d68-a897-381edde0bbcc",
30          "is_ffpe": false,
31          "pathology_report_uuid": null,
32          "created_datetime": null,
33          "tumor_descriptor": null,
34          "sample_type": "Solid Tissue Normal",
35          "state": null,
36          "current_weight": null,
37          "composition": null,
38          "time_between_clamping_and_freezing": null,
39          "shortest_dimension": null,
40          "tumor_code": null,
41          "tissue_type": null,
42          "days_to_sample_procurement": null,
43          "freezing_method": null,
44          "preservation_method": null,
45          "days_to_collection": 5980,
46          "initial_weight": 810.0,
47          "longest_dimension": null
48        }
49      ]
50    }
51  ],
52  "submitter_id": "32872121-d38a-4128-b96a-698a6f18f29d",
53  "type": "aligned_reads",
54  "file_state": "processed",
55  "experimental_strategy": "WXS"
56 },
57 "warnings": {}
58 }

```

Size and From

GDC API provides a pagination feature that limits the number of results returned by the API. It is implemented using `size` and `from` query parameters.

The `size` query parameter specifies the maximum number of results to return. Default `size` is 10. If the number of query results

is greater than `size`, only some of the results will be returned.

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 `cases` endpoint, then setting `from` to 11 will return results 12 to 20. The `from` parameter can be used in conjunction with the `size` parameter to return a specific subset of results.

Example

```
1 curl 'https://api.gdc.cancer.gov/files?fields=file_name&from=0&size=2&pretty=true'
```

```
1 import requests
2 import json
3
4 files_endpt = 'https://api.gdc.cancer.gov/files'
5 params = {'fields':'file_name',
6           'from':0, 'size':2}
7 response = requests.get(files_endpt, params = params)
8 print json.dumps(response.json(), indent=2)
```

```
1 {
2   "data": {
3     "hits": [
4       {
5         "file_name":
6           "unc.edu.276a1e00-cf3a-4463-a97b-d544381219ea.2363081.rsem.isoforms.normalized_results"
7       },
8       {
9         "file_name": "nationwidechildrens.org_clinical.TCGA-EY-A5W2.xml"
10      }
11     ],
12     "pagination": {
13       "count": 2,
14       "sort": "",
15       "from": 0,
16       "pages": 300936,
17       "total": 601872,
18       "page": 1,
19       "size": 2
20     }
21   },
22   "warnings": {}
23 }
```

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

```
1 import requests
2 import json
3
4 files_endpt = 'https://api.gdc.cancer.gov/files'
5 params = {'fields':'file_name',
6           'from':101, 'size':5}
7 response = requests.get(files_endpt, params = params)
8 print json.dumps(response.json(), indent=2)
```

```
1 {
2   "data": {
3     "hits": [
4       {
```

```

5     "file_name": "OCULI_p_TCGA_159_160_SNP_N_GenomeWideSNP_6_E09_831242.grch38.seg.txt",
6     "id": "1d959137-d8e6-4336-b357-8ab9c88eeca8"
7 },
8 {
9     "file_name":
10     "jhu-usc.edu_SKCM.HumanMethylation450.3.lvl-3.TCGA-EE-A3JI-06A-11D-A21B-05.gdc_hg38.txt",
11     "id": "9c02ec95-4aa3-4112-8823-c0fa87f71773"
12 },
13 {
14     "file_name":
15     "jhu-usc.edu_LAML.HumanMethylation450.2.lvl-3.TCGA-AB-3002-03A-01D-0742-05.gdc_hg38.txt",
16     "id": "731c3560-bcef-4ebf-bfbc-7320399a5bcb"
17 },
18 {
19     "file_name": "CUSKS_p_TCGAb47_SNP_1N_GenomeWideSNP_6_B03_628222.grch38.seg.txt",
20     "id": "a6f73a3e-faf8-49d9-9b68-77781bd302df"
21 },
22 {
23     "file_name": "5496e9f1-a383-4874-95bb-f4d1b33f4594.vcf",
24     "id": "5496e9f1-a383-4874-95bb-f4d1b33f4594"
25 }
26 ],
27 "pagination": {
28     "count": 5,
29     "sort": "",
30     "from": 101,
31     "page": 21,
32     "total": 274724,
33     "pages": 54945,
34     "size": 5
35 }
36 }

```

Sort

The `sort` query parameter sorts the results by a specific field, and with the sort direction specified using the `:asc` (ascending) or `:desc` (descending) prefix, e.g. `sort=field:desc`. A list of all valid *field* names is available in Appendix A; the list can also be accessed programmatically at the `__mapping` endpoint.

Example

Sort cases by `submitter_id` in ascending order:

```
1 curl 'https://api.gdc.cancer.gov/cases?fields=submitter_id&sort=submitter_id:asc&pretty=true'
```

```

1 import requests
2 import json
3
4 cases_endpt = 'https://api.gdc.cancer.gov/cases'
5 params = {'fields': 'submitter_id',
6           'sort': 'submitter_id:asc'}
7 response = requests.get(cases_endpt, params = params)
8 print json.dumps(response.json(), indent=2)

```

```

1 {
2   "data": {

```

```

3  "hits": [
4    {
5      "id": "f7af65fc-97e3-52ce-aa2c-b707650e747b",
6      "submitter_id": "TARGET-00-NAAEMA"
7    },
8    {
9      "id": "513d0a2a-3c94-5a36-97a4-24c3656fc66e",
10     "submitter_id": "TARGET-00-NAAEMB"
11    },
12    {
13     "id": "b5f20676-727b-50b0-9b5a-582cd8572d6d",
14     "submitter_id": "TARGET-00-NAAEMC"
15    },
16    {
17     "id": "0c0b183f-0d4a-5a9d-9888-0617cebcc462",
18     "submitter_id": "TARGET-20-PABGKN"
19    },
20    {
21     "id": "0f5ed7a7-226d-57bc-a4ce-8a6b18560c55",
22     "submitter_id": "TARGET-20-PABHET"
23    },
24    {
25     "id": "b2a560a4-5e52-5d78-90ef-d680fbaf44d0",
26     "submitter_id": "TARGET-20-PABHKY"
27    },
28    {
29     "id": "1e5c8323-383d-51a0-9199-1b9504b29c7e",
30     "submitter_id": "TARGET-20-PABLDZ"
31    },
32    {
33     "id": "c550a267-30bd-5bf3-9699-61341559e0d5",
34     "submitter_id": "TARGET-20-PACDZR"
35    },
36    {
37     "id": "0fe29a81-74fc-5158-ae13-0437bc272805",
38     "submitter_id": "TARGET-20-PACEGD"
39    },
40    {
41     "id": "dd2b23ec-46f4-56b2-9429-6015c6dc730f",
42     "submitter_id": "TARGET-20-PADDXZ"
43    }
44  ],
45  "pagination": {
46    "count": 10,
47    "sort": "submitter_id:asc",
48    "from": 0,
49    "page": 1,
50    "total": 14551,
51    "pages": 1456,
52    "size": 10
53  }
54 },
55 "warnings": {}
56 }

```

Facets

The `facets` parameter provides aggregate information for a specified field. It provides all values that exist for that field, and the number of entities (cases, projects, files, or annotations) that this value. The primary intended use of this parameter is for displaying aggregate information in the GDC Data Portal.

The `facets` parameter can be used in conjunction with the `filters` parameter to get aggregate information for a set of search results. The following limitations apply when using `facets` and `filters` together:

1. The `filters` object's top level operator must be `and`, and the internal filters must be limited to: `=`, `!=`, `in`, `exclude`, `is`, and `not`.
2. The information provided by `facets` for a given field will disregard any filters applied to that same field.

Example

This is an example of a request for a count of projects in each program.

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

```
1 import requests
2 import json
3
4 projects_endpt = 'https://api.gdc.cancer.gov/projects'
5 params = {'facets': 'program.name',
6          'from': 0, 'size': 0,
7          'sort': 'program.name:asc'}
8 response = requests.get(projects_endpt, params = params)
9 print json.dumps(response.json(), indent=2)
```

```
1 {
2   "data": {
3     "pagination": {
4       "count": 0,
5       "sort": "program.name:asc",
6       "from": 0,
7       "page": 1,
8       "total": 39,
9       "pages": 39,
10      "size": 0
11    },
12    "hits": [],
13    "aggregations": {
14      "program.name": {
15        "buckets": [
16          {
17            "key": "TCGA",
18            "doc_count": 33
19          },
20          {
21            "key": "TARGET",
22            "doc_count": 6
23          }
24        ]
25      }
26    }
27  },
28  "warnings": {}
29 }
```


Example

In this sample POST request, both `filters` and `facets` parameters are used. Note that `facets` ignores the `primary_site` filter.

```
1 {
2   "filters":{
3     "op":"and",
4     "content":[
5       {
6         "op":"=",
7         "content":{
8           "field":"cases.project.primary_site",
9           "value":"Kidney"
10        }
11      },
12      {
13        "op":"=",
14        "content":{
15          "field":"project.program.name",
16          "value":"TCGA"
17        }
18      }
19    ]
20  },
21  "size":"0",
22  "facets":"project.primary_site",
23  "pretty":"true"
24 }
```

```
1 curl --request POST --header "Content-Type: application/json" --data @Payload
   'https://api.gdc.cancer.gov/v0/cases'
```

```
1 {
2   "data": {
3     "pagination": {
4       "count": 0,
5       "sort": "",
6       "from": 0,
7       "page": 1,
8       "total": 941,
9       "pages": 941,
10      "size": 0
11    },
12    "hits": [],
13    "aggregations": {
14      "project.primary_site": {
15        "buckets": [
16          {
17            "key": "Brain",
18            "doc_count": 1133
19          },
20          {
21            "key": "Breast",
22            "doc_count": 1098
23          },
24          {
25            "key": "Lung",
26            "doc_count": 1089
27          }
28        ]
29      }
30    }
31  }
```

```
28     {
29         "key": "Kidney",
30         "doc_count": 941
31     },
32     {
33         "key": "Colorectal",
34         "doc_count": 635
35     },
36     {
37         "key": "Uterus",
38         "doc_count": 617
39     },
40     {
41         "key": "Ovary",
42         "doc_count": 608
43     },
44     {
45         "key": "Head and Neck",
46         "doc_count": 528
47     },
48     {
49         "key": "Thyroid",
50         "doc_count": 507
51     },
52     {
53         "key": "Prostate",
54         "doc_count": 500
55     },
56     {
57         "key": "Stomach",
58         "doc_count": 478
59     },
60     {
61         "key": "Skin",
62         "doc_count": 470
63     },
64     {
65         "key": "Bladder",
66         "doc_count": 412
67     },
68     {
69         "key": "Liver",
70         "doc_count": 377
71     },
72     {
73         "key": "Cervix",
74         "doc_count": 308
75     },
76     {
77         "key": "Adrenal Gland",
78         "doc_count": 271
79     },
80     {
81         "key": "Soft Tissue",
82         "doc_count": 261
83     },
84     {
85         "key": "Bone Marrow",
```



```

1 {
2   "size":"20000",
3   "pretty":"TRUE",
4   "fields":"submitter_id,disease_type",
5   "format":"TSV",
6   "filters":{
7     "op":"=",
8     "content":{
9       "field":"disease_type",
10      "value":"*Adenocarcinoma"
11    }
12  }
13 }

```

Quicksearch Endpoint

The GDC Portal has a quicksearch functionality that allows for a project, case, or file to be queried from a search box. This function calls the `/v0/all` endpoint, which retrieves the top cases, files, and projects that match to the query. The quicksearch can also be used programmatically through the API. For example, a search term of 'TCGA' would produce the following query:

```
1 curl "https://api.gdc.cancer.gov/v0/all?query=TCGA&size=5"
```

```

1 {
2   "data": {
3     "query": {
4       "hits": [
5         {
6           "disease_type": [
7             "Esophageal Carcinoma"
8           ],
9           "id": "UHJvamVjdDpUQ0dBLUVTQOE=",
10          "name": "Esophageal Carcinoma",
11          "primary_site": [
12            "Esophagus"
13          ],
14          "project_id": "TCGA-ESCA"
15        },
16        {
17          "disease_type": [
18            "Head and Neck Squamous Cell Carcinoma"
19          ],
20          "id": "UHJvamVjdDpUQ0dBLUxOUOM=",
21          "name": "Head and Neck Squamous Cell Carcinoma",
22          "primary_site": [
23            "Head and Neck"
24          ],
25          "project_id": "TCGA-HNSC"
26        },
27        {
28          "disease_type": [
29            "Liver Hepatocellular Carcinoma"
30          ],
31          "id": "UHJvamVjdDpUQ0dBLUxJSEM=",
32          "name": "Liver Hepatocellular Carcinoma",
33          "primary_site": [
34            "Liver"

```

```

35     ],
36     "project_id": "TCGA-LIHC"
37   },
38   {
39     "disease_type": [
40       "Colon Adenocarcinoma"
41     ],
42     "id": "UHJvamVjdDpUQ0dBLUNPQUQ=",
43     "name": "Colon Adenocarcinoma",
44     "primary_site": [
45       "Colorectal"
46     ],
47     "project_id": "TCGA-COAD"
48   },
49   {
50     "disease_type": [
51       "Adrenocortical Carcinoma"
52     ],
53     "id": "UHJvamVjdDpUQ0dBLUFDQw==",
54     "name": "Adrenocortical Carcinoma",
55     "primary_site": [
56       "Adrenal Gland"
57     ],
58     "project_id": "TCGA-ACC"
59   }
60 ]
61 }
62 }
63 }

```

This endpoint can be used to quickly retrieve information about a file. For example, if a user wanted to know the UUID for `nationwidechildrens.org_biospecimen.TCGA-EL-A4K1.xml`, the following query could be used to quickly retrieve it programmatically:

```
1 curl "https://api.gdc.cancer.gov/v0/all?query=nationwidechildrens.org_biospecimen.TCGA-EL-A4K1.xml&size=5"
```

```

1 {
2   "data": {
3     "query": {
4       "hits": [
5         {
6           "file_id": "2a7a354b-e497-4ae6-8a85-a170951596c1",
7           "file_name": "nationwidechildrens.org_biospecimen.TCGA-EL-A4K1.xml",
8           "id": "RmlsZToyYTdhMzU0Yi1lNDk3LTRhZTYtOGE4NS1hMTcwOTUxNTk2YzE=",
9           "submitter_id": null
10        }
11      ]
12    }
13  }
14 }

```

Additional Examples

More examples of API functionality described in this section are provided in Additional Examples.

Chapter 3

Downloading Files

Downloading Files

The GDC API implements file download functionality using `data` and `manifest` endpoints. The `data` endpoint allows users to download files stored in the GDC by specifying file UUID(s). The `manifest` endpoint generates a download manifest file that can be used with the GDC Data Transfer Tool to transfer large volumes of data.

Note: Downloading controlled access data requires the use of an authentication token. See [Getting Started: Authentication](#) for details.

Note: Requests to download data from the GDC Legacy Archive may be directed to `legacy/data` or `data`. See [Getting Started: Legacy Archive](#) for details.

Data endpoint

To download a file, users can pass UUID(s) to the `data` endpoint. If a single UUID is provided, the API will return the associated file. If a comma-separated list of UUIDs is provided, the API will return an archive file containing the requested files.

The `data` endpoint supports GET and POST requests as demonstrated in the following examples.

Downloading a Single File using GET

This example demonstrates downloading a single file from the GDC. Here we pass the file's UUID to the `data` endpoint with a GET request.

```
1 curl --remote-name --remote-header-name
   'https://api.gdc.cancer.gov/data/5b2974ad-f932-499b-90a3-93577a9f0573'
2
3 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
4          Dload  Upload   Total     Spent    Left     Speed
3 100 6111k  100 6111k    0     0   414k      0  0:00:14  0:00:14  --:--:--  412k
4 curl: Saved to filename '14-3-3_beta-R-V_GBL1112940.tif'
```

Related Files

If the `related_files=true` parameter is specified, the following related files, if available, will be included in the download package by the GDC API:

- BAM index files (BAI files)

- Metadata files (such as SRA XML or MAGE-TAB files)

For example, this request will download a legacy copy number segmentation file and its associated MAGE-TAB metadata file:

```
1 curl --remote-name --remote-header-name
   'https://api.gdc.cancer.gov/data/7efc039a-fde3-4bc1-9433-2fc6b5e3ffa5?related_files=true'

1 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
2                                     Dload  Upload  Total  Spent    Left  Speed
3 100 65353    0 65353    0    0 65353    0 --:--:-- --:--:-- --:--:-- 102k
4 curl: Saved to filename 'gdc_download_20180830_131817.826097.tar.gz'
```

Downloading Multiple Files using GET

This example demonstrates downloading multiple files from the GDC using a GET request. The GDC API returns a `.tar.gz` archive containing the downloaded files.

```
1 curl --remote-name --remote-header-name
   'https://api.gdc.cancer.gov/data/e3228020-1c54-4521-9182-1ea14c5dc0f7,18e1e38e-0f0a-4a0e-918f-08e6201ea140'

1 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
2                                     Dload  Upload  Total  Spent    Left  Speed
3 100 287k    0 287k    0    0 30131    0 --:--:-- 0:00:09 --:--:-- 42759
4 curl: Saved to filename 'gdc_download_064d1aa8cc8cbab33e93979bebbf7d6af2d6a802.tar.gz'
```

Note: This method supports downloading a limited number of files at one time. To download a large number of files, please use **POST**.

Downloading an Uncompressed Group of Files

If the `?tarfile` parameter is specified to a data endpoint download query all files requested in the download string will be bundled in a single tar file rather than a `tar.gz` file which is the default behavior.

```
1 curl --remote-name --remote-header-name
   'https://api.gdc.cancer.gov/data/1da7105a-f0ff-479d-9f82-6c1d94456c91,77e73cc4-ff31-449e-8e3c-7ae5ce57838c?tarfile'
```

Downloading Multiple Files using POST

The following two examples demonstrate downloading multiple files from the GDC using a POST request that contains a payload in one of two formats: percent-encoded form data or JSON. The GDC API returns a `.tar.gz` archive containing the downloaded files.

POST request with form data payload

POST requests that carry a payload of percent-encoded form data must include the HTTP header `Content-Type: application/x-www-form-urlencoded`.

The payload is a string in the following format:

```
1 ids=UUID1&ids=UUID2&ids=UUID3...
```

where UUID corresponds to the UUIDs of the files to be downloaded.

In this example we use `curl` to download a set of files from the GDC Legacy Archive. The payload is stored in a plain text file named `Payload`; `curl` includes the `Content-Type: application/x-www-form-urlencoded` header by default.

```
1 ids=556e5e3f-0ab9-4b6c-aa62-c42f6a6cf20c&ids=e0de63e2-02f3-4309-9b24-69f4c24e85fc&ids=f1a06178-2ec2-4b06-83f3-3a...
```

```

1 curl --remote-name --remote-header-name --request POST 'https://api.gdc.cancer.gov/data' --data @Payload
2
3 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
4 curl: Saved to filename 'gdc_download_20180830_132402.379282.tar.gz'

```

POST request with JSON payload

POST requests that carry a JSON payload must include the HTTP header `Content-Type: application/json`.

The payload is a string in the following format:

```

1 {
2   "ids": [
3     "UUID1",
4     "UUID2",
5     ...
6     "UUID3"
7   ]
8 }

```

where UUID corresponds to the UUIDs of the files to be downloaded.

In this example we use `curl` to download a set of files from the GDC Legacy Archive; the payload is stored in a plain text file named `Payload`.

```

1 {
2   "ids": [
3     "556e5e3f-0ab9-4b6c-aa62-c42f6a6cf20c",
4     "e0de63e2-02f3-4309-9b24-69f4c24e85fc",
5     "f1a06178-2ec2-4b06-83f3-3aedac332cfe",
6     "11a8aca0-c8e6-4ff8-8ab6-fe18a1b8ba82",
7     "69a69c84-00de-45ff-b397-fd2b6713ed4f",
8     "9ec48233-395d-401e-b205-951c971f8dd4",
9     "93129547-378c-4b69-b858-532abfff678e",
10    "8d4277e9-a472-4590-886d-24dc2538ea65",
11    "6733b412-56da-4f1c-a12b-ff804cb656d7",
12    "a72eec98-c5e0-4866-8953-765780acb6c1",
13    "e77b2294-1bdd-4fba-928a-d81d2622312f",
14    "965e01fc-318e-4c02-a801-d6fad60bfae4",
15    "21ad5409-fe0b-4728-97e4-15520b9fc287",
16    "1a777521-277c-4aeb-baf1-66871a7c2d2a",
17    "c13a3449-9e0d-45a9-bcc0-518f55e45c8a",
18    "5f2d329b-d59d-4112-b490-5114b830e34d",
19    "bb966617-6c1f-4bb0-a1ed-ceb37ecade67",
20    "05d11519-2b33-4742-aa87-3934632f2f2b",
21    "39bfafe2-9628-434e-bd72-148051a47477",
22    "481bea69-3cd5-45f3-8a52-2d4cc8fc8df7",
23    "f95e407b-de69-416c-920c-6be8c9414862",
24    "75940293-8fa6-47f9-ad5d-155b61933fdc",
25    "e8e84ccf-f8a8-4551-9257-ef731d02116f",
26    "e4991159-f088-4a2a-88b7-38d6ac47c6bc"
27   ]
28 }

```

```

1 curl --remote-name --remote-header-name --request POST --header 'Content-Type: application/json' --data
   @request.txt 'https://api.gdc.cancer.gov/data'

```



```

1  % Total    % Received % Xferd  Average Speed   Time    Time       Time  Current
2                                 Dload  Upload    Total   Spent    Left   Speed
3 100 2562k    0 2561k  100  1145    788k    352  0:00:03  0:00:03  --:--:--  788k
4 curl: Saved to filename 'gdc_download_20160701_011007.tar.gz'

```

Downloading Controlled-access Files

To download controlled-access files, a valid authentication token must be passed to the GDC API using the `X-Auth-Token` HTTP header:

```

1 token=$(cat gdc-token-text-file.txt)
2
3 curl --remote-name --remote-header-name --header "X-Auth-Token: $token"
   'https://api.gdc.cancer.gov/data/0eccf79d-1f1e-4205-910f-8e126b08276e'

```

```

1  % Total    % Received % Xferd  Average Speed   Time    Time       Time  Current
2                                 Dload  Upload    Total   Spent    Left   Speed
3 100 31.4M  100 31.4M    0     0   290k      0  0:01:50  0:01:50  --:--:--  172k
4 curl: Saved to filename
   'ACOLD_p_TCGA_Batch17_SNP_N_GenomeWideSNP_6_A03_466078.tangent.copynumber.data.txt'

```

Manifest endpoint

The `manifest` endpoint generates a download manifest file that can be used with the GDC Data Transfer Tool. The Data Transfer Tool is recommended for transferring large volumes of data. The GDC API can also generate a download manifest from a list of results that match a Search and Retrieval query. To do this, append `&return_type=manifest` to the end of the query.

Using the manifest endpoint

The `manifest` endpoint allows users to create a download manifest, which can be used with the GDC Data Transfer Tool to download a large volume of data. The `manifest` endpoint generates a manifest file from a comma-separated list of UUIDs.

```

1 curl --remote-name --remote-header-name
   'https://api.gdc.cancer.gov/v0/manifest/ae9db773-78ab-48d0-972d-debe1bedd37d,3d815e6e-db97-419d-ad7f-dba4e40
2
1  % Total    % Received % Xferd  Average Speed   Time    Time       Time  Current
2                                 Dload  Upload    Total   Spent    Left   Speed
3 100   274  100   274    0     0   1042      0  --:--:--  --:--:--  --:--:--  1041
4 curl: Saved to filename 'gdc_manifest_20160428_234614.txt'

```

The `manifest` endpoint also supports HTTP POST requests in the same format as the `data` endpoint; see [above](#) for details.

Using `return_type=manifest`

Alternatively, users can create a manifest by appending `&return_type=manifest` to a Search and Retrieval query. In this example, we generate a download manifest for RNA-seq data files from solid tissue normal samples, that are part of the TCGA-KIRC project:

```

1 curl --remote-name --remote-header-name
   'https://api.gdc.cancer.gov/files?filters=%7B%22op%22%3A%22and%22%2C%22content%22%3A%5B%7B%22op%22%3A%22%3D%
2
1  % Total    % Received % Xferd  Average Speed   Time    Time       Time  Current
2                                 Dload  Upload    Total   Spent    Left   Speed
3 100 40663    0 40663    0     0  77109      0  --:--:--  --:--:--  --:--:--  77306
4 curl: Saved to filename 'gdc_manifest.2016-06-28T13:26:33.850459.tsv'

```

Chapter 4

Data Analysis

Data Analysis

The GDC DAVE tools use the same API as the rest of the Data Portal and takes advantage of several new endpoints. Similar to the [GDC Data Portal Exploration](#) feature, the GDC data analysis endpoints allow API users to programmatically explore data in the GDC using advanced filters at a gene and mutation level. Survival analysis data is also available.

Endpoints

The following data analysis endpoints are available from the GDC API:

Node	Endpoint	Description
Genes	/genes	Allows users to access summary information about each gene using its Ensembl ID.
	/ssms	Allows users to access information about each somatic mutation. For example, a <code>ssm</code> would represent the transition of C to T at position 52000 of chromosome 1.
	/ssms/<ssm_id>	Get information about a specific ssm using a <code><ssm_id></code> , often supplemented with the <code>expand</code> option to show fields of interest.
	/ssm_occurrences	A <code>ssm</code> entity as applied to a single instance (case). An example of a <code>ssm occurrence</code> would be that the transition of C to T at position 52000 of chromosome 1 occurred in patient TCGA-XX-XXXX.
CNVS	/ssm_occurrences/<ssm_occurrences_id>	Get information about a specific ssm occurrence using a <code><ssm_occurrences_id></code> , often supplemented with the <code>expand</code> option to show fields of interest.
	/cnvs	Allows users to access data about copy number variations (cnvs). This data will be specific to cnvs and not a specific case.
	/cnvs/<cnv_id>	Get information about a specific copy number variation using a <code><cnv_id></code> , often supplemented with the <code>expand</code> option to show fields of interest.
	/cnvs/ids	This endpoint will retrieve nodes that contain the queried <code>cnv_id</code> . This is accomplished by adding the query parameter: <code>/cnvs/ids?query=<cnv_id></code> .
	/cnv_occurrences	A <code>cnv</code> entity as applied to a single case.
	/cnv_occurrences/<cnv_occurrence_id>	Get information about a specific copy number variation occurrence using a <code><cnv_occurrence_id></code> , often supplemented with the <code>expand</code> option to show fields of interest.

Node	Endpoint	Description
	<code>/cnv_occurrences/ids</code>	This endpoint will retrieve nodes that contain the queried <code>cnv_occurrence_id</code> . This is accomplished by adding the query parameter: <code>/cnv_occurrences/ids?query=<cnv_occurrences_id></code>
Analysis	<code>/analysis/top_cases_counts_by_genes</code>	Returns the number of cases with a mutation in each gene listed in the <code>gene_ids</code> parameter for each project. Note that this endpoint cannot be used with the <code>format</code> or <code>fields</code> parameters.
	<code>/analysis/top_mutated_genes_by_project</code>	Returns a list of genes that have the most mutations within a given project.
	<code>/analysis/top_mutated_cases_by_gene</code>	Generates information about the cases that are most affected by mutations in a given number of genes
	<code>/analysis/mutated_cases_count_by_project</code>	Returns counts for the number of cases that have associated <code>ssm</code> data in each project. The number of affected cases can be found under “ <code>case_with_ssm</code> ”: {“ <code>doc_count</code> ”: \$ <code>case_count</code> }.
	<code>/analysis/survival</code>	Survival plots can be generated in the Data Portal for different subsets of data, based upon many query factors such as variants, disease type and projects. This endpoint can be used to programmatically retrieve the raw data to generate these plots and apply different filters to the data. (see Survival Example)

The methods for retrieving information from these endpoints are very similar to those used for the `cases` and `files` endpoints. These methods are explored in depth in the [API Search and Retrieval](#) documentation. The `_mapping` parameter can also be used with each of these endpoints to generate a list of potential fields. For example:

```
https://api.gdc.cancer.gov/ssms/_mapping
```

While it is not an endpoint, the `observation` entity is featured in the visualization section of the API. The `observation` entity provides information from the MAF file, such as read depth and normal genotype, that supports the validity of the associated `ssm`. An example is demonstrated below:

```
1 curl
   https://api.gdc.cancer.gov/ssms/57bb3f2e-ec05-52c2-ab02-7065b7d24849?expand=occurrence.case.observation.read

1 {
2   "data": {
3     "ncbi_build": "GRCh38",
4     "occurrence": [
5       {
6         "case": {
7           "observation": [
8             {
9               "read_depth": {
10                "t_ref_count": 321,
11                "t_alt_count": 14,
12                "t_depth": 335,
13                "n_depth": 115
14              }
15            }
16          ]
17        }
18      }
19    ]
20  }
```

```

18     }
19   ],
20   "tumor_allele": "G",
21   "mutation_type": "Simple Somatic Mutation",
22   "end_position": 14304578,
23   "reference_allele": "C",
24   "ssm_id": "57bb3f2e-ec05-52c2-ab02-7065b7d24849",
25   "start_position": 14304578,
26   "mutation_subtype": "Single base substitution",
27   "cosmic_id": null,
28   "genomic_dna_change": "chr5:g.14304578C>G",
29   "gene_aa_change": [
30     "TRIO L229V",
31     "TRIO L437V",
32     "TRIO L447V",
33     "TRIO L496V"
34   ],
35   "chromosome": "chr5"
36 },
37 "warnings": {}
38 }

```

Genes Endpoint Examples

Example 1: A user would like to access information about the gene ZMPSTE24, which has an Ensembl gene ID of ENSG0000084073. This would be accomplished by appending ENSG0000084073 (gene_id) to the genes endpoint.

```
1 curl "https://api.gdc.cancer.gov/genes/ENSG0000084073?pretty=true"
```

```

1 {
2   "data": {
3     "canonical_transcript_length": 3108,
4     "description": "This gene encodes a member of the peptidase M48A family. The encoded protein is a
                    zinc metalloproteinase involved in the two step post-translational proteolytic cleavage of
                    carboxy terminal residues of farnesylated prelamin A to form mature lamin A. Mutations in this
                    gene have been associated with mandibuloacral dysplasia and restrictive dermopathy. [provided by
                    RefSeq, Jul 2008]",
5     "cytoband": [
6       "1p34.2"
7     ],
8     "gene_start": 40258107,
9     "canonical_transcript_length_genomic": 36078,
10    "gene_id": "ENSG0000084073",
11    "gene_strand": 1,
12    "canonical_transcript_length_cds": 1425,
13    "gene_chromosome": "1",
14    "synonyms": [
15      "FACE-1",
16      "HGPS",
17      "PRO1",
18      "STE24",
19      "Ste24p"
20    ],
21    "is_cancer_gene_census": null,
22    "biotype": "protein_coding",
23    "gene_end": 40294184,

```

```

24   "canonical_transcript_id": "ENST00000372759",
25   "symbol": "ZMPSTE24",
26   "name": "zinc metalloproteinase STE24"
27 },
28 "warnings": {}
29 }

```

Example 2: A user wants a subset of elements such as a list of coordinates for all genes on chromosome 7. The query can be filtered for only results from chromosome 7 using a JSON-formatted query that is URL-encoded.

```

1 curl
   "https://api.gdc.cancer.gov/genes?pretty=true&fields=gene_id,symbol,gene_start,gene_end&format=tsv&size=2000"

```

gene_start	gene_end	symbol	id
28995231	29195451	CPVL	ENSG00000106066
33014114	33062797	NT5C3A	ENSG00000122643
143052320	143053347	OR6V1	ENSG00000225781
100400826	100428992	ZCWPW1	ENSG00000078487
73861159	73865893	WBSCR28	ENSG00000175877
64862999	64864370	EEF1DP4	ENSG00000213640
159231435	159233377	PIP5K1P2	ENSG00000229435
141972631	141973773	TAS2R38	ENSG00000257138
16646131	16706523	BZW2	ENSG00000136261
149239651	149255609	ZNF212	ENSG00000170260
57405025	57405090	MIR3147	ENSG00000266168
130393771	130442433	CEP41	ENSG00000106477
150800403	150805120	TMEM176A	ENSG00000002933
93591573	93911265	GNGT1	ENSG00000127928
117465784	117715971	CFTR	ENSG00000001626
5879827	5886362	OCM	ENSG00000122543
144118461	144119360	OR2A15P	ENSG00000239981
30424527	30478784	NOD1	ENSG00000106100
137227341	137343865	PTN	ENSG00000105894
84876554	84876956	HMG2P11	ENSG00000232605
107470018	107475659	GPR22	ENSG00000172209
31330711	31330896	RP11-463M14.1	ENSG00000271027
78017057	79453574	MAGI2	ENSG00000187391
55736779	55739605	CICP11	ENSG00000237799
142111749	142222324	RP11-1220K2.2	ENSG00000257743

```

27 (truncated)

```

Simple Somatic Mutation Endpoint Examples

Example 1: Similar to the /genes endpoint, a user would like to retrieve information about the mutation based on its COSMIC ID. This would be accomplished by creating a JSON filter, which will then be encoded to URL for the curl command.

```

1 {
2   "op": "in",
3   "content": {
4     "field": "cosmic_id",
5     "value": [
6       "COSM4860838"
7     ]
8   }
9 }

```

```

1 curl
   'https://api.gdc.cancer.gov/ssms?pretty=true&filters=%7B%0A%22op%22%3A%22in%22%2C%0A%22content%22%3A%7B%0A%22

```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "ncbi_build": "GRCh38",
6         "mutation_type": "Simple Somatic Mutation",
7         "mutation_subtype": "Single base substitution",
8         "end_position": 62438203,
9         "reference_allele": "C",
10        "ssm_id": "8b3c1a7a-e4e0-5200-9d46-5767c2982145",
11        "start_position": 62438203,
12        "cosmic_id": [
13          "COSM4860838",
14          "COSM731764",
15          "COSM731765"
16        ],
17        "id": "8b3c1a7a-e4e0-5200-9d46-5767c2982145",
18        "tumor_allele": "T",
19        "gene_aa_change": [
20          "CADPS G1147G",
21          "CADPS G1187G",
22          "CADPS G1217G",
23          "CADPS G1226G",
24          "CADPS G127G",
25          "CADPS G218G",
26          "CADPS G95G"
27        ],
28        "chromosome": "chr3",
29        "genomic_dna_change": "chr3:g.62438203C>T"
30      }
31    ],
32    "pagination": {
33      "count": 1,
34      "sort": "",
35      "from": 0,
36      "page": 1,
37      "total": 1,
38      "pages": 1,
39      "size": 10
40    }
41  },
42  "warnings": {}
43 }

```

Example 2: Based on the previous example's `ssm_id` (8b3c1a7a-e4e0-5200-9d46-5767c2982145), a user would like to look at the consequences and the VEP impact due to this ssm.

```

1 curl
2   'https://api.gdc.cancer.gov/ssms/8b3c1a7a-e4e0-5200-9d46-5767c2982145?pretty=true&expand=consequence.transcr

1 {
2   "data": {
3     "consequence": [
4       {
5         "transcript": {
6           "aa_start": 127,
7           "consequence_type": "synonymous_variant",
8           "aa_end": 127,
9           "transcript_id": "ENST00000466621",

```

```

10     "is_canonical": false,
11     "aa_change": "G127G",
12     "annotation": {
13         "vep_impact": "LOW"
14     },
15     "ref_seq_accession": ""
16 }
17 },
18 {
19     "transcript": {
20         "aa_start": 95,
21         "consequence_type": "synonymous_variant",
22         "aa_end": 95,
23         "transcript_id": "ENST00000613879",
24         "is_canonical": false,
25         "aa_change": "G95G",
26         "annotation": {
27             "vep_impact": "LOW"
28         },
29         "ref_seq_accession": ""
30     }
31 },
32 {
33     "transcript": {
34         "aa_start": 218,
35         "consequence_type": "synonymous_variant",
36         "aa_end": 218,
37         "transcript_id": "ENST00000473635",
38         "is_canonical": false,
39         "aa_change": "G218G",
40         "annotation": {
41             "vep_impact": "LOW"
42         },
43         "ref_seq_accession": ""
44     }
45 },
46 {
47     "transcript": {
48         "aa_start": null,
49         "consequence_type": "non_coding_transcript_exon_variant",
50         "aa_end": null,
51         "transcript_id": "ENST00000474560",
52         "is_canonical": false,
53         "aa_change": null,
54         "annotation": {
55             "vep_impact": "MODIFIER"
56         },
57         "ref_seq_accession": ""
58     }
59 },
60 {
61     "transcript": {
62         "aa_start": 1226,
63         "consequence_type": "synonymous_variant",
64         "aa_end": 1226,
65         "transcript_id": "ENST00000383710",
66         "is_canonical": true,
67         "aa_change": "G1226G",

```

```

68     "annotation": {
69         "vep_impact": "LOW"
70     },
71     "ref_seq_accession": "NM_003716.3"
72 }
73 },
74 {
75     "transcript": {
76         "aa_start": 1187,
77         "consequence_type": "synonymous_variant",
78         "aa_end": 1187,
79         "transcript_id": "ENST00000283269",
80         "is_canonical": false,
81         "aa_change": "G1187G",
82         "annotation": {
83             "vep_impact": "LOW"
84         },
85         "ref_seq_accession": "NM_183394.2"
86     }
87 },
88 {
89     "transcript": {
90         "aa_start": 1147,
91         "consequence_type": "synonymous_variant",
92         "aa_end": 1147,
93         "transcript_id": "ENST00000357948",
94         "is_canonical": false,
95         "aa_change": "G1147G",
96         "annotation": {
97             "vep_impact": "LOW"
98         },
99         "ref_seq_accession": "NM_183393.2"
100    }
101 },
102 {
103     "transcript": {
104         "aa_start": 1217,
105         "consequence_type": "synonymous_variant",
106         "aa_end": 1217,
107         "transcript_id": "ENST00000612439",
108         "is_canonical": false,
109         "aa_change": "G1217G",
110         "annotation": {
111             "vep_impact": "LOW"
112         },
113         "ref_seq_accession": ""
114     }
115 }
116 ]
117 }

```

Simple Somatic Mutation Occurrence Endpoint Examples

Example 1: A user wants to determine the chromosome in case TCGA-DU-6407 that contains the greatest number of `ssms`. As this relates to mutations that are observed in a case, the `ssm_occurrences` endpoint is used.

```
1 {
```



```

2  "op": "in",
3  "content": {
4    "field": "case.submitter_id",
5    "value": ["TCGA-DU-6407"]
6  }
7 }

```

```

1 curl
   https://api.gdc.cancer.gov/ssm_occurrences?format=tsv&fields=ssm.chromosome&size=5000&filters=%7B%0D%0A%22%

```

```

1 ssm.chromosome id
2 chr3      552c09d1-69b1-5c04-b543-524a6feae3eb
3 chr10     391011ff-c1fd-5e2a-a128-652bc660f64c
4 chr10     1378cbc4-af88-55bb-b2e5-185bb4246d7a
5 chr10     3a2b3870-a395-5bc3-8c8f-0d40b0f2202c
6 chr1      4a93d7a5-988d-5055-80da-999dc3b45d80
7 chrX      22a07c7c-16ba-51df-a9a9-1e41e2a45225
8 chr12     dbc5eafa-ea26-5f1c-946c-b6974a345b69
9 chr11     02ae553d-1f27-565d-96c5-2c3cfca7264a
10 chr2      faee73a9-4804-58ea-a91f-18c3d901774f
11 chr6      97c5b38b-fc96-57f5-8517-cc702b3aa70a
12 chr17     0010a89d-9434-5d97-8672-36ee394767d0
13 chr19     f08dcc53-eadc-5ceb-bf31-f6b38629e4cb
14 chrX      19ca262d-b354-54a0-b582-c4719e37e91d
15 chr19     c44a93a1-5c73-5cff-b40e-98ce7e5fe57b
16 chr3      b67f31b5-0341-518e-8fcc-811cd2e36af1
17 chr1      94abd5fd-d539-5a4a-8719-9615cf7cec5d
18 chr17     1476a543-2951-5ec4-b165-67551b47d810
19 chr2      b4822fc9-f0cc-56fd-9d97-f916234e309d
20 chr2      3548ecfe-5186-51e7-8f40-37f4654cd260
21 chr16     105e7811-4601-5ccb-ae93-e7107923599e
22 chr2      99b3aad4-d368-506d-99d6-047cbe5dff0f
23 chr13     9dc3f7cd-9efa-530a-8524-30d067e49d54
24 chr21     1267330b-ae6d-5e25-b19e-34e98523679e
25 chr16     c77f7ce5-fbe6-5da4-9a7b-b528f8e530cb
26 chr10     2cb06277-993e-5502-b2c5-263037c45d18
27 chr17     d25129ad-3ad7-584f-bdeb-fba5c3881d32
28 chr17     a76469cb-973c-5d4d-bf82-7cf4e8f6c129
29 chr10     727c9d57-7b74-556f-aa5b-e1ca1f76d119
30 chr15     b4a86ffd-e60c-5c9c-aaa1-9e9f02d86116
31 chr5      3a023e72-da92-54f7-aa18-502c1076b2b0

```

Example 2: A user has retrieved a `ssm_occurrence`, and would like to determine if that case also has tissue slides and transcriptome profiling data.

```

1 curl
   'https://api.gdc.cancer.gov/ssm_occurrences/6fd8527d-5c40-5604-8fa9-0ce798eec231?pretty=true&expand=case,cas

```

```

1 {
2   "data": {
3     "case": {
4       "disease_type": "Nevi and Melanomas",
5       "updated_datetime": "2018-09-06T18:42:50.098635-05:00",
6       "created_datetime": null,
7       "summary": {
8         "experimental_strategies": [
9           {
10            "file_count": 3,
11            "experimental_strategy": "miRNA-Seq"

```

```

12     },
13     {
14         "file_count": 1,
15         "experimental_strategy": "Tissue Slide"
16     },
17     {
18         "file_count": 18,
19         "experimental_strategy": "WXS"
20     },
21     {
22         "file_count": 1,
23         "experimental_strategy": "Diagnostic Slide"
24     },
25     {
26         "file_count": 4,
27         "experimental_strategy": "RNA-Seq"
28     },
29     {
30         "file_count": 4,
31         "experimental_strategy": "Genotyping Array"
32     },
33     {
34         "file_count": 1,
35         "experimental_strategy": "Methylation Array"
36     }
37 ]
38 },
39 "state": "released",
40 "case_id": "590b5e18-d837-4c0e-becf-80520db57c0f",
41 "primary_site": "Skin",
42 "submitter_id": "TCGA-Z2-A8RT",
43 "available_variation_data": [
44     "cnv",
45     "ssm"
46 ]
47 },
48 "ssm_occurrence_id": "6fd8527d-5c40-5604-8fa9-0ce798eec231"
49 }

```

Copy Number Variation Endpoint Examples

Example 1: A user is interested in finding the first 30 cnvs found on chromosome 4 that have a cnv loss.

```

1 {
2     "op": "and",
3     "content": [
4         {
5             "op": "in",
6             "content": {
7                 "field": "chromosome",
8                 "value": [
9                     "4"
10                ]
11            }
12        },
13        {

```

```

14         "op": "in",
15         "content": {
16             "field": "cnv_change",
17             "value": [
18                 "Loss"
19             ]
20         }
21     }
22 ]
23 }

```

```

1 curl
  'https://api.gdc.cancer.gov/cnvs?filters=%7B%0D%0A+++%22op%22%3A+%22and%22%2C%0D%0A+++%22content%22%3A+%5B%

```

1	ncbi_build	cnv_id	gene_level_cn	cnv_change	end_position	start_position	id	chromosome
2	GRCh38	d18e0dc8-7d56-5d9e-84fd-4f2cf3353c66	True	Loss	88211	53285		
		d18e0dc8-7d56-5d9e-84fd-4f2cf3353c66	4					
3	GRCh38	357a6606-8a64-5827-b776-e71f44b7e05f	True	Loss	163989	124480		
		357a6606-8a64-5827-b776-e71f44b7e05f	4					
4	GRCh38	eda45f5f-6a57-5fae-b8ad-5d67a14423f1	True	Loss	305321	270675		
		eda45f5f-6a57-5fae-b8ad-5d67a14423f1	4					
5	GRCh38	64d82c29-0f20-5a8f-8599-7afb550ab403	True	Loss	384864	337814		
		64d82c29-0f20-5a8f-8599-7afb550ab403	4					
6	GRCh38	f9d24781-34cb-51ff-99c2-84c83a8348ac	True	Loss	499156	425815		
		f9d24781-34cb-51ff-99c2-84c83a8348ac	4					
7	GRCh38	56209b45-3b2c-5862-85bb-362722bae857	True	Loss	540196	499210		
		56209b45-3b2c-5862-85bb-362722bae857	4					
8	GRCh38	04b976d8-90ad-501d-b672-e14816582339	True	Loss	670782	625584		
		04b976d8-90ad-501d-b672-e14816582339	4					
9	GRCh38	574939d6-bf4f-57e9-9c86-629b3d8de664	True	Loss	674338	672436		
		574939d6-bf4f-57e9-9c86-629b3d8de664	4					
10	GRCh38	b2ebf724-0a08-542e-ad1e-392a30208140	True	Loss	682033	673580		
		b2ebf724-0a08-542e-ad1e-392a30208140	4					
11	GRCh38	4e37e683-6f9f-5e80-8e3b-78d0cdf3c28e	True	Loss	689441	681829		
		4e37e683-6f9f-5e80-8e3b-78d0cdf3c28e	4					
12	GRCh38	06837ab7-8242-518f-a24c-dce8a0140b01	True	Loss	770640	705748		
		06837ab7-8242-518f-a24c-dce8a0140b01	4					
13	GRCh38	9f877f14-55ea-5e19-afa0-d294d1700b4b	True	Loss	826198	784957		
		9f877f14-55ea-5e19-afa0-d294d1700b4b	4					
14	GRCh38	bde18311-8a8a-52ef-bcc0-3b6660509df0	True	Loss	932373	849276		
		bde18311-8a8a-52ef-bcc0-3b6660509df0	4					
15	GRCh38	31c65477-0e54-5be3-b1f6-3f249850ef79	True	Loss	958656	932387		
		31c65477-0e54-5be3-b1f6-3f249850ef79	4					
16	GRCh38	c26f1b4d-d4c3-5685-8789-fb0051f8a188	True	Loss	986895	958887		
		c26f1b4d-d4c3-5685-8789-fb0051f8a188	4					
17	GRCh38	0aa931e9-7ec1-57e7-9cb9-ec66a8da5689	True	Loss	993440	979073		
		0aa931e9-7ec1-57e7-9cb9-ec66a8da5689	4					
18	GRCh38	162a9e1d-e1ee-5478-9291-6ba8082d5776	True	Loss	1004506	986997		
		162a9e1d-e1ee-5478-9291-6ba8082d5776	4					
19	GRCh38	6a4d4aef-2289-54f5-b78b-797db8c3a9f2	True	Loss	1026897	1009936		
		6a4d4aef-2289-54f5-b78b-797db8c3a9f2	4					
20	GRCh38	3c26920b-fb93-5595-81a0-770df0c88246	True	Loss	1113562	1056250		
		3c26920b-fb93-5595-81a0-770df0c88246	4					
21	GRCh38	7036724d-1a73-5b2b-ae02-c2dc5b3333d7	True	Loss	1208962	1166932		
		7036724d-1a73-5b2b-ae02-c2dc5b3333d7	4					
22	GRCh38	30b408be-db7b-579b-bbde-4a265c6291ce	True	Loss	1249953	1211448		
		30b408be-db7b-579b-bbde-4a265c6291ce	4					

23	GRCh38	a7c6f097-bba8-5859-838d-8b3b4610c9e6	True	Loss	1340147	1289851
		a7c6f097-bba8-5859-838d-8b3b4610c9e6	4			
24	GRCh38	8fd4f4e8-ddf3-574b-ac19-3112a2778b22	True	Loss	1388049	1347266
		8fd4f4e8-ddf3-574b-ac19-3112a2778b22	4			
25	GRCh38	2315f6cc-9d91-58b8-9f3e-f0d36cd6846c	True	Loss	1395989	1391552
		2315f6cc-9d91-58b8-9f3e-f0d36cd6846c	4			
26	GRCh38	1480d682-fe0e-5ba1-bf4e-ac84945f194a	True	Loss	1406331	1402932
		1480d682-fe0e-5ba1-bf4e-ac84945f194a	4			
27	GRCh38	280e825e-1c51-506b-a4b5-3dc85fd79cbe	True	Loss	1684302	1617915
		280e825e-1c51-506b-a4b5-3dc85fd79cbe	4			
28	GRCh38	607e36e3-6b1d-5564-9670-759668053ceb	True	Loss	1712555	1692800
		607e36e3-6b1d-5564-9670-759668053ceb	4			
29	GRCh38	93b6ccc4-d88d-5040-936f-a23c9006a965	True	Loss	1721358	1715952
		93b6ccc4-d88d-5040-936f-a23c9006a965	4			
30	GRCh38	f6f660d2-5a68-5e49-92b1-a816be39e0fe	True	Loss	1745176	1721490
		f6f660d2-5a68-5e49-92b1-a816be39e0fe	4			
31	GRCh38	a0c069d1-dcb0-5833-8fff-211cd6e3719a	True	Loss	1808872	1793307
		a0c069d1-dcb0-5833-8fff-211cd6e3719a	4			

Example 2: A user wants to determine the location and identity of the gene affected by the cnv 5052be09-2bbe-5175-a0ae-fc568ea75339 and determine whether the gene is found within the Cancer Gene Census.

```
1 curl
   'https://api.gdc.cancer.gov/cnvs/5052be09-2bbe-5175-a0ae-fc568ea75339?pretty=true&expand=consequence.gene'
```

```
1 {
2   "data": {
3     "ncbi_build": "GRCh38",
4     "cnv_id": "5052be09-2bbe-5175-a0ae-fc568ea75339",
5     "gene_level_cn": true,
6     "cnv_change": "Gain",
7     "end_position": 110346681,
8     "start_position": 110338506,
9     "consequence": [
10      {
11        "gene": {
12          "symbol": "RBM15",
13          "is_cancer_gene_census": "True",
14          "biotype": "protein_coding",
15          "gene_id": "ENSG00000162775"
16        }
17      }
18    ],
19    "chromosome": "1"
20  }
```

Copy Number Variation Occurrence Endpoint Examples

Example 1: A user is interested in finding cases that have both cnv and ssm data for females diagnosed with Squamous Cell Neoplasms and have a cnv gain change on chromosome 9. It is important to note that for a case like this, where multiple arguments are need for one filtered field, it is easier for the API to have multiple filters for the same field, `case.available_variation_data` in this example, than having one filter with multiple arguments.

```
1 {
2   "op": "and",
3   "content": [
4     {
```

```

5     "op": "in",
6     "content": {
7         "field": "cnv.cnv_change",
8         "value": [
9             "Gain"
10        ]
11    }
12 },
13 {
14     "op": "in",
15     "content": {
16         "field": "case.demographic.gender",
17         "value": [
18             "female"
19        ]
20    }
21 },
22 {
23     "op": "in",
24     "content": {
25         "field": "case.available_variation_data",
26         "value": [
27             "cnv"
28        ]
29    }
30 },
31 {
32     "op": "in",
33     "content": {
34         "field": "case.available_variation_data",
35         "value": [
36             "ssm"
37        ]
38    }
39 },
40 {
41     "op": "in",
42     "content": {
43         "field": "cnv.chromosome",
44         "value": [
45             "9"
46        ]
47    }
48 },
49 {
50     "op": "in",
51     "content": {
52         "field": "case.disease_type",
53         "value": [
54             "Squamous Cell Neoplasms"
55        ]
56    }
57 }
58 ]
59 }

```

1 curl

'https://api.gdc.cancer.gov/cnv_occurrences?filters=%7B%0D%0A++++%22op%22%3A+%22and%22%2C%0D%0A++++%22content

1	case.case_id	case.available_variation_data.1	case.available_variation_data.0	id
2	638035f6-2909-4a44-980f-468ac5d74e18	ssm cnv	e76d2aaf-f951-5a51-a949-a241dba61f73	
3	ad98977b-e159-410a-b8c2-f4e8a07f9784	ssm cnv	ff3506b8-ee80-570f-ad2d-4ab4a7363b82	
4	c83c52f4-3815-4f49-8218-cf80aaa62e2f	ssm cnv	e73696c5-386f-5cae-aa10-f8628f32ee0e	
5	dac27c24-cdbf-4527-9214-178fde3d098a	ssm cnv	77885824-fae1-5116-9851-694255249cc8	
6	0e91d7b5-ce35-4671-ab9f-cfd5369b557c	ssm cnv	526529ae-8e59-597e-aea1-cc0b06a82e76	
7	ea34663c-f40e-4a3e-9ac0-65d5e9eef12b	ssm cnv	e4a0c034-44d4-5dea-912a-ce331d9a9512	
8	05026179-b1da-411e-a286-89727b1ae380	ssm cnv	30bdc04c-54a5-53ca-bdd0-b808f23da266	
9	f1a1bbf9-4751-4fb4-8a2b-19f8d4ba57bd	ssm cnv	02e3fbb3-da8f-5983-8d10-189e641ddf11	
10	a6ec75d4-1c90-4527-bfae-aa91d2dae082	ssm cnv	94b0e8be-1130-5b88-9103-6756bdabf67b	
11	107f6b9a-2883-4499-a40a-ec25bc834a06	ssm cnv	ad831f27-e6f5-5b78-8a15-0b652621ea4c	

Example 2: A user is interested in the first cnv occurrence (e76d2aaf-f951-5a51-a949-a241dba61f73) from the previous example, and would like to know more about the case exposures and demographics.

```
1 curl
   'https://api.gdc.cancer.gov/cnv_occurrences/e76d2aaf-f951-5a51-a949-a241dba61f73?pretty=true&expand=cnv,case'
```

```
1 {
2   "data": {
3     "cnv": {
4       "ncbi_build": "GRCh38",
5       "cnv_id": "0d475712-c11e-51fb-b6e6-407d12978057",
6       "gene_level_cn": true,
7       "cnv_change": "Gain",
8       "end_position": 133348131,
9       "variant_status": "Tumor only",
10      "start_position": 133338323,
11      "chromosome": "9"
12    },
13    "case": {
14      "disease_type": "Squamous Cell Neoplasms",
15      "updated_datetime": "2018-09-06T11:07:45.510627-05:00",
16      "created_datetime": null,
17      "demographic": {
18        "updated_datetime": "2018-09-06T11:07:45.510627-05:00",
19        "created_datetime": null,
20        "gender": "female",
21        "year_of_birth": 1954,
22        "submitter_id": "TCGA-EA-A3HR_demographic",
23        "state": "released",
24        "race": "white",
25        "demographic_id": "dd8576a8-bd62-55e7-b0df-7233ceded2fb",
26        "ethnicity": "not hispanic or latino",
27        "year_of_death": null
28      },
29      "submitter_id": "TCGA-EA-A3HR",
30      "state": "released",
31      "case_id": "638035f6-2909-4a44-980f-468ac5d74e18",
32      "primary_site": "Cervix uteri",
33      "available_variation_data": [
34        "cnv",
35        "ssm"
36      ],
37      "exposures": [
38        {
39          "cigarettes_per_day": null,
40          "weight": 86,
41          "updated_datetime": "2018-09-06T11:07:45.510627-05:00",
```

```

42     "created_datetime": null,
43     "alcohol_intensity": null,
44     "bmi": 40,
45     "years_smoked": null,
46     "submitter_id": "TCGA-EA-A3HR_exposure",
47     "alcohol_history": null,
48     "state": "released",
49     "tobacco_smoking_status": null,
50     "tobacco_smoking_onset_year": null,
51     "tobacco_smoking_quit_year": null,
52     "exposure_id": "0e7265ab-bf65-50c7-bf33-96a7ac452d7c",
53     "height": 146,
54     "pack_years_smoked": null
55   }
56 ]
57 },
58 "cnv_occurrence_id": "e76d2aaf-f951-5a51-a949-a241dba61f73"
59 }

```

Analysis Endpoints

In addition to the `ssms`, `ssm_occurrences`, and `genes` endpoints mentioned previously, several `/analysis` endpoints were designed to quickly retrieve specific datasets used for visualization display.

Example 1: The `/analysis/top_cases_counts_by_genes` endpoint gives the number of cases with a mutation in each gene listed in the `gene_ids` parameter for each project. Note that this endpoint cannot be used with the `format` or `fields` parameters. In this instance, the query will produce the number of cases in each projects with mutations in the gene `ENSG00000155657`.

```
1 curl "https://api.gdc.cancer.gov/analysis/top_cases_counts_by_genes?gene_ids=ENSG00000155657&pretty=true"
```

This JSON-formatted output is broken up by project. For an example, see the following text:

```

1     "genes": {
2       "my_genes": {
3         "gene_id": {
4           "buckets": [
5             {
6               "key": "ENSG00000155657",
7               "doc_count": 45
8             }
9           ],
10          "sum_other_doc_count": 0,
11          "doc_count_error_upper_bound": 0
12        },
13        "doc_count": 45
14      },
15      "doc_count": 12305
16    },
17    "key": "TCGA-GBM",
18    "doc_count": 45
19  }

```

This portion of the output shows TCGA-GBM including 45 cases that have `ssms` in the gene `ENSG00000155657`.

Example 2: The following demonstrates a use of the `/analysis/top_mutated_genes_by_project` endpoint. This will output the genes that are mutated in the most cases in “TCGA-DLBC” and will count the mutations that have a `HIGH` or `MODERATE` impact on gene function. Note that the `score` field does not represent the number of mutations in a given gene, but a calculation that is used to determine which genes have the greatest number of unique mutations.

```

1 {
2   "op": "AND",
3   "content": [
4     {
5       "op": "in",
6       "content": {
7         "field": "case.project.project_id",
8         "value": [
9           "TCGA-DLBC"
10        ]
11      }
12    },
13    {
14      "op": "in",
15      "content": {
16        "field": "case.ssm.consequence.transcript.annotation.impact",
17        "value": [
18          "HIGH",
19          "MODERATE"
20        ]
21      }
22    }
23  ]
24 }

```

```

1 curl
   "https://api.gdc.cancer.gov/analysis/top_mutated_genes_by_project?fields=gene_id,symbol&filters=%7B%22op%22%3A%5B%7B%22op%22%3A%22in%22%2C%22content%22%3A%7B%22field%22%3A%22case.project.project_id%22%2C%22value%22%3A%5B%22TCGA-DLBC%22%5D%7D%2C%7B%22op%22%3A%22in%22%2C%22content%22%3A%7B%22field%22%3A%22case.ssm.consequence.transcript.annotation.impact%22%2C%22value%22%3A%5B%22HIGH%22%2C%22MODERATE%22%5D%7D%5D%7D"

```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "_score": 14.0,
6         "symbol": "IGHV2-70",
7         "gene_id": "ENSG00000274576"
8       },
9       {
10        "_score": 14.0,
11        "symbol": "IGLV3-1",
12        "gene_id": "ENSG00000211673"
13      },
14      {
15        "_score": 14.0,
16        "symbol": "IGHM",
17        "gene_id": "ENSG00000211899"
18      },
19      {
20        "_score": 11.0,
21        "symbol": "KMT2D",
22        "gene_id": "ENSG00000167548"
23      },
24      {
25        "_score": 11.0,
26        "symbol": "IGLL5",
27        "gene_id": "ENSG00000254709"
28      },
29      {
30        "_score": 11.0,

```



```

31     "symbol": "BTG2",
32     "gene_id": "ENSG00000159388"
33   },
34   {
35     "_score": 9.0,
36     "symbol": "CARD11",
37     "gene_id": "ENSG00000198286"
38   },
39   {
40     "_score": 9.0,
41     "symbol": "IGHG1",
42     "gene_id": "ENSG00000211896"
43   },
44   {
45     "_score": 9.0,
46     "symbol": "IGLC2",
47     "gene_id": "ENSG00000211677"
48   },
49   {
50     "_score": 9.0,
51     "symbol": "LRP1B",
52     "gene_id": "ENSG00000168702"
53   }
54 ],
55 "pagination": {
56   "count": 10,
57   "sort": "None",
58   "from": 0,
59   "page": 1,
60   "total": 3214,
61   "pages": 322,
62   "size": 10
63 }
64 },
65 "warnings": {}
66 }

```

Example 3: The `/analysis/top_mutated_cases_by_gene` endpoint will generate information about the cases that are most affected by mutations in a given number of genes. Below, the file count for each category is given for the cases most affected by mutations in these 50 genes. The size of the output is limited to two cases with the `size=2` parameter, but a higher value can be set by the user.

```

1 curl
   "https://api.gdc.cancer.gov/analysis/top_mutated_cases_by_gene?fields=diagnoses.days_to_death,diagnoses.age_

1 {
2   "data": {
3     "hits": [
4       {
5         "_score": 7.0,
6         "diagnoses": [
7           {
8             "days_to_death": null,
9             "vital_status": "alive",
10            "age_at_diagnosis": 18691,
11            "primary_diagnosis": "c83.3"
12          }
13        ],
14        "case_id": "eda9496e-be80-4a13-bf06-89f0cc9e937f",

```

```

15     "demographic": {
16         "gender": "male",
17         "race": "white",
18         "ethnicity": "hispanic or latino"
19     },
20     "summary": {
21         "data_categories": [
22             {
23                 "file_count": 1,
24                 "data_category": "DNA Methylation"
25             },
26             {
27                 "file_count": 5,
28                 "data_category": "Transcriptome Profiling"
29             },
30             {
31                 "file_count": 1,
32                 "data_category": "Biospecimen"
33             },
34             {
35                 "file_count": 16,
36                 "data_category": "Simple Nucleotide Variation"
37             },
38             {
39                 "file_count": 1,
40                 "data_category": "Clinical"
41             },
42             {
43                 "file_count": 4,
44                 "data_category": "Copy Number Variation"
45             },
46             {
47                 "file_count": 4,
48                 "data_category": "Raw Sequencing Data"
49             }
50         ]
51     }
52 },
53 {
54     "_score": 4.0,
55     "diagnoses": [
56         {
57             "days_to_death": null,
58             "vital_status": "alive",
59             "age_at_diagnosis": 27468,
60             "primary_diagnosis": "c83.3"
61         }
62     ],
63     "case_id": "a43e5f0e-a21f-48d8-97e0-084d413680b7",
64     "demographic": {
65         "gender": "male",
66         "race": "white",
67         "ethnicity": "not hispanic or latino"
68     },
69     "summary": {
70         "data_categories": [
71             {
72                 "file_count": 1,

```

```

73     "data_category": "DNA Methylation"
74   },
75   {
76     "file_count": 5,
77     "data_category": "Transcriptome Profiling"
78   },
79   {
80     "file_count": 1,
81     "data_category": "Biospecimen"
82   },
83   {
84     "file_count": 16,
85     "data_category": "Simple Nucleotide Variation"
86   },
87   {
88     "file_count": 1,
89     "data_category": "Clinical"
90   },
91   {
92     "file_count": 4,
93     "data_category": "Copy Number Variation"
94   },
95   {
96     "file_count": 4,
97     "data_category": "Raw Sequencing Data"
98   }
99 ]
100 }
101 }
102 ],
103 "pagination": {
104   "count": 2,
105   "sort": "None",
106   "from": 0,
107   "page": 1,
108   "total": 27,
109   "pages": 14,
110   "size": 2
111 }
112 },
113 "warnings": {}
114 }

```

Example 4: The `/analysis/mutated_cases_count_by_project` endpoint produces counts for the number of cases that have associated `ssm` data in each project. The number of affected cases can be found under `"case_with_ssm": {"doc_count": $case_count}`.

```
1 curl "https://api.gdc.cancer.gov/analysis/mutated_cases_count_by_project?size=0&pretty=true"
```

```

1 {
2   "hits": {
3     "hits": [],
4     "total": 14551,
5     "max_score": 0.0
6   },
7   "_shards": {
8     "successful": 9,
9     "failed": 0,
10    "total": 9

```

```

11 },
12 "took": 4,
13 "aggregations": {
14   "projects": {
15     "buckets": [
16       {
17         "case_summary": {
18           "case_with_ssm": {
19             "doc_count": 216
20           },
21           "doc_count": 637
22         },
23         "key": "TARGET-NBL",
24         "doc_count": 1127
25       },
26       {
27         "case_summary": {
28           "case_with_ssm": {
29             "doc_count": 1044
30           },
31           "doc_count": 7625
32         },
33         "key": "TCGA-BRCA",
34         "doc_count": 1098
35       },
36       {
37         "case_summary": {
38           "case_with_ssm": {
39             "doc_count": 8
40           },
41           "doc_count": 579
42         },
43         "key": "TARGET-AML",
44         "doc_count": 988
45       },
46       {
47         "case_summary": {
48           "case_with_ssm": {
49             "doc_count": 34
50           },
51           "doc_count": 290
52         },
53         "key": "TARGET-WT",
54         "doc_count": 652
55       },
56       {
57         "case_summary": {
58           "case_with_ssm": {
59             "doc_count": 396
60           },
61           "doc_count": 3197
62         },
63         "key": "TCGA-GBM",
64         "doc_count": 617
65       },
66       {
67         "case_summary": {
68           "case_with_ssm": {

```

```

69         "doc_count": 443
70     },
71     "doc_count": 3880
72 },
73 "key": "TCGA-OV",
74 "doc_count": 608
75 },
76 {
77     "case_summary": {
78         "case_with_ssm": {
79             "doc_count": 569
80         },
81         "doc_count": 3874
82     },
83     "key": "TCGA-LUAD",
84     "doc_count": 585
85 },
86 {
87     "case_summary": {
88         "case_with_ssm": {
89             "doc_count": 542
90         },
91         "doc_count": 3874
92     },
93     "key": "TCGA-UCEC",
94     "doc_count": 560
95 },
96 {
97     "case_summary": {
98         "case_with_ssm": {
99             "doc_count": 339
100        },
101        "doc_count": 3547
102    },
103    "key": "TCGA-KIRC",
104    "doc_count": 537
105 },
106 {
107     "case_summary": {
108         "case_with_ssm": {
109             "doc_count": 510
110         },
111         "doc_count": 3671
112     },
113     "key": "TCGA-HNSC",
114     "doc_count": 528
115 },
116 {
117     "case_summary": {
118         "case_with_ssm": {
119             "doc_count": 513
120         },
121         "doc_count": 3606
122     },
123     "key": "TCGA-LGG",
124     "doc_count": 516
125 },
126 {

```

```

127     "case_summary": {
128         "case_with_ssm": {
129             "doc_count": 496
130         },
131         "doc_count": 3536
132     },
133     "key": "TCGA-THCA",
134     "doc_count": 507
135 },
136 {
137     "case_summary": {
138         "case_with_ssm": {
139             "doc_count": 497
140         },
141         "doc_count": 3520
142     },
143     "key": "TCGA-LUSC",
144     "doc_count": 504
145 },
146 {
147     "case_summary": {
148         "case_with_ssm": {
149             "doc_count": 498
150         },
151         "doc_count": 3490
152     },
153     "key": "TCGA-PRAD",
154     "doc_count": 500
155 },
156 {
157     "case_summary": {
158         "case_with_ssm": {
159             "doc_count": 470
160         },
161         "doc_count": 3289
162     },
163     "key": "TCGA-SKCM",
164     "doc_count": 470
165 },
166 {
167     "case_summary": {
168         "case_with_ssm": {
169             "doc_count": 433
170         },
171         "doc_count": 3188
172     },
173     "key": "TCGA-COAD",
174     "doc_count": 461
175 },
176 {
177     "case_summary": {
178         "case_with_ssm": {
179             "doc_count": 441
180         },
181         "doc_count": 3095
182     },
183     "key": "TCGA-STAD",
184     "doc_count": 443

```

```

185     },
186     {
187         "case_summary": {
188             "case_with_ssm": {
189                 "doc_count": 412
190             },
191             "doc_count": 2884
192         },
193         "key": "TCGA-BLCA",
194         "doc_count": 412
195     },
196     {
197         "case_summary": {
198             "case_with_ssm": {
199                 "doc_count": 0
200             },
201             "doc_count": 0
202         },
203         "key": "TARGET-OS",
204         "doc_count": 381
205     },
206     {
207         "case_summary": {
208             "case_with_ssm": {
209                 "doc_count": 375
210             },
211             "doc_count": 2635
212         },
213         "key": "TCGA-LIHC",
214         "doc_count": 377
215     },
216     {
217         "case_summary": {
218             "case_with_ssm": {
219                 "doc_count": 305
220             },
221             "doc_count": 2142
222         },
223         "key": "TCGA-CESC",
224         "doc_count": 307
225     },
226     {
227         "case_summary": {
228             "case_with_ssm": {
229                 "doc_count": 288
230             },
231             "doc_count": 2033
232         },
233         "key": "TCGA-KIRP",
234         "doc_count": 291
235     },
236     {
237         "case_summary": {
238             "case_with_ssm": {
239                 "doc_count": 255
240             },
241             "doc_count": 1821
242         },

```

```

243     "key": "TCGA-SARC",
244     "doc_count": 261
245 },
246 {
247     "case_summary": {
248         "case_with_ssm": {
249             "doc_count": 149
250         },
251         "doc_count": 1192
252     },
253     "key": "TCGA-LAML",
254     "doc_count": 200
255 },
256 {
257     "case_summary": {
258         "case_with_ssm": {
259             "doc_count": 184
260         },
261         "doc_count": 1293
262     },
263     "key": "TCGA-ESCA",
264     "doc_count": 185
265 },
266 {
267     "case_summary": {
268         "case_with_ssm": {
269             "doc_count": 183
270         },
271         "doc_count": 1285
272     },
273     "key": "TCGA-PAAD",
274     "doc_count": 185
275 },
276 {
277     "case_summary": {
278         "case_with_ssm": {
279             "doc_count": 179
280         },
281         "doc_count": 1253
282     },
283     "key": "TCGA-PCPG",
284     "doc_count": 179
285 },
286 {
287     "case_summary": {
288         "case_with_ssm": {
289             "doc_count": 158
290         },
291         "doc_count": 1169
292     },
293     "key": "TCGA-READ",
294     "doc_count": 172
295 },
296 {
297     "case_summary": {
298         "case_with_ssm": {
299             "doc_count": 150
300         },

```



```

301     "doc_count": 1018
302   },
303   "key": "TCGA-TGCT",
304   "doc_count": 150
305 },
306 {
307   "case_summary": {
308     "case_with_ssm": {
309       "doc_count": 123
310     },
311     "doc_count": 867
312   },
313   "key": "TCGA-THYM",
314   "doc_count": 124
315 },
316 {
317   "case_summary": {
318     "case_with_ssm": {
319       "doc_count": 66
320     },
321     "doc_count": 556
322   },
323   "key": "TCGA-KICH",
324   "doc_count": 113
325 },
326 {
327   "case_summary": {
328     "case_with_ssm": {
329       "doc_count": 92
330     },
331     "doc_count": 620
332   },
333   "key": "TCGA-ACC",
334   "doc_count": 92
335 },
336 {
337   "case_summary": {
338     "case_with_ssm": {
339       "doc_count": 83
340     },
341     "doc_count": 605
342   },
343   "key": "TCGA-MESO",
344   "doc_count": 87
345 },
346 {
347   "case_summary": {
348     "case_with_ssm": {
349       "doc_count": 80
350     },
351     "doc_count": 560
352   },
353   "key": "TCGA-UVM",
354   "doc_count": 80
355 },
356 {
357   "case_summary": {
358     "case_with_ssm": {

```

```

359         "doc_count": 0
360     },
361     "doc_count": 163
362 },
363 "key": "TARGET-RT",
364 "doc_count": 75
365 },
366 {
367     "case_summary": {
368         "case_with_ssm": {
369             "doc_count": 48
370         },
371         "doc_count": 346
372     },
373     "key": "TCGA-DLBC",
374     "doc_count": 58
375 },
376 {
377     "case_summary": {
378         "case_with_ssm": {
379             "doc_count": 57
380         },
381         "doc_count": 399
382     },
383     "key": "TCGA-UCS",
384     "doc_count": 57
385 },
386 {
387     "case_summary": {
388         "case_with_ssm": {
389             "doc_count": 51
390         },
391         "doc_count": 306
392     },
393     "key": "TCGA-CHOL",
394     "doc_count": 51
395 },
396 {
397     "case_summary": {
398         "case_with_ssm": {
399             "doc_count": 0
400         },
401         "doc_count": 13
402     },
403     "key": "TARGET-CCSK",
404     "doc_count": 13
405 }
406 ],
407 "sum_other_doc_count": 0,
408 "doc_count_error_upper_bound": 0
409 }
410 },
411 "timed_out": false
412 }

```

Survival Analysis Endpoint

[Survival plots](#) are generated for different subsets of data, based on variants or projects, in the GDC Data Portal. The `/analysis/survival` endpoint can be used to programmatically retrieve the raw data used to generate these plots and apply different filters. Note that the `fields` and `format` parameters cannot be modified.

Example 1: A user wants to download data to generate a survival plot for cases from the project TCGA-DLBC.

```
1 curl
   "https://api.gdc.cancer.gov/analysis/survival?filters=%5B%7B%22op%22%3A%22%3D%22%2C%22content%22%3A%7B%22fie

1 {
2   "overallStats": {},
3   "results": [
4     {
5       "donors": [
6         {
7           "survivalEstimate": 1,
8           "id": "dc87a809-95de-4eb7-a1c2-2650475f2d7e",
9           "censored": true,
10          "time": 1
11        },
12        {
13          "survivalEstimate": 1,
14          "id": "4dd86ebd-ef16-4b2b-9ea0-5d1d7afef257",
15          "censored": true,
16          "time": 17
17        },
18        {
19          "survivalEstimate": 1,
20          "id": "0bf573ac-cd1e-42d8-90cf-b30d7b08679c",
21          "censored": false,
22          "time": 58
23        },
24        {
25          "survivalEstimate": 0.9777777777777777,
26          "id": "f978cb0f-d319-4c01-b4c5-23ae1403a106",
27          "censored": true,
28          "time": 126
29        },
30        {
31          "survivalEstimate": 0.9777777777777777,
32          "id": "a43e5f0e-a21f-48d8-97e0-084d413680b7",
33          "censored": true,
34          "time": 132
35        },
36        {
37          "survivalEstimate": 0.9777777777777777,
38          "id": "1843c82e-7a35-474f-9f79-c0a9af9aa09c",
39          "censored": true,
40          "time": 132
41        },
42        {
43          "survivalEstimate": 0.9777777777777777,
44          "id": "0030a28c-81aa-44b0-8be0-b35e1dcbf98c",
45          "censored": false,
46          "time": 248
47        },
48        {
```

```

49     "survivalEstimate": 0.9539295392953929,
50     "id": "f553f1a9-ecf2-4783-a609-6adca7c4c597",
51     "censored": true,
52     "time": 298
53 },
54 {
55     "survivalEstimate": 0.9539295392953929,
56     "id": "f784bc3a-751b-4025-aab2-0af2f6f24266",
57     "censored": false,
58     "time": 313
59 },
60 {
61     "survivalEstimate": 0.929469807518588,
62     "id": "29e3d122-15a1-4235-a356-b1a9f94ceb39",
63     "censored": true,
64     "time": 385
65 },
66 {
67     "survivalEstimate": 0.929469807518588,
68     "id": "0e251c03-bf86-4ed8-b45d-3cbc97160502",
69     "censored": false,
70     "time": 391
71 },
72 {
73     "survivalEstimate": 0.9043490019099776,
74     "id": "e6365b38-bc44-400c-b4aa-18ce8ff5bfce",
75     "censored": true,
76     "time": 427
77 },
78 {
79     "survivalEstimate": 0.9043490019099776,
80     "id": "b56bdbdb-43af-4a03-a072-54dd22d7550c",
81     "censored": true,
82     "time": 553
83 },
84 {
85     "survivalEstimate": 0.9043490019099776,
86     "id": "31bbad4e-3789-42ec-9faa-1cb86970f723",
87     "censored": false,
88     "time": 595
89 },
90 {
91     "survivalEstimate": 0.8777505018538018,
92     "id": "0e9fcccc-0630-408d-a121-2c6413824cb7",
93     "censored": true,
94     "time": 679
95 },
96 {
97     "survivalEstimate": 0.8777505018538018,
98     "id": "a5b188f0-a6d3-4d4a-b04f-36d47ec05338",
99     "censored": false,
100    "time": 708
101 },
102 {
103    "survivalEstimate": 0.8503207986708705,
104    "id": "ed746cb9-0f2f-48ce-923a-3a9f9f00b331",
105    "censored": true,
106    "time": 719

```

```

107 },
108 {
109     "survivalEstimate": 0.8503207986708705,
110     "id": "c85f340e-584b-4f3b-b6a5-540491fc8ad2",
111     "censored": true,
112     "time": 730
113 },
114 {
115     "survivalEstimate": 0.8503207986708705,
116     "id": "69f23725-adca-48ac-9b33-80a7aae24cfe",
117     "censored": true,
118     "time": 749
119 },
120 {
121     "survivalEstimate": 0.8503207986708705,
122     "id": "67325322-483f-443f-9ffa-2a20d108a2fb",
123     "censored": true,
124     "time": 751
125 },
126 {
127     "survivalEstimate": 0.8503207986708705,
128     "id": "eda9496e-be80-4a13-bf06-89f0cc9e937f",
129     "censored": true,
130     "time": 765
131 },
132 {
133     "survivalEstimate": 0.8503207986708705,
134     "id": "25ff86af-beb4-480c-b706-f3fe0306f7cf",
135     "censored": true,
136     "time": 788
137 },
138 {
139     "survivalEstimate": 0.8503207986708705,
140     "id": "1d0db5d7-39ca-466d-96b3-0d278c5ea768",
141     "censored": true,
142     "time": 791
143 },
144 {
145     "survivalEstimate": 0.8503207986708705,
146     "id": "c8cde9ea-89e9-4ee8-8a46-417a48f6d3ab",
147     "censored": true,
148     "time": 832
149 },
150 {
151     "survivalEstimate": 0.8503207986708705,
152     "id": "f0a326d2-1f3e-4a5d-bca8-32aacc52338",
153     "censored": true,
154     "time": 946
155 },
156 {
157     "survivalEstimate": 0.8503207986708705,
158     "id": "a8e2df1e-4042-42af-9231-3a00e83489f0",
159     "censored": true,
160     "time": 965
161 },
162 {
163     "survivalEstimate": 0.8503207986708705,
164     "id": "e56e4d9c-052e-4ec6-a81b-dbd53e9c8ffe",

```

```

165     "censored": true,
166     "time": 972
167   },
168   {
169     "survivalEstimate": 0.8503207986708705,
170     "id": "45b0cf9f-a879-417f-8f39-7770552252c0",
171     "censored": true,
172     "time": 982
173   },
174   {
175     "survivalEstimate": 0.8503207986708705,
176     "id": "1f971af1-6772-4fe6-8d35-bbe527a037fe",
177     "censored": true,
178     "time": 1081
179   },
180   {
181     "survivalEstimate": 0.8503207986708705,
182     "id": "33365d22-cb83-4d8e-a2d1-06b675f75f6e",
183     "censored": true,
184     "time": 1163
185   },
186   {
187     "survivalEstimate": 0.8503207986708705,
188     "id": "6a21c948-cd85-4150-8c01-83017d7dc1ed",
189     "censored": false,
190     "time": 1252
191   },
192   {
193     "survivalEstimate": 0.8003019281608192,
194     "id": "f855dad1-6ffc-493e-ba6c-970874bc9210",
195     "censored": true,
196     "time": 1299
197   },
198   {
199     "survivalEstimate": 0.8003019281608192,
200     "id": "c1c06604-5ae2-4a53-b9c0-eb210d38e3f0",
201     "censored": true,
202     "time": 1334
203   },
204   {
205     "survivalEstimate": 0.8003019281608192,
206     "id": "58e66976-4507-4552-ac53-83a49a142dde",
207     "censored": true,
208     "time": 1373
209   },
210   {
211     "survivalEstimate": 0.8003019281608192,
212     "id": "ea54dbad-1b23-41cc-9378-d4002a8fca51",
213     "censored": true,
214     "time": 1581
215   },
216   {
217     "survivalEstimate": 0.8003019281608192,
218     "id": "d7df78b5-24f1-4ff4-bd9b-f0e6bec8289a",
219     "censored": true,
220     "time": 1581
221   },
222   {

```

```

223     "survivalEstimate": 0.8003019281608192,
224     "id": "29aff186-c321-4ff9-b81b-105e27e620ff",
225     "censored": true,
226     "time": 1617
227   },
228   {
229     "survivalEstimate": 0.8003019281608192,
230     "id": "5eff68ff-f6c3-40c9-9fc8-00e684a7b712",
231     "censored": true,
232     "time": 1739
233   },
234   {
235     "survivalEstimate": 0.8003019281608192,
236     "id": "f8cf647b-1447-4ac3-8c43-bef07765cabf",
237     "censored": true,
238     "time": 2131
239   },
240   {
241     "survivalEstimate": 0.8003019281608192,
242     "id": "c3d662ee-48d0-454a-bb0c-77d3338d3747",
243     "censored": true,
244     "time": 2983
245   },
246   {
247     "survivalEstimate": 0.8003019281608192,
248     "id": "6e9437f0-a4ed-475c-ab0e-bf1431c70a90",
249     "censored": true,
250     "time": 3333
251   },
252   {
253     "survivalEstimate": 0.8003019281608192,
254     "id": "fdec74f-ac4e-46b1-b23a-5f7fde96ef9f",
255     "censored": true,
256     "time": 3394
257   },
258   {
259     "survivalEstimate": 0.8003019281608192,
260     "id": "a468e725-ad4b-411d-ac5c-2eacc68ec580",
261     "censored": false,
262     "time": 3553
263   },
264   {
265     "survivalEstimate": 0.6402415425286554,
266     "id": "1ea575f1-f731-408b-a629-f5f4abab569e",
267     "censored": true,
268     "time": 3897
269   },
270   {
271     "survivalEstimate": 0.6402415425286554,
272     "id": "7a589441-11ef-4158-87e7-3951d86bc2aa",
273     "censored": true,
274     "time": 4578
275   },
276   {
277     "survivalEstimate": 0.6402415425286554,
278     "id": "3622cf29-600f-4410-84d4-a9afeb41c475",
279     "censored": true,
280     "time": 5980

```

```

281     },
282     {
283       "survivalEstimate": 0.6402415425286554,
284       "id": "3f5a897d-1eaa-4d4c-8324-27ac07c90927",
285       "censored": false,
286       "time": 6425
287     }
288   ],
289   "meta": {
290     "id": 140429063094496
291   }
292 }
293 ]
294 }

```

Example 2: Here the survival endpoint is used to compare two survival plots for TCGA-BRCA cases. One plot will display survival information about cases with a particular mutation (in this instance: chr3:g.179234297A>G) and the other plot will display information about cases without that mutation. This type of query will also print the results of a chi-squared analysis between the two subsets of cases.

```

1 [
2   {
3     "op": "and",
4     "content": [
5       {
6         "op": "=",
7         "content": {
8           "field": "cases.project.project_id",
9           "value": "TCGA-BRCA"
10        }
11      },
12      {
13        "op": "=",
14        "content": {
15          "field": "gene.ssm.ssm_id",
16          "value": "edd1ae2c-3ca9-52bd-a124-b09ed304fcc2"
17        }
18      }
19    ]
20  },
21  {
22    "op": "and",
23    "content": [
24      {
25        "op": "=",
26        "content": {
27          "field": "cases.project.project_id",
28          "value": "TCGA-BRCA"
29        }
30      },
31      {
32        "op": "excludeifany",
33        "content": {
34          "field": "gene.ssm.ssm_id",
35          "value": "edd1ae2c-3ca9-52bd-a124-b09ed304fcc2"
36        }
37      }
38    ]
39  }

```


40]

```
1 curl
  "https://api.gdc.cancer.gov/analysis/survival?filters=%5B%7B%22op%22%3A%22and%22%2C%22content%22%3A%5B%7B%22
1 {
2   "overallStats": {
3     "degreesFreedom": 1,
4     "chiSquared": 0.8577589072612264,
5     "pValue": 0.35436660628146011
6   },
7   "results": [
8     {
9       "donors": [
10        {
11          "survivalEstimate": 1,
12          "id": "a991644b-3ee6-4cda-acf0-e37de48a49fc",
13          "censored": true,
14          "time": 10
15        },
16        {
17          "survivalEstimate": 1,
18          "id": "2e1e3bf0-1708-4b65-936c-48b89eb8966a",
19          "censored": true,
20          "time": 19
21        }
22      ],
23    },
24    "meta": {
25      "id": 140055251282040
26    }
27  ],
28  {
29    "donors": [
30      {
31        "survivalEstimate": 1,
32        "id": "5e4187c9-98f8-4bdb-a8da-6a914e96f47a",
33        "censored": true,
34        "time": -31
35      }
36    ],
37    "meta": {
38      "id": 140055251282040
39    }
40  ]
41 }
```

The output represents two sets of coordinates delimited as objects with the `donors` tag. One set of coordinates will generate a survival plot representing TCGA-BRCA cases that have the mutation of interest and the other will generate a survival plot for the remaining cases in TCGA-BRCA.

Example 3: Custom survival plots can be generated using the GDC API. For example, a user could generate survival plot data comparing patients with a mutation in genes associated with a biological pathway with patients without mutations in that pathway. The following example compares a patient with at least one mutation in either gene `ENSG00000141510` or `ENSG00000155657` with patients that do not have mutations in these genes.

```
1 [
2   {
3     "op": "and",
4     "content": [
5       {
6         "op": "=",
7         "content": {
8           "field": "cases.project.project_id",
9         }
10      }
11    ]
12  }
13 ]
```

```

9         "value": "TCGA-BRCA"
10     }
11 },
12 {
13     "op": "=",
14     "content": {
15         "field": "gene.gene_id",
16         "value": ["ENSG00000141510", "ENSG00000155657"]
17     }
18 }
19 ]
20 },
21 {
22     "op": "and",
23     "content": [
24         {
25             "op": "=",
26             "content": {
27                 "field": "cases.project.project_id",
28                 "value": "TCGA-BRCA"
29             }
30         },
31         {
32             "op": "excludeifany",
33             "content": {
34                 "field": "gene.gene_id",
35                 "value": ["ENSG00000141510", "ENSG00000155657"]
36             }
37         }
38     ]
39 }
40 ]

```

```

1 curl
  "https://api.gdc.cancer.gov/analysis/survival?filters=%5B%0D%0A%7B%0D%0A%22op%22%3A%22and%22%2C%0D%0A%22cont

```

Example 4: Survival plots can be even more customizable when sets of case IDs are used. Two sets of case IDs (or barcodes) can be retrieved in a separate step based on custom criteria and compared in a survival plot. See below for an example query.

```

1 [{
2     "op": "=",
3     "content": {
4         "field": "cases.submitter_id",
5         "value": ["TCGA-HT-A74J", "TCGA-43-A56U", "TCGA-GM-A3XL", "TCGA-A1-A0SQ", "TCGA-K1-A6RV", "TCGA-J2-A4AD", "TCGA-X
6     }
7 },
8 {
9     "op": "=",
10    "content": {
11        "field": "cases.submitter_id",
12        "value": ["TCGA-55-5899", "TCGA-55-6642", "TCGA-55-7907", "TCGA-67-6216", "TCGA-75-5146", "TCGA-49-4510", "TCGA-7
13    }
14 }]

```

```

1 curl
  "https://api.gdc.cancer.gov/analysis/survival?filters=%5B%7B%22op%22%3A%22%3D%22%2C%22content%22%3A%7B%22fie

```

Chapter 5

BAM Slicing

BAM Slicing

The GDC API provides remote BAM slicing functionality that enables downloading of specific parts of a BAM file instead of the whole file. This functionality can be accessed at the `slicing` endpoint, using a syntax similar to that of widely used bioinformatics tools such as `samtools`.

About the slicing endpoint

The `slicing` endpoint accepts HTTP GET requests in the form of a URL, and HTTP POST requests that carry a JSON payload. POST requests are more appropriate in cases where query parameters make the GET URL very long.

The response will be a BAM-formatted file containing the header of the source BAM file, as well as any alignment records that are found to overlap the specified regions, sorted by chromosomal coordinate.

Please note the following:

- The functionality of this API differs from the usual functionality of `samtools` in that alignment records that overlap multiple regions will not be returned multiple times.
- A request with no region or gene specified will return the BAM header, which makes it easy to inspect the references to which the alignment records were aligned.
- A request for regions that are not included in the source BAM is not considered an error, and is treated the same as if no records existed for the region.
- Examples provided for BAM slicing functionality are intended for use with GDC harmonized data (i.e. BAM files available in the GDC Data Portal). Slicing of unharmonized BAM files (i.e. BAM files in the GDC Legacy Portal) is not supported.
- Bam slicing does not create an associated bam index (.bai) file. For applications requiring a .bai file users will need to generate this file from the bam slice using a tool and command such as `samtools index`.

Query Parameters

The following query parameters and JSON fields are supported:

Description	Query Parameter	JSON Field	Query format
entire chromosome, or a position or region on the chromosome, specified using chromosomal coordinates	region	regions	region=(:-)??
region specified using a HGNC / GENCODE v22 gene name	gencode	gencode	gencode=

NOTE: The successfully sliced BAM will contain all reads that overlap (entirely or partially) with the specified region or gene. It is possible to specify an open-ended region, e.g. `chr2:10000`, which would return all reads that (completely or partially) overlap with the region of chromosome 2 from position 10,000 to the end of the chromosome.

JSON Schema

JSON payloads can be syntactically verified using the following JSON schema:

```
1 {
2   "$schema": "http://json-schema.org/schema#",
3   "type": "object",
4   "properties": {
5     "regions": {
6       "type": "array",
7       "items": {
8         "type": "string",
9         "pattern": "^[a-zA-Z0-9]+(:([0-9]+)?(-[0-9]+)?)?$"
10      }
11    },
12    "gencode": {
13      "type": "array",
14      "items": {
15        "type": "string"
16      }
17    }
18  }
19 }
```

Examples: Specifying a region

The following two requests are examples of BAM slicing using region(s).

```
1
2 token=$(<gdc-token-text-file.txt)
3
4 curl --header "X-Auth-Token: $token"
   'https://api.gdc.cancer.gov/slicing/view/df80679e-c4d3-487b-934c-fcc782e5d46e?region=chr1&region=chr2:10000&'
   --output get_regions_slice.bam
```

```
1 {
2   "regions": [
3     "chr1",
4     "chr2:10000",
5     "chr3:10000-20000"
6   ]
7 }
```

```
1 token=$(<gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request POST
   https://api.gdc.cancer.gov/slicing/view/9ca90dfa-e62f-4f9c-9946-dfcecfd3ca4d --header "Content-Type:
   application/json" -d@Payload --output post_regions_slice.bam
```

```
1 Response:
2 HTTP/1.1 206
3
4 <bam_data_stream>
```

Examples: Specifying a gene

The following two requests are examples of BAM slicing using HGNC / GENCODE v22 gene name(s).

```
1 token=$(cat gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token"
   'https://api.gdc.cancer.gov/slicing/view/df80679e-c4d3-487b-934c-fcc782e5d46e?gencode=BRCA1' --output
   get_brca1_slice.bam
```

```
1 {
2   "gencode": [
3     "BRCA1",
4     "BRCA2"
5   ]
6 }
```

```
1 curl --header "X-Auth-Token: $token" --request POST
   https://api.gdc.cancer.gov/slicing/view/df80679e-c4d3-487b-934c-fcc782e5d46e --header "Content-Type:
   application/json" -d@Payload --output post_brca12_slice.bam
```

```
1 Response:
2 HTTP/1.1 206
3
4 <bam_data_stream>
```

After downloading, the sliced BAM file can be converted to SAM using the following command if `samtools` is installed on the user's system:

```
1 samtools view -h brca1_slice.bam -o brca1_slice.sam
```

Errors

When slicing cannot be performed, the GDC API will provide JSON error responses and HTTP error codes.

JSON Error Responses

JSON error responses have the following structure:

```
1 {
2   "error": "<error-message>"
3 }
```

For example, when making a request for a protected BAM without supplying a GDC authentication token:

```
1 curl https://api.gdc.cancer.gov/v0/slicing/view/15b0bf8e-ff20-41ab-8366-a495c11b30be
```

```
1 HTTP/1.1 403 FORBIDDEN
2 {
3   "error": "Please specify a X-Auth-Token"
4 }
```

HTTP error codes

Potential HTTP error codes include:

Error Code	Description
400	Bad Request – The regions specified are malformed
403	Unauthorized – The user could not be authenticated
403	Unauthorized – The user is not authorized for access to the source BAM
404	Not Found – No BAM is specified
404	Not Found – No BAM can be found for the specified GDC BAM ID
504	BAI Not Found – No BAI can be found for the BAM

Transfer Errors

In the case that an error occurs during transfer of the resulting BAM, the BGZF EOF marker will not be present. This early truncation of the BAM file will cause errors if the file is used as input to other programs. For example, `samtools` will provide the error “EOF marker is absent”.

Early truncation can arise when connection is interrupted or when slicing fails due to BAM corruption.

Chapter 6

Submission

Submission

Overview

The GDC Submission API uses methods and endpoints that are distinct from those that drive the functionality of the GDC Data Portal. In particular, data and metadata that are in the process of being submitted can only be queried using GraphQL.

This section describes the GDC API's submission functionality, including methods for submitting, deleting, updating, searching, and retrieving data and metadata.

Submission endpoint

Constructing the endpoint URL

The endpoint for submitting data to a specific project in the GDC is constructed as follows:

```
1 https://api.gdc.cancer.gov/[API_version/]submission/Program.name/Project.code
```

where `API_version/` is the optional API version component (see [Getting Started]).

The values of `Program.name` and `Project.code` can be obtained from the project URL on the GDC Data Submission Portal:

```
1 https://portal.gdc.cancer.gov/submission/Program.name/Project.code/dashboard
```

For more information about program name and project code see [The GDC Data Model section](#).

Example

The following are URL examples for a project with `Program.name` “TCGA” and `Project.code` “ALCH”:

- Submission Portal URL: `https://portal.gdc.cancer.gov/submission/TCGA/ALCH/dashboard`
- API submission endpoint (versioned): `https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH`
- API submission endpoint (unversioned): `https://api.gdc.cancer.gov/submission/TCGA/ALCH`

Submission Formats

Metadata Formats

JSON and TSV

The GDC API accepts project metadata in JSON and TSV formats for the purpose of creating entities in the GDC Data Model. This includes clinical and biospecimen metadata such as disease name and stage, patient age, sample type, and certain details about the types of data collected. Upon successful data submission and project release, this metadata is indexed and becomes available for queries by data users via the GDC Data Portal and the GDC API. See [GDC Data Model](#) (below) for information on accepted metadata elements and instructions for obtaining templates for metadata submission.

Content-Type Header JSON is the default format for metadata submission. Submission API calls with JSON payloads should include the HTTP header `Content-Type: application/json`. Requests with TSV payloads must instead include the header `Content-Type: text/tsv`.

Binary Mode Metadata files must be uploaded in raw, unencoded form. Binary mode should be used, if available, to ensure that file contents are not encoded by the upload tool before transmission. For example, when using the `curl` command-line tool, the `--data-binary` switch should be used instead of `--data`. The `--data-binary` switch is required for uploading TSV files.

BCR XML

While JSON and TSV are the recommended formats for submitting metadata, the GDC API can also extract metadata elements from BCR XML files. Users wishing to submit metadata as BCR XML must contact GDC User Services and ensure that appropriate element mapping is in place before initiating XML submission.

To submit BCR XML, make PUT requests with the `Content-Type: application/xml` header to the following URLs, replacing `Program.name` and `Project.code` as described in Submission Endpoint (above):

0. For Biospecimen BCR XML: `https://api.gdc.cancer.gov/v0/submission/Program.name/Project.code/xml/biospecimen/`
1. For Clinical BCR XML: `https://api.gdc.cancer.gov/v0/submission/Program.name/Project.code/xml/clinical/bcr/`

Biospecimen BCR XML creates Case entities in the GDC Data Model, whereas Clinical BCR XML does not. Unless the associated cases already exist in the GDC, Biospecimen BCR XML must be uploaded before Clinical BCR XML.

BCR XML files can be submitted in dry run mode, described [below](#), by appending `_dry_run` to the above URLs.

The following is a sample shell command for submitting an XML file:

```
1 curl --request PUT --header "X-Auth-Token: $token" --header 'Content-Type: application/xml'
   --data-binary @biospecimen.xml
   'https://api.gdc.cancer.gov/v0/submission/GDC/INTERNAL/xml/biospecimen/bcr/_dry_run'
```

NOTE: A typical BCR XML file contains more information than what is extracted and indexed by the GDC. XML files submitted to the above endpoints are not retained or distributed to GDC data users, so the same files should also be submitted as data files (i.e. as clinical or biospecimen supplements).

Data File Formats

The GDC API accepts a variety of data files after their metadata has been registered: BAM and FASTQ files, clinical and biospecimen supplements, slide images, and other file types. Supported data file formats are listed on the [GDC website](#).

GDC Data Model

Submitters should review the GDC Data Model documentation and the GDC Data Dictionary before initiating submission.

UUIDs

Submitters can assign UUIDs to all submittable entities other than those that correspond to files (entities in categories `data_file` or `metadata_file`). If the submitter does not provide a UUID, it will be assigned by the GDC and returned in the API response upon successful completion of the submission transaction. See Appendix C for details of the API response format. To learn more about UUIDs see the [GDC Data Model documentation](#).

Submitter IDs

In addition to `id`, many entities also include a `submitter_id` field. This field can contain any string (e.g. a “barcode”) that the submitter wishes to use to identify the entity. Typically this string identifies a corresponding entry in submitter’s records. The GDC’s only requirement with respect to `submitter_id` is that it be a string that is unique for all entities within a project. The GDC Submission API requires a `submitter_id` for most entities.

Note: For `case` entities, `submitter_id` must correspond to a `submitted_subject_id` of a study participant registered with the project in dbGaP.

GDC Data Dictionary Endpoints

Information in the GDC Data Dictionary can be accessed programmatically as described below.

Submission Templates

Submission templates are accessible programmatically at the `templates` endpoint. Template format (`json`, `tsv` or `csv`) is specified using the `format` parameter.

For example, the JSON template for `case` entities can be obtained from:

```
1 https://api.gdc.cancer.gov/v0/submission/template/case?format=json
```

In addition to `case`, templates for the following entities can be downloaded

Biospecimen:

```
1 sample
2 portion
3 analyte
4 aliquot
5 read_group
```

Clinical:

```
1 slide
2 demographic
3 diagnosis
4 exposure
5 family_history
6 treatment
7 follow_up
8 molecular_test
```

Data Files:

```
1 analysis_metadata
2 biospecimen_supplement
3 clinical_supplement
4 experiment_metadata
5 pathology_report
```

```
6 run_metadata
7 slide_image
8 submitted_unaligned_reads
9 submitted_aligned_reads
10 submitted_genomic_profile
```

Entity JSON Schemas

The entire collection of GDC entity schemas can be downloaded from the `dictionary` endpoint:

```
1 https://api.gdc.cancer.gov/v0/submission/_dictionary/_all
```

Individual schemas can be downloaded by specifying entity type. For example, the JSON schema for `case` entities can be found at:

```
1 https://api.gdc.cancer.gov/v0/submission/_dictionary/case
```

Making Requests to the Submission API

Requests to create or update entities in the GDC must specify the entity `type`, the entity `id` or `submitter_id`, relationships (links) that the entity has to existing entities in the GDC Data Model, and entity properties as defined by the GDC Data Dictionary. To delete entities, only the `id` property is required. The general format of GDC API submission requests and responses is provided in Appendix C.

Submission Transactions

Submission of data to the GDC involves a series of transactions initiated by the submitter, that create and link entities according to the GDC Data Model. With the exception of `program`, which is an administrative entity created by the GDC, all new entities must be linked, at creation, to existing entities or to new entities being created in the same transaction. For example, a submitter cannot create a `portion` entity unless the submitter either (1) has previously created the corresponding `case` and `sample` entities, or (2) is creating those entities in the same transaction. This also means that entities cannot be deleted if they have “child” entities attached to them.

If multiple entities are being created and/or updated in a transaction, and an error is encountered for one of the entities, then the transaction will fail and no changes will be made to the GDC.

Dry Run Transactions

The `submission` endpoint provides a `_dry_run` mode that simulates submission transactions without making changes to the GDC. This mode is activated by appending `_dry_run` to the end of a submission endpoint.

The following is an example of a POST request, that simulates creating an entity in dry run mode:

```
1 {
2   "project_id": "TCGA-ALCH",
3   "type": "case",
4   "submitter_id": "TCGA-ALCH-000001",
5   "projects": {
6     "code": "ALCH"
7   }
8 }
```

```
1 token=$(cat gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request POST --data-binary @Request --header 'Content-Type:
   application/json' https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH/_dry_run
```

```

1 {
2   "cases_related_to_created_entities_count": 0,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 200,
5   "created_entity_count": 1,
6   "entities": [
7     {
8       "action": "create",
9       "errors": [],
10      "id": "61f48d1c-9439-448c-a90c-d6dbe76b3654",
11      "related_cases": [],
12      "type": "case",
13      "unique_keys": [
14        {
15          "project_id": "TCGA-ALCH",
16          "submitter_id": "TCGA-ALCH-000001"
17        }
18      ],
19      "valid": true,
20      "warnings": []
21    }
22  ],
23  "entity_error_count": 0,
24  "message": "Transaction would have been successful. User selected dry run option, transaction aborted,
25            no data written to database.",
26  "success": true,
27  "transaction_id": null,
28  "transactional_error_count": 0,
29  "transactional_errors": [],
30  "updated_entity_count": 0
}

```

Dry Run Commit

For convenience, the GDC enables users to commit earlier `_dry_run` transactions instead of uploading the same data again to execute the changes. This `commit` action is allowed on transactions that (1) have not been previously committed and (2) were successful `_dry_run` transactions.

Note that the `commit` action is a separate transaction with its own transaction id, and it can be executed **asynchronously**. If the state of the submission project has changed in a way that would make the original `_dry_run` transaction invalid if it were run again (e.g. entities with the same `submitter_id` have since been created in another transaction), then then `commit` action will fail.

To commit a transaction, submit a POST or PUT request to `/submission/Program.name/Project.code/transactions/transaction_id/commit`, replacing `Program.name`, `Project.code`, and `transaction_id` with values associated with the transaction.

```

1 token=$(<gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request POST
4     https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH/transactions/467/commit?async=true

```

```

1 {
2   "code": 200,
3   "message": "Transaction submitted.",
4   "transaction_id": 468,
5 }

```

Dry Run Close

The GDC Submission API also provides a `close` action on `_dry_run` transactions. This `close` action is allowed on `_dry_run` transactions that have not been previously closed. Closing a `_dry_run` transaction prevents it from being committed in the future.

To close a transaction, submit a POST or PUT request to `/submission/Program.name/Project.code/transactions/transaction_id/`, replacing `Program.name`, `Project.code`, and `transaction_id` with values associated with the transaction.

```
1 token=$(<gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request POST
   https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH/transactions/467/close
```

```
1 {
2   "code": 200,
3   "message": "Closed transaction.",
4   "transaction_id": 467
5 }
```

Asynchronous Transactions

The `submission` endpoint provides an asynchronous mode that provides immediate response and executes submission transactions in the background. This mode is activated by appending `?async=true` to the end of a submission endpoint. The API will respond with the `transaction_id` which can be used to look up the result of the transaction at a later time via the GraphQL endpoint. If the server has too many asynchronous jobs scheduled already, your request to schedule a transaction may fail.

Example

The following is an example of a PUT request, that creates a case asynchronously:

```
1 {
2   "project_id": "TCGA-ALCH",
3   "type": "case",
4   "submitter_id": "TCGA-ALCH-000001",
5   "projects": {
6     "code": "ALCH"
7   }
8 }
```

```
1 token=$(<gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request POST --data-binary @Request --header 'Content-Type:
   application/json' https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH?async=true
```

```
1 {
2   "code": 200,
3   "message": "Transaction submitted.",
4   "transaction_id": 467,
5 }
```

The following is a GraphQL request that looks up the state of the above transaction:

```
1 query {
2   transaction_log(id: 467) {
3     is_dry_run
4     committed_by
5     state
6   }
7 }
```

```

1 {
2   "data": {
3     "transaction_log": [
4       {
5         "committed_by": null,
6         "is_dry_run": false,
7         "state": "FAILED"
8       }
9     ]
10  }
11 }

```

Transaction Status

The following transaction fields can be queried using GraphQL and are helpful in determining the status of a transaction:

Field	Type	Description
<code>id</code>	ID	Transaction identifier
<code>is_dry_run</code>	Boolean	Indicates whether the transaction is a dry run
<code>closed</code>	Boolean	For dry run transactions, indicates whether the transaction has been closed to prevent it from being committed in the future.
<code>committable</code>	Boolean	Indicates whether the transaction can be committed (i.e. it is a successful dry run transaction that has not been committed previously and has not been closed)
<code>state</code>	String	Indicates the state of the transaction: <code>PENDING</code> , <code>SUCCEEDED</code> , <code>FAILED</code> (due to user error), or <code>ERRORED</code> (due to system error)
<code>committed_by</code>	ID	The ID of the transaction that committed this transaction

Note: To check whether a dry run transaction was committed successfully, check the `state` of the transaction that executed the commit. The `state` of the dry run transaction itself does not represent the status of a subsequent commit.

Creating and Updating Entities

The GDC Submission API supports HTTP POST and HTTP PUT methods for creating entities:

- **POST** will create entities that do not exist, and will fail if any of the entities in the transaction already exist in the GDC.
- **PUT** will create new entities and update existing entities, and identify which entities were created or updated in the API response.

The GDC suggests using POST for creating new entities, and using PUT only for updating entities. This helps to avoid inadvertent entity updates that can occur when using PUT for creating entities.

Note: Once a relationship has been created between two entities, it cannot be removed by updating an entity. To remove a relationship, the child entity must be **deleted**.

Example: Creating and Updating Case Entities (JSON)

In this example, a case entity is created using POST. Then an attempt is made to create the same entity again using POST, resulting in an error. Then the originally created entity is updated (with the same information) using PUT.

The JSON in the request was generated using the case JSON template that can be obtained from the GDC Data Dictionary Viewer and from <https://api.gdc.cancer.gov/v0/submission/template/case?format=json>.

Note: For case entities, `submitter_id` must correspond to a `submitted_subject_id` of a study participant registered with the project in dbGaP.

```
1 {
2   "type": "case",
3   "submitter_id": "TCGA-ALCH-000001",
4   "projects": {
5     "code": "ALCH"
6   }
7
8 }
```

```
1 token=$(cat gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request POST --data-binary @Request --header 'Content-Type:
   application/json' https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH
```

```
1 {
2   "cases_related_to_created_entities_count": 0,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 201,
5   "created_entity_count": 1,
6   "entities": [
7     {
8       "action": "create",
9       "errors": [],
10      "id": "fbf69646-5904-4f95-92d6-692bde658f05",
11      "related_cases": [],
12      "type": "case",
13      "unique_keys": [
14        {
15          "project_id": "TCGA-ALCH",
16          "submitter_id": "TCGA-ALCH-000001"
17        }
18      ],
19      "valid": true,
20      "warnings": []
21    }
22  ],
23   "entity_error_count": 0,
24   "message": "Transaction successful.",
25   "success": true,
26   "transaction_id": 215,
27   "transactional_error_count": 0,
28   "transactional_errors": [],
29   "updated_entity_count": 0
30 }
```

```
1 curl --header "X-Auth-Token: $token" --request POST --data-binary @Request --header 'Content-Type:
   application/json' https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH
```

```
1 {
2   "cases_related_to_created_entities_count": 0,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 400,
```

```

5  "created_entity_count": 0,
6  "entities": [
7    {
8      "action": null,
9      "errors": [
10     {
11       "keys": [
12         "id"
13       ],
14       "message": "Cannot create entity that already exists. Try updating entity (PUT instead of
15         POST)",
16       "type": "NOT_UNIQUE"
17     }
18   ],
19   "id": null,
20   "related_cases": [],
21   "type": "case",
22   "unique_keys": [
23     {
24       "project_id": "TCGA-ALCH",
25       "submitter_id": "TCGA-ALCH-000001"
26     }
27   ],
28   "valid": false,
29   "warnings": []
30 }
31 ],
32 "entity_error_count": 1,
33 "message": "Transaction aborted due to 1 invalid entity.",
34 "success": false,
35 "transaction_id": null,
36 "transactional_error_count": 0,
37 "transactional_errors": [],
38 "updated_entity_count": 0
39 }

```

```

1 curl --header "X-Auth-Token: $token" --request PUT --data-binary @Request --header 'Content-Type:
2 application/json' https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH

```

```

1 {
2   "cases_related_to_created_entities_count": 0,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 200,
5   "created_entity_count": 0,
6   "entities": [
7     {
8       "action": "update",
9       "errors": [],
10      "id": "fbf69646-5904-4f95-92d6-692bde658f05",
11      "related_cases": [],
12      "type": "case",
13      "unique_keys": [
14        {
15          "project_id": "TCGA-ALCH",
16          "submitter_id": "TCGA-ALCH-000001"
17        }
18      ],
19      "valid": true,

```

```

20     "warnings": []
21   }
22 ],
23 "entity_error_count": 0,
24 "message": "Transaction successful.",
25 "success": true,
26 "transaction_id": 216,
27 "transactional_error_count": 0,
28 "transactional_errors": [],
29 "updated_entity_count": 1
30 }

```

Example: Creating an Aliquot Entity (JSON)

In this example, an `aliquot` entity and a `sample` entity are created in a single transaction. The `aliquot` is linked to `sample` which is linked to `case`. The first request is an example of using `submitter_id` properties to link entities together. The second request is an example of using UUIDs for creating the links.

Request 1: Creating Links Using `submitter_id`

```

1 [
2   {
3     "type": "sample",
4     "submitter_id": "TCGA-ALCH-000001-SAMPLE000001",
5     "sample_type": "Primary Tumor",
6     "sample_type_id": "01",
7     "cases": {
8       "submitter_id": "TCGA-ALCH-000001"
9     }
10  },
11  {
12     "type": "aliquot",
13     "submitter_id": "TCGA-ALCH-000001-SAMPLE000001-ALIQUOT000001",
14     "samples": {
15       "submitter_id": "TCGA-ALCH-000001-SAMPLE000001"
16     }
17  }
18 ]

```

```

1 token=$(cat gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request POST --data-binary @Request --header 'Content-Type:
  application/json' https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH

```

```

1 {
2   "cases_related_to_created_entities_count": 1,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 201,
5   "created_entity_count": 2,
6   "entities": [
7     {
8       "action": "create",
9       "errors": [],
10      "id": "48270338-6464-448f-bbef-b09d4f80b11b",
11      "related_cases": [
12        {

```



```

13     "id": "fbf69646-5904-4f95-92d6-692bde658f05",
14     "submitter_id": "TCGA-ALCH-000001"
15   }
16 ],
17 "type": "sample",
18 "unique_keys": [
19   {
20     "project_id": "TCGA-ALCH",
21     "submitter_id": "TCGA-ALCH-000001-SAMPLE000001"
22   }
23 ],
24 "valid": true,
25 "warnings": []
26 },
27 {
28   "action": "create",
29   "errors": [],
30   "id": "7af58da0-cb3e-43e2-a074-4bd8f27565ba",
31   "related_cases": [
32     {
33       "id": "fbf69646-5904-4f95-92d6-692bde658f05",
34       "submitter_id": "TCGA-ALCH-000001"
35     }
36   ],
37   "type": "aliquot",
38   "unique_keys": [
39     {
40       "project_id": "TCGA-ALCH",
41       "submitter_id": "TCGA-ALCH-000001-SAMPLE000001-ALIQUOT000001"
42     }
43   ],
44   "valid": true,
45   "warnings": []
46   }
47 ],
48 "entity_error_count": 0,
49 "message": "Transaction successful.",
50 "success": true,
51 "transaction_id": 222,
52 "transactional_error_count": 0,
53 "transactional_errors": [],
54 "updated_entity_count": 0
55 }

```

Request 2: Creating Links Using UUID

```

1 [
2   {
3     "type": "sample",
4     "submitter_id": "TCGA-ALCH-000001-SAMPLE000001",
5     "id": "2aa7a07b-e706-4eef-aeba-b849972423a0",
6     "sample_type": "Primary Tumor",
7     "sample_type_id": "01",
8     "cases": {
9       "id": "fbf69646-5904-4f95-92d6-692bde658f05"
10    }
11  },
12  {

```

```

13   "type": "aliquot",
14   "submitter_id": "TCGA-ALCH-000001-SAMPLE000001-ALIQUOT000001",
15   "samples": {
16     "id": "2aa7a07b-e706-4eef-aeba-b849972423a0"
17   }
18 }
19 ]

```

```

1 token=$(cat gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request POST --data-binary @Request --header 'Content-Type:
  application/json' https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH

```

```

1 {
2   "cases_related_to_created_entities_count": 1,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 201,
5   "created_entity_count": 2,
6   "entities": [
7     {
8       "action": "create",
9       "errors": [],
10      "id": "2aa7a07b-e706-4eef-aeba-b849972423a0",
11      "related_cases": [
12        {
13          "id": "fbf69646-5904-4f95-92d6-692bde658f05",
14          "submitter_id": "TCGA-ALCH-000001"
15        }
16      ],
17      "type": "sample",
18      "unique_keys": [
19        {
20          "project_id": "TCGA-ALCH",
21          "submitter_id": "TCGA-ALCH-000001-SAMPLE000001"
22        }
23      ],
24      "valid": true,
25      "warnings": []
26    },
27    {
28      "action": "create",
29      "errors": [],
30      "id": "545096d5-ce1c-433f-80f0-fd0b04b56cb6",
31      "related_cases": [
32        {
33          "id": "fbf69646-5904-4f95-92d6-692bde658f05",
34          "submitter_id": "TCGA-ALCH-000001"
35        }
36      ],
37      "type": "aliquot",
38      "unique_keys": [
39        {
40          "project_id": "TCGA-ALCH",
41          "submitter_id": "TCGA-ALCH-000001-SAMPLE000001-ALIQUOT000001"
42        }
43      ],
44      "valid": true,
45      "warnings": []

```

```

46   }
47 ],
48 "entity_error_count": 0,
49 "message": "Transaction successful.",
50 "success": true,
51 "transaction_id": 219,
52 "transactional_error_count": 0,
53 "transactional_errors": [],
54 "updated_entity_count": 0
55 }

```

Example: Creating Two Samples (TSV)

In this example, a TSV file containing metadata for two samples is uploaded to the GDC in dry run mode.

```

1 type   project_id submitter_id  cases.submitter_id sample_type sample_type_id tumor_descriptor
2 sample GDC-INTERNAL  GDC-INTERNAL-000022-sampleA GDC-INTERNAL-000022 Additional Metastatic 01
3 sample GDC-INTERNAL  GDC-INTERNAL-000022-sampleB GDC-INTERNAL-000022 Solid Tissue Normal 02

```

```

1 curl --header "X-Auth-Token: $token" --header 'Content-Type: text/tsv' --request PUT --data-binary
   @Samples.tsv 'https://api.gdc.cancer.gov/submission/GDC/INTERNAL/_dry_run'

```

```

1 {
2   "cases_related_to_created_entities_count": 1,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 200,
5   "created_entity_count": 2,
6   "entities": [
7     {
8       "action": "create",
9       "errors": [],
10      "id": "b55e10af-5b7f-48f1-b230-0f8e6b7a7afe",
11      "related_cases": [
12        {
13          "id": "6e2e3b31-c5d2-45df-a911-eb3577640b70",
14          "submitter_id": "GDC-INTERNAL-000022"
15        }
16      ],
17      "type": "sample",
18      "unique_keys": [
19        {
20          "project_id": "GDC-INTERNAL",
21          "submitter_id": "GDC-INTERNAL-000022-sampleA"
22        }
23      ],
24      "valid": true,
25      "warnings": []
26    },
27    {
28      "action": "create",
29      "errors": [],
30      "id": "15076660-fccc-4406-b981-c745eb992034",
31      "related_cases": [
32        {
33          "id": "6e2e3b31-c5d2-45df-a911-eb3577640b70",
34          "submitter_id": "GDC-INTERNAL-000022"
35        }
36      ],

```

```

37     "type": "sample",
38     "unique_keys": [
39       {
40         "project_id": "GDC-INTERNAL",
41         "submitter_id": "GDC-INTERNAL-000022-sampleB"
42       }
43     ],
44     "valid": true,
45     "warnings": []
46   }
47 ],
48 "entity_error_count": 0,
49 "message": "Transaction would have been successful. User selected dry run option, transaction aborted,
50   no data written to database.",
51 "success": true,
52 "transaction_id": 51284,
53 "transactional_error_count": 0,
54 "transactional_errors": [],
55 "updated_entity_count": 0
56 }

```

Example: Updating a Sample Entity (JSON)

Entities can be updated using a very similar process to what is shown above.

Updating a sample

New nodes are created in Request1. Nodes in state `validated` are updated in Request2.

```

1 [
2   {
3     "type": "case",
4     "submitter_id": "QA-REGRESSION-0002",
5     "projects": {
6       "code": "REGRESSION"
7     }
8   },
9   {
10    "type": "sample",
11    "submitter_id": "QA-REGRESSION-0002-SAMPLE000001",
12    "sample_type": "Primary Tumor",
13    "sample_type_id": "01",
14    "cases": {
15      "submitter_id": "QA-REGRESSION-0002"
16    }
17  },
18  {
19    "type": "aliquot",
20    "submitter_id": "QA-REGRESSION-0002-SAMPLE000001-ALIQUOT000001",
21    "samples": {
22      "submitter_id": "QA-REGRESSION-0002-SAMPLE000001"
23    }
24  }
25 ]

```

```
1 token=$(<gdc-token-text-file.txt)
```

```
2
3 curl --header "X-Auth-Token: $token" --request POST --data-binary @sample.json --header 'Content-Type:
  application/json' https://api.gdc.cancer.gov/v0/submission/QA/REGRESSION
```

```
1 {
2   "cases_related_to_created_entities_count": 1,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 201,
5   "created_entity_count": 3,
6   "entities": [
7     {
8       "action": "create",
9       "errors": [],
10      "id": "3a750ae8-8e63-472e-852e-8e514a0c1550",
11      "related_cases": [],
12      "type": "case",
13      "unique_keys": [
14        {
15          "project_id": "QA-REGRESSION",
16          "submitter_id": "QA-REGRESSION-0002"
17        }
18      ],
19      "valid": true,
20      "warnings": []
21    },
22    {
23      "action": "create",
24      "errors": [],
25      "id": "8a1872e6-c5e6-4f39-b9fe-15ecf45715c7",
26      "related_cases": [
27        {
28          "id": "3a750ae8-8e63-472e-852e-8e514a0c1550",
29          "submitter_id": "QA-REGRESSION-0002"
30        }
31      ],
32      "type": "sample",
33      "unique_keys": [
34        {
35          "project_id": "QA-REGRESSION",
36          "submitter_id": "QA-REGRESSION-0002-SAMPLE000001"
37        }
38      ],
39      "valid": true,
40      "warnings": []
41    },
42    {
43      "action": "create",
44      "errors": [],
45      "id": "e9279137-92b4-41ab-be28-a03e32e6fac7",
46      "related_cases": [
47        {
48          "id": "3a750ae8-8e63-472e-852e-8e514a0c1550",
49          "submitter_id": "QA-REGRESSION-0002"
50        }
51      ],
52      "type": "aliquot",
53      "unique_keys": [
54        {
```

```

55     "project_id": "QA-REGRESSION",
56     "submitter_id": "QA-REGRESSION-0002-SAMPLE000001-ALIQUOTO00001"
57   }
58 ],
59   "valid": true,
60   "warnings": []
61 }
62 ],
63 "entity_error_count": 0,
64 "message": "Transaction successful.",
65 "success": true,
66 "transaction_id": 920117,
67 "transactional_error_count": 0,
68 "transactional_errors": [],
69 "updated_entity_count": 0
70 }

```

```

1 [
2   {
3     "type": "case",
4     "submitter_id": "QA-REGRESSION-0002",
5     "projects": {
6       "code": "REGRESSION"
7     }
8   },
9   {
10    "type": "sample",
11    "submitter_id": "QA-REGRESSION-0002-SAMPLE000001",
12    "sample_type": "Primary Tumor",
13    "days_to_collection": 5,
14    "sample_type_id": "01",
15    "cases": {
16      "submitter_id": "QA-REGRESSION-0002"
17    }
18  },
19  {
20    "type": "aliquot",
21    "submitter_id": "QA-REGRESSION-0002-SAMPLE000001-ALIQUOTO00001",
22    "samples": {
23      "submitter_id": "QA-REGRESSION-0002-SAMPLE000001"
24    }
25  }
26 ]

```

```

1 token=$(cat gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request PUT --data-binary @sample2.json --header 'Content-Type:
  application/json' https://api.gdc.cancer.gov/v0/submission/QA/REGRESSION

```

```

1 {
2   "cases_related_to_created_entities_count": 0,
3   "cases_related_to_updated_entities_count": 1,
4   "code": 200,
5   "created_entity_count": 0,
6   "entities": [
7     {
8       "action": "update",
9       "errors": [],

```

```

10 "id": "3a750ae8-8e63-472e-852e-8e514a0c1550",
11 "related_cases": [],
12 "type": "case",
13 "unique_keys": [
14   {
15     "project_id": "QA-REGRESSION",
16     "submitter_id": "QA-REGRESSION-0002"
17   }
18 ],
19 "valid": true,
20 "warnings": []
21 },
22 {
23   "action": "update",
24   "errors": [],
25   "id": "8a1872e6-c5e6-4f39-b9fe-15ecf45715c7",
26   "related_cases": [
27     {
28       "id": "3a750ae8-8e63-472e-852e-8e514a0c1550",
29       "submitter_id": "QA-REGRESSION-0002"
30     }
31   ],
32   "type": "sample",
33   "unique_keys": [
34     {
35       "project_id": "QA-REGRESSION",
36       "submitter_id": "QA-REGRESSION-0002-SAMPLE000001"
37     }
38   ],
39   "valid": true,
40   "warnings": []
41 },
42 {
43   "action": "update",
44   "errors": [],
45   "id": "e9279137-92b4-41ab-be28-a03e32e6fac7",
46   "related_cases": [
47     {
48       "id": "3a750ae8-8e63-472e-852e-8e514a0c1550",
49       "submitter_id": "QA-REGRESSION-0002"
50     }
51   ],
52   "type": "aliquot",
53   "unique_keys": [
54     {
55       "project_id": "QA-REGRESSION",
56       "submitter_id": "QA-REGRESSION-0002-SAMPLE000001-ALIQUOT000001"
57     }
58   ],
59   "valid": true,
60   "warnings": []
61 }
62 ],
63 "entity_error_count": 0,
64 "message": "Transaction successful.",
65 "success": true,
66 "transaction_id": 920120,
67 "transactional_error_count": 0,

```

```
68 "transactional_errors": [],
69 "updated_entity_count": 3
70 }
```

Updating a sample using a BCR XML

Entities are created in Command1. These entities are later released. Command2 demonstrates updating entity information via XML submission.

```
1 curl --request PUT --header "X-Auth-Token: $token" --header 'Content-Type: application/xml'
   --data-binary @BCR_biospecimen.xml
   'https://api.gdc.cancer.gov/v0/submission/QA/REGRESSION/xml/biospecimen/bcr/'
```

```
1 {{
2   "cases_related_to_created_entities_count": 1,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 200,
5   "created_entity_count": 28,
6   "entities": [
7     {
8       "action": "create",
9       "errors": [],
10      "id": "b69b96e0-4b45-5a99-b862-97ab9cdf0c88",
11      "related_cases": [
12        {
13          "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
14          "submitter_id": "QA-REGRESSION-0003"
15        }
16      ],
17      "type": "aliquot",
18      "unique_keys": [
19        {
20          "project_id": "QA-REGRESSION",
21          "submitter_id": "TCGA-BP-5184-01A-01D-2101-10"
22        }
23      ],
24      "valid": true,
25      "warnings": []
26    },
27    {
28      "action": "create",
29      "errors": [],
30      "id": "b1118ebf-7e66-5823-a6ce-eeeb7547aad7",
31      "related_cases": [
32        {
33          "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
34          "submitter_id": "QA-REGRESSION-0003"
35        }
36      ],
37      "type": "slide",
38      "unique_keys": [
39        {
40          "project_id": "QA-REGRESSION",
41          "submitter_id": "TCGA-BP-5184-11A-01-TS1"
42        }
43      ],
44      "valid": true,
```



```

45     "warnings": []
46 },
47 {
48     "action": "create",
49     "errors": [],
50     "id": "d6a4b230-bbd6-5b1e-88ae-476ec1ade556",
51     "related_cases": [
52         {
53             "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
54             "submitter_id": "QA-REGRESSION-0003"
55         }
56     ],
57     "type": "aliquot",
58     "unique_keys": [
59         {
60             "project_id": "QA-REGRESSION",
61             "submitter_id": "TCGA-BP-5184-01A-01D-1422-02"
62         }
63     ],
64     "valid": true,
65     "warnings": []
66 },
67 {
68     "action": "create",
69     "errors": [],
70     "id": "9861336b-0359-521d-81d1-423262ef2560",
71     "related_cases": [
72         {
73             "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
74             "submitter_id": "QA-REGRESSION-0003"
75         }
76     ],
77     "type": "analyte",
78     "unique_keys": [
79         {
80             "project_id": "QA-REGRESSION",
81             "submitter_id": "TCGA-BP-5184-01A-01D"
82         }
83     ],
84     "valid": true,
85     "warnings": []
86 },
87 {
88     "action": "create",
89     "errors": [],
90     "id": "151b99c8-16fa-5074-a2f5-2aef7c43d0ec",
91     "related_cases": [
92         {
93             "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
94             "submitter_id": "QA-REGRESSION-0003"
95         }
96     ],
97     "type": "analyte",
98     "unique_keys": [
99         {
100             "project_id": "QA-REGRESSION",
101             "submitter_id": "TCGA-BP-5184-11A-01D"
102         }

```

```

103     ],
104     "valid": true,
105     "warnings": []
106   },
107   {
108     "action": "create",
109     "errors": [],
110     "id": "2baf9dc2-649b-5182-8592-feef9fb72b5a",
111     "related_cases": [
112       {
113         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
114         "submitter_id": "QA-REGRESSION-0003"
115       }
116     ],
117     "type": "portion",
118     "unique_keys": [
119       {
120         "project_id": "QA-REGRESSION",
121         "submitter_id": "TCGA-BP-5184-11A-01"
122       }
123     ],
124     "valid": true,
125     "warnings": []
126   },
127   {
128     "action": "create",
129     "errors": [],
130     "id": "01d8e68d-7299-5b10-b545-afab0144ca1c",
131     "related_cases": [
132       {
133         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
134         "submitter_id": "QA-REGRESSION-0003"
135       }
136     ],
137     "type": "aliquot",
138     "unique_keys": [
139       {
140         "project_id": "QA-REGRESSION",
141         "submitter_id": "TCGA-BP-5184-11A-01D-1423-01"
142       }
143     ],
144     "valid": true,
145     "warnings": []
146   },
147   {
148     "action": "create",
149     "errors": [],
150     "id": "e13264ea-2ff1-52d6-b781-65d21c2d24a9",
151     "related_cases": [
152       {
153         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
154         "submitter_id": "QA-REGRESSION-0003"
155       }
156     ],
157     "type": "aliquot",
158     "unique_keys": [
159       {
160         "project_id": "QA-REGRESSION",

```

```

161     "submitter_id": "TCGA-BP-5184-01A-01R-1425-13"
162   }
163 ],
164 "valid": true,
165 "warnings": []
166 },
167 {
168   "action": "create",
169   "errors": [],
170   "id": "a7c6332c-04ad-5563-8ca6-51813122bc03",
171   "related_cases": [
172     {
173       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
174       "submitter_id": "QA-REGRESSION-0003"
175     }
176   ],
177   "type": "aliquot",
178   "unique_keys": [
179     {
180       "project_id": "QA-REGRESSION",
181       "submitter_id": "TCGA-BP-5184-11A-01D-2101-10"
182     }
183   ],
184   "valid": true,
185   "warnings": []
186 },
187 {
188   "action": "create",
189   "errors": [],
190   "id": "69f19d8d-13a5-5f0d-ac82-d4de075fd51a",
191   "related_cases": [
192     {
193       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
194       "submitter_id": "QA-REGRESSION-0003"
195     }
196   ],
197   "type": "aliquot",
198   "unique_keys": [
199     {
200       "project_id": "QA-REGRESSION",
201       "submitter_id": "TCGA-BP-5184-11A-01D-1429-08"
202     }
203   ],
204   "valid": true,
205   "warnings": []
206 },
207 {
208   "action": "create",
209   "errors": [],
210   "id": "11251eba-5a20-5747-8d30-0eec4d8fd8e2",
211   "related_cases": [
212     {
213       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
214       "submitter_id": "QA-REGRESSION-0003"
215     }
216   ],
217   "type": "analyte",
218   "unique_keys": [

```

```

219     {
220         "project_id": "QA-REGRESSION",
221         "submitter_id": "TCGA-BP-5184-01A-01W"
222     }
223 ],
224 "valid": true,
225 "warnings": []
226 },
227 {
228     "action": "create",
229     "errors": [],
230     "id": "2c038da9-5373-538a-bd15-38522ef49ab1",
231     "related_cases": [
232         {
233             "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
234             "submitter_id": "QA-REGRESSION-0003"
235         }
236     ],
237     "type": "sample",
238     "unique_keys": [
239         {
240             "project_id": "QA-REGRESSION",
241             "submitter_id": "TCGA-BP-5184-11A"
242         }
243     ],
244     "valid": true,
245     "warnings": []
246 },
247 {
248     "action": "create",
249     "errors": [],
250     "id": "d012beaf-5e0c-559a-b8d5-d4960912774b",
251     "related_cases": [
252         {
253             "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
254             "submitter_id": "QA-REGRESSION-0003"
255         }
256     ],
257     "type": "sample",
258     "unique_keys": [
259         {
260             "project_id": "QA-REGRESSION",
261             "submitter_id": "TCGA-BP-5184-01A"
262         }
263     ],
264     "valid": true,
265     "warnings": []
266 },
267 {
268     "action": "create",
269     "errors": [],
270     "id": "5ca816e8-1325-5986-beaf-8f300caa42c3",
271     "related_cases": [
272         {
273             "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
274             "submitter_id": "QA-REGRESSION-0003"
275         }
276     ],

```

```

277 "type": "slide",
278 "unique_keys": [
279   {
280     "project_id": "QA-REGRESSION",
281     "submitter_id": "TCGA-BP-5184-01A-01-TS1"
282   }
283 ],
284 "valid": true,
285 "warnings": []
286 },
287 {
288   "action": "create",
289   "errors": [],
290   "id": "fe1047d0-82ae-513e-8b3c-353f5022ecc0",
291   "related_cases": [
292     {
293       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
294       "submitter_id": "QA-REGRESSION-0003"
295     }
296   ],
297   "type": "portion",
298   "unique_keys": [
299     {
300       "project_id": "QA-REGRESSION",
301       "submitter_id": "TCGA-BP-5184-01A-01"
302     }
303   ],
304   "valid": true,
305   "warnings": []
306 },
307 {
308   "action": "create",
309   "errors": [],
310   "id": "6d4054dd-c30a-5572-af39-cb6910a67bcc",
311   "related_cases": [
312     {
313       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
314       "submitter_id": "QA-REGRESSION-0003"
315     }
316   ],
317   "type": "aliquot",
318   "unique_keys": [
319     {
320       "project_id": "QA-REGRESSION",
321       "submitter_id": "TCGA-BP-5184-01A-01W-1477-10"
322     }
323   ],
324   "valid": true,
325   "warnings": []
326 },
327 {
328   "action": "create",
329   "errors": [],
330   "id": "6684b446-ece3-57c1-93b2-a4885f829707",
331   "related_cases": [
332     {
333       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
334       "submitter_id": "QA-REGRESSION-0003"

```

```

335     }
336 ],
337 "type": "aliquot",
338 "unique_keys": [
339   {
340     "project_id": "QA-REGRESSION",
341     "submitter_id": "TCGA-BP-5184-11A-01D-1424-05"
342   }
343 ],
344 "valid": true,
345 "warnings": []
346 },
347 {
348   "action": "create",
349   "errors": [],
350   "id": "f52da5fd-2267-517c-b09c-d7d3074e4f13",
351   "related_cases": [
352     {
353       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
354       "submitter_id": "QA-REGRESSION-0003"
355     }
356   ],
357   "type": "aliquot",
358   "unique_keys": [
359     {
360       "project_id": "QA-REGRESSION",
361       "submitter_id": "TCGA-BP-5184-01A-01D-1429-08"
362     }
363   ],
364   "valid": true,
365   "warnings": []
366 },
367 {
368   "action": "create",
369   "errors": [],
370   "id": "778d5545-f35d-54a1-bd7e-e148aa048046",
371   "related_cases": [
372     {
373       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
374       "submitter_id": "QA-REGRESSION-0003"
375     }
376   ],
377   "type": "portion",
378   "unique_keys": [
379     {
380       "project_id": "QA-REGRESSION",
381       "submitter_id": "TCGA-BP-5184-01A-21-1740-20"
382     }
383   ],
384   "valid": true,
385   "warnings": []
386 },
387 {
388   "action": "create",
389   "errors": [],
390   "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
391   "related_cases": [],
392   "type": "case",

```

```

393 "unique_keys": [
394   {
395     "project_id": "QA-REGRESSION",
396     "submitter_id": "QA-REGRESSION-0003"
397   }
398 ],
399 "valid": true,
400 "warnings": []
401 },
402 {
403   "action": "create",
404   "errors": [],
405   "id": "e1a7ac18-8473-56f4-8f49-42c427f2b6ff",
406   "related_cases": [
407     {
408       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
409       "submitter_id": "QA-REGRESSION-0003"
410     }
411   ],
412   "type": "aliquot",
413   "unique_keys": [
414     {
415       "project_id": "QA-REGRESSION",
416       "submitter_id": "TCGA-BP-5184-11A-01D-1422-02"
417     }
418   ],
419   "valid": true,
420   "warnings": []
421 },
422 {
423   "action": "create",
424   "errors": [],
425   "id": "f2531646-dc01-5d89-9ecb-3031716adb96",
426   "related_cases": [
427     {
428       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
429       "submitter_id": "QA-REGRESSION-0003"
430     }
431   ],
432   "type": "aliquot",
433   "unique_keys": [
434     {
435       "project_id": "QA-REGRESSION",
436       "submitter_id": "TCGA-BP-5184-01A-01R-1426-07"
437     }
438   ],
439   "valid": true,
440   "warnings": []
441 },
442 {
443   "action": "create",
444   "errors": [],
445   "id": "9682a18b-7d3e-514b-9bf2-666b4f697140",
446   "related_cases": [
447     {
448       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
449       "submitter_id": "QA-REGRESSION-0003"
450     }

```

```

451 ],
452 "type": "aliquot",
453 "unique_keys": [
454   {
455     "project_id": "QA-REGRESSION",
456     "submitter_id": "TCGA-BP-5184-01A-01D-1424-05"
457   }
458 ],
459 "valid": true,
460 "warnings": []
461 },
462 {
463   "action": "create",
464   "errors": [],
465   "id": "91bc6c5c-b340-52fb-9ab2-fec7e9312cf6",
466   "related_cases": [
467     {
468       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
469       "submitter_id": "QA-REGRESSION-0003"
470     }
471   ],
472   "type": "analyte",
473   "unique_keys": [
474     {
475       "project_id": "QA-REGRESSION",
476       "submitter_id": "TCGA-BP-5184-11A-01W"
477     }
478   ],
479   "valid": true,
480   "warnings": []
481 },
482 {
483   "action": "create",
484   "errors": [],
485   "id": "dadca81e-7a7d-5944-abc8-63c8acbdbfd6",
486   "related_cases": [
487     {
488       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
489       "submitter_id": "QA-REGRESSION-0003"
490     }
491   ],
492   "type": "analyte",
493   "unique_keys": [
494     {
495       "project_id": "QA-REGRESSION",
496       "submitter_id": "TCGA-BP-5184-01A-01R"
497     }
498   ],
499   "valid": true,
500   "warnings": []
501 },
502 {
503   "action": "create",
504   "errors": [],
505   "id": "15546b81-fe15-5838-aa1a-97ad29e36b15",
506   "related_cases": [
507     {
508       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",

```



```

509     "submitter_id": "QA-REGRESSION-0003"
510   }
511 ],
512 "type": "aliquot",
513 "unique_keys": [
514   {
515     "project_id": "QA-REGRESSION",
516     "submitter_id": "TCGA-BP-5184-01A-01D-1423-01"
517   }
518 ],
519 "valid": true,
520 "warnings": []
521 },
522 {
523   "action": "create",
524   "errors": [],
525   "id": "3a7c43e0-4582-57dd-8d2b-7b2a92c30b21",
526   "related_cases": [
527     {
528       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
529       "submitter_id": "QA-REGRESSION-0003"
530     }
531   ],
532   "type": "slide",
533   "unique_keys": [
534     {
535       "project_id": "QA-REGRESSION",
536       "submitter_id": "TCGA-BP-5184-01A-01-BS1"
537     }
538   ],
539   "valid": true,
540   "warnings": []
541 },
542 {
543   "action": "create",
544   "errors": [],
545   "id": "ab513f26-8cba-50f2-bac9-59fc50f5c201",
546   "related_cases": [
547     {
548       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
549       "submitter_id": "QA-REGRESSION-0003"
550     }
551   ],
552   "type": "aliquot",
553   "unique_keys": [
554     {
555       "project_id": "QA-REGRESSION",
556       "submitter_id": "TCGA-BP-5184-11A-01W-1477-10"
557     }
558   ],
559   "valid": true,
560   "warnings": []
561 }
562 ],
563 "entity_error_count": 0,
564 "message": "Transaction successful.",
565 "success": true,
566 "transaction_id": 920139,

```

```
567 "transactional_error_count": 0,
568 "transactional_errors": [],
569 "updated_entity_count": 0
570 ]}]
```

1 Command2

```
2 curl --request PUT --header "X-Auth-Token: $token" --header 'Content-Type: application/xml'
  --data-binary @BCR_biospecimen_updated.xml
  'https://api.gdc.cancer.gov/v0/submission/QA/REGRESSION/xml/biospecimen/bcr'
```

```
1 {
2   "cases_related_to_created_entities_count": 0,
3   "cases_related_to_updated_entities_count": 1,
4   "code": 200,
5   "created_entity_count": 0,
6   "entities": [
7     {
8       "action": "update",
9       "errors": [],
10      "id": "b69b96e0-4b45-5a99-b862-97ab9cdf0c88",
11      "related_cases": [
12        {
13          "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
14          "submitter_id": "QA-REGRESSION-0003"
15        }
16      ],
17      "type": "aliquot",
18      "unique_keys": [
19        {
20          "project_id": "QA-REGRESSION",
21          "submitter_id": "TCGA-BP-5184-01A-01D-2101-10"
22        }
23      ],
24      "valid": true,
25      "warnings": []
26    },
27    {
28      "action": "update",
29      "errors": [],
30      "id": "b1118ebf-7e66-5823-a6ce-eeeb7547aad7",
31      "related_cases": [
32        {
33          "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
34          "submitter_id": "QA-REGRESSION-0003"
35        }
36      ],
37      "type": "slide",
38      "unique_keys": [
39        {
40          "project_id": "QA-REGRESSION",
41          "submitter_id": "TCGA-BP-5184-11A-01-TS1"
42        }
43      ],
44      "valid": true,
45      "warnings": []
46    },
47    {
48      "action": "update",
```

```

49 "errors": [],
50 "id": "d6a4b230-bbd6-5b1e-88ae-476ec1ade556",
51 "related_cases": [
52   {
53     "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
54     "submitter_id": "QA-REGRESSION-0003"
55   }
56 ],
57 "type": "aliquot",
58 "unique_keys": [
59   {
60     "project_id": "QA-REGRESSION",
61     "submitter_id": "TCGA-BP-5184-01A-01D-1422-02"
62   }
63 ],
64 "valid": true,
65 "warnings": []
66 },
67 {
68   "action": "update",
69   "errors": [],
70   "id": "9861336b-0359-521d-81d1-423262ef2560",
71   "related_cases": [
72     {
73       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
74       "submitter_id": "QA-REGRESSION-0003"
75     }
76 ],
77 "type": "analyte",
78 "unique_keys": [
79   {
80     "project_id": "QA-REGRESSION",
81     "submitter_id": "TCGA-BP-5184-01A-01D"
82   }
83 ],
84 "valid": true,
85 "warnings": []
86 },
87 {
88   "action": "update",
89   "errors": [],
90   "id": "151b99c8-16fa-5074-a2f5-2aef7c43d0ec",
91   "related_cases": [
92     {
93       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
94       "submitter_id": "QA-REGRESSION-0003"
95     }
96 ],
97 "type": "analyte",
98 "unique_keys": [
99   {
100     "project_id": "QA-REGRESSION",
101     "submitter_id": "TCGA-BP-5184-11A-01D"
102   }
103 ],
104 "valid": true,
105 "warnings": []
106 },

```

```

107 {
108   "action": "update",
109   "errors": [],
110   "id": "2baf9dc2-649b-5182-8592-feef9fb72b5a",
111   "related_cases": [
112     {
113       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
114       "submitter_id": "QA-REGRESSION-0003"
115     }
116   ],
117   "type": "portion",
118   "unique_keys": [
119     {
120       "project_id": "QA-REGRESSION",
121       "submitter_id": "TCGA-BP-5184-11A-01"
122     }
123   ],
124   "valid": true,
125   "warnings": []
126 },
127 {
128   "action": "update",
129   "errors": [],
130   "id": "01d8e68d-7299-5b10-b545-afab0144ca1c",
131   "related_cases": [
132     {
133       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
134       "submitter_id": "QA-REGRESSION-0003"
135     }
136   ],
137   "type": "aliquot",
138   "unique_keys": [
139     {
140       "project_id": "QA-REGRESSION",
141       "submitter_id": "TCGA-BP-5184-11A-01D-1423-01"
142     }
143   ],
144   "valid": true,
145   "warnings": []
146 },
147 {
148   "action": "update",
149   "errors": [],
150   "id": "e13264ea-2ff1-52d6-b781-65d21c2d24a9",
151   "related_cases": [
152     {
153       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
154       "submitter_id": "QA-REGRESSION-0003"
155     }
156   ],
157   "type": "aliquot",
158   "unique_keys": [
159     {
160       "project_id": "QA-REGRESSION",
161       "submitter_id": "TCGA-BP-5184-01A-01R-1425-13"
162     }
163   ],
164   "valid": true,

```

```

165     "warnings": []
166   },
167   {
168     "action": "update",
169     "errors": [],
170     "id": "a7c6332c-04ad-5563-8ca6-51813122bc03",
171     "related_cases": [
172       {
173         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
174         "submitter_id": "QA-REGRESSION-0003"
175       }
176     ],
177     "type": "aliquot",
178     "unique_keys": [
179       {
180         "project_id": "QA-REGRESSION",
181         "submitter_id": "TCGA-BP-5184-11A-01D-2101-10"
182       }
183     ],
184     "valid": true,
185     "warnings": []
186   },
187   {
188     "action": "update",
189     "errors": [],
190     "id": "69f19d8d-13a5-5f0d-ac82-d4de075fd51a",
191     "related_cases": [
192       {
193         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
194         "submitter_id": "QA-REGRESSION-0003"
195       }
196     ],
197     "type": "aliquot",
198     "unique_keys": [
199       {
200         "project_id": "QA-REGRESSION",
201         "submitter_id": "TCGA-BP-5184-11A-01D-1429-08"
202       }
203     ],
204     "valid": true,
205     "warnings": []
206   },
207   {
208     "action": "update",
209     "errors": [],
210     "id": "11251eba-5a20-5747-8d30-0eec4d8fd8e2",
211     "related_cases": [
212       {
213         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
214         "submitter_id": "QA-REGRESSION-0003"
215       }
216     ],
217     "type": "analyte",
218     "unique_keys": [
219       {
220         "project_id": "QA-REGRESSION",
221         "submitter_id": "TCGA-BP-5184-01A-01W"
222       }

```

```

223     ],
224     "valid": true,
225     "warnings": []
226   },
227   {
228     "action": "update",
229     "errors": [],
230     "id": "2c038da9-5373-538a-bd15-38522ef49ab1",
231     "related_cases": [
232       {
233         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
234         "submitter_id": "QA-REGRESSION-0003"
235       }
236     ],
237     "type": "sample",
238     "unique_keys": [
239       {
240         "project_id": "QA-REGRESSION",
241         "submitter_id": "TCGA-BP-5184-11A"
242       }
243     ],
244     "valid": true,
245     "warnings": []
246   },
247   {
248     "action": "update",
249     "errors": [],
250     "id": "d012beaf-5e0c-559a-b8d5-d4960912774b",
251     "related_cases": [
252       {
253         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
254         "submitter_id": "QA-REGRESSION-0003"
255       }
256     ],
257     "type": "sample",
258     "unique_keys": [
259       {
260         "project_id": "QA-REGRESSION",
261         "submitter_id": "TCGA-BP-5184-01A"
262       }
263     ],
264     "valid": true,
265     "warnings": []
266   },
267   {
268     "action": "update",
269     "errors": [],
270     "id": "5ca816e8-1325-5986-beaf-8f300caa42c3",
271     "related_cases": [
272       {
273         "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
274         "submitter_id": "QA-REGRESSION-0003"
275       }
276     ],
277     "type": "slide",
278     "unique_keys": [
279       {
280         "project_id": "QA-REGRESSION",

```

```

281     "submitter_id": "TCGA-BP-5184-01A-01-TS1"
282   }
283 ],
284 "valid": true,
285 "warnings": []
286 },
287 {
288   "action": "update",
289   "errors": [],
290   "id": "fe1047d0-82ae-513e-8b3c-353f5022ecc0",
291   "related_cases": [
292     {
293       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
294       "submitter_id": "QA-REGRESSION-0003"
295     }
296   ],
297   "type": "portion",
298   "unique_keys": [
299     {
300       "project_id": "QA-REGRESSION",
301       "submitter_id": "TCGA-BP-5184-01A-01"
302     }
303   ],
304   "valid": true,
305   "warnings": []
306 },
307 {
308   "action": "update",
309   "errors": [],
310   "id": "6d4054dd-c30a-5572-af39-cb6910a67bcc",
311   "related_cases": [
312     {
313       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
314       "submitter_id": "QA-REGRESSION-0003"
315     }
316   ],
317   "type": "aliquot",
318   "unique_keys": [
319     {
320       "project_id": "QA-REGRESSION",
321       "submitter_id": "TCGA-BP-5184-01A-01W-1477-10"
322     }
323   ],
324   "valid": true,
325   "warnings": []
326 },
327 {
328   "action": "update",
329   "errors": [],
330   "id": "6684b446-ece3-57c1-93b2-a4885f829707",
331   "related_cases": [
332     {
333       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
334       "submitter_id": "QA-REGRESSION-0003"
335     }
336   ],
337   "type": "aliquot",
338   "unique_keys": [

```

```

339     {
340         "project_id": "QA-REGRESSION",
341         "submitter_id": "TCGA-BP-5184-11A-01D-1424-05"
342     }
343 ],
344 "valid": true,
345 "warnings": []
346 },
347 {
348     "action": "update",
349     "errors": [],
350     "id": "f52da5fd-2267-517c-b09c-d7d3074e4f13",
351     "related_cases": [
352         {
353             "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
354             "submitter_id": "QA-REGRESSION-0003"
355         }
356     ],
357     "type": "aliquot",
358     "unique_keys": [
359         {
360             "project_id": "QA-REGRESSION",
361             "submitter_id": "TCGA-BP-5184-01A-01D-1429-08"
362         }
363     ],
364     "valid": true,
365     "warnings": []
366 },
367 {
368     "action": "update",
369     "errors": [],
370     "id": "778d5545-f35d-54a1-bd7e-e148aa048046",
371     "related_cases": [
372         {
373             "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
374             "submitter_id": "QA-REGRESSION-0003"
375         }
376     ],
377     "type": "portion",
378     "unique_keys": [
379         {
380             "project_id": "QA-REGRESSION",
381             "submitter_id": "TCGA-BP-5184-01A-21-1740-20"
382         }
383     ],
384     "valid": true,
385     "warnings": []
386 },
387 {
388     "action": "update",
389     "errors": [],
390     "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
391     "related_cases": [],
392     "type": "case",
393     "unique_keys": [
394         {
395             "project_id": "QA-REGRESSION",
396             "submitter_id": "QA-REGRESSION-0003"

```



```

397     }
398   ],
399   "valid": true,
400   "warnings": []
401 },
402 {
403   "action": "update",
404   "errors": [],
405   "id": "e1a7ac18-8473-56f4-8f49-42c427f2b6ff",
406   "related_cases": [
407     {
408       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
409       "submitter_id": "QA-REGRESSION-0003"
410     }
411   ],
412   "type": "aliquot",
413   "unique_keys": [
414     {
415       "project_id": "QA-REGRESSION",
416       "submitter_id": "TCGA-BP-5184-11A-01D-1422-02"
417     }
418   ],
419   "valid": true,
420   "warnings": []
421 },
422 {
423   "action": "update",
424   "errors": [],
425   "id": "f2531646-dc01-5d89-9ecb-3031716adb96",
426   "related_cases": [
427     {
428       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
429       "submitter_id": "QA-REGRESSION-0003"
430     }
431   ],
432   "type": "aliquot",
433   "unique_keys": [
434     {
435       "project_id": "QA-REGRESSION",
436       "submitter_id": "TCGA-BP-5184-01A-01R-1426-07"
437     }
438   ],
439   "valid": true,
440   "warnings": []
441 },
442 {
443   "action": "update",
444   "errors": [],
445   "id": "9682a18b-7d3e-514b-9bf2-666b4f697140",
446   "related_cases": [
447     {
448       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
449       "submitter_id": "QA-REGRESSION-0003"
450     }
451   ],
452   "type": "aliquot",
453   "unique_keys": [
454     {

```

```

455     "project_id": "QA-REGRESSION",
456     "submitter_id": "TCGA-BP-5184-01A-01D-1424-05"
457   }
458 ],
459 "valid": true,
460 "warnings": []
461 },
462 {
463   "action": "update",
464   "errors": [],
465   "id": "91bc6c5c-b340-52fb-9ab2-fec7e9312cf6",
466   "related_cases": [
467     {
468       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
469       "submitter_id": "QA-REGRESSION-0003"
470     }
471   ],
472   "type": "analyte",
473   "unique_keys": [
474     {
475       "project_id": "QA-REGRESSION",
476       "submitter_id": "TCGA-BP-5184-11A-01W"
477     }
478   ],
479   "valid": true,
480   "warnings": []
481 },
482 {
483   "action": "update",
484   "errors": [],
485   "id": "dadca81e-7a7d-5944-abc8-63c8acbdbfd6",
486   "related_cases": [
487     {
488       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
489       "submitter_id": "QA-REGRESSION-0003"
490     }
491   ],
492   "type": "analyte",
493   "unique_keys": [
494     {
495       "project_id": "QA-REGRESSION",
496       "submitter_id": "TCGA-BP-5184-01A-01R"
497     }
498   ],
499   "valid": true,
500   "warnings": []
501 },
502 {
503   "action": "update",
504   "errors": [],
505   "id": "15546b81-fe15-5838-aa1a-97ad29e36b15",
506   "related_cases": [
507     {
508       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
509       "submitter_id": "QA-REGRESSION-0003"
510     }
511   ],
512   "type": "aliquot",

```

```

513 "unique_keys": [
514   {
515     "project_id": "QA-REGRESSION",
516     "submitter_id": "TCGA-BP-5184-01A-01D-1423-01"
517   }
518 ],
519 "valid": true,
520 "warnings": []
521 },
522 {
523   "action": "update",
524   "errors": [],
525   "id": "3a7c43e0-4582-57dd-8d2b-7b2a92c30b21",
526   "related_cases": [
527     {
528       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
529       "submitter_id": "QA-REGRESSION-0003"
530     }
531   ],
532   "type": "slide",
533   "unique_keys": [
534     {
535       "project_id": "QA-REGRESSION",
536       "submitter_id": "TCGA-BP-5184-01A-01-BS1"
537     }
538   ],
539   "valid": true,
540   "warnings": []
541 },
542 {
543   "action": "update",
544   "errors": [],
545   "id": "ab513f26-8cba-50f2-bac9-59fc50f5c201",
546   "related_cases": [
547     {
548       "id": "3128d9bf-71fd-4edb-8d07-98f53ef0432d",
549       "submitter_id": "QA-REGRESSION-0003"
550     }
551   ],
552   "type": "aliquot",
553   "unique_keys": [
554     {
555       "project_id": "QA-REGRESSION",
556       "submitter_id": "TCGA-BP-5184-11A-01W-1477-10"
557     }
558   ],
559   "valid": true,
560   "warnings": []
561 }
562 ],
563 "entity_error_count": 0,
564 "message": "Transaction would have been successful. User selected dry run option, transaction aborted,
565           no data written to database.",
566 "success": true,
567 "transaction_id": 922373,
568 "transactional_error_count": 0,
569 "transactional_errors": [],
570 "updated_entity_count": 28

```

Retrieving Entities

Entities Endpoint

JSON objects representing submitted entities can be retrieved with a GET request to the `entities` endpoint. This endpoint retrieves entities by UUID. A single UUID or a comma-separated list of UUIDs can be passed to this endpoint as a query.

```
1 token=$(cat gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token"
   https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH/entities/fbf69646-5904-4f95-92d6-692bde658f05
```

```
1 {
2   "entities": [
3     {
4       "program": "TCGA",
5       "project": "ALCH",
6       "properties": {
7         "created_datetime": "2016-04-14T08:44:43.361800-05:00",
8         "id": "fbf69646-5904-4f95-92d6-692bde658f05",
9         "project_id": "TCGA-ALCH",
10        "projects": [
11          {
12            "id": "d9906779-f1da-5d9f-9caa-6d5ecb2e3cd6",
13            "submitter_id": null
14          }
15        ],
16        "state": "validated",
17        "submitter_id": "TCGA-ALCH-000001",
18        "type": "case",
19        "updated_datetime": "2016-04-14T21:29:28.401212-05:00"
20      }
21    ]
22  }
23 }
```

Export Endpoint

The `export` endpoint provides additional functionality for exporting entities from the GDC submission system. The `ids` parameter accepts a UUID or a comma-separated list of UUIDs. The `format` parameter allows the user to specify the preferred format of the API response: JSON, TSV, or CSV. When the `with_children` parameter is set to `with_children`, the response includes the metadata stored in all “child” entities of the entity being requested. The `export` endpoint accepts GET requests.

```
1 token=$(cat gdc-token-text-file.txt)
2
3
4 curl --header "X-Auth-Token: $token"
   'https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH/export?ids=11f8321-832f-4a8b-8384-a2f6256557e0&format=json'
```

```
1 {
2   "case": [
3     {
4       "tissue_source_sites": [],
```

```

5     "submitter_id": "TCGA-ALCH-000026",
6     "project_id": "TCGA-ALCH",
7     "type": "case",
8     "id": "11f83251-832f-4a8b-8384-a2f6256557e0",
9     "projects": [
10      {
11        "code": "ALCH",
12        "id": "d9906779-f1da-5d9f-9caa-6d5ecb2e3cd6"
13      }
14    ]
15  },
16 ],
17 "sample": [
18   {
19     "sample_type_id": "10",
20     "time_between_excision_and_freezing": null,
21     "oct_embedded": "false",
22     "tumor_code_id": null,
23     "submitter_id": "Blood-00001_api26",
24     "intermediate_dimension": null,
25     "id": "23308708-6a63-471e-947c-6a93c6e85983",
26     "time_between_clamping_and_freezing": null,
27     "pathology_report_uuid": null,
28     "tumor_descriptor": null,
29     "sample_type": "Blood Derived Normal",
30     "project_id": "TCGA-ALCH",
31     "current_weight": null,
32     "composition": null,
33     "is_ffpe": null,
34     "shortest_dimension": null,
35     "tumor_code": null,
36     "tissue_type": null,
37     "days_to_sample_procurement": null,
38     "cases": [
39       {
40         "id": "11f83251-832f-4a8b-8384-a2f6256557e0",
41         "submitter_id": "TCGA-ALCH-000026"
42       }
43     ],
44     "freezing_method": null,
45     "type": "sample",
46     "preservation_method": null,
47     "days_to_collection": null,
48     "initial_weight": null,
49     "longest_dimension": null
50   }
51 ],
52 "read_group": [
53   {
54     "library_name": "Solexa-34688",
55     "is_paired_end": true,
56     "size_selection_range": null,
57     "adapter_sequence": null,
58     "library_strand": null,
59     "submitter_id": "Blood-00001-aliquot_lane1_barcode26",
60     "library_preparation_kit_name": null,
61     "adapter_name": null,
62     "target_capture_kit_name": null,

```

```

63 "includes_spike_ins": null,
64 "library_preparation_kit_version": null,
65 "id": "90163202-cfd7-4f6a-8214-e7e4e924d3a6",
66 "spike_ins_concentration": null,
67 "target_capture_kit_vendor": null,
68 "read_length": 75,
69 "sequencing_date": "2010-08-04",
70 "spike_ins_fasta": null,
71 "to_trim_adapter_sequence": null,
72 "RIN": null,
73 "platform": "Illumina",
74 "library_selection": "Hybrid_Selection",
75 "library_strategy": "WXS",
76 "library_preparation_kit_catalog_number": null,
77 "target_capture_kit_target_region": null,
78 "fastq_name": null,
79 "target_capture_kit_version": null,
80 "aliquots": [
81   {
82     "id": "e66dee54-5f4c-4471-9e08-dba0f6cdaaa4",
83     "submitter_id": "Blood-00001-aliquot26"
84   }
85 ],
86 "read_group_name": "205DD.3-2",
87 "library_preparation_kit_vendor": null,
88 "project_id": "TCGA-ALCH",
89 "type": "read_group",
90 "target_capture_kit_catalog_number": null,
91 "instrument_model": "Illumina HiSeq 2000",
92 "base_caller_name": null,
93 "experiment_name": "Resequencing",
94 "flow_cell_barcode": "205DDABXX",
95 "sequencing_center": "BI",
96 "base_caller_version": null
97 }
98 ],
99 "aliquot": [
100  {
101    "source_center": "23",
102    "centers": [],
103    "analytes": [],
104    "submitter_id": "Blood-00001-aliquot26",
105    "amount": 10,
106    "samples": [
107      {
108        "id": "23308708-6a63-471e-947c-6a93c6e85983",
109        "submitter_id": "Blood-00001_api26"
110      }
111    ],
112    "concentration": 0.07,
113    "project_id": "TCGA-ALCH",
114    "type": "aliquot",
115    "id": "e66dee54-5f4c-4471-9e08-dba0f6cdaaa4"
116  }
117 ],
118 "submitted_unaligned_reads": [
119  {
120    "read_groups": [

```

```

121     {
122       "id": "90163202-cfd7-4f6a-8214-e7e4e924d3a6",
123       "submitter_id": "Blood-00001-aliquot_lane1_barcode26"
124     }
125   ],
126   "data_type": "Unaligned Reads",
127   "file_name": "dummy.fastq",
128   "md5sum": "70c48a8a670ed2a02327601a10038d06",
129   "data_format": "FASTQ",
130   "submitter_id": "Blood-00001-aliquot_lane1_barcode26.fastq",
131   "state_comment": null,
132   "data_category": "Sequencing Data",
133   "file_size": 38,
134   "project_id": "TCGA-ALCH",
135   "type": "submitted_unaligned_reads",
136   "id": "6d45f2a0-8161-42e3-97e6-e058ac18f3f3",
137   "experimental_strategy": "WGS"
138 },
139 {
140   "read_groups": [
141     {
142       "id": "90163202-cfd7-4f6a-8214-e7e4e924d3a6",
143       "submitter_id": "Blood-00001-aliquot_lane1_barcode26"
144     }
145   ],
146   "data_type": "Unaligned Reads",
147   "file_name": "dummy.fastq",
148   "md5sum": "70c48a8a670ed2a02327601a10038d06",
149   "data_format": "FASTQ",
150   "submitter_id": "Blood-00001-aliquot_lane1_barcode27.fastq",
151   "state_comment": null,
152   "data_category": "Sequencing Data",
153   "file_size": 38,
154   "project_id": "TCGA-ALCH",
155   "type": "submitted_unaligned_reads",
156   "id": "4faabdd6-45bb-4259-8868-13d5b1149748",
157   "experimental_strategy": "WGS"
158 }
159 ]
160 }

```

GraphQL

Submitters can use the GraphQL query language for advanced search and retrieval of data from the GDC Submission Portal. See [GraphQL](#) for more information.

Deleting Entities

The `entities` endpoint can also be used to delete entities. This is accomplished using a DELETE request to the endpoint, specifying the entity's UUID. If an entity cannot be deleted because it is linked to child entities, the GDC Submission API will respond with an error providing a list of entities that must be deleted prior to deleting the subject entity.

A subgraph (a parent along with all of its child entities) can be deleted in a single transaction by passing a comma-separated list of UUIDs to the `entities` endpoint.

Entities in submitted state (assigned when the project has been submitted) cannot be deleted.

```

1 token=$(gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --request DELETE
   https://api.gdc.cancer.gov/v0/submission/TCGA/ALCH/entities/67782964-0065-491d-b051-2ae404bb734d

```

```

1 {
2   "code": 200,
3   "deleted_entity_count": 1,
4   "dependent_ids": "",
5   "entities": [
6     {
7       "action": "delete",
8       "errors": [],
9       "id": "67782964-0065-491d-b051-2ae404bb734d",
10      "related_cases": [],
11      "type": "case",
12      "valid": true,
13      "warnings": []
14    }
15  ],
16  "entity_error_count": 0,
17  "message": "Successfully deleted 1 entities",
18  "success": true,
19  "transaction_id": 192,
20  "transactional_error_count": 0,
21  "transactional_errors": []
22 }

```

Working With Files

Uploading Data Files

Experimental data files like BAM and FASTQ can be uploaded directly to the API using the `files` endpoint, by specifying the UUID of the corresponding `data_file` entity. Binary upload mode must be used if available. Uploading large files may be more efficiently performed using the GDC Data Transfer Tool.

```

1 token=$(gdc-token-text-file.txt)
2
3 curl --header "X-Auth-Token: $token" --output needed\_to\_show\_progress\_bar.log --request PUT --data-binary
   @GDC-INTERNAL-000084-S1-Q1-RG1.fastq.zip
   https://api.gdc.cancer.gov/v0/submission/GDC/INTERNAL/files/c414a205-376e-4993-af48-2a4689eb433e &&
   rm needed\_to\_show\_progress\_bar.log
4
5   "&& rm needed\_to\_show\_progress\_bar.log" at the end of the command above
6   removes the temporary file required to show upload progress bar. This
7   will not work on Windows platforms. Windows users must remove this
8   string and can delete the file manually.

```

Upload Manifest

The `manifest` endpoint generates a manifest for uploading files using the GDC Data Transfer Tool. It requires a comma-separated list of file UUIDs to generate a manifest.

```

1 https://api.gdc.cancer.gov/v0/submission/PROGRAM/PROJECT/manifest?ids=bf0751ca-fc3b-4760-b876-0fefce040be5,90163

```


Uploading New Versions of Data Files

The GDC Submission system supports submitting updated versions of files. For example, you may want to submit an updated version of a clinical supplement file that contains new clinical information about a patient. If a file is in file_state `validated` then you would simply delete and upload a new copy of this file. No additional version of the file will be created in this case. The UUID of the node stays the same.

However, if a file is in file_state `submitted` or `validated` and state `released` then a different process is required. In this situation simply upload a new template containing updated metadata (e.g. `md5sum` or `file_size`). A new node (with a new UUID) will automatically be created that is linked to the previous version. Once this new file is indexed and released to users they will be able to query the new UUID in the `/files` endpoint and both versions' UUID in the `files/versions` or `/history` endpoint. In the example below we register a file, upload the file, and register a new version of this file.

```
1 [
2   {
3     "data_type": "Clinical Supplement",
4     "file_name": "nationwidechildrens.org_clinical.TCGA-4G-AAZT-.xml",
5     "md5sum": "ecaaa87613ba03c971bfefdb6f693959",
6     "data_format": "BCR XML",
7     "submitter_id":
8       "nationwidechildrens.org_CHOL.bio.Level_1.428.25.0.tar.gz_nationwidechildrens.org_clinical.TCGA-4G-AAZT-
9     "archives": [],
10    "data_category": "Clinical",
11    "file_size": 39195,
12    "cases": [
13      {
14        "id": "b10c64c2-7fd2-4210-b975-034affb14b57",
15        "submitter_id": "TCGA-4G-AAZT"
16      }
17    ],
18    "project_id": "TCGA-CHOL",
19    "type": "clinical_supplement"
20 ]
```

```
1 curl --header "X-Auth-Token: $token" --header 'Content-Type: json' --request PUT --data-binary @clin.json
   'https://api.gdc.cancer.gov/submission/TCGA/CHOL'
```

```
1 {
2   "cases_related_to_created_entities_count": 1,
3   "cases_related_to_updated_entities_count": 0,
4   "code": 200,
5   "created_entity_count": 1,
6   "entities": [
7     {
8       "action": "create",
9       "errors": [],
10      "id": "d65c15d9-9e33-4a0b-863d-605ad6155506",
11      "related_cases": [
12        {
13          "id": "b10c64c2-7fd2-4210-b975-034affb14b57",
14          "submitter_id": "TCGA-4G-AAZT"
15        }
16      ],
17      "type": "clinical_supplement",
18      "unique_keys": [
19        {
20          "project_id": "TCGA-CHOL",
```

```

21     "submitter_id":
22         "nationwidechildrens.org_CHOL.bio.Level_1.428.25.0.tar.gz_nationwidechildrens.org_clinical.TCGA-4G
23     },
24     "valid": true,
25     "warnings": []
26 }
27 ],
28 "entity_error_count": 0,
29 "message": "Transaction successful.",
30 "success": true,
31 "transaction_id": 922606,
32 "transactional_error_count": 0,
33 "transactional_errors": [],
34 "updated_entity_count": 0
35 }

```

```

1 [
2 {
3     "data_type": "Clinical Supplement",
4     "file_name": "nationwidechildrens.org_clinical.TCGA-4G-AAZT-.xml",
5     "md5sum": "93e306e5e621d3cacb363e2be96ca3cd",
6     "data_format": "BCR XML",
7     "submitter_id":
8         "nationwidechildrens.org_CHOL.bio.Level_1.428.25.0.tar.gz_nationwidechildrens.org_clinical.TCGA-4G-AAZT-
9     "archives": [],
10    "data_category": "Clinical",
11    "file_size": 39197,
12    "cases": [
13        {
14            "id": "b10c64c2-7fd2-4210-b975-034affb14b57",
15            "submitter_id": "TCGA-4G-AAZT"
16        }
17    ],
18    "project_id": "TCGA-CHOL",
19    "type": "clinical_supplement"
20 }

```

```

1 curl --header "X-Auth-Token: $token" --header 'Content-Type: json' --request PUT --data-binary
   @clin_v2.json 'https://api.gdc.cancer.gov/submission/TCGA/CHOL'

```

```

1 {
2     "cases_related_to_created_entities_count": 0,
3     "cases_related_to_updated_entities_count": 0,
4     "code": 200,
5     "created_entity_count": 0,
6     "entities": [
7         {
8             "action": "version",
9             "errors": [],
10            "id": "32e9fd2c-877a-4700-a06f-bb34e0590ca5",
11            "related_cases": [
12                {
13                    "id": "b10c64c2-7fd2-4210-b975-034affb14b57",
14                    "submitter_id": "TCGA-4G-AAZT"
15                }
16            ],

```

```

17     "type": "clinical_supplement",
18     "unique_keys": [
19       {
20         "project_id": "TCGA-CHOL",
21         "submitter_id":
22           "nationwidechildrens.org_CHOL.bio.Level_1.428.25.0.tar.gz_nationwidechildrens.org_clinical.TCGA-40
23       },
24     "valid": true,
25     "warnings": []
26   }
27 ],
28 "entity_error_count": 0,
29 "message": "Transaction successful.",
30 "success": true,
31 "transaction_id": 922607,
32 "transactional_error_count": 0,
33 "transactional_errors": [],
34 "updated_entity_count": 0
35 }

```

Downloading Files

Files in `file_state = validated` can be downloaded by the submitter using the API or the Data Transfer Tool. This is done in a similar manner as files available in the Data Portal, but will require submission access to the particular project in dbGaP as opposed to downloader access. File UUIDs can be found in the original upload manifest file, the submission portal, or by API calls. See [Downloading Files](#) for details.

Deleting Files

Uploaded files can be deleted by deleting the entity that corresponds to the file. See [Deleting Entities](#) for details.

Querying Submitted Data Using GraphQL

GraphQL Overview

[GraphQL](#) is a query language that makes it easy to search and retrieve data from graph data structures such as the GDC Data Model.

Unlike the methods outlined in [Search and Retrieval](#), which provide access to public releases (or snapshots) of GDC data, the `/graphql` endpoint of GDC Submission API makes it possible for submitters to access “live” data, which provides a real-time view of the state of entities in a project.

NOTE: Access to GDC Submission API GraphQL service is limited to authorized and authenticated submitters. Submitters may only access data in their own project using GraphQL.

GraphQL IDE

The GDC GraphQL IDE is an instance of [GraphiQL](#), an in-browser GraphQL IDE that facilitates construction and execution of GraphQL queries. The GDC GraphQL IDE provides tab-completion and syntax checking using schema from the GDC Data Dictionary. It can be found at <https://portal.gdc.cancer.gov/submission/graphiql>.

Before interacting directly with the GDC Submission API’s GraphQL endpoint, users are encouraged to become familiar with executing queries using the GDC GraphQL IDE.

GraphQL Endpoint

GDC data submitters can access the GDC Submission API GraphQL endpoint at:

```
1 https://api.gdc.cancer.gov/[API_version/]submission/graphql
```

where **API_version/** is the optional API version component (see [Getting Started]).

NOTE: An authentication token is required for all requests to the `graphql` endpoint. Queries are restricted to those projects for which the submitter has obtained authorization.

Constructing a Query

When sending GraphQL requests to the API directly, the bare GraphQL query must be wrapped in a “query” JSON object as shown below:

When using the GDC GraphQL IDE, the bare JSON query must be used without a JSON wrapper.

Bare GraphQL query

In its simplest form, a GraphQL query is a **selection set** (curly brackets) that encloses a set of **fields**. The selection set defines the set of information that is to be retrieved. Furthermore, in GraphQL fields are conceptually equivalent to functions that retrieve additional fields and, in some cases, can take arguments. So each field in a selection set can have its own selection set, thereby creating a nested query structure that can navigate complex data relationships. See [GraphQL Specification](#) for further details.

In GDC GraphQL IDE, a root field (field within the outermost/umbrella selection set) typically corresponds to an entity, whereas fields inside nested selection sets are typically a combination of entities and entity properties.

The “Docs” panel on the right-hand side of the GDC GraphQL IDE allows users to discover the fields that can be queried with GraphQL. Note that the panel contains a lot of information and users may experience a delay before it is displayed.

A simple GraphQL query looks like this:

```
1 {
2   case (project_id: "TCGA-ALCH", first: 0) {
3     id
4     submitter_id
5   }
6 }
7 _case_count (project_id: "TCGA-ALCH")
8 }
```

The query above has two root fields: `case` and `_case_count`. The `case` field corresponds to the `case` entity in the GDC Data Model. The query supplies two arguments to the field:

1. `project_id: "TCGA-ALCH"`, which requests only cases in the TCGA-ALCH project.
2. `first: 0`, which requests that the API provide all results in the response, without pagination (a nonzero positive integer value of `first` specifies the number of results to return, 10 by default; “pages” are selected using `offset`).

The `_case_count` field is a special field that returns the number of cases that match the supplied argument.

The bare query above can be used as is in the GraphQL IDE. In order to pass this query to the GDC API directly, it needs to be further processed as described below.

Passing GraphQL queries to GDC API directly

Before a bare GraphQL query is passed to the GDC API, it must be processed as follows:

1. [Escape](#) the query using JSON string rules
2. Wrap the query in a “[query](#)” [JSON object](#).
3. Pass the query to the `graphql` endpoint in an HTTP POST request.

Using the `case` and `_case_count` example above as the starting point, the results are as follows:

```
1 {
2   case (project_id: "TCGA-ALCH", first: 0) {
3     id
4     submitter_id
5   }
6 }
7 _case_count (project_id: "TCGA-ALCH")
8 }
```

```
1 {\n\tcase (project_id: \"TCGA-ALCH\", first: 0) {\n\t\tid\n\t\tsubmitter_id\n\t}\n\t_case_count
   (project_id: \"TCGA-ALCH\")\n}
```

```
1 {
2   "query": "{\n\tcase (project_id: \"TCGA-ALCH\", first: 0)
   {\n\t\tid\n\t\tsubmitter_id\n\t}\n\t_case_count (project_id: \"TCGA-ALCH\")\n}",
3   "variables": null
4 }
```

```
1 token=$(<gdc-token-text-file.txt)
2
3 curl --request POST --header "X-Auth-Token: $token" 'https://api.gdc.cancer.gov/v0/submission/graphql'
   --data-binary @Query_json
```

```
1 {
2   "data": {
3     "_case_count": 20,
4     "case": [
5       {
6         "id": "700d1110-b6b4-4251-89d4-fa6f0698e3f8",
7         "submitter_id": "TCGA-ALCH-000004"
8       },
9       {
10        "id": "be01357d-7348-40b4-a997-8a61ae7af17d",
11        "submitter_id": "TCGA-ALCH-000005"
12      },
13      {
14        "id": "e5638697-6ef3-4bf8-a373-102519093f33",
15        "submitter_id": "TCGA-ALCH-000008"
16      },
17      {
18        "id": "4871d41a-680e-4fd0-901c-b06f06ecae33",
19        "submitter_id": "TCGA-ALCH-000007"
20      },
21      {
22        "id": "2f18c2c1-bff2-43b6-9702-e138c72d8c6b",
23        "submitter_id": "TCGA-ALCH-000009"
24      },
25      {
```

```
26     "id": "ec83e038-4f01-47a6-bc69-47fb297d0282",
27     "submitter_id": "TCGA-ALCH-000006"
28   },
29   {
30     "id": "e4642952-d259-4be1-9c53-ed95aa1fc50b",
31     "submitter_id": "TCGA-ALCH-000011"
32   },
33   {
34     "id": "8bcaf0b3-21d0-45c6-87ee-c997efb417dc",
35     "submitter_id": "TCGA-ALCH-000010"
36   },
37   {
38     "id": "83de027e-bcbf-4239-975b-7e8ced82448e",
39     "submitter_id": "TCGA-ALCH-000013"
40   },
41   {
42     "id": "bbd91cc1-06e2-4e60-8b93-e09c3b16f00c",
43     "submitter_id": "TCGA-ALCH-000014"
44   },
45   {
46     "id": "574fd163-4368-440c-9548-d76a0fbc9056",
47     "submitter_id": "TCGA-ALCH-000015"
48   },
49   {
50     "id": "47c92cdd-ff11-4c25-b0f0-0f7671144271",
51     "submitter_id": "TCGA-ALCH-000016"
52   },
53   {
54     "id": "9f13caab-1fda-4b2a-b500-f79dc978c6c1",
55     "submitter_id": "TCGA-ALCH-000017"
56   },
57   {
58     "id": "9418f194-8741-44db-bd8f-36f4fd8c3bf2",
59     "submitter_id": "TCGA-ALCH-000018"
60   },
61   {
62     "id": "6fb2a018-c5f3-45e5-81d3-e58e7e4bf921",
63     "submitter_id": "TCGA-ALCH-000019"
64   },
65   {
66     "id": "70236972-e796-414a-9b7a-3b29b849ba7c",
67     "submitter_id": "TCGA-ALCH-000020"
68   },
69   {
70     "id": "6f78e86f-9e31-4af5-a0d9-b8970ece476d",
71     "submitter_id": "TCGA-ALCH-000021"
72   },
73   {
74     "id": "c6fcb2f0-c6bb-4b40-a761-bae3e63869cb",
75     "submitter_id": "TCGA-ALCH-000002"
76   },
77   {
78     "id": "67782964-0065-491d-b051-2ae404bb734d",
79     "submitter_id": "TCGA-ALCH-000001"
80   },
81   {
82     "id": "b45d2891-ba81-4ecc-a250-c58060934227",
83     "submitter_id": "TCGA-ALCH-000012"
```

```

84     }
85   ]
86 }
87 }

```

Additional Examples

Example: File UUID

GraphQL query to find the file UUID based on file `submitter_id`:

```

1 {
2
3   submitted_unaligned_reads (project_id: "GDC-INTERNAL", submitter_id:
4     "Blood-00001-aliquot_lane1_barcode23.fastq") {
5     id
6     submitter_id
7     file_name
8     project_id
9 }

```

```

1 {
2   "query": "{\n \n submitted_unaligned_reads (project_id: \"GDC-INTERNAL\", submitter_id:
3     \"Blood-00001-aliquot_lane1_barcode23.fastq\") {\n   id\n   submitter_id\n   file_name\n
4     project_id\n}\n}",
5   "variables": null
6 }

```

```

1 curl --request POST --header "X-Auth-Token: $token" 'https://api.gdc.cancer.gov/v0/submission/graphql'
2 --data-binary @escaped_GraphQL

```

```

1 {
2   "data": {
3     "submitted_unaligned_reads": [
4       {
5         "file_name": "dummy.fastq",
6         "id": "616eab2f-791a-4641-8cd6-ee195a10a201",
7         "project_id": "GDC-INTERNAL",
8         "submitter_id": "Blood-00001-aliquot_lane1_barcode23.fastq"
9       }
10    ]
11  }
12 }

```

Example: Case Without Diagnosis

GraphQL query for any one case in 'TCGA-LUAD' without Diagnosis information:

```

1 {
2   case (project_id: "TCGA-LUAD", without_links: ["diagnoses"], first: 1) {
3     submitter_id
4   }
5 }

```

```

1 {
2   "data": {
3     "case": [
4       {
5         "submitter_id": "TCGA-17-Z050"
6       }
7     ]
8   }
9 }

```

Example: Number of Cases Without Diagnosis

GraphQL query for the number of cases in 'TCGA-LUAD' without Diagnosis information:

```

1 {
2   _case_count (project_id: "TCGA-LUAD", without_links: ["diagnoses"])
3 }

```

```

1 {
2   "data": {
3     "_case_count": 5
4   }
5 }

```

Example: Aliquot State

Query for the state of aliquots belonging to case with submitter_id: "TCGA-ALCH-000001":

```

1 {
2   aliquot(with_path_to: {type: "case", submitter_id:"TCGA-ALCH-000001"}) {
3     id release_state
4   }
5 }

```

```

1 {
2   "data": {
3     "aliquot": [
4       {
5         "id": "7af58da0-cb3e-43e2-a074-4bd8f27565ba",
6         "state": "validated"
7       }
8     ]
9   }
10 }

```

Example: Aliases

GraphQL query that uses a GraphQL fragment to get specific properties from two portions and give them aliases in the response:

```

1 {
2   some_portion: portion (first: 1) {
3     ...portionProperties
4   }
5   specific_portion: portion(submitter_id: "TCGA-67-6217-01A-13-2191-20") {
6     ...portionProperties
7   }

```



```

8 }
9
10 fragment portionProperties on portion {
11   submitter_id
12   is_ffpe
13 }

1 {
2   "data": {
3     "some_portion": [
4       {
5         "is_ffpe": false,
6         "submitter_id": "TCGA-62-A471-10A-01"
7       }
8     ],
9     "specific_portion": [
10      {
11        "is_ffpe": false,
12        "submitter_id": "TCGA-67-6217-01A-13-2191-20"
13      }
14    ]
15  }
16 }

```

Example: Biospecimen Tree

GraphQL Query for a case in “TCGA-LUAD” and return a biospecimen tree:

```

1 {
2   case(project_id: "TCGA-LUAD", first: 1) {
3     id
4     samples(first: 1) {
5       id
6       portions(first: 1) {
7         id
8         analytes(first: 1) {
9           id
10          aliquots(first: 1) {
11            id
12          }
13        }
14      }
15    }
16  }
17 }

1 {
2   "data": {
3     "case": [
4       {
5         "id": "19ca36e6-2154-4224-89b1-117a4a4407f6",
6         "samples": [
7           {
8             "id": "5e2625d2-290d-48cd-af5c-27dc8e3c8b6a",
9             "portions": [
10              {
11                "analytes": [

```

```
12     {
13       "aliquots": [
14         {
15           "id": "8e1820d5-dcd8-4760-9962-221e2b71d4b9"
16         }
17       ],
18       "id": "6449533c-e52a-4e58-bae7-0732f48153ef"
19     }
20   ],
21   "id": "26b75643-8fcd-445e-a0e0-9868cac589ea"
22 }
23 ]
24 }
25 ]
26 }
27 ]
28 }
29 }
```

Chapter 7

Python Examples

Using Python to Query the GDC API

Python can be a versatile tool for retrieving information from the GDC API and performing downstream processing. This page details some examples that demonstrate the basic API queries using Python. The examples in this guide will use the [requests](#) Python library and should be compatible with Python3.

Querying Metadata

Python can be used with the GDC API to retrieve metadata that is indexed in the GDC Database. See the Search and Retrieval section of the API documentation for specific details about parameters and usage.

A Basic Query

This example passes some basic parameters (fields, format, size) to the `cases` endpoint and prints the results in a tab-delimited format. Note that the `fields` parameter needs to be a string comprising comma-delimited field names.

1 Choose the Python tab to view script.

```
1 import requests
2 import json
3
4 cases_endpt = 'https://api.gdc.cancer.gov/cases'
5
6 ## The 'fields' parameter is passed as a comma-separated string of single names.
7 fields = [
8     "submitter_id",
9     "case_id",
10    "primary_site",
11    "disease_type",
12    "diagnoses.vital_status"
13 ]
14
15 fields = ','.join(fields)
16
17 params = {
18     "fields": fields,
19     "format": "TSV",
20     "size": "100"
21 }
```

```

22
23 response = requests.get(cases_endpt, params = params)
24
25 print(response.content)

```

[Download Script](#)

A Filtered Query

In the next example, a `filters` parameter is added to the script. This parameter is passed as a Python dictionary object. The filter used in this example will only display cases that come from a kidney disease study (`primary_site: Kidney`).

1 Choose the Python tab to view script.

```

1 import requests
2 import json
3
4 fields = [
5     "submitter_id",
6     "case_id",
7     "primary_site",
8     "disease_type",
9     "diagnoses.vital_status"
10 ]
11
12 fields = ",".join(fields)
13
14 cases_endpt = "https://api.gdc.cancer.gov/cases"
15
16 filters = {
17     "op": "in",
18     "content":{
19         "field": "primary_site",
20         "value": ["Kidney"]
21     }
22 }
23
24 ## With a GET request, the filters parameter needs to be converted
25 ## from a dictionary to JSON-formatted string
26
27 params = {
28     "filters": json.dumps(filters),
29     "fields": fields,
30     "format": "TSV",
31     "size": "100"
32 }
33
34 response = requests.get(cases_endpt, params = params)
35
36 print(response.content)

```

[Download Script](#)

Complex Filters

The following example uses the `and` operator in the filter to return information about files that 1) were produced using RNA-Seq, 2) are downloadable in BAM format, and 3) originate from lung cancer patients. Note that these three filters are nested within a list in the highest level `content` key.

1 Choose the Python tab to view script.

```
1 import requests
2 import json
3
4 fields = [
5     "file_name",
6     "cases.submitter_id",
7     "cases.samples.sample_type",
8     "cases.disease_type",
9     "cases.project.project_id"
10 ]
11
12 fields = ",".join(fields)
13
14 files_endpt = "https://api.gdc.cancer.gov/files"
15
16 ## This set of filters is nested under an 'and' operator.
17 filters = {
18     "op": "and",
19     "content": [
20         {
21             "op": "in",
22             "content": {
23                 "field": "cases.project.primary_site",
24                 "value": ["Lung"]
25             }
26         },
27         {
28             "op": "in",
29             "content": {
30                 "field": "files.experimental_strategy",
31                 "value": ["RNA-Seq"]
32             }
33         },
34         {
35             "op": "in",
36             "content": {
37                 "field": "files.data_format",
38                 "value": ["BAM"]
39             }
40         }
41     ]
42 }
43
44 ## A POST is used, so the filter parameters can be passed directly as a Dict object.
45 params = {
46     "filters": filters,
47     "fields": fields,
48     "format": "TSV",
49     "size": "2000"
50 }
51
52 ## The parameters are passed to 'json' rather than 'params' in this case
53 response = requests.post(files_endpt, headers = {"Content-Type": "application/json"}, json = params)
54
55 print(response.content.decode("utf-8"))
```

[Download Script](#)

Downloading Files

GDC files can also be downloaded from the API and saved locally using Python scripts. See the File Download section of the API documentation for more information.

A Simple Download Request

An open-access GDC file can be downloaded by appending the file UUID to the data endpoint URL.

1 Choose the Python tab to view script.

```
1 import requests
2 import json
3 import re
4
5 file_id = "b658d635-258a-4f6f-8377-767a43771fe4"
6
7 data_endpt = "https://api.gdc.cancer.gov/data/{}".format(file_id)
8
9 response = requests.get(data_endpt, headers = {"Content-Type": "application/json"})
10
11 ## The file name can be found in the header within the Content-Disposition key.
12 response_head_cd = response.headers["Content-Disposition"]
13
14 file_name = re.findall("filename=(.+)", response_head_cd)[0]
15
16 with open(file_name, "wb") as output_file:
17     output_file.write(response.content)
```

[Download Script](#)

Passing a Token to Download a Controlled-Access File

A token can be passed to the script by specifying a plain text file that contains only the GDC token. A token can be downloaded by logging into the GDC Data Portal. See the [Data Security](#) documentation for more details.

1 Choose the Python tab to view script.

```
1 import requests
2 import json
3
4 '''
5 This script will not work until $TOKEN_FILE_PATH
6 is replaced with an actual path.
7 '''
8 token_file = "$TOKEN_FILE_PATH"
9
10 file_id = "11443f3c-9b8b-4e47-b5b7-529468fec098"
11
12 data_endpt = "https://api.gdc.cancer.gov/slicing/view/{}".format(file_id)
13
14 with open(token_file, "r") as token:
15     token_string = str(token.read().strip())
16
```

```

17 params = {"gencode": ["BRCA1", "BRCA2"]}
18
19 response = requests.post(data_endpt,
20                          data = json.dumps(params),
21                          headers = {
22                              "Content-Type": "application/json",
23                              "X-Auth-Token": token_string
24                          })
25
26 file_name = "brca_slices.bam"
27
28 with open(file_name, "wb") as output_file:
29     output_file.write(response.content)

```

[Download Script](#)

Post Request to Download Multiple Files

This example uses a Python list to specify a set of file UUIDs. The list in the example was populated manually but could potentially be populated programmatically from an external list or API call.

1 Choose the Python tab to view script.

```

1 import requests
2 import json
3 import re
4
5 data_endpt = "https://api.gdc.cancer.gov/data"
6
7 ids = [
8     "b658d635-258a-4f6f-8377-767a43771fe4",
9     "3968213d-b293-4b3d-8033-5b5a0ca07b6c"
10 ]
11
12 params = {"ids": ids}
13
14 response = requests.post(data_endpt,
15                          data = json.dumps(params),
16                          headers={
17                              "Content-Type": "application/json"
18                          })
19
20 response_head_cd = response.headers["Content-Disposition"]
21
22 file_name = re.findall("filename=(.+)", response_head_cd)[0]
23
24 with open(file_name, "wb") as output_file:
25     output_file.write(response.content)

```

[Download Script](#)

Downloading a Set of Files Based on a Filter

Here a list of files based on a set of filters are downloaded. File UUIDs are retrieved based on the filters. These UUIDs are then passed to the data endpoint to download the correct files.

1 Choose the Python tab to view script.

```

1 import requests
2 import json
3 import re
4
5 files_endpt = "https://api.gdc.cancer.gov/files"
6
7 filters = {
8     "op": "and",
9     "content": [
10        {
11            "op": "in",
12            "content": {
13                "field": "cases.project.primary_site",
14                "value": ["Lung"]
15            }
16        },
17        {
18            "op": "in",
19            "content": {
20                "field": "cases.demographic.race",
21                "value": ["white"]
22            }
23        },
24        {
25            "op": "in",
26            "content": {
27                "field": "cases.demographic.gender",
28                "value": ["female"]
29            }
30        },
31        {
32            "op": "in",
33            "content": {
34                "field": "files.analysis.workflow_type",
35                "value": ["HTSeq - FPKM"]
36            }
37        }
38    ]
39 }
40
41 ## Here a GET is used, so the filter parameters should be passed as a JSON string.
42
43 params = {
44     "filters": json.dumps(filters),
45     "fields": "file_id",
46     "format": "JSON",
47     "size": "1000"
48 }
49
50 response = requests.get(files_endpt, params = params)
51
52 file_uuid_list = []
53
54 ## This step populates the download list with the file_ids from the previous query
55 for file_entry in json.loads(response.content.decode("utf-8"))["data"]["hits"]:
56     file_uuid_list.append(file_entry["file_id"])
57
58 data_endpt = "https://api.gdc.cancer.gov/data"

```



```

59
60 params = {"ids": file_uuid_list}
61
62 response = requests.post(data_endpt, data = json.dumps(params), headers = {"Content-Type":
    "application/json"})
63
64 response_head_cd = response.headers["Content-Disposition"]
65
66 file_name = re.findall("filename=(.+)", response_head_cd)[0]
67
68 with open(file_name, "wb") as output_file:
69     output_file.write(response.content)

```

[Download Script](#)

BAM Slicing

The GDC BAM Slicing feature can also be accessed through Python. Below is an example of a basic BAM slicing command.

1 Choose the Python tab to view script.

```

1 import requests
2 import json
3
4 '''
5 This script will not work until $TOKEN_FILE_PATH
6 is replaced with an actual path.
7 '''
8 token_file = "$TOKEN_FILE_PATH"
9
10 file_id = "11443f3c-9b8b-4e47-b5b7-529468fec098"
11
12 data_endpt = "https://api.gdc.cancer.gov/slicing/view/{}".format(file_id)
13
14 with open(token_file,"r") as token:
15     token_string = str(token.read().strip())
16
17 params = {"gencode": ["BRCA1", "BRCA2"]}
18
19 response = requests.post(data_endpt,
20     data = json.dumps(params),
21     headers = {
22         "Content-Type": "application/json",
23         "X-Auth-Token": token_string
24     })
25
26 file_name = "brca_slices.bam"
27
28 with open(file_name, "wb") as output_file:
29     output_file.write(response.content)

```

[Download Script](#)

The same region(s) across multiple BAM files can be retrieved using a for-loop within a Python script.

1 Choose the Python tab to view script.

```

1 import requests
2 import json

```

```

3
4 '''
5 This script will not work until $TOKEN_FILE_PATH
6 is replaced with an actual path.
7 '''
8 token_file = "$TOKEN_FILE_PATH"
9
10 file_ids = [
11     "11443f3c-9b8b-4e47-b5b7-529468fec098",
12     "1f103620-bb34-46f1-b565-94f0027e396d",
13     "ca549554-a244-4209-9086-92add7bb7109"
14 ]
15
16 for file_id in file_ids:
17
18     data_endpt = "https://api.gdc.cancer.gov/slicing/view/{}".format(file_id)
19
20     with open(token_file, "r") as token:
21         token_string = str(token.read().strip())
22
23     params = {
24         "regions": ["chr1:1-20000", "chr10:129000-160000"]
25     }
26
27     response = requests.post(data_endpt,
28                             data = json.dumps(params),
29                             headers = {
30                                 "Content-Type": "application/json",
31                                 "X-Auth-Token": token_string
32                             })
33
34     file_name = "{}_region_slices.bam".format(file_id)
35
36     with open(file_name, "wb") as output_file:
37         output_file.write(response.content)

```

[Download Script](#)

Basic Troubleshooting

The following script should produce an unformatted JSON string with information about the API status. Run this script to verify that a valid connection is being made to the GDC API.

```

1 import requests
2 status_endpt = "https://api.gdc.cancer.gov/status"
3 response = requests.get(status_endpt)
4 print(response.content)

```

Chapter 8

GraphQL Examples

Introduction to GDC GraphQL

[GraphQL](#) is a query language for APIs. The [GDC REST API](#) has structured and specifically defined query parameters as well as endpoints that have set requests and responses. The GDC GraphQL provides advanced GDC users greater flexibility to specify the data they would like to be returned. This allows queries to be cleaner and easier to understand, especially when combining multiple queries into one request.

To produce queries in a visual interface, the GDC recommends using [GraphiQL](#). See below for the correct endpoint URLs.

Using GDC GraphQL vs GDC REST API

If the query requires only a subset of the data to be returned, GDC GraphQL may speed up requests as GraphQL queries return only the specified data. This may require less work on the GDC server-side to fulfill those requests. Conversely, if an entire data-set is required for each request, the GDC REST API may be a better fit. No matter which method is used, the data returned by the GDC REST API and the GraphQL query will be identical as they query the same source.

GDC GraphQL Overview

GraphQL is not a storage model or a database query language. The graph refers to graph structures defined in the schema, where nodes define objects and edges define relationships between objects. The API traverses and returns application data based on the schema definitions, independent of how the data is stored.

GDC GraphQL Endpoints

The GDC GraphQL has only two endpoints: * **GDC Search and Retrieval Endpoint:** <https://api.gdc.cancer.gov/v0/graphql>
* **GDC Submission Endpoint:** <https://api.gdc.cancer.gov/v0/submission/graphql>

This page covers the search and retrieval endpoint, see the GDC Submission API for additional details on the submission endpoint.

GDC GraphQL Schema

All GDC GraphQL queries are validated and executed against the [GDC GraphQL schema](#). Because the GraphQL parameters are discoverable, the GDC GraphQL schema can be queried for details about itself.

The `__schema` keyword can be queried to list all types defined in the schema and retrieve details about each:

```

1 {
2   __schema {
3     types {
4       name
5       kind
6       fields {
7         name
8       }
9     }
10  }
11 }

```

The `__type` keyword can also be queried to retrieve details about any type such as “Explore” or “Case”:

```

1
2 {
3   __type(name: "Explore") {
4     name
5     kind
6     description
7     fields {
8       name
9     }
10  }
11 }

```

```

1 {
2   __type(name: "Case") {
3     name
4     kind
5     description
6     fields {
7       name
8     }
9   }
10 }

```

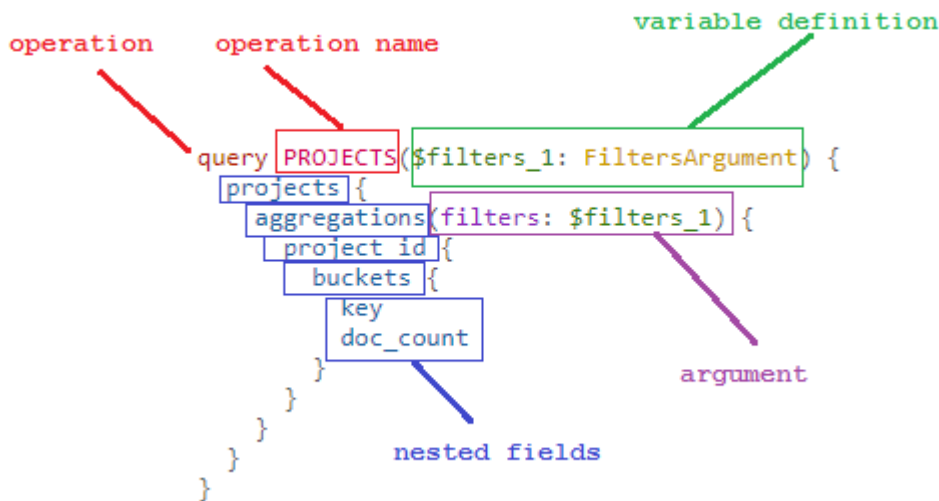
Basic GraphQL queries in GDC

The two types of allowed operations in GDC GraphQL API are queries and mutations. Comparing GraphQL to REST, queries operate like GET requests, while mutations operate like POST/PATCH/DELETE.

Note: This guide does not cover GDC GraphQL mutation operations.

GraphQL queries return only the data that is specified. Queries are built by specifying fields within fields (also called nested *subfields*) until only scalars are returned. Scalars are primitive values such as: `Int`, `Float`, `String`, `Boolean`, or `ID`.

Anatomy of a typical GDC GraphQL Query



- **Operation type:** Describes what type of operation that is being performed, such as query, mutation, or subscription
- **Operation name:** Similar to a function name, gives queries meaningful names
- **Field:** Denotes the specific fields on objects that will be included with the response data
- **Arguments:** A set of key-value pairs associated with a specific field. The parameters can be literal values or variables. **NOTE:** Arguments can appear on any field, even fields nested deep in an operation.
- **Variable definitions:** As GraphQL is strong typed, it validates the variable being passed dynamically. **NOTE:** Variables are passed separately from the query document as JSON such as:

```
1 { "filters_1": { "op": "in", "content": { "field": "projects.program.name", "value": ["TARGET"]} } }
```

GDC GraphQL Examples

Nodes And Edges Example

A very powerful feature of GDC GraphQL API is that the graph structures defined in the [GDC GraphQL schema](#) can be queried and traversed. In these queries, nodes define objects and edges define relationships between objects.

```
1
2 query PROJECTS_EDGES($filters_1: FiltersArgument) {
3   projects {
4     hits(filters: $filters_1) {
5       total
6       edges {
7         node {
8           primary_site
9           disease_type
10          project_id
11          dbgap_accession_number
12        }
13      }
14    }
15  }
16 }
17
```

```

18 variable:
19 { "filters_1": {"op": "in", "content": {"field": "projects.primary_site", "value": ["Kidney"]}}}

```

Query Case File Counts

```

1 query CaseFileCounts($filters: FiltersArgument) {
2   viewer {
3     repository {
4       cases {
5         hits(first: 1, filters: $filters) {
6           edges {
7             node {
8               case_id
9               files {
10                hits(first: 0) {
11                  total
12                }
13              }
14              summary {
15                experimental_strategies {
16                  experimental_strategy
17                  file_count
18                }
19                data_categories {
20                  data_category
21                  file_count
22                }
23              }
24            }
25          }
26        }
27      }
28    }
29  }
30 }
31
32 variable:
33 {"filters":{"op":"in","content":{"field":"cases.case_id","value":["dcd5860c-7e3a-44f3-a732-fe92fe3fe300"]}}}

```

Query Simple Static Mutations Based on Gene IDs

```

1
2 query PROJECTS_EDGES($filters_2: FiltersArgument) {
3   explore {
4     ssms {
5       hits(filters: $filters_2) {
6         total
7         edges {
8           node {
9             ssm_id
10            gene_aa_change
11          }
12        }
13      }
14    }
15  }

```

```
16 }
17
18 variable:
19 {"filters_2":
    {"op": "in", "content": {"field": "consequence.transcript.gene.gene_id", "value": ["ENSG00000155657"]}}}
```

Chapter 9

System Information

System Information

Overview

The GDC API offers endpoints that provide information about the system. These endpoints are described below.

GDC Notifications Endpoint

The `notifications` endpoint provides current user-facing notifications.

GDC notifications have a corresponding `level` with the following meanings:

Level	Meaning
INFO	Non-essential information, e.g. regarding a new dataset
WARNING	Important user information, e.g. regarding a dataset to be removed
ERROR	Important system information, e.g. regarding a GDC component
DEBUG	Unimportant system information, e.g. testing the notification system

Notifications will indicate the GDC `components` to which they apply:

Component	Description
PORTAL	The GDC Data Portal
LEGACY	The GDC Legacy Archive
SUBMISSION	The GDC Data Submission Portal
DOCUMENTATION	The GDC documentation site that contains GDC user guides, release notes, and the GDC Data Dictionary
WEBSITE	The GDC project website that includes information about the system. This does not include any of the above-listed GDC components.

Sample Request


```
1 curl --request GET https://api.gdc.cancer.gov/v0/notifications
```

```
1 {
2   "data": [
3     {
4       "level": "INFO",
5       "components": [
6         "SUBMISSION_API",
7         "LEGACY_API"
8       ],
9       "message": "The system is up!"
10    }
11  ]
12 }
```

API Status Endpoint

The status endpoint provides information about the current status and version of the GDC API.

Sample Request

```
1 curl https://api.gdc.cancer.gov/status
```

```
1 import requests
2 import json
3
4 status_endpt = 'https://api.gdc.cancer.gov/status'
5 response = requests.get(status_endpt)
6 print json.dumps(response.json(), indent=2)
```

```
1 {
2   "commit": "74e1e3583c0f39fbf2149322addb7378206be3b9",
3   "status": "OK",
4   "tag": "1.2.0",
5   "version": 1
6 }
```

Chapter 10

Additional Examples

Additional Examples

Data Search and Retrieval

Endpoint Examples

This section contains additional examples for using endpoints.

Project Endpoint Example

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

```
1 curl 'https://api.gdc.cancer.gov/projects?from=0&size=5&sort=project.name:asc&pretty=true'
```

```
1 {
2   "data": {
3     "hits": [
4       {
5         "state": "legacy",
6         "project_id": "TARGET-AML",
7         "primary_site": "Blood",
8         "disease_type": "Acute Myeloid Leukemia",
9         "name": "Acute Myeloid Leukemia"
10      },
11     {
12       "state": "legacy",
13       "project_id": "TCGA-LAML",
14       "primary_site": "Blood",
15       "disease_type": "Acute Myeloid Leukemia",
16       "name": "Acute Myeloid Leukemia"
17     },
18     {
19       "state": "legacy",
20       "project_id": "TARGET-AML-IF",
21       "primary_site": "Blood",
22       "disease_type": "Acute Myeloid Leukemia Induction Failure",
23       "name": "Acute Myeloid Leukemia Induction Failure"
24     },
25     {
26       "state": "legacy",
```

```

27     "project_id": "TARGET-ALL-P2",
28     "primary_site": "Blood",
29     "disease_type": "Acute Lymphoblastic Leukemia",
30     "name": "Acute Lymphoblastic Leukemia - Phase II"
31 },
32 {
33     "state": "legacy",
34     "project_id": "TARGET-ALL-P1",
35     "primary_site": "Blood",
36     "disease_type": "Acute Lymphoblastic Leukemia",
37     "name": "Acute Lymphoblastic Leukemia - Phase I"
38 }
39 ],
40 "pagination": {
41     "count": 5,
42     "sort": "project.name:asc",
43     "from": 0,
44     "pages": 10,
45     "total": 46,
46     "page": 1,
47     "size": 5
48 }
49 },
50 "warnings": {}
51 }

```

Files Endpoint Example

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.

```
1 curl 'https://api.gdc.cancer.gov/files?from=0&size=2&sort=file_size:asc&pretty=true'
```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "data_type": "Raw Simple Somatic Mutation",
6         "updated_datetime": "2017-03-04T16:45:40.925270-06:00",
7         "file_name": "9f78a291-2d50-472c-8f56-5f8fbd09ab2a.snp.Somatic.hc.vcf.gz",
8         "submitter_id": "TCGA-13-0757-01A-01W-0371-08_TCGA-13-0757-10A-01W-0371-08_varscan",
9         "file_id": "9f78a291-2d50-472c-8f56-5f8fbd09ab2a",
10        "file_size": 1120,
11        "id": "9f78a291-2d50-472c-8f56-5f8fbd09ab2a",
12        "created_datetime": "2016-05-04T14:50:54.560567-05:00",
13        "md5sum": "13c1ceb3519615e2c67128b350365fbf",
14        "data_format": "VCF",
15        "acl": [
16          "phs000178"
17        ],
18        "access": "controlled",
19        "state": "live",
20        "data_category": "Simple Nucleotide Variation",
21        "type": "simple_somatic_mutation",
22        "file_state": "submitted",
23        "experimental_strategy": "WXS"
24      },
25      {

```

```

26     "data_type": "Raw Simple Somatic Mutation",
27     "updated_datetime": "2017-03-04T16:45:40.925270-06:00",
28     "file_name": "7780009b-abb6-460b-903d-accdac626c2e.snp.Somatic.hc.vcf.gz",
29     "submitter_id": "TCGA-HC-8261-01A-11D-2260-08_TCGA-HC-8261-10A-01D-2260-08_varscan",
30     "file_id": "7780009b-abb6-460b-903d-accdac626c2e",
31     "file_size": 1237,
32     "id": "7780009b-abb6-460b-903d-accdac626c2e",
33     "created_datetime": "2016-05-08T13:54:38.369393-05:00",
34     "md5sum": "fd9bb46c8022b96af730c48dc00e2c41",
35     "data_format": "VCF",
36     "acl": [
37         "phs000178"
38     ],
39     "access": "controlled",
40     "state": "live",
41     "data_category": "Simple Nucleotide Variation",
42     "type": "simple_somatic_mutation",
43     "file_state": "submitted",
44     "experimental_strategy": "WXS"
45 }
46 ],
47 "pagination": {
48     "count": 2,
49     "sort": "file_size:asc",
50     "from": 0,
51     "page": 1,
52     "total": 274724,
53     "pages": 137362,
54     "size": 2
55 }
56 },
57 "warnings": {}
58 }

```

Cases Endpoint Example

This example is a query for cases contained in GDC. It returns only the first five files.

```
1 curl 'https://api.gdc.cancer.gov/cases?from=0&size=5&pretty=true'
```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "updated_datetime": "2017-03-09T10:01:14.834935-06:00",
6         "submitter_analyte_ids": [
7           "TCGA-ER-A193-06A-12D",
8           "TCGA-ER-A193-06A-12R",
9           "TCGA-ER-A193-06A-12W",
10          "TCGA-ER-A193-10A-01W",
11          "TCGA-ER-A193-10A-01D"
12        ],
13        "analyte_ids": [
14          "62e14ca4-95f5-4af3-848f-83f7273c3b70",
15          "6178b8aa-6afb-4951-bc92-bf9bfc57b9c7",
16          "e16b701c-7809-4fb5-a9e0-4ff71e5d1d84",
17          "5bfa8c9f-6797-4b2b-9122-854f8ab3bbba",

```

```

18     "9b73d64e-c973-45b6-be31-a486fb8d1708"
19 ],
20 "submitter_id": "TCGA-ER-A193",
21 "case_id": "8ab09143-daf6-40a9-85d3-0fe9de7b3e06",
22 "id": "8ab09143-daf6-40a9-85d3-0fe9de7b3e06",
23 "disease_type": "Skin Cutaneous Melanoma",
24 "sample_ids": [
25     "378b3d8a-adbb-4912-a0bf-6b74a282113e",
26     "7a384d44-8b05-4197-9921-7d020ada2437"
27 ],
28 "portion_ids": [
29     "6680bbf2-9cf1-4f93-9ec3-04318cffb5ba",
30     "690d3b12-a61d-42fd-af2a-5a7a9a3e5de8",
31     "824d724e-6836-423e-a751-fee3260ef4d2"
32 ],
33 "submitter_portion_ids": [
34     "TCGA-ER-A193-06A-21-A20N-20",
35     "TCGA-ER-A193-10A-01",
36     "TCGA-ER-A193-06A-12"
37 ],
38 "created_datetime": null,
39 "slide_ids": [
40     "d2751354-a8b7-4f7a-a4f1-d062de5ceb14"
41 ],
42 "state": "live",
43 "aliquot_ids": [
44     "dc9f9544-6c76-4b45-b5c3-dd2fec5acfe",
45     "390b3574-ba23-4ecb-acf8-f5ad8a958bd2",
46     "33f43961-b32d-46fc-ba11-264f1101e78d",
47     "cd17367c-3270-42ae-8ac5-941a3453ea33",
48     "b17269a2-79aa-459e-9c3d-589b7efe6fd9",
49     "28a7d729-7555-4545-924b-3dec49b54230",
50     "13256e77-0b0b-49e3-9959-3b6730d68732",
51     "87ca642a-dd4c-47ea-b81f-2d3402f2157a",
52     "8a1bfe0e-c97a-41c4-815f-cf5bb5cfc69f",
53     "5e1e9c82-99fd-49de-9dfb-a349d4d8ac94",
54     "67f00459-e423-4900-be23-9283b0478620",
55     "d939c477-a01f-4d54-bcfb-c9fdd957f2ec"
56 ],
57 "primary_site": "Skin",
58 "submitter_aliquot_ids": [
59     "TCGA-ER-A193-06A-12D-A18Y-02",
60     "TCGA-ER-A193-10A-01D-A193-01",
61     "TCGA-ER-A193-10A-01D-A190-02",
62     "TCGA-ER-A193-06A-12D-A197-08",
63     "TCGA-ER-A193-06A-12R-A18S-07",
64     "TCGA-ER-A193-06A-12W-A20H-08",
65     "TCGA-ER-A193-10A-01D-A199-08",
66     "TCGA-ER-A193-10A-01D-A38R-08",
67     "TCGA-ER-A193-10A-01W-A20J-08",
68     "TCGA-ER-A193-06A-12R-A18V-13",
69     "TCGA-ER-A193-06A-12D-A19C-05",
70     "TCGA-ER-A193-06A-12D-A191-01"
71 ],
72 "submitter_sample_ids": [
73     "TCGA-ER-A193-10A",
74     "TCGA-ER-A193-06A"
75 ],

```

```

76     "submitter_slide_ids": [
77         "TCGA-ER-A193-06A-01-TSA"
78     ]
79 },
80 {
81     "updated_datetime": "2017-03-04T16:39:19.244769-06:00",
82     "submitter_analyte_ids": [
83         "TCGA-VR-AA4G-10A-01W",
84         "TCGA-VR-AA4G-01A-11R",
85         "TCGA-VR-AA4G-10A-01D",
86         "TCGA-VR-AA4G-01A-11D",
87         "TCGA-VR-AA4G-01A-11W"
88     ],
89     "analyte_ids": [
90         "152d7d7a-c746-4b58-8c3f-4252454c7b7c",
91         "9090d556-bd2e-4851-8a0c-46e22cc61408",
92         "7118f4c3-b635-4428-8240-8db85281f2d9",
93         "1d8223ff-685a-4427-a3d1-f53887f2a19d",
94         "60dfb30a-bea0-426d-b11d-d5813ba39cfc"
95     ],
96     "submitter_id": "TCGA-VR-AA4G",
97     "case_id": "df5bd25c-d70b-4126-89cb-6c838044ae3b",
98     "id": "df5bd25c-d70b-4126-89cb-6c838044ae3b",
99     "disease_type": "Esophageal Carcinoma",
100    "sample_ids": [
101        "21456849-38a9-4190-9ece-ed69b3c24fda",
102        "6ee6d239-2af6-41cd-bc32-c5cdaf7742b0"
103    ],
104    "portion_ids": [
105        "484b40d5-d77c-4e6f-9e80-1ef27ffbc8a5",
106        "fdc56e67-52ab-44fd-823a-5a3124876ff7"
107    ],
108    "submitter_portion_ids": [
109        "TCGA-VR-AA4G-10A-01",
110        "TCGA-VR-AA4G-01A-11"
111    ],
112    "created_datetime": null,
113    "slide_ids": [
114        "e950eba2-7d6e-4ffd-a2d5-e0eb6486848a"
115    ],
116    "state": "live",
117    "aliquot_ids": [
118        "db6beed3-a5a2-469f-8dc8-00d838c1f37f",
119        "f5db4d36-034b-429b-a7be-26a872b702ee",
120        "16421a96-b843-4f7e-9f7c-64d2fb5b2a25",
121        "5d938cb5-7064-40bc-877d-57faa94c3333",
122        "d231404d-ece5-43c0-a8a3-e9f294ceb777",
123        "8c77dc3e-2ea3-4626-88f5-e74f242bedf3",
124        "993624d4-1c28-41a5-a0b6-094a0e442c36",
125        "105a18c9-df7e-4573-b1a2-6a987e57d553",
126        "af81c3bb-3b9e-41cb-b85a-b55c6437d05b",
127        "38938066-5fd9-415c-b00e-65efff14085e",
128        "20139afe-ad04-4571-b779-0c4a51e74ada"
129    ],
130    "primary_site": "Esophagus",
131    "submitter_aliquot_ids": [
132        "TCGA-VR-AA4G-10A-01W-A44M-09",
133        "TCGA-VR-AA4G-01A-11D-A37B-01",

```

```

134     "TCGA-VR-AA4G-01A-11D-A37D-05",
135     "TCGA-VR-AA4G-10A-01D-A37F-09",
136     "TCGA-VR-AA4G-01A-11D-A37R-26",
137     "TCGA-VR-AA4G-01A-11R-A37J-13",
138     "TCGA-VR-AA4G-01A-11R-A37I-31",
139     "TCGA-VR-AA4G-01A-11D-A37C-09",
140     "TCGA-VR-AA4G-10A-01D-A37R-26",
141     "TCGA-VR-AA4G-10A-01D-A37E-01",
142     "TCGA-VR-AA4G-01A-11W-A44L-09"
143 ],
144 "submitter_sample_ids": [
145     "TCGA-VR-AA4G-01A",
146     "TCGA-VR-AA4G-10A"
147 ],
148 "submitter_slide_ids": [
149     "TCGA-VR-AA4G-01A-01-TS1"
150 ]
151 },
152 {
153     "updated_datetime": "2017-03-04T16:39:19.244769-06:00",
154     "submitter_analyte_ids": [
155         "TCGA-D1-A174-01A-11D",
156         "TCGA-D1-A174-01A-11W",
157         "TCGA-D1-A174-10A-01D",
158         "TCGA-D1-A174-10A-01W",
159         "TCGA-D1-A174-01A-11R"
160     ],
161     "analyte_ids": [
162         "96203028-f824-4a90-9758-22340285062c",
163         "f4878e33-b773-43b5-83a5-9fd8e539e668",
164         "8627ccd0-0575-4d03-b589-ca45642d523d",
165         "1183f7c6-992d-4084-946e-adce7c52f9cc",
166         "5343f6a8-8ac2-4446-ace5-a27d21e76844"
167     ],
168     "submitter_id": "TCGA-D1-A174",
169     "case_id": "fc7315b0-9f48-4206-b197-2268c0518eb4",
170     "id": "fc7315b0-9f48-4206-b197-2268c0518eb4",
171     "disease_type": "Uterine Corpus Endometrial Carcinoma",
172     "sample_ids": [
173         "df9a1f44-9b3f-48b2-96af-54aaabdfd243",
174         "ad5a9cb6-b3f9-4651-b6d1-13c78010bd88"
175     ],
176     "portion_ids": [
177         "79dd516c-bae3-4f6e-b4cb-901de030acb7",
178         "6e55e6d9-902f-439b-b6f1-ca296c123fd3"
179     ],
180     "submitter_portion_ids": [
181         "TCGA-D1-A174-01A-11",
182         "TCGA-D1-A174-10A-01"
183     ],
184     "created_datetime": null,
185     "slide_ids": [
186         "7602727e-b46d-40fc-bd03-5ccf631041f8"
187     ],
188     "state": "live",
189     "aliquot_ids": [
190         "5c15542b-cd63-44b5-b278-e211410fb0aa",
191         "d661cfb9-248a-49e6-b0db-865ca257e8dc",

```

```

192     "83bd3bdb-9bd3-46fa-888c-f6f5efec530f",
193     "c46551c9-c0d0-4140-8d0a-946b53e504e2",
194     "96b511df-3a69-4168-908c-662060b4f976",
195     "0182d4e1-f835-46b5-a8f0-53decf5868de",
196     "e9563a06-0b86-4986-976e-43d4040f1d61",
197     "6bb2de6e-5b85-4e97-a930-1f2c6bf663a1",
198     "f6ee5558-a1b6-4b11-8f48-c17186fff39a",
199     "67f6f0d9-6581-4946-a9c7-a6629da86888",
200     "39e9a948-054a-4b50-b108-7d7aee686363",
201     "ddb4ca26-655d-4bdc-a00d-7caf26cadafe"
202 ],
203 "primary_site": "Uterus",
204 "submitter_aliquot_ids": [
205     "TCGA-D1-A174-01A-11D-A12F-02",
206     "TCGA-D1-A174-01A-01D-YYYY-23",
207     "TCGA-D1-A174-01A-11W-A139-09",
208     "TCGA-D1-A174-10A-01W-A139-09",
209     "TCGA-D1-A174-01A-11D-A12K-05",
210     "TCGA-D1-A174-10A-01D-A12F-02",
211     "TCGA-D1-A174-10A-01D-A12G-01",
212     "TCGA-D1-A174-01A-11R-A12I-07",
213     "TCGA-D1-A174-01A-11D-A12J-09",
214     "TCGA-D1-A174-10A-01D-A12J-09",
215     "TCGA-D1-A174-01A-11R-A12H-13",
216     "TCGA-D1-A174-01A-11D-A12G-01"
217 ],
218 "submitter_sample_ids": [
219     "TCGA-D1-A174-01A",
220     "TCGA-D1-A174-10A"
221 ],
222 "submitter_slide_ids": [
223     "TCGA-D1-A174-01A-01-TS1"
224 ]
225 },
226 {
227     "updated_datetime": "2017-03-04T16:39:19.244769-06:00",
228     "submitter_analyte_ids": [
229         "TCGA-XM-A8RL-10A-01D",
230         "TCGA-XM-A8RL-01A-11R",
231         "TCGA-XM-A8RL-01A-11D"
232     ],
233     "analyte_ids": [
234         "2c483e72-92b0-425d-ac1b-b75a169cf531",
235         "57f88d4f-8b3a-4349-88b0-3d2e58a95ed9",
236         "499bfbe1-639c-479c-abaa-42cbb11c0568"
237     ],
238     "submitter_id": "TCGA-XM-A8RL",
239     "case_id": "dd240b82-b1d6-4c0f-aa3e-6fcfe1364ec1",
240     "id": "dd240b82-b1d6-4c0f-aa3e-6fcfe1364ec1",
241     "disease_type": "Thymoma",
242     "sample_ids": [
243         "cb091cc1-7bbe-43a4-8460-01215af3aa21",
244         "cabc9729-c1e1-4f08-9959-985dcb7a00d5"
245     ],
246     "portion_ids": [
247         "e8ea57c9-729e-46ea-b1da-2db7a00b02bc",
248         "8e2edb92-753f-4cb0-a5b8-8c45dbefaf36",
249         "650fa4f2-9fa2-4d3a-8b63-ff4a9bd8c33e"

```



```

250 ],
251 "submitter_portion_ids": [
252     "TCGA-XM-A8RL-01A-21-A45R-20",
253     "TCGA-XM-A8RL-10A-01",
254     "TCGA-XM-A8RL-01A-11"
255 ],
256 "created_datetime": null,
257 "slide_ids": [
258     "08cedd34-aafd-4b47-891f-cf66ee1f627b"
259 ],
260 "state": "live",
261 "aliquot_ids": [
262     "df9d8553-8d5b-4c65-8b28-74030a8f8e76",
263     "47b7f634-b36f-49e9-a4dc-d8f5508fdc0a",
264     "e692ebed-9721-40db-8986-fcaba07d68f1",
265     "189ee080-95d1-4ccb-8618-955605c7bd55",
266     "83af7ff3-45be-4378-a8b5-5dff3584e95d",
267     "42ebb1f0-e236-48ae-847f-69a153969903",
268     "e8a4938f-6b93-4ad1-9324-31c97dd1d477"
269 ],
270 "primary_site": "Thymus",
271 "submitter_aliquot_ids": [
272     "TCGA-XM-A8RL-10A-01D-A426-09",
273     "TCGA-XM-A8RL-01A-11D-A423-09",
274     "TCGA-XM-A8RL-01A-11D-A422-01",
275     "TCGA-XM-A8RL-01A-11R-A42C-07",
276     "TCGA-XM-A8RL-10A-01D-A425-01",
277     "TCGA-XM-A8RL-01A-11R-A42W-13",
278     "TCGA-XM-A8RL-01A-11D-A424-05"
279 ],
280 "submitter_sample_ids": [
281     "TCGA-XM-A8RL-10A",
282     "TCGA-XM-A8RL-01A"
283 ],
284 "submitter_slide_ids": [
285     "TCGA-XM-A8RL-01A-01-TSA"
286 ]
287 },
288 {
289     "updated_datetime": "2017-03-04T16:39:19.244769-06:00",
290     "submitter_analyte_ids": [
291         "TCGA-BO-5120-01A-01W",
292         "TCGA-BO-5120-01A-01D",
293         "TCGA-BO-5120-01A-01R",
294         "TCGA-BO-5120-11A-01W",
295         "TCGA-BO-5120-11A-01D"
296     ],
297     "analyte_ids": [
298         "996336e6-fad7-4100-96ae-60adb5c276f1",
299         "0eb7da02-0b90-4f6d-abd2-b048a9cb2995",
300         "fa2861b9-67c1-486a-a1e0-95d8f8adf65b",
301         "7e9f5639-a462-493e-98f8-1b7aeec383c7",
302         "d51e9fd4-0c99-49ec-9de5-db3946b0bf43"
303     ],
304     "submitter_id": "TCGA-BO-5120",
305     "case_id": "c5bf474c-6919-47b4-ba59-34ab20c087d5",
306     "id": "c5bf474c-6919-47b4-ba59-34ab20c087d5",
307     "disease_type": "Kidney Renal Clear Cell Carcinoma",

```

```

308     "sample_ids": [
309         "b50d3c6f-fdec-488b-ab26-a9b690fad34f",
310         "f3148210-ecae-4314-b5f8-9bee2315a093"
311     ],
312     "portion_ids": [
313         "b8fcbf00-4c5a-42c3-95e9-fb6e169a8da9",
314         "34443e91-0210-4477-9511-53026ae62b38",
315         "e466f011-79a1-4158-b796-f8e9dda32d68"
316     ],
317     "submitter_portion_ids": [
318         "TCGA-BO-5120-01A-01",
319         "TCGA-BO-5120-11A-01",
320         "TCGA-BO-5120-01A-21-1740-20"
321     ],
322     "created_datetime": null,
323     "slide_ids": [
324         "e5a29e92-4125-4acb-a797-86822b4961a2",
325         "78d873e0-037f-4aef-8725-7c651598b1f8",
326         "43d8cec7-f5a0-45d5-a5f8-cc77d6b7b539"
327     ],
328     "state": "live",
329     "aliquot_ids": [
330         "b35280fe-dbfa-4e45-8f49-3d0489e68743",
331         "a2e3a2f2-c32b-44a1-9b29-911145d700b8",
332         "a064d108-e8b2-46fa-b277-0a7a89904a3a",
333         "59be71a1-50e3-4565-852a-173afc8a6851",
334         "136dff0e-b181-49c9-8305-b3289625ea2e",
335         "8fbb983b-53ad-44a9-976a-7945628eaa51",
336         "cecf40f8-7301-4db9-b276-a14317d4dd59",
337         "fac8b066-bf2c-4f08-b42b-251035596a28",
338         "fa55c92f-54e8-436b-b8c4-04cb68a24e93",
339         "007e3098-aaf9-4ee7-9ae1-f94b131a5ae0",
340         "6ce58fbc-6742-4ade-84b0-cd025266e030",
341         "9668e15e-a3fa-4ead-ad42-322c5700e0db",
342         "c1167003-0730-41d5-bdd5-1cbf501c1463",
343         "73aab074-cbd1-45f2-8266-9ef6f7c559bc"
344     ],
345     "primary_site": "Kidney",
346     "submitter_aliquot_ids": [
347         "TCGA-BO-5120-11A-01D-1416-02",
348         "TCGA-BO-5120-11A-01D-2099-10",
349         "TCGA-BO-5120-11A-01D-1418-05",
350         "TCGA-BO-5120-01A-01W-1475-10",
351         "TCGA-BO-5120-01A-01D-1421-08",
352         "TCGA-BO-5120-01A-01D-1416-02",
353         "TCGA-BO-5120-01A-01R-1419-13",
354         "TCGA-BO-5120-01A-01R-1420-07",
355         "TCGA-BO-5120-11A-01D-1421-08",
356         "TCGA-BO-5120-01A-01D-1417-01",
357         "TCGA-BO-5120-01A-01D-1418-05",
358         "TCGA-BO-5120-11A-01W-1475-10",
359         "TCGA-BO-5120-01A-01D-2099-10",
360         "TCGA-BO-5120-11A-01D-1417-01"
361     ],
362     "submitter_sample_ids": [
363         "TCGA-BO-5120-11A",
364         "TCGA-BO-5120-01A"
365     ],

```

```

366     "submitter_slide_ids": [
367         "TCGA-BO-5120-11A-01-TS1",
368         "TCGA-BO-5120-01A-01-BS1",
369         "TCGA-BO-5120-01A-01-TS1"
370     ]
371 }
372 ],
373 "pagination": {
374     "count": 5,
375     "sort": "",
376     "from": 0,
377     "page": 1,
378     "total": 14551,
379     "pages": 2911,
380     "size": 5
381 }
382 },
383 "warnings": {}
384 }

```

Annotations Endpoint Example

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

```
1 curl 'https://api.gdc.cancer.gov/annotations?from=0&size=2&pretty=true'
```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "category": "History of unacceptable prior treatment related to a prior/other malignancy",
6         "status": "Approved",
7         "entity_id": "51c37449-6a2e-4c3d-a7cc-06f901e1224f",
8         "classification": "Notification",
9         "entity_type": "case",
10        "created_datetime": "2014-06-16T00:00:00",
11        "annotation_id": "3d086829-de62-5d08-b848-ce0724188ff0",
12        "notes": "unknown treatment history",
13        "updated_datetime": "2017-03-09T12:32:36.305475-06:00",
14        "submitter_id": "20743",
15        "state": "submitted",
16        "case_id": "51c37449-6a2e-4c3d-a7cc-06f901e1224f",
17        "case_submitter_id": "TCGA-AG-A014",
18        "entity_submitter_id": "TCGA-AG-A014",
19        "id": "3d086829-de62-5d08-b848-ce0724188ff0"
20      },
21      {
22        "category": "Center QC failed",
23        "status": "Approved",
24        "entity_id": "733f0607-6c6b-4385-9868-fa6f155a9a2e",
25        "classification": "CenterNotification",
26        "entity_type": "aliquot",
27        "created_datetime": "2012-07-20T00:00:00",
28        "annotation_id": "5cf05f41-ce70-58a3-8ecb-6bfaf6264437",
29        "notes": "RNA-seq:INSUFFICIENT INPUT MATERIAL,LOW SEQUENCE YIELD/DIVERSITY;LOW 5/3 COVERAGE
30        "updated_datetime": "2017-03-09T13:51:45.396638-06:00",

```

```

31     "submitter_id": "8764",
32     "state": "submitted",
33     "case_id": "3e8a51bf-7e1f-4eab-af83-3c60d04db1bf",
34     "case_submitter_id": "TCGA-13-0913",
35     "entity_submitter_id": "TCGA-13-0913-02A-01R-1564-13",
36     "id": "5cf05f41-ce70-58a3-8ecb-6bfaf6264437"
37   }
38 ],
39 "pagination": {
40   "count": 2,
41   "sort": "",
42   "from": 0,
43   "page": 1,
44   "total": 2361,
45   "pages": 1181,
46   "size": 2
47 }
48 },
49 "warnings": {}
50 }

```

Filters Examples

This section contains additional examples for using the `filters` parameter.

Example: Basic syntax

The following is an example of `filters` syntax, including the JSON object passed to the `filters` parameter, the corresponding API query, and the JSON object returned by the API. The example finds projects where the primary site is Blood.

```

1 {
2   "op": "and",
3   "content": [
4     {
5       "op": "in",
6       "content": {
7         "field": "primary_site",
8         "value": [
9           "Blood"
10        ]
11      }
12    }
13  ]
14 }

```

```

1 curl
   'https://api.gdc.cancer.gov/projects?filters=%7b%0d%0a%22op%22%3a%22and%22%2c%0d%0a%22content%22%3a%5b%

```

```

1 {
2   "data": {
3     "hits": [
4       {
5         "dbgap_accession_number": "phs000465",
6         "disease_type": [
7           "Acute Myeloid Leukemia"
8         ],

```

```

9     "released": true,
10    "state": "legacy",
11    "primary_site": [
12      "Blood"
13    ],
14    "project_id": "TARGET-AML",
15    "id": "TARGET-AML",
16    "name": "Acute Myeloid Leukemia"
17  }
18 ],
19 "pagination": {
20   "count": 1,
21   "sort": "",
22   "from": 0,
23   "page": 1,
24   "total": 1,
25   "pages": 1,
26   "size": 10
27 }
28 },
29 "warnings": {}
30 }

```

Example: Filter cases keeping only 'male'

This is an example of a value-based filter:

```

1 {
2   "op" : "=",
3   "content" : {
4     "field" : "cases.demographic.gender" ,
5     "value" : [ "male" ]
6   }
7 }

```

```

1 curl
   'https://api.gdc.cancer.gov/cases?filters=%7b%0d%0a+++%22op%22+%3a+%22%3d%22+%2c%0d%0a+++%22content%22+%3a+%

```

Example: Filter using a range

This is an example of filtering for age at diagnosis. The request is for cases where the age at diagnosis is between 40 and 70 years.
>**Note:** `age_at_diagnosis` is expressed in days.

```

1 {
2   "op": "and",
3   "content": [
4     {
5       "op": ">=",
6       "content": {
7         "field": "cases.diagnoses.age_at_diagnosis",
8         "value": [
9           14600
10        ]
11      }
12    },
13    {
14      "op": "<=",

```

```

15     "content": {
16         "field": "cases.diagnoses.age_at_diagnosis",
17         "value": [
18             25550
19         ]
20     }
21 }
22 ]
23 }

```

```

1 curl
  'https://api.gdc.cancer.gov/cases?filters=%7B%22op%22:%22and%22,%22content%22:%5B%7B%22op%22:%22%3E%3D%22,%2

```

Example: Multiple fields

Filter projects for primary_site being Kidney or Brain and program.name being TCGA

```

1 {
2   "op" : "and" ,
3   "content" : [{
4     "op" : "in" ,
5     "content" : {
6       "field" : "primary_site" ,
7       "value" : [
8         "Kidney" ,
9         "Brain"
10      ]
11    }
12  }, {
13    "op" : "in" ,
14    "content" : {
15      "field" : "program.name" ,
16      "value" : [
17        "TCGA"
18      ]
19    }
20  }
21 }

```

```

1 curl
  'https://api.gdc.cancer.gov/projects?filters=%7B%22op%22:%3A%22and%22%2C%22content%22:%3A%5B%7B%22op%22:%3A%22i

```

Chapter 11

Appendix A: Available Fields

Appendix A: Available Fields

The GDC API's search and retrieval endpoints provide access to fields that correspond to properties defined in the GDC Data Dictionary. This appendix contains a list of fields available at each endpoint, and a list of field groups accessible via the [expand parameter](#).

Field Listing by Endpoint

Project Fields

Field Name

dbgap_accession_number

disease_type

name

primary_site

project_id

released

state

program.dbgap_accession_number

program.name

program.program_id

summary.case_count

summary.file_count

summary.file_size

summary.data_categories.case_count

summary.data_categories.data_category

summary.data_categories.file_count

summary.experimental_strategies.case_count

summary.experimental_strategies.experimental_strategy

summary.experimental_strategies.file_count

Case Fields

Field Name |
— |
aliquot_ids |
analyte_ids |
case_id |
created_datetime |
days_to_index |
portion_ids |
sample_ids |
slide_ids |
state |
submitter_aliquot_ids |
submitter_analyte_ids |
submitter_id |
submitter_portion_ids |
submitter_sample_ids |
submitter_slide_ids |
updated_datetime |
annotations.annotation_id |
annotations.case_id |
annotations.case_submitter_id |
annotations.category |
annotations.classification |
annotations.created_datetime |
annotations.creator |
annotations.entity_id |
annotations.entity_submitter_id |
annotations.entity_type |
annotations.legacy_created_datetime |
annotations.legacy_updated_datetime |
annotations.notes |
annotations.state |
annotations.status |
annotations.submitter_id |
annotations.updated_datetime |
demographic.created_datetime |
demographic.demographic_id |
demographic.ethnicity |
demographic.gender |
demographic.race |
demographic.state |
demographic.submitter_id |
demographic.updated_datetime |
demographic.year_of_birth |
demographic.year_of_death |
diagnoses.age_at_diagnosis |
diagnoses.classification_of_tumor |
diagnoses.created_datetime |
diagnoses.days_to_birth |
diagnoses.days_to_death |
diagnoses.days_to_last_follow_up |
diagnoses.days_to_last_known_disease_status |
diagnoses.days_to_recurrence |
diagnoses.diagnosis_id |
diagnoses.last_known_disease_status |
diagnoses.morphology |

diagnoses.primary_diagnosis |
diagnoses.prior_malignancy |
diagnoses.progression_or_recurrence |
diagnoses.site_of_resection_or_biopsy |
diagnoses.state |
diagnoses.submitter_id |
diagnoses.tissue_or_organ_of_origin |
diagnoses.tumor_grade |
diagnoses.tumor_stage |
diagnoses.updated_datetime |
diagnoses.vital_status |
diagnoses.treatments.created_datetime |
diagnoses.treatments.days_to_treatment |
diagnoses.treatments.state |
diagnoses.treatments.submitter_id |
diagnoses.treatments.therapeutic_agents |
diagnoses.treatments.treatment_id |
diagnoses.treatments.treatment_intent_type |
diagnoses.treatments.treatment_or_therapy |
diagnoses.treatments.updated_datetime |
exposures.alcohol_history |
exposures.alcohol_intensity |
exposures.bmi |
exposures.cigarettes_per_day |
exposures.created_datetime |
exposures.exposure_id |
exposures.height |
exposures.state |
exposures.submitter_id |
exposures.updated_datetime |
exposures.weight |
exposures.years_smoked |
family_histories.created_datetime |
family_histories.family_history_id |
family_histories.relationship_age_at_diagnosis |
family_histories.relationship_gender |
family_histories.relationship_primary_diagnosis |
family_histories.relationship_type |
family_histories.relative_with_cancer_history |
family_histories.state |
family_histories.submitter_id |
family_histories.updated_datetime |
files.access |
files.acl |
files.created_datetime |
files.data_category |
files.data_format |
files.data_type |
files.error_type |
files.experimental_strategy |
files.file_id |
files.file_name |
files.file_size |
files.file_state |
files.md5sum |
files.origin |
files.platform |
files.revision |

files.state |
files.state_comment |
files.submitter_id |
files.tags |
files.type |
files.updated_datetime |
files.analysis.analysis_id |
files.analysis.analysis_type |
files.analysis.created_datetime |
files.analysis.state |
files.analysis.submitter_id |
files.analysis.updated_datetime |
files.analysis.workflow_end_datetime |
files.analysis.workflow_link |
files.analysis.workflow_start_datetime |
files.analysis.workflow_type |
files.analysis.workflow_version |
files.analysis.input_files.access |
files.analysis.input_files.created_datetime |
files.analysis.input_files.data_category |
files.analysis.input_files.data_format |
files.analysis.input_files.data_type |
files.analysis.input_files.error_type |
files.analysis.input_files.experimental_strategy |
files.analysis.input_files.file_id |
files.analysis.input_files.file_name |
files.analysis.input_files.file_size |
files.analysis.input_files.file_state |
files.analysis.input_files.md5sum |
files.analysis.input_files.platform |
files.analysis.input_files.revision |
files.analysis.input_files.state |
files.analysis.input_files.state_comment |
files.analysis.input_files.submitter_id |
files.analysis.input_files.updated_datetime |
files.analysis.metadata.read_groups.adapter_name |
files.analysis.metadata.read_groups.adapter_sequence |
files.analysis.metadata.read_groups.base_caller_name |
files.analysis.metadata.read_groups.base_caller_version |
files.analysis.metadata.read_groups.created_datetime |
files.analysis.metadata.read_groups.experiment_name |
files.analysis.metadata.read_groups.flow_cell_barcode |
files.analysis.metadata.read_groups.includes_spike_ins |
files.analysis.metadata.read_groups.instrument_model |
files.analysis.metadata.read_groups.is_paired_end |
files.analysis.metadata.read_groups.library_name |
files.analysis.metadata.read_groups.library_preparation_kit_catalog_number |
files.analysis.metadata.read_groups.library_preparation_kit_name |
files.analysis.metadata.read_groups.library_preparation_kit_vendor |
files.analysis.metadata.read_groups.library_preparation_kit_version |
files.analysis.metadata.read_groups.library_selection |
files.analysis.metadata.read_groups.library_strand |
files.analysis.metadata.read_groups.library_strategy |
files.analysis.metadata.read_groups.platform |
files.analysis.metadata.read_groups.read_group_id |
files.analysis.metadata.read_groups.read_group_name |
files.analysis.metadata.read_groups.read_length |
files.analysis.metadata.read_groups.RIN |

files.analysis.metadata.read_groups.sequencing_center |
files.analysis.metadata.read_groups.sequencing_date |
files.analysis.metadata.read_groups.size_selection_range |
files.analysis.metadata.read_groups.spike_ins_concentration |
files.analysis.metadata.read_groups.spike_ins_fasta |
files.analysis.metadata.read_groups.state |
files.analysis.metadata.read_groups.submitter_id |
files.analysis.metadata.read_groups.target_capture_kit_catalog_number |
files.analysis.metadata.read_groups.target_capture_kit_name |
files.analysis.metadata.read_groups.target_capture_kit_target_region |
files.analysis.metadata.read_groups.target_capture_kit_vendor |
files.analysis.metadata.read_groups.target_capture_kit_version |
files.analysis.metadata.read_groups.to_trim_adapter_sequence |
files.analysis.metadata.read_groups.updated_datetime |
files.analysis.metadata.read_groups.read_group_qcs.adapter_content |
files.analysis.metadata.read_groups.read_group_qcs.basic_statistics |
files.analysis.metadata.read_groups.read_group_qcs.created_datetime |
files.analysis.metadata.read_groups.read_group_qcs.encoding |
files.analysis.metadata.read_groups.read_group_qcs.fastq_name |
files.analysis.metadata.read_groups.read_group_qcs.kmer_content |
files.analysis.metadata.read_groups.read_group_qcs.overrepresented_sequences |
files.analysis.metadata.read_groups.read_group_qcs.per_base_n_content |
files.analysis.metadata.read_groups.read_group_qcs.per_base_sequence_content |
files.analysis.metadata.read_groups.read_group_qcs.per_base_sequence_quality |
files.analysis.metadata.read_groups.read_group_qcs.per_sequence_gc_content |
files.analysis.metadata.read_groups.read_group_qcs.per_sequence_quality_score |
files.analysis.metadata.read_groups.read_group_qcs.per_tile_sequence_quality |
files.analysis.metadata.read_groups.read_group_qcs.percent_gc_content |
files.analysis.metadata.read_groups.read_group_qcs.read_group_qc_id |
files.analysis.metadata.read_groups.read_group_qcs.sequence_duplication_levels |
files.analysis.metadata.read_groups.read_group_qcs.sequence_length_distribution |
files.analysis.metadata.read_groups.read_group_qcs.state |
files.analysis.metadata.read_groups.read_group_qcs.submitter_id |
files.analysis.metadata.read_groups.read_group_qcs.total_sequences |
files.analysis.metadata.read_groups.read_group_qcs.updated_datetime |
files.analysis.metadata.read_groups.read_group_qcs.workflow_end_datetime |
files.analysis.metadata.read_groups.read_group_qcs.workflow_link |
files.analysis.metadata.read_groups.read_group_qcs.workflow_start_datetime |
files.analysis.metadata.read_groups.read_group_qcs.workflow_type |
files.analysis.metadata.read_groups.read_group_qcs.workflow_version |
files.archive.archive_id |
files.archive.created_datetime |
files.archive.data_category |
files.archive.data_format |
files.archive.data_type |
files.archive.error_type |
files.archive.file_name |
files.archive.file_size |
files.archive.file_state |
files.archive.md5sum |
files.archive.revision |
files.archive.state |
files.archive.state_comment |
files.archive.submitter_id |
files.archive.updated_datetime |
files.cases.aliquot_ids |
files.cases.analyte_ids |
files.cases.case_id |

files.cases.created_datetime |
files.cases.days_to_index |
files.cases.portion_ids |
files.cases.sample_ids |
files.cases.slide_ids |
files.cases.state |
files.cases.submitter_aliquot_ids |
files.cases.submitter_analyte_ids |
files.cases.submitter_id |
files.cases.submitter_portion_ids |
files.cases.submitter_sample_ids |
files.cases.submitter_slide_ids |
files.cases.updated_datetime |
files.cases.annotations.annotation_id |
files.cases.annotations.case_id |
files.cases.annotations.case_submitter_id |
files.cases.annotations.category |
files.cases.annotations.classification |
files.cases.annotations.created_datetime |
files.cases.annotations.creator |
files.cases.annotations.entity_id |
files.cases.annotations.entity_submitter_id |
files.cases.annotations.entity_type |
files.cases.annotations.legacy_created_datetime |
files.cases.annotations.legacy_updated_datetime |
files.cases.annotations.notes |
files.cases.annotations.state |
files.cases.annotations.status |
files.cases.annotations.submitter_id |
files.cases.annotations.updated_datetime |
files.cases.demographic.created_datetime |
files.cases.demographic.demographic_id |
files.cases.demographic.ethnicity |
files.cases.demographic.gender |
files.cases.demographic.race |
files.cases.demographic.state |
files.cases.demographic.submitter_id |
files.cases.demographic.updated_datetime |
files.cases.demographic.year_of_birth |
files.cases.demographic.year_of_death |
files.cases.diagnoses.age_at_diagnosis |
files.cases.diagnoses.classification_of_tumor |
files.cases.diagnoses.created_datetime |
files.cases.diagnoses.days_to_birth |
files.cases.diagnoses.days_to_death |
files.cases.diagnoses.days_to_last_follow_up |
files.cases.diagnoses.days_to_last_known_disease_status |
files.cases.diagnoses.days_to_recurrence |
files.cases.diagnoses.diagnosis_id |
files.cases.diagnoses.last_known_disease_status |
files.cases.diagnoses.morphology |
files.cases.diagnoses.primary_diagnosis |
files.cases.diagnoses.prior_malignancy |
files.cases.diagnoses.progression_or_recurrence |
files.cases.diagnoses.site_of_resection_or_biopsy |
files.cases.diagnoses.state |
files.cases.diagnoses.submitter_id |
files.cases.diagnoses.tissue_or_organ_of_origin |

files.cases.diagnoses.tumor_grade |
files.cases.diagnoses.tumor_stage |
files.cases.diagnoses.updated_datetime |
files.cases.diagnoses.vital_status |
files.cases.diagnoses.treatments.created_datetime |
files.cases.diagnoses.treatments.days_to_treatment |
files.cases.diagnoses.treatments.state |
files.cases.diagnoses.treatments.submitter_id |
files.cases.diagnoses.treatments.therapeutic_agents |
files.cases.diagnoses.treatments.treatment_id |
files.cases.diagnoses.treatments.treatment_intent_type |
files.cases.diagnoses.treatments.treatment_or_therapy |
files.cases.diagnoses.treatments.updated_datetime |
files.cases.exposures.alcohol_history |
files.cases.exposures.alcohol_intensity |
files.cases.exposures.bmi |
files.cases.exposures.cigarettes_per_day |
files.cases.exposures.created_datetime |
files.cases.exposures.exposure_id |
files.cases.exposures.height |
files.cases.exposures.state |
files.cases.exposures.submitter_id |
files.cases.exposures.updated_datetime |
files.cases.exposures.weight |
files.cases.exposures.years_smoked |
files.cases.family_histories.created_datetime |
files.cases.family_histories.family_history_id |
files.cases.family_histories.relationship_age_at_diagnosis |
files.cases.family_histories.relationship_gender |
files.cases.family_histories.relationship_primary_diagnosis |
files.cases.family_histories.relationship_type |
files.cases.family_histories.relative_with_cancer_history |
files.cases.family_histories.state |
files.cases.family_histories.submitter_id |
files.cases.family_histories.updated_datetime |
files.cases.files.created_datetime |
files.cases.files.error_type |
files.cases.files.file_id |
files.cases.files.file_name |
files.cases.files.file_size |
files.cases.files.file_state |
files.cases.files.md5sum |
files.cases.files.state |
files.cases.files.state_comment |
files.cases.files.submitter_id |
files.cases.files.updated_datetime |
files.cases.project.dbgap_accession_number |
files.cases.project.disease_type |
files.cases.project.name |
files.cases.project.primary_site |
files.cases.project.project_id |
files.cases.project.released |
files.cases.project.state |
files.cases.project.program.dbgap_accession_number |
files.cases.project.program.name |
files.cases.project.program.program_id |
files.cases.samples.composition |
files.cases.samples.created_datetime |

files.cases.samples.current_weight |
files.cases.samples.days_to_collection |
files.cases.samples.days_to_sample_procurement |
files.cases.samples.freezing_method |
files.cases.samples.initial_weight |
files.cases.samples.intermediate_dimension |
files.cases.samples.is_ffpe |
files.cases.samples.longest_dimension |
files.cases.samples.oct_embedded |
files.cases.samples.pathology_report_uuid |
files.cases.samples.preservation_method |
files.cases.samples.sample_id |
files.cases.samples.sample_type |
files.cases.samples.sample_type_id |
files.cases.samples.shortest_dimension |
files.cases.samples.state |
files.cases.samples.submitter_id |
files.cases.samples.time_between_clamping_and_freezing |
files.cases.samples.time_between_excision_and_freezing |
files.cases.samples.tissue_type |
files.cases.samples.tumor_code |
files.cases.samples.tumor_code_id |
files.cases.samples.tumor_descriptor |
files.cases.samples.updated_datetime |
files.cases.samples.annotations.annotation_id |
files.cases.samples.annotations.case_id |
files.cases.samples.annotations.case_submitter_id |
files.cases.samples.annotations.category |
files.cases.samples.annotations.classification |
files.cases.samples.annotations.created_datetime |
files.cases.samples.annotations.creator |
files.cases.samples.annotations.entity_id |
files.cases.samples.annotations.entity_submitter_id |
files.cases.samples.annotations.entity_type |
files.cases.samples.annotations.legacy_created_datetime |
files.cases.samples.annotations.legacy_updated_datetime |
files.cases.samples.annotations.notes |
files.cases.samples.annotations.state |
files.cases.samples.annotations.status |
files.cases.samples.annotations.submitter_id |
files.cases.samples.annotations.updated_datetime |
files.cases.samples.portions.created_datetime |
files.cases.samples.portions.creation_datetime |
files.cases.samples.portions.is_ffpe |
files.cases.samples.portions.portion_id |
files.cases.samples.portions.portion_number |
files.cases.samples.portions.state |
files.cases.samples.portions.submitter_id |
files.cases.samples.portions.updated_datetime |
files.cases.samples.portions.weight |
files.cases.samples.portions.analytes.a260_a280_ratio |
files.cases.samples.portions.analytes.amount |
files.cases.samples.portions.analytes.analyte_id |
files.cases.samples.portions.analytes.analyte_type |
files.cases.samples.portions.analytes.analyte_type_id |
files.cases.samples.portions.analytes.concentration |
files.cases.samples.portions.analytes.created_datetime |
files.cases.samples.portions.analytes.spectrophotometer_method |

files.cases.samples.portions.analytes.state |
files.cases.samples.portions.analytes.submitter_id |
files.cases.samples.portions.analytes.updated_datetime |
files.cases.samples.portions.analytes.well_number |
files.cases.samples.portions.analytes.aliquots.aliquot_id |
files.cases.samples.portions.analytes.aliquots.amount |
files.cases.samples.portions.analytes.aliquots.analyte_type |
files.cases.samples.portions.analytes.aliquots.analyte_type_id |
files.cases.samples.portions.analytes.aliquots.concentration |
files.cases.samples.portions.analytes.aliquots.created_datetime |
files.cases.samples.portions.analytes.aliquots.source_center |
files.cases.samples.portions.analytes.aliquots.state |
files.cases.samples.portions.analytes.aliquots.submitter_id |
files.cases.samples.portions.analytes.aliquots.updated_datetime |
files.cases.samples.portions.analytes.aliquots.annotations.annotation_id |
files.cases.samples.portions.analytes.aliquots.annotations.case_id |
files.cases.samples.portions.analytes.aliquots.annotations.case_submitter_id |
files.cases.samples.portions.analytes.aliquots.annotations.category |
files.cases.samples.portions.analytes.aliquots.annotations.classification |
files.cases.samples.portions.analytes.aliquots.annotations.created_datetime |
files.cases.samples.portions.analytes.aliquots.annotations.creator |
files.cases.samples.portions.analytes.aliquots.annotations.entity_id |
files.cases.samples.portions.analytes.aliquots.annotations.entity_submitter_id |
files.cases.samples.portions.analytes.aliquots.annotations.entity_type |
files.cases.samples.portions.analytes.aliquots.annotations.legacy_created_datetime |
files.cases.samples.portions.analytes.aliquots.annotations.legacy_updated_datetime |
files.cases.samples.portions.analytes.aliquots.annotations.notes |
files.cases.samples.portions.analytes.aliquots.annotations.state |
files.cases.samples.portions.analytes.aliquots.annotations.status |
files.cases.samples.portions.analytes.aliquots.annotations.submitter_id |
files.cases.samples.portions.analytes.aliquots.annotations.updated_datetime |
files.cases.samples.portions.analytes.aliquots.center.center_id |
files.cases.samples.portions.analytes.aliquots.center.center_type |
files.cases.samples.portions.analytes.aliquots.center.code |
files.cases.samples.portions.analytes.aliquots.center.name |
files.cases.samples.portions.analytes.aliquots.center.namespace |
files.cases.samples.portions.analytes.aliquots.center.short_name |
files.cases.samples.portions.analytes.annotations.annotation_id |
files.cases.samples.portions.analytes.annotations.case_id |
files.cases.samples.portions.analytes.annotations.case_submitter_id |
files.cases.samples.portions.analytes.annotations.category |
files.cases.samples.portions.analytes.annotations.classification |
files.cases.samples.portions.analytes.annotations.created_datetime |
files.cases.samples.portions.analytes.annotations.creator |
files.cases.samples.portions.analytes.annotations.entity_id |
files.cases.samples.portions.analytes.annotations.entity_submitter_id |
files.cases.samples.portions.analytes.annotations.entity_type |
files.cases.samples.portions.analytes.annotations.legacy_created_datetime |
files.cases.samples.portions.analytes.annotations.legacy_updated_datetime |
files.cases.samples.portions.analytes.annotations.notes |
files.cases.samples.portions.analytes.annotations.state |
files.cases.samples.portions.analytes.annotations.status |
files.cases.samples.portions.analytes.annotations.submitter_id |
files.cases.samples.portions.analytes.annotations.updated_datetime |
files.cases.samples.portions.annotations.annotation_id |
files.cases.samples.portions.annotations.case_id |
files.cases.samples.portions.annotations.case_submitter_id |
files.cases.samples.portions.annotations.category |

files.cases.samples.portions.annotations.classification |
files.cases.samples.portions.annotations.created_datetime |
files.cases.samples.portions.annotations.creator |
files.cases.samples.portions.annotations.entity_id |
files.cases.samples.portions.annotations.entity_submitter_id |
files.cases.samples.portions.annotations.entity_type |
files.cases.samples.portions.annotations.legacy_created_datetime |
files.cases.samples.portions.annotations.legacy_updated_datetime |
files.cases.samples.portions.annotations.notes |
files.cases.samples.portions.annotations.state |
files.cases.samples.portions.annotations.status |
files.cases.samples.portions.annotations.submitter_id |
files.cases.samples.portions.annotations.updated_datetime |
files.cases.samples.portions.center.center_id |
files.cases.samples.portions.center.center_type |
files.cases.samples.portions.center.code |
files.cases.samples.portions.center.name |
files.cases.samples.portions.center.namespace |
files.cases.samples.portions.center.short_name |
files.cases.samples.portions.slides.created_datetime |
files.cases.samples.portions.slides.number_proliferating_cells |
files.cases.samples.portions.slides.percent_eosinophil_infiltration |
files.cases.samples.portions.slides.percent_granulocyte_infiltration |
files.cases.samples.portions.slides.percent_inflam_infiltration |
files.cases.samples.portions.slides.percent_lymphocyte_infiltration |
files.cases.samples.portions.slides.percent_monocyte_infiltration |
files.cases.samples.portions.slides.percent_necrosis |
files.cases.samples.portions.slides.percent_neutrophil_infiltration |
files.cases.samples.portions.slides.percent_normal_cells |
files.cases.samples.portions.slides.percent_stromal_cells |
files.cases.samples.portions.slides.percent_tumor_cells |
files.cases.samples.portions.slides.percent_tumor_nuclei |
files.cases.samples.portions.slides.section_location |
files.cases.samples.portions.slides.slide_id |
files.cases.samples.portions.slides.state |
files.cases.samples.portions.slides.submitter_id |
files.cases.samples.portions.slides.updated_datetime |
files.cases.samples.portions.slides.annotations.annotation_id |
files.cases.samples.portions.slides.annotations.case_id |
files.cases.samples.portions.slides.annotations.case_submitter_id |
files.cases.samples.portions.slides.annotations.category |
files.cases.samples.portions.slides.annotations.classification |
files.cases.samples.portions.slides.annotations.created_datetime |
files.cases.samples.portions.slides.annotations.creator |
files.cases.samples.portions.slides.annotations.entity_id |
files.cases.samples.portions.slides.annotations.entity_submitter_id |
files.cases.samples.portions.slides.annotations.entity_type |
files.cases.samples.portions.slides.annotations.legacy_created_datetime |
files.cases.samples.portions.slides.annotations.legacy_updated_datetime |
files.cases.samples.portions.slides.annotations.notes |
files.cases.samples.portions.slides.annotations.state |
files.cases.samples.portions.slides.annotations.status |
files.cases.samples.portions.slides.annotations.submitter_id |
files.cases.samples.portions.slides.annotations.updated_datetime |
files.cases.summary.file_count |
files.cases.summary.file_size |
files.cases.summary.data_categories.data_category |
files.cases.summary.data_categories.file_count |

files.cases.summary.experimental_strategies.experimental_strategy |
files.cases.summary.experimental_strategies.file_count |
files.cases.tissue_source_site.bcr_id |
files.cases.tissue_source_site.code |
files.cases.tissue_source_site.name |
files.cases.tissue_source_site.project |
files.cases.tissue_source_site.tissue_source_site_id |
files.center.center_id |
files.center.center_type |
files.center.code |
files.center.name |
files.center.namespace |
files.center.short_name |
files.downstream_analyses.analysis_id |
files.downstream_analyses.analysis_type |
files.downstream_analyses.created_datetime |
files.downstream_analyses.state |
files.downstream_analyses.submitter_id |
files.downstream_analyses.updated_datetime |
files.downstream_analyses.workflow_end_datetime |
files.downstream_analyses.workflow_link |
files.downstream_analyses.workflow_start_datetime |
files.downstream_analyses.workflow_type |
files.downstream_analyses.workflow_version |
files.downstream_analyses.output_files.access |
files.downstream_analyses.output_files.created_datetime |
files.downstream_analyses.output_files.data_category |
files.downstream_analyses.output_files.data_format |
files.downstream_analyses.output_files.data_type |
files.downstream_analyses.output_files.error_type |
files.downstream_analyses.output_files.experimental_strategy |
files.downstream_analyses.output_files.file_id |
files.downstream_analyses.output_files.file_name |
files.downstream_analyses.output_files.file_size |
files.downstream_analyses.output_files.file_state |
files.downstream_analyses.output_files.md5sum |
files.downstream_analyses.output_files.platform |
files.downstream_analyses.output_files.revision |
files.downstream_analyses.output_files.state |
files.downstream_analyses.output_files.state_comment |
files.downstream_analyses.output_files.submitter_id |
files.downstream_analyses.output_files.updated_datetime |
files.index_files.access |
files.index_files.created_datetime |
files.index_files.data_category |
files.index_files.data_format |
files.index_files.data_type |
files.index_files.error_type |
files.index_files.experimental_strategy |
files.index_files.file_id |
files.index_files.file_name |
files.index_files.file_size |
files.index_files.file_state |
files.index_files.md5sum |
files.index_files.platform |
files.index_files.revision |
files.index_files.state |
files.index_files.state_comment |

files.index_files.submitter_id |
files.index_files.updated_datetime |
files.metadata_files.access |
files.metadata_files.created_datetime |
files.metadata_files.data_category |
files.metadata_files.data_format |
files.metadata_files.data_type |
files.metadata_files.error_type |
files.metadata_files.file_id |
files.metadata_files.file_name |
files.metadata_files.file_size |
files.metadata_files.file_state |
files.metadata_files.md5sum |
files.metadata_files.state |
files.metadata_files.state_comment |
files.metadata_files.submitter_id |
files.metadata_files.type |
files.metadata_files.updated_datetime |
project.dbgap_accession_number |
project.disease_type |
project.name |
project.primary_site |
project.project_id |
project.released |
project.state |
project.program.dbgap_accession_number |
project.program.name |
project.program.program_id |
samples.composition |
samples.created_datetime |
samples.current_weight |
samples.days_to_collection |
samples.days_to_sample_procurement |
samples.freezing_method |
samples.initial_weight |
samples.intermediate_dimension |
samples.is_ffpe |
samples.longest_dimension |
samples.oct_embedded |
samples.pathology_report_uuid |
samples.preservation_method |
samples.sample_id |
samples.sample_type |
samples.sample_type_id |
samples.shortest_dimension |
samples.state |
samples.submitter_id |
samples.time_between_clamping_and_freezing |
samples.time_between_excision_and_freezing |
samples.tissue_type |
samples.tumor_code |
samples.tumor_code_id |
samples.tumor_descriptor |
samples.updated_datetime |
samples.annotations.annotation_id |
samples.annotations.case_id |
samples.annotations.case_submitter_id |
samples.annotations.category |

samples.annotations.classification |
samples.annotations.created_datetime |
samples.annotations.creator |
samples.annotations.entity_id |
samples.annotations.entity_submitter_id |
samples.annotations.entity_type |
samples.annotations.legacy_created_datetime |
samples.annotations.legacy_updated_datetime |
samples.annotations.notes |
samples.annotations.state |
samples.annotations.status |
samples.annotations.submitter_id |
samples.annotations.updated_datetime |
samples.portions.created_datetime |
samples.portions.creation_datetime |
samples.portions.is_ffpe |
samples.portions.portion_id |
samples.portions.portion_number |
samples.portions.state |
samples.portions.submitter_id |
samples.portions.updated_datetime |
samples.portions.weight |
samples.portions.analytes.a260_a280_ratio |
samples.portions.analytes.amount |
samples.portions.analytes.analyte_id |
samples.portions.analytes.analyte_type |
samples.portions.analytes.analyte_type_id |
samples.portions.analytes.concentration |
samples.portions.analytes.created_datetime |
samples.portions.analytes.spectrophotometer_method |
samples.portions.analytes.state |
samples.portions.analytes.submitter_id |
samples.portions.analytes.updated_datetime |
samples.portions.analytes.well_number |
samples.portions.analytes.aliquots.aliquot_id |
samples.portions.analytes.aliquots.amount |
samples.portions.analytes.aliquots.analyte_type |
samples.portions.analytes.aliquots.analyte_type_id |
samples.portions.analytes.aliquots.concentration |
samples.portions.analytes.aliquots.created_datetime |
samples.portions.analytes.aliquots.source_center |
samples.portions.analytes.aliquots.state |
samples.portions.analytes.aliquots.submitter_id |
samples.portions.analytes.aliquots.updated_datetime |
samples.portions.analytes.aliquots.annotations.annotation_id |
samples.portions.analytes.aliquots.annotations.case_id |
samples.portions.analytes.aliquots.annotations.case_submitter_id |
samples.portions.analytes.aliquots.annotations.category |
samples.portions.analytes.aliquots.annotations.classification |
samples.portions.analytes.aliquots.annotations.created_datetime |
samples.portions.analytes.aliquots.annotations.creator |
samples.portions.analytes.aliquots.annotations.entity_id |
samples.portions.analytes.aliquots.annotations.entity_submitter_id |
samples.portions.analytes.aliquots.annotations.entity_type |
samples.portions.analytes.aliquots.annotations.legacy_created_datetime |
samples.portions.analytes.aliquots.annotations.legacy_updated_datetime |
samples.portions.analytes.aliquots.annotations.notes |
samples.portions.analytes.aliquots.annotations.state |

samples.portions.analytes.aliquots.annotations.status |
samples.portions.analytes.aliquots.annotations.submitter_id |
samples.portions.analytes.aliquots.annotations.updated_datetime |
samples.portions.analytes.aliquots.center.center_id |
samples.portions.analytes.aliquots.center.center_type |
samples.portions.analytes.aliquots.center.code |
samples.portions.analytes.aliquots.center.name |
samples.portions.analytes.aliquots.center.namespace |
samples.portions.analytes.aliquots.center.short_name |
samples.portions.analytes.annotations.annotation_id |
samples.portions.analytes.annotations.case_id |
samples.portions.analytes.annotations.case_submitter_id |
samples.portions.analytes.annotations.category |
samples.portions.analytes.annotations.classification |
samples.portions.analytes.annotations.created_datetime |
samples.portions.analytes.annotations.creator |
samples.portions.analytes.annotations.entity_id |
samples.portions.analytes.annotations.entity_submitter_id |
samples.portions.analytes.annotations.entity_type |
samples.portions.analytes.annotations.legacy_created_datetime |
samples.portions.analytes.annotations.legacy_updated_datetime |
samples.portions.analytes.annotations.notes |
samples.portions.analytes.annotations.state |
samples.portions.analytes.annotations.status |
samples.portions.analytes.annotations.submitter_id |
samples.portions.analytes.annotations.updated_datetime |
samples.portions.annotations.annotation_id |
samples.portions.annotations.case_id |
samples.portions.annotations.case_submitter_id |
samples.portions.annotations.category |
samples.portions.annotations.classification |
samples.portions.annotations.created_datetime |
samples.portions.annotations.creator |
samples.portions.annotations.entity_id |
samples.portions.annotations.entity_submitter_id |
samples.portions.annotations.entity_type |
samples.portions.annotations.legacy_created_datetime |
samples.portions.annotations.legacy_updated_datetime |
samples.portions.annotations.notes |
samples.portions.annotations.state |
samples.portions.annotations.status |
samples.portions.annotations.submitter_id |
samples.portions.annotations.updated_datetime |
samples.portions.center.center_id |
samples.portions.center.center_type |
samples.portions.center.code |
samples.portions.center.name |
samples.portions.center.namespace |
samples.portions.center.short_name |
samples.portions.slides.created_datetime |
samples.portions.slides.number_proliferating_cells |
samples.portions.slides.percent_eosinophil_infiltration |
samples.portions.slides.percent_granulocyte_infiltration |
samples.portions.slides.percent_inflam_infiltration |
samples.portions.slides.percent_lymphocyte_infiltration |
samples.portions.slides.percent_monocyte_infiltration |
samples.portions.slides.percent_necrosis |
samples.portions.slides.percent_neutrophil_infiltration |

samples.portions.slides.percent_normal_cells |
samples.portions.slides.percent_stromal_cells |
samples.portions.slides.percent_tumor_cells |
samples.portions.slides.percent_tumor_nuclei |
samples.portions.slides.section_location |
samples.portions.slides.slide_id |
samples.portions.slides.state |
samples.portions.slides.submitter_id |
samples.portions.slides.updated_datetime |
samples.portions.slides.annotations.annotation_id |
samples.portions.slides.annotations.case_id |
samples.portions.slides.annotations.case_submitter_id |
samples.portions.slides.annotations.category |
samples.portions.slides.annotations.classification |
samples.portions.slides.annotations.created_datetime |
samples.portions.slides.annotations.creator |
samples.portions.slides.annotations.entity_id |
samples.portions.slides.annotations.entity_submitter_id |
samples.portions.slides.annotations.entity_type |
samples.portions.slides.annotations.legacy_created_datetime |
samples.portions.slides.annotations.legacy_updated_datetime |
samples.portions.slides.annotations.notes |
samples.portions.slides.annotations.state |
samples.portions.slides.annotations.status |
samples.portions.slides.annotations.submitter_id |
samples.portions.slides.annotations.updated_datetime |
summary.file_count |
summary.file_size |
summary.data_categories.data_category |
summary.data_categories.file_count |
summary.experimental_strategies.experimental_strategy |
summary.experimental_strategies.file_count |
tissue_source_site.bcr_id |
tissue_source_site.code |
tissue_source_site.name |
tissue_source_site.project |
tissue_source_site.tissue_source_site_id |

File Fields

Field Name

access
acl
created_datetime
data_category
data_format
data_type
error_type
experimental_strategy
file_id
file_name
file_size

Field Name

file_state
md5sum
origin
platform
revision
state
state_comment
submitter_id
tags
type
updated_datetime
analysis.analysis_id
analysis.analysis_type
analysis.created_datetime
analysis.state
analysis.submitter_id
analysis.updated_datetime
analysis.workflow_end_datetime
analysis.workflow_link
analysis.workflow_start_datetime
analysis.workflow_type
analysis.workflow_version
analysis.input_files.access
analysis.input_files.created_datetime
analysis.input_files.data_category
analysis.input_files.data_format
analysis.input_files.data_type
analysis.input_files.error_type
analysis.input_files.experimental_strategy
analysis.input_files.file_id
analysis.input_files.file_name
analysis.input_files.file_size
analysis.input_files.file_state
analysis.input_files.md5sum
analysis.input_files.platform
analysis.input_files.revision
analysis.input_files.state
analysis.input_files.state_comment
analysis.input_files.submitter_id

Field Name

analysis.input_files.updated_datetime
analysis.metadata.read_groups.adapter_name
analysis.metadata.read_groups.adapter_sequence
analysis.metadata.read_groups.base_caller_name
analysis.metadata.read_groups.base_caller_version
analysis.metadata.read_groups.created_datetime
analysis.metadata.read_groups.experiment_name
analysis.metadata.read_groups.flow_cell_barcode
analysis.metadata.read_groups.includes_spike_ins
analysis.metadata.read_groups.instrument_model
analysis.metadata.read_groups.is_paired_end
analysis.metadata.read_groups.library_name
analysis.metadata.read_groups.library_preparation_kit_catalog_number
analysis.metadata.read_groups.library_preparation_kit_name
analysis.metadata.read_groups.library_preparation_kit_vendor
analysis.metadata.read_groups.library_preparation_kit_version
analysis.metadata.read_groups.library_selection
analysis.metadata.read_groups.library_strand
analysis.metadata.read_groups.library_strategy
analysis.metadata.read_groups.platform
analysis.metadata.read_groups.read_group_id
analysis.metadata.read_groups.read_group_name
analysis.metadata.read_groups.read_length
analysis.metadata.read_groups.RIN
analysis.metadata.read_groups.sequencing_center
analysis.metadata.read_groups.sequencing_date
analysis.metadata.read_groups.size_selection_range
analysis.metadata.read_groups.spike_ins_concentration
analysis.metadata.read_groups.spike_ins_fasta
analysis.metadata.read_groups.state
analysis.metadata.read_groups.submitter_id
analysis.metadata.read_groups.target_capture_kit_catalog_number
analysis.metadata.read_groups.target_capture_kit_name
analysis.metadata.read_groups.target_capture_kit_target_region
analysis.metadata.read_groups.target_capture_kit_vendor
analysis.metadata.read_groups.target_capture_kit_version
analysis.metadata.read_groups.to_trim_adapter_sequence
analysis.metadata.read_groups.updated_datetime
analysis.metadata.read_groups.read_group_qcs.adapter_content

Field Name

analysis.metadata.read_groups.read_group_qcs.basic_statistics
analysis.metadata.read_groups.read_group_qcs.created_datetime
analysis.metadata.read_groups.read_group_qcs.encoding
analysis.metadata.read_groups.read_group_qcs.fastq_name
analysis.metadata.read_groups.read_group_qcs.kmer_content
analysis.metadata.read_groups.read_group_qcs.overrepresented_sequences
analysis.metadata.read_groups.read_group_qcs.per_base_n_content
analysis.metadata.read_groups.read_group_qcs.per_base_sequence_content
analysis.metadata.read_groups.read_group_qcs.per_base_sequence_quality
analysis.metadata.read_groups.read_group_qcs.per_sequence_gc_content
analysis.metadata.read_groups.read_group_qcs.per_sequence_quality_score
analysis.metadata.read_groups.read_group_qcs.per_tile_sequence_quality
analysis.metadata.read_groups.read_group_qcs.percent_gc_content
analysis.metadata.read_groups.read_group_qcs.read_group_qc_id
analysis.metadata.read_groups.read_group_qcs.sequence_duplication_levels
analysis.metadata.read_groups.read_group_qcs.sequence_length_distribution
analysis.metadata.read_groups.read_group_qcs.state
analysis.metadata.read_groups.read_group_qcs.submitter_id
analysis.metadata.read_groups.read_group_qcs.total_sequences
analysis.metadata.read_groups.read_group_qcs.updated_datetime
analysis.metadata.read_groups.read_group_qcs.workflow_end_datetime
analysis.metadata.read_groups.read_group_qcs.workflow_link
analysis.metadata.read_groups.read_group_qcs.workflow_start_datetime
analysis.metadata.read_groups.read_group_qcs.workflow_type
analysis.metadata.read_groups.read_group_qcs.workflow_version
annotations.annotation_id
annotations.case_id
annotations.case_submitter_id
annotations.category
annotations.classification
annotations.created_datetime
annotations.creator
annotations.entity_id
annotations.entity_submitter_id
annotations.entity_type
annotations.legacy_created_datetime
annotations.legacy_updated_datetime
annotations.notes
annotations.state

Field Name

annotations.status
annotations.submitter_id
annotations.updated_datetime
archive.archive_id
archive.created_datetime
archive.data_category
archive.data_format
archive.data_type
archive.error_type
archive.file_name
archive.file_size
archive.file_state
archive.md5sum
archive.revision
archive.state
archive.state_comment
archive.submitter_id
archive.updated_datetime
associated_entities.case_id
associated_entities.entity_id
associated_entities.entity_submitter_id
associated_entities.entity_type
cases.aliquot_ids
cases.analyte_ids
cases.case_id
cases.created_datetime
cases.days_to_index
cases.portion_ids
cases.sample_ids
cases.slide_ids
cases.state
cases.submitter_aliquot_ids
cases.submitter_analyte_ids
cases.submitter_id
cases.submitter_portion_ids
cases.submitter_sample_ids
cases.submitter_slide_ids
cases.updated_datetime
cases.annotations.annotation_id

Field Name

cases.annotations.case_id
cases.annotations.case_submitter_id
cases.annotations.category
cases.annotations.classification
cases.annotations.created_datetime
cases.annotations.creator
cases.annotations.entity_id
cases.annotations.entity_submitter_id
cases.annotations.entity_type
cases.annotations.legacy_created_datetime
cases.annotations.legacy_updated_datetime
cases.annotations.notes
cases.annotations.state
cases.annotations.status
cases.annotations.submitter_id
cases.annotations.updated_datetime
cases.demographic.created_datetime
cases.demographic.demographic_id
cases.demographic.ethnicity
cases.demographic.gender
cases.demographic.race
cases.demographic.state
cases.demographic.submitter_id
cases.demographic.updated_datetime
cases.demographic.year_of_birth
cases.demographic.year_of_death
cases.diagnoses.age_at_diagnosis
cases.diagnoses.classification_of_tumor
cases.diagnoses.created_datetime
cases.diagnoses.days_to_birth
cases.diagnoses.days_to_death
cases.diagnoses.days_to_last_follow_up
cases.diagnoses.days_to_last_known_disease_status
cases.diagnoses.days_to_recurrence
cases.diagnoses.diagnosis_id
cases.diagnoses.last_known_disease_status
cases.diagnoses.morphology
cases.diagnoses.primary_diagnosis
cases.diagnoses.prior_malignancy

Field Name

cases.diagnoses.progression_or_recurrence
cases.diagnoses.site_of_resection_or_biopsy
cases.diagnoses.state
cases.diagnoses.submitter_id
cases.diagnoses.tissue_or_organ_of_origin
cases.diagnoses.tumor_grade
cases.diagnoses.tumor_stage
cases.diagnoses.updated_datetime
cases.diagnoses.vital_status
cases.diagnoses.treatments.created_datetime
cases.diagnoses.treatments.days_to_treatment
cases.diagnoses.treatments.state
cases.diagnoses.treatments.submitter_id
cases.diagnoses.treatments.therapeutic_agents
cases.diagnoses.treatments.treatment_id
cases.diagnoses.treatments.treatment_intent_type
cases.diagnoses.treatments.treatment_or_therapy
cases.diagnoses.treatments.updated_datetime
cases.exposures.alcohol_history
cases.exposures.alcohol_intensity
cases.exposures.bmi
cases.exposures.cigarettes_per_day
cases.exposures.created_datetime
cases.exposures.exposure_id
cases.exposures.height
cases.exposures.state
cases.exposures.submitter_id
cases.exposures.updated_datetime
cases.exposures.weight
cases.exposures.years_smoked
cases.family_histories.created_datetime
cases.family_histories.family_history_id
cases.family_histories.relationship_age_at_diagnosis
cases.family_histories.relationship_gender
cases.family_histories.relationship_primary_diagnosis
cases.family_histories.relationship_type
cases.family_histories.relative_with_cancer_history
cases.family_histories.state
cases.family_histories.submitter_id

Field Name

cases.family_histories.updated_datetime
cases.files.created_datetime
cases.files.error_type
cases.files.file_id
cases.files.file_name
cases.files.file_size
cases.files.file_state
cases.files.md5sum
cases.files.state
cases.files.state_comment
cases.files.submitter_id
cases.files.updated_datetime
cases.project.dbgap_accession_number
cases.project.disease_type
cases.project.name
cases.project.primary_site
cases.project.project_id
cases.project.released
cases.project.state
cases.project.program.dbgap_accession_number
cases.project.program.name
cases.project.program.program_id
cases.samples.composition
cases.samples.created_datetime
cases.samples.current_weight
cases.samples.days_to_collection
cases.samples.days_to_sample_procurement
cases.samples.freezing_method
cases.samples.initial_weight
cases.samples.intermediate_dimension
cases.samples.is_ffpe
cases.samples.longest_dimension
cases.samples.oct_embedded
cases.samples.pathology_report_uuid
cases.samples.preservation_method
cases.samples.sample_id
cases.samples.sample_type
cases.samples.sample_type_id
cases.samples.shortest_dimension

Field Name

cases.samples.state
cases.samples.submitter_id
cases.samples.time_between_clamping_and_freezing
cases.samples.time_between_excision_and_freezing
cases.samples.tissue_type
cases.samples.tumor_code
cases.samples.tumor_code_id
cases.samples.tumor_descriptor
cases.samples.updated_datetime
cases.samples.annotations.annotation_id
cases.samples.annotations.case_id
cases.samples.annotations.case_submitter_id
cases.samples.annotations.category
cases.samples.annotations.classification
cases.samples.annotations.created_datetime
cases.samples.annotations.creator
cases.samples.annotations.entity_id
cases.samples.annotations.entity_submitter_id
cases.samples.annotations.entity_type
cases.samples.annotations.legacy_created_datetime
cases.samples.annotations.legacy_updated_datetime
cases.samples.annotations.notes
cases.samples.annotations.state
cases.samples.annotations.status
cases.samples.annotations.submitter_id
cases.samples.annotations.updated_datetime
cases.samples.portions.created_datetime
cases.samples.portions.creation_datetime
cases.samples.portions.is_ffpe
cases.samples.portions.portion_id
cases.samples.portions.portion_number
cases.samples.portions.state
cases.samples.portions.submitter_id
cases.samples.portions.updated_datetime
cases.samples.portions.weight
cases.samples.portions.analytes.a260_a280_ratio
cases.samples.portions.analytes.amount
cases.samples.portions.analytes.analyte_id
cases.samples.portions.analytes.analyte_type

Field Name

cases.samples.portions.analytes.analyte_type_id
cases.samples.portions.analytes.concentration
cases.samples.portions.analytes.created_datetime
cases.samples.portions.analytes.spectrophotometer_method
cases.samples.portions.analytes.state
cases.samples.portions.analytes.submitter_id
cases.samples.portions.analytes.updated_datetime
cases.samples.portions.analytes.well_number
cases.samples.portions.analytes.aliquots.aliquot_id
cases.samples.portions.analytes.aliquots.amount
cases.samples.portions.analytes.aliquots.analyte_type
cases.samples.portions.analytes.aliquots.analyte_type_id
cases.samples.portions.analytes.aliquots.concentration
cases.samples.portions.analytes.aliquots.created_datetime
cases.samples.portions.analytes.aliquots.source_center
cases.samples.portions.analytes.aliquots.state
cases.samples.portions.analytes.aliquots.submitter_id
cases.samples.portions.analytes.aliquots.updated_datetime
cases.samples.portions.analytes.aliquots.annotations.annotation_id
cases.samples.portions.analytes.aliquots.annotations.case_id
cases.samples.portions.analytes.aliquots.annotations.case_submitter_id
cases.samples.portions.analytes.aliquots.annotations.category
cases.samples.portions.analytes.aliquots.annotations.classification
cases.samples.portions.analytes.aliquots.annotations.created_datetime
cases.samples.portions.analytes.aliquots.annotations.creator
cases.samples.portions.analytes.aliquots.annotations.entity_id
cases.samples.portions.analytes.aliquots.annotations.entity_submitter_id
cases.samples.portions.analytes.aliquots.annotations.entity_type
cases.samples.portions.analytes.aliquots.annotations.legacy_created_datetime
cases.samples.portions.analytes.aliquots.annotations.legacy_updated_datetime
cases.samples.portions.analytes.aliquots.annotations.notes
cases.samples.portions.analytes.aliquots.annotations.state
cases.samples.portions.analytes.aliquots.annotations.status
cases.samples.portions.analytes.aliquots.annotations.submitter_id
cases.samples.portions.analytes.aliquots.annotations.updated_datetime
cases.samples.portions.analytes.aliquots.center.center_id
cases.samples.portions.analytes.aliquots.center.center_type
cases.samples.portions.analytes.aliquots.center.code
cases.samples.portions.analytes.aliquots.center.name

Field Name

cases.samples.portions.analytes.aliquots.center.namespace
cases.samples.portions.analytes.aliquots.center.short_name
cases.samples.portions.analytes.annotations.annotation_id
cases.samples.portions.analytes.annotations.case_id
cases.samples.portions.analytes.annotations.case_submitter_id
cases.samples.portions.analytes.annotations.category
cases.samples.portions.analytes.annotations.classification
cases.samples.portions.analytes.annotations.created_datetime
cases.samples.portions.analytes.annotations.creator
cases.samples.portions.analytes.annotations.entity_id
cases.samples.portions.analytes.annotations.entity_submitter_id
cases.samples.portions.analytes.annotations.entity_type
cases.samples.portions.analytes.annotations.legacy_created_datetime
cases.samples.portions.analytes.annotations.legacy_updated_datetime
cases.samples.portions.analytes.annotations.notes
cases.samples.portions.analytes.annotations.state
cases.samples.portions.analytes.annotations.status
cases.samples.portions.analytes.annotations.submitter_id
cases.samples.portions.analytes.annotations.updated_datetime
cases.samples.portions.annotations.annotation_id
cases.samples.portions.annotations.case_id
cases.samples.portions.annotations.case_submitter_id
cases.samples.portions.annotations.category
cases.samples.portions.annotations.classification
cases.samples.portions.annotations.created_datetime
cases.samples.portions.annotations.creator
cases.samples.portions.annotations.entity_id
cases.samples.portions.annotations.entity_submitter_id
cases.samples.portions.annotations.entity_type
cases.samples.portions.annotations.legacy_created_datetime
cases.samples.portions.annotations.legacy_updated_datetime
cases.samples.portions.annotations.notes
cases.samples.portions.annotations.state
cases.samples.portions.annotations.status
cases.samples.portions.annotations.submitter_id
cases.samples.portions.annotations.updated_datetime
cases.samples.portions.center.center_id
cases.samples.portions.center.center_type
cases.samples.portions.center.code

Field Name

cases.samples.portions.center.name
cases.samples.portions.center.namespace
cases.samples.portions.center.short_name
cases.samples.portions.slides.created_datetime
cases.samples.portions.slides.number_proliferating_cells
cases.samples.portions.slides.percent_eosinophil_infiltration
cases.samples.portions.slides.percent_granulocyte_infiltration
cases.samples.portions.slides.percent_inflam_infiltration
cases.samples.portions.slides.percent_lymphocyte_infiltration
cases.samples.portions.slides.percent_monocyte_infiltration
cases.samples.portions.slides.percent_necrosis
cases.samples.portions.slides.percent_neutrophil_infiltration
cases.samples.portions.slides.percent_normal_cells
cases.samples.portions.slides.percent_stromal_cells
cases.samples.portions.slides.percent_tumor_cells
cases.samples.portions.slides.percent_tumor_nuclei
cases.samples.portions.slides.section_location
cases.samples.portions.slides.slide_id
cases.samples.portions.slides.state
cases.samples.portions.slides.submitter_id
cases.samples.portions.slides.updated_datetime
cases.samples.portions.slides.annotations.annotation_id
cases.samples.portions.slides.annotations.case_id
cases.samples.portions.slides.annotations.case_submitter_id
cases.samples.portions.slides.annotations.category
cases.samples.portions.slides.annotations.classification
cases.samples.portions.slides.annotations.created_datetime
cases.samples.portions.slides.annotations.creator
cases.samples.portions.slides.annotations.entity_id
cases.samples.portions.slides.annotations.entity_submitter_id
cases.samples.portions.slides.annotations.entity_type
cases.samples.portions.slides.annotations.legacy_created_datetime
cases.samples.portions.slides.annotations.legacy_updated_datetime
cases.samples.portions.slides.annotations.notes
cases.samples.portions.slides.annotations.state
cases.samples.portions.slides.annotations.status
cases.samples.portions.slides.annotations.submitter_id
cases.samples.portions.slides.annotations.updated_datetime
cases.summary.file_count

Field Name

cases.summary.file_size
cases.summary.data_categories.data_category
cases.summary.data_categories.file_count
cases.summary.experimental_strategies.experimental_strategy
cases.summary.experimental_strategies.file_count
cases.tissue_source_site.bcr_id
cases.tissue_source_site.code
cases.tissue_source_site.name
cases.tissue_source_site.project
cases.tissue_source_site.tissue_source_site_id
center.center_id
center.center_type
center.code
center.name
center.namespace
center.short_name
downstream_analyses.analysis_id
downstream_analyses.analysis_type
downstream_analyses.created_datetime
downstream_analyses.state
downstream_analyses.submitter_id
downstream_analyses.updated_datetime
downstream_analyses.workflow_end_datetime
downstream_analyses.workflow_link
downstream_analyses.workflow_start_datetime
downstream_analyses.workflow_type
downstream_analyses.workflow_version
downstream_analyses.output_files.access
downstream_analyses.output_files.created_datetime
downstream_analyses.output_files.data_category
downstream_analyses.output_files.data_format
downstream_analyses.output_files.data_type
downstream_analyses.output_files.error_type
downstream_analyses.output_files.experimental_strategy
downstream_analyses.output_files.file_id
downstream_analyses.output_files.file_name
downstream_analyses.output_files.file_size
downstream_analyses.output_files.file_state
downstream_analyses.output_files.md5sum

Field Name

downstream_analyses.output_files.platform
downstream_analyses.output_files.revision
downstream_analyses.output_files.state
downstream_analyses.output_files.state_comment
downstream_analyses.output_files.submitter_id
downstream_analyses.output_files.updated_datetime
index_files.access
index_files.created_datetime
index_files.data_category
index_files.data_format
index_files.data_type
index_files.error_type
index_files.experimental_strategy
index_files.file_id
index_files.file_name
index_files.file_size
index_files.file_state
index_files.md5sum
index_files.platform
index_files.revision
index_files.state
index_files.state_comment
index_files.submitter_id
index_files.updated_datetime
metadata_files.access
metadata_files.created_datetime
metadata_files.data_category
metadata_files.data_format
metadata_files.data_type
metadata_files.error_type
metadata_files.file_id
metadata_files.file_name
metadata_files.file_size
metadata_files.file_state
metadata_files.md5sum
metadata_files.state
metadata_files.state_comment
metadata_files.submitter_id
metadata_files.type

Field Name

metadata_files.updated_datetime

Annotation Fields

Field Name |

— |
annotation_id |
case_id |
case_submitter_id |
category |
classification |
created_datetime |
entity_id |
entity_submitter_id |
entity_type |
legacy_created_datetime |
legacy_updated_datetime |
notes |
state |
status |
submitter_id |
updated_datetime |
project.code |
project.dbgap_accession_number |
project.disease_type |
project.name |
project.primary_site |
project.program.dbgap_accession_number |
project.program.name |
project.program.program_id |
project.project_id |
project.released |
project.state |

Field Group Listing by Endpoint

Projects Field Groups

Field Group Name

program

summary

summary.data_categories

summary.experimental_strategies

Cases Field Groups

Field Group Name

annotations
demographic
diagnoses
diagnoses.treatments
exposures
family_histories
files
files.analysis
files.analysis.input_files
files.analysis.metadata
files.analysis.metadata.read_groups
files.analysis.metadata.read_groups.read_group_qcs
files.archive
files.cases
files.cases.annotations
files.cases.demographic
files.cases.diagnoses
files.cases.diagnoses.treatments
files.cases.exposures
files.cases.family_histories
files.cases.files
files.cases.project
files.cases.project.program
files.cases.samples
files.cases.samples.annotations
files.cases.samples.portions
files.cases.samples.portions.analytes
files.cases.samples.portions.analytes.aliquots
files.cases.samples.portions.analytes.aliquots.annotations
files.cases.samples.portions.analytes.aliquots.center
files.cases.samples.portions.analytes.annotations
files.cases.samples.portions.annotations
files.cases.samples.portions.center
files.cases.samples.portions.slides
files.cases.samples.portions.slides.annotations
files.cases.summary
files.cases.summary.data_categories
files.cases.summary.experimental_strategies
files.cases.tissue_source_site

Field Group Name

files.center

files.downstream_analyses

files.downstream_analyses.output_files

files.index_files

files.metadata_files

project

project.program

samples

samples.annotations

samples.portions

samples.portions.analytes

samples.portions.analytes.aliquots

samples.portions.analytes.aliquots.annotations

samples.portions.analytes.aliquots.center

samples.portions.analytes.annotations

samples.portions.annotations

samples.portions.center

samples.portions.slides

samples.portions.slides.annotations

summary

summary.data_categories

summary.experimental_strategies

tissue_source_site

Files Field Groups

Field Group Name

analysis

analysis.input_files

analysis.metadata

analysis.metadata.read_groups

analysis.metadata.read_groups.read_group_qcs

annotations

archive

associated_entities

cases

cases.annotations

cases.demographic

cases.diagnoses

Field Group Name

cases.diagnoses.treatments
cases.exposures
cases.family_histories
cases.files
cases.project
cases.project.program
cases.samples
cases.samples.annotations
cases.samples.portions
cases.samples.portions.analytes
cases.samples.portions.analytes.aliquots
cases.samples.portions.analytes.aliquots.annotations
cases.samples.portions.analytes.aliquots.center
cases.samples.portions.analytes.annotations
cases.samples.portions.annotations
cases.samples.portions.center
cases.samples.portions.slides
cases.samples.portions.slides.annotations
cases.summary
cases.summary.data_categories
cases.summary.experimental_strategies
cases.tissue_source_site
center
downstream_analyses
downstream_analyses.output_files
index_files
metadata_files

Annotations Field Groups

Field Group Name

project
project.program

GDC Conventions Supplemental

Field Group Name |

— | — |

File size | Listed in bytes |

Submitter id | String supplied by submitter |

Datetime | Combination of date and time of day in the form [-]CCYY-MM-DDThh:mm:ss |
Id | GDC supplied UUID ID |

Chapter 12

Appendix B: Key Terms

Appendix B: Key Terms

The following table provides definitions and explanations for terms and acronyms relevant to the content presented within this document.

Term	Definition
ACL	Access Control List
API	Application Programming Interface
BAM	Bidirectional Associative Memory
BCR	Biospecimen Core Resource
CSV	Comma Separated Values
DCC	Data Coordinating Center
eRA	Electronic Research Administration
GDC	Genomic Data Commons
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
MAGE-TAB	Microarray Gene Expression - Tabular format
NCI	National Cancer Institute
NIH	National Institutes of Health
Op	Operations
REST	Representational State Transfer
TARGET	Therapeutically Applicable Research to Generate Effective Treatments
TCGA	The Cancer Genome Atlas
TSV	Tab Separated Values
UUID	Universally Unique Identifier
URL	Universal Resource Locator
XML	Extensible Markup Language

Chapter 13

Appendix C: Format of Submission Queries and Responses

Appendix C: Format of Submission Requests and Responses

Format of Submission Request

The general format of JSON objects submitted to the GDC API is as follows:

```
1 {
2   "type": string,
3   "id": string,
4   "submitter_id": string,
5   "<properties>": any type,
6   "<relationship_name>": [
7     {
8       "id": string,
9       "submitter_id": string
10    },
11    ...
12  ]
13 }
```

The request must specify either an `id` or a `submitter_id`.

id: A string specifying the `id` of the node that the user is creating or updating. This is the persistent GDC UUID4 for the node. If it is preferred to refer to the node using a custom id, users can do so with the `submitter_id` field (described below).

submitter_id: A string specifying the custom id of the object the user is creating or updating. This is not the official GDC ID for the node.

<properties>: These key-value pairs will be applied to properties on the referenced node.

<relationship_name>: A JSON object that specifies a relationship (link) between the node and other nodes. Links are typically established using the `submitter_id` or `id` of the neighboring node.

Format of API Response to a Submission Request

The following fields are included in all API responses to submission requests.

```
1 {
2   "cases_related_to_created_entities_count": int,
3   "cases_related_to_updated_entities_count": int,
```

```

4  "code": int,
5  "created_entity_count": int,
6  "entities": [entities],
7  "entity_error_count": int,
8  "message": string,
9  "success": boolean,
10 "transaction_id": string,
11 "transactional_error_count": int,
12 "transactional_errors": [transactional_errors],
13 "updated_entity_count": int
14 }

```

cases_related_to_created_entities_count: Number of cases related to the created entities.

cases_related_to_updated_entities_count: Number of cases related to the updated entities.

code: The HTTP status code of the response message.

created_entity_count: Number of entities created.

entities: A list of entities of the form:

```

1 {
2  "action": string,
3  "errors": [entity_errors],
4  "id": string,
5  "related_cases": [object],
6  "type": string,
7  "unique_keys": [unique_keys],
8  "valid": boolean,
9  "warnings": [object]
10 }

```

entity_errors: A list of errors that occurred while parsing, validating, or performing a CRUD operation on a specific entity. Entity errors are of the form:

```

1 {
2  "keys": [string],
3  "message": string
4 }

```

unique_keys: Properties, or combinations of properties, that can be used to uniquely identify the node in the GDC. Unique_keys are of the form:

```

1 {
2  "project_id": string,
3  "submitter_id": string
4 }

```

entity_error_count: Number of entities that were not successful.

message: A human-readable message describing the transaction.

success: A boolean value stating whether the transaction was successful. If the value is False, then no changes will be made to the database.

transaction_id: A string specifying the transaction id.

transactional_error_count: Number of transactional errors that occurred.

transactional_errors: A list of transactional errors that have occurred. These errors are errors that are not specific to an individual entity. Transactional errors are of the form:

```

1 {
2  "message": string
3 }

```

updated_entity_count: Number of existing entities updated by the transaction.

Error Types

EntityNotFoundError A referenced entity was not found among existing entities and entities specified in the transaction.

MissingPropertyError A required property was not provided.

ValidationError A provided property did not pass a validation test.

Status Messages

API responses will contain a status for each entity specified in the request:

success: The desired transaction was successful and the entity's state was modified in the GDC.

valid: The desired transaction was not successful, but the transaction was not aborted because of this entity.

error: The desired transaction was not successful, and the transaction was aborted because of this entity. This entity did not pass validation or an internal error occurred when attempting to complete the transaction. The error state will be accompanied by a list of errors recorded about the entity (see label-error-messages).

Note: GDC API requests are transactional. An error with processing a node specified in the transaction will abort the transaction and will result in no changes being applied for any node involved in the transaction.

Chapter 14

Release Notes

API Release Notes

Version	Date
v1.19.0	February 20, 2019
v1.18.0	December 18, 2018
v1.17.0	November 7, 2018
v1.16.0	September 27, 2018
v1.15.0	August 23, 2018
v1.14.1	May 21, 2018
v1.14.0	April 23, 2018
v1.13.0	February 15, 2018
v1.12.0	December 21, 2017
v1.11.0	November 16, 2017
v1.10.0	August 22, 2017
v1.9.0	June 29, 2017
v1.8.0	May 9, 2017
v1.7.1	March 16, 2017
v1.5.0	October 31, 2016
v1.4.0	September 23, 2016
v1.3.1	September 7, 2016
v1.2.0	August 9, 2016
v1.1.0	May 25, 2016
v1.0.1	May 16, 2016

v1.19.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** February 20, 2019

New Features and Changes

- Added API features to support controlled access DAVE
- Updated API query endpoints to handle filtering of queries based on tokens
- Created login notification Endpoint
- Added hashing and logging for similar ES queries

Bugs Fixed Since Last Release

- Fixed bug where quick search ES query grows with each request
- Fixed bug where new file versions could be created when exactly the same existing metadata is uploaded
- Fixed bug where submitting to specific projects produced error that data already existed

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.18.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** December 18, 2018

New Features and Changes

- Update to auth for GDC Pre-Release Data Portal

Bugs Fixed Since Last Release

- None

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.17.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** November 7, 2018

New Features and Changes

- Created new index `cnv_centric`
- Created new index `cnv_occurrence_centric`
- Created new REST API endpoints for CNV
- Created mapping from aliquot to case for occurrence on `cnv_centric`
- Created new graphql endpoints for CNV
- Updated index `case_centric` to add `cnv`
- Updated index `gene_centric` to add `cnv`

Bugs Fixed Since Last Release

- Fixed bug to prevent users from deleting files in state submitted or released

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.16.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** September 27, 2018

New Features and Changes

- `Tissue_type` is now parsed from the BCR Biospecimen submission

Bugs Fixed Since Last Release

- Re-attach the `/files` path to the legacy blueprints in `gdcapi`
- Fix Internal Server Error when exporting JSON in Data Portal

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.15.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** August 23, 2018

New Features and Changes

- Added support for file versioning
- Added new API endpoints in support of versioning: history, files/versions, and files/versions/manifest
- Updated the submit and release process
- Created admin endpoint to allow for modifying file and project flags and states
- Populate version and release information for all existing files

Bugs Fixed Since Last Release

- None

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.14.1

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** May 21, 2018

New Features and Changes

- None

Bugs Fixed Since Last Release

- Update to XML submission to correctly import biospecimen relationships
- GraphQL submission queries return incorrect information

Known Issues and Workarounds

- The error message for not including a token is incorrect for controlled access API requests (e.g. bam slicing). It states that your token is invalid, but it should request that a token be supplied.
- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.14.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** April 23, 2018

New Features and Changes

- Updated support of BCR XML submission for Next Generation Cancer Models

Bugs Fixed Since Last Release

- None

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.13.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** February 15, 2018

New Features and Changes

- Reduced number of submission portal timeouts by optimizing graphql API requests

Bugs Fixed Since Last Release

- Fixed bug preventing submission of clinical data via BCR XML

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.12.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** December 21, 2017

New Features and Changes

- None

Bugs Fixed Since Last Release

- Updated submit project button on submission portal to work more quickly for projects with large numbers of nodes

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.11.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** November 16, 2017

New Features and Changes

- Added support for histogram aggregation for numeric fields

Bugs Fixed Since Last Release

- None

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.10.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** August 22, 2017

New Features and Changes

- Longitudinal clinical data can now be submitted to the GDC
- Created new entities and reorganized the location of many properties in the GDC Dictionary. A list of deprecated properties and their new locations can be found [here](#). Note that the deprecated properties still contain all information that they previously did. In a later release we will map the data from the old location to the new. Eventually, we will suspend submission to the old properties and will remove them entirely from the GDC.
- GDC Data Dictionary Changes Details
 - Created follow_up entity to support longitudinal clinical data
 - Deprecated clinical test entity
 - Modified acceptable values for Read Group properties
 - Modified Diagnosis entity
 - Modified Treatment entity
 - Modified Demographic entity
 - Modified Case entity
 - Added new tumor code, tumor id, and sample types to Sample entity to support OCG

- Added property `days_to_diagnosis` to Diagnosis entity
- Created Somatic Mutation Index entity
- Updated CaDSR CDE links in data dictionary
- Added new sample type `tumor` to sample entity
- Made `classification_of_tumor` on diagnosis entity non-required
- Added support for FM-AD to Genomic Profile Harmonization Workflow entity
- Added data type `Gene Level Copy Number Scores` to Copy Number Segment entity

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.9.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** June 29, 2017

New Features and Changes

- Updated ability to search data based on genes and mutations have been released. Discussion of the new endpoints and their use can be found in the Data Portal User Guide. These feature are limited to those projects for which the GDC contains open-access MAF files.
- API request parameter `from` now returns first record when `from=0`. Previously, `from=1` was the first record. `from=0` is now the default.

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.8.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** May 9, 2017

New Features and Changes

- Users can now control whether a set of files will be compressed or not when downloading. For further details see the [API User Guide](#).

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.7.1

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** March 16, 2017

New Features and Changes

- Submission: Due to Data Dictionary updates new submission templates may be required for users submitting JSON and TSV formats
- Submission: Entities in submitted state (assigned when the project has been submitted) cannot be deleted.
- Submission: When attempting to delete an entity that has child entities not specified in the request, an error message is generated that will include all of the child entities’ UUIDs.
- Submission: Entities associated with files uploaded to the GDC object store cannot be deleted until the associated file has been deleted.
- Re-enable Review, Submit, and Release functions for submission
- GDC Data Dictionary Changes
- Added “submittable” property to all entities
- Changed Read Group to category biospecimen
- Added many new clinical properties available for submission
- Added sample codes from Office of Cancer Genomics (OCG) to analyte and aliquot
- Slides can now be attached to sample rather than just portion
- `sample_type_id` is no longer required when submitting sample entities
- `analyte_type_id` is no longer required when submitting aliquot and analyte entities
- Clinical Test Entity is created for storing results of a variety of potential clinical tests related to the diagnosis -
- Genomic Profiling Report entity created for storing particular derived sequencing results
- Structural Variation entity created
- Project entity includes new field “Intended Release Date”
- Project entity includes new field “Releasable”

Bugs Fixed Since Last Release

- Fixed bug where boolean properties were not accepted with TSV submission

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.5.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** October 31, 2016

New Features and Changes

- API responds with an error when the request specifies an unsupported combination of **filters** and **facets**.
- In TSV submissions, trailing and leading whitespace, including non-ASCII whitespace characters, are stripped from property names and values.
- For released projects, any updates to previously submitted entities (i.e. "state": "submitted") will be included in the following GDC data release.
- Performance improvements for manifest generation.

Bugs Fixed Since Last Release

- Uploading certain unsupported metadata files caused the associated submission transactions to remain stuck in pending state.

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of "IS MISSING" or "NOT MISSING".
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.4.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** September 23, 2016

New Features and Changes

- Submission transaction log includes additional information to assist in tracking.
- Submission project state transitions are disabled temporarily while project release features are being improved.
- GDC data dictionary changes:
 - The **submittable** property was added to all entity types in the GDC data model. It indicates whether the entity type can be submitted by users.
 - Category of Read Group entities in the GDC Data Model has changed from **data_bundle** to **biospecimen**.
 - Analyte entities support an expanded set of **analyte_type** values.

Bugs Fixed Since Last Release

- None to report

Known Issues and Workarounds

- API search & retrieval queries that do not include a **sort** parameter may return results in different order each time they are executed. This is a particular problem for paginated responses (i.e. responses to queries for which the number of results is greater than the **size** parameter).
 - **Workaround:** Include a **sort** parameter in API search & retrieval queries.
- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.3.1

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** September 7, 2016

New Features and Changes

- Successful **_dry_run** submission transactions can be committed to the GDC data model without having to re-upload metadata. The transactions can also be closed to prevent them from being committed in the future.
- Submission transactions can be submitted in asynchronous mode. In this mode, the GDC API will issue an immediate acknowledgement of the transaction, along with the **transaction_id**. The status of the transaction can be verified by the user at a later time by specifying the **transaction_id**. Users submitting large transactions may find this mode helpful.
- New submission transaction properties can be queried with GraphQL
- GDC Data Dictionary changes:
 - Clinical Supplement entities can have **data_format** set to OMF.
 - Biospecimen Supplement entities can have **data_format** set to SSF or PPS.
 - Read group **instrument_model** can be set to “Illumina HiSeq 4000”.
 - Category of Slide entities in the GDC Data Model has changed from **data_bundle** to **biospecimen**.

Bugs Fixed Since Last Release

- Incorrect BMI calculation in the import of BCR XML files.

Known Issues and Workarounds

- API search & retrieval queries that do not include a **sort** parameter may return results in different order each time they are executed. This is a particular problem for paginated responses (i.e. responses to queries for which the number of results is greater than the **size** parameter).
 - **Workaround:** Include a **sort** parameter in API search & retrieval queries.
- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.2.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** August 9, 2016

New Features and Changes

- Tarballs generated by the **data** endpoint in response to multi-file data download requests now include a folder structure that puts each file in a folder whose name is the file's UUID.
- UUIDs in clinical XML files are no longer treated in a case-sensitive way by the **submission** endpoint.
- Improved performance of **submission** endpoint for transactions that include many cases.
- Speed improvements for the **submission** endpoint.
- BCR XML is no longer validated against its XSD at submission.

Bugs Fixed Since Last Release

- Fixed handling of POST requests to address problems with cart functionality in older versions of Firefox
- Files of category **related_files** can now be downloaded from the **data** endpoint.
- Allowed submission by projects in certain dbGaP registration states that were previously blocked.

Known Issues and Workarounds

- API search & retrieval queries that do not include a **sort** parameter may return results in different order each time they are executed. This is a particular problem for paginated responses (i.e. responses to queries for which the number of results is greater than the **size** parameter).
 - **Workaround:** Include a **sort** parameter in API search & retrieval queries.
- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.1.0

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** May 25, 2016

New Features and Changes

- BAM index files (.bai) are now automatically downloaded with parent BAM.

Bugs Fixed Since Last Release

- None to report

Bugs Fixed Since Last Release

- Sorting by file **submitter_id** no longer causes an internal server error
- BAM index files are now included with harmonized BAM files
- Certain very long API requests will time out. It is recommended to break up into a series of smaller requests.

Known Issues and Workarounds

- Fields are not counted as missing if parent field is also missing. This may occur with queries of nested fields in the Data Portal Advanced Search or an API query using a filter. This behavior could impact results reported using search parameters of “IS MISSING” or “NOT MISSING”.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.

v1.0.1

- **GDC Product:** Application Programming Interface (API)
- **Release Date:** May 16, 2016

New Features and Changes

- HTTP interface that uses JSON as the primary data exchange format
- Programmatic access to functionality provided by GDC Data and Submission portals, via **projects**, **cases**, **files**, **annotations**, **data**, **slicing**, **status**, and **submission** endpoints
- Programmatic access to GDC Legacy Archive via **legacy** endpoint
- Token-based authentication for secure access to controlled data and to submission functionality
- RESTful search that supports simple and complex queries via **filters**, **fields**, and **facets** parameters, and **project**, **files**, **cases**, and **annotations** endpoints.
- Search results can be sorted using **sort** parameter, paginated using **size** and **from** parameters, and output in JSON, TSV, and XML using **format** and **pretty** parameters.
- **_mapping** endpoint enables user discovery of fields available for data search and retrieval operations
- Support for downloading of individual files and of archives containing multiple files
- Generation of download and upload manifests for use with the GDC Data Transfer Tool
- BAM slicing functionality for downloading part(s) of a BAM file specified using chromosomal coordinates or HGNC gene names
- Transactional submission system that links individual data elements according to a graph-based GDC Data Model
- Two data entity identifiers: UUIDs, which are consistent across GDC, and Submitter IDs, for compatibility with submitters' tracking systems

Bugs Fixed Since Last Release

- None to report

Known Issues and Workarounds

- Use of non-ascii characters in token passed to Data Transfer Tool will produce incorrect error message “Internal server error: Auth service temporarily unavailable”.
- Use of a decimal in an integer search field produces unexpected error.
- Certain very large API requests will time out. It is recommended to break up very large requests into a series of smaller requests.