Correlation Engine Query API
Version 0.30

Illumina Confidential. Do Not Distribute.
Table of Contents

Overview of the Correlation Engine Query API .............................................................. 6
Documentation Conventions ............................................................................................ 6
Sample Workflow ............................................................................................................. 7
Running a Correlation ..................................................................................................... 7
Filtering the Results ....................................................................................................... 8
Obtaining Study Details .................................................................................................. 10
Correlating the Biosets in the Study ............................................................................. 12
Obtaining Bioset Details ............................................................................................... 13
Obtaining Bioset Features ............................................................................................. 14
Obtaining Projects ......................................................................................................... 16
Obtaining Studies ........................................................................................................... 16
Downloading Data ......................................................................................................... 18
General Concepts ........................................................................................................... 19
API Key .......................................................................................................................... 19
URLs ............................................................................................................................... 19
API Version Numbers .................................................................................................... 19
Output Formats ................................................................................................................ 20
JSON Formats .................................................................................................................. 20
XML format ...................................................................................................................... 21
Translating JSON results into XML ................................................................................ 22
Raw Results ..................................................................................................................... 22
The Correlation Engine API Testbed ............................................................................... 23
Entering your keys .......................................................................................................... 23
Selecting the Format ....................................................................................................... 24
Selecting a Command ..................................................................................................... 25
Choosing Parameters .................................................................................................... 25
Seeing the URL and Running the Command ................................................................. 26
Common URL Parameters ............................................................................................. 29
Query Parameter: q .......................................................................................................... 29
Paging Parameters: fr and nr ....................................................................................... 29
Correlation context ........................................................................................................ 29
Context groups ............................................................................................................... 29
Data Types ...................................................................................................................... 31
Basic Data Types ........................................................................................................... 32
  integer ......................................................................................................................... 32
  float ............................................................................................................................. 32
  string ........................................................................................................................... 32
  array ........................................................................................................................... 32
  map ............................................................................................................................... 33
  object ........................................................................................................................... 33
  null fields .................................................................................................................... 33
Response Formats .......................................................................................................... 34
Success Response .......................................................................................................... 34
The Error Response ....................................................................................................... 34
The Raw Response ........................................................................................................ 35
Correlation Engine Query API Commands ................................................................. 36
Command Overview ........................................................................................................ 36
  Correlation commands ........................................................................................... 36
  Data access commands ......................................................................................... 36
  Other query commands ......................................................................................... 36
  Filter IDs ................................................................................................................ 37
  Miscellaneous commands ...................................................................................... 37
Details of the Commands ............................................................................................. 37
  alldatatypes ........................................................................................................... 37
  allexperimentdesigns ....................................................................................... 38
Correlation Engine Query API

GeneGWASBioSetObject ........................................................................................................ 98
GenomeBuild ....................................................................................................................... 99
IdAndName ........................................................................................................................... 99
LiteratureItem ....................................................................................................................... 100
OntologyInfoObject ........................................................................................................... 100
OntologyHierarchyObject ................................................................................................... 100
OntologyChildNode ............................................................................................................ 101
Organism ............................................................................................................................. 101
RelatedGene ......................................................................................................................... 101
Score .................................................................................................................................... 102
Sequence ............................................................................................................................. 102
ShortBioSet .......................................................................................................................... 102
ShortStudy ........................................................................................................................... 102
ShortTrial ............................................................................................................................. 103
Statistic .................................................................................................................................. 104
Study ..................................................................................................................................... 104
StudyAnnotation .................................................................................................................... 106
StudyBioSet ........................................................................................................................... 107
Tag .......................................................................................................................................... 107
TagCloudItem ....................................................................................................................... 107
ApiPMFacetGroup ................................................................................................................. 108
ApiPMFacetElement .............................................................................................................. 109
ApiTagCloudElement .......................................................................................................... 109
allstudy .................................................................................................................................... 109
Project .................................................................................................................................... 110
GeneGeneralInfo ................................................................................................................. 110
CategoryStudy ..................................................................................................................... 116
Change History ..................................................................................................................... 117
Version 0.30 ......................................................................................................................... 117
Version 0.29 ......................................................................................................................... 117
Version 0.28 ......................................................................................................................... 117
Version 0.27 ......................................................................................................................... 117
Version 0.26 ......................................................................................................................... 117
  • Extend the knockdownatlas command to include biostIds and gene perturbation type for each

CategoryGeneObject ............................................................................................................. 117
Version 0.25 ......................................................................................................................... 117
Version 0.24 ......................................................................................................................... 117
Version 0.23 ......................................................................................................................... 117
Version 0.22 ......................................................................................................................... 118
Version 0.21 ......................................................................................................................... 118
Version 0.20 ......................................................................................................................... 118
Version 0.19 ......................................................................................................................... 118
Version 0.18 ......................................................................................................................... 118
Version 0.17 ......................................................................................................................... 118
Version 0.16 ......................................................................................................................... 118
Version 0.15 ......................................................................................................................... 118
Version 0.14 ......................................................................................................................... 118
Version 0.13 ......................................................................................................................... 119
Version 0.12 ......................................................................................................................... 119
Version 0.11 ......................................................................................................................... 119
Version 0.10 ......................................................................................................................... 119
Version 0.09 ......................................................................................................................... 119
Version 0.08 ......................................................................................................................... 119
Version 0.7.1 ......................................................................................................................... 119
Version 0.7 ........................................................................................................................... 119
Version 0.6 ........................................................................................................................... 119
Version 0.5 ........................................................................................................................... 119
Overview of the Correlation Engine Query API

The Correlation Engine Query API can be used to query the Correlation Engine system using a REST based API. Requests are made by opening URLs on the Correlation Engine server. The response is available in either JSON\(^1\) or XML formats.

The API mimics the flow of the Correlation Engine web pages. For example, on a Data Correlations page a user queries the correlations based on a query term, e.g. a disease or gene. The results show a list of studies, correlated to the query term. The user may page through the results, and may apply filters, such as an organism or keywords. The user may expand a study to see the biosets in that study, and how they correlate to the original query term. The user may also request additional details about any study or bioset.

Similarly, there is a correlation request in the API to obtain correlation results for a query term. The correlation results provide a list of studies. The request may include parameters in the correlation request to apply filters, or to page through the results.

Once a list of studies has been returned, a correlationbiosets request may be sent to obtain the biosets in a given study, and how they correlate to the original query term. The study and bioset requests can then be sent to obtain more details about a specific study and bioset.

**Documentation Conventions**

In this document, code samples, URLs, parameter names and values are shown in fixed type. Text in normal fixed type represents literal values. Text in **bold fixed type** represents a value that will need to be replaced with the actual value. For example, in the URL:

```
```

*company, command, version, key,* and *parameters* will all be replaced with actual values, for example, company will be replaced with your company name. Please note that the examples show public data that may have been changed or deleted. If an example does not work with the ID shown, please query again to get current IDs.

---

\(^1\) [http://www.json.org/](http://www.json.org/)
Sample Workflow

This section will walk through some sample queries. Some of the concepts discussed may make more sense after reading the rest of the document. When reading this document the first time, it may make sense to skim this section, to get an idea of some of the commands. You can return to it in more detail after reading the other chapters.

The workflow will run through the following steps:
1. Retrieve studies correlated to a specific query term.
2. Filter those studies by species and keyword.
3. Obtain information about one of those studies.
4. Correlate that study against the original query term, obtaining the biosets and their scores.
5. Retrieve generic details about one of those biosets.
6. Retrieve the features in that bioset.
7. Download raw data for the bioset.

For readability, all results will be shown in formatted JSON, although, since this is supposed to represent actual queries, the fmt parameter will not be included in the URLs shown. If you are trying to reproduce these results in a browser, you should add fmt=prettyjson to all URLs. Also, you will need to substitute your own company's prefix and API key in the URL. If this paragraph is unclear, read it after reading General Concepts.

Due to space constraints, many of the sample results are in this document truncated. Typically only one or two results will be shown instead of the default ten. Long descriptions will be truncated to a single line. Actual query results may be different from those shown here. Lastly, in the sample results, fields that will be referred to elsewhere in the discussion are highlighted.

Running a Correlation

To run a query against breast cancer, use the correlation command:

https://<company>.ussc.informatics.illumina.com/c/nbapi/correlation.api?v=0&apik ey=key&q=breast+cancer

Note that since the query term contain spaces, the q parameter must be URL encoded.

The results provide a list of studies correlated to breast cancer:

```json
{"command": "correlation",
"parameters": {
    "apikey": "...",
    "nr": "2",
    "q": "breast cancer",
    "v": "0"
},
"result": {
    "context": "cT1icmVhc3QlMjBjYW5jZXImdHlwZT1kaXNlYXNlJm1kPTE0MjAyNw==",
    "studies": [
        {
            "score": null,
            "study": {
                "authors": null,
                "dataType": {
                    "id": 1,
                    "name": "RNA expression"
                },
                "id": 1430,
```
Filtering the Results

In this example, we will filter the above results to mouse studies containing the term "transgenic". First it is necessary to obtain the organism ID for mouse. The allorganisms command provides a list of all organisms in the Correlation Engine system:

https://<company>.ussc.informatics.illumina.com/c/nbapi/allorganisms.api?v=0&apikey=key

Here are the results (which can be cached, since they change very infrequently).

```json
{
  "command": "allorganisms",
  "parameters": {
    "apikey": "...",
    "v": "0"
  },
  "result": {
    "list": [
      {
        "id": 1,
        "name": "Homo sapiens"
      }
    ]
  }
}
```
The organism ID for mouse is 2. To run the **correlation** with the appropriate filter, use the following command:

```
https://<company>.ussc.informatics.illumina.com/c/nbapi/correlation.api?v=0&apikey=key&q=breast+cancer&organism=2&tf=transgenic&context=cT1icmVhc3QlMjBjYW5jZXImdHlwZT1kaXNlYXRlJmlkPTE0MjAyNw==
```

Note the additional parameters of `organism=2` specifying mouse, and `tf=transgenic` specifying the text filter. Note also the additional `context` parameter. The original query for "breast cancer" provided a `context` field in the response. This context should be passed to subsequent queries for the same query term. See [Correlation context](#) for more details.
Obtaining Study Details

Suppose you want more information about the first study in the above results. Note that the ID returned for that study is 1498. To obtain the additional information, run the study command specifying studyid=1498:

```
https://<company>.ussc.informatics.illumina.com/c/nbapi/study.api?v=0&apikey=key&studyid=1498
```

The results show information about this study. It is important to note that this is generic information about the study itself, outside the context of the query for breast cancer.

```json
{
    "command": "study",
    "parameters": {
        "studyid": "1498",
        "apikey": "...",
        "v": "0"
    },
    "result": {
        "associatedFiles": [
            {
                "id": 24091,
                "name": "RNA expression",
                "id": 1498,
                "institution": "Case Western Reserve School ...",
                "organism": {
                    "id": 2,
                    "name": "Mus musculus"
                },
                "shortDescription": "To identify early events of erbB2-induced ...",
                "title": "ErbB2-Neu mammary tumor progression"
            }
        ],
        "totalResults": 6
    }
}
```
"mimetype": "image/png",
"name": "Stat_Report_GSE4779_Boxplot_logbase2.png"
"type": "related"
},
{
"id": 24100,
"mimetype": "image/png",
"name": "Stat_Report_GSE4779/GPL1352_comparison_1_histogram.png"
"type": "related"
}
{
"id": b257,
"mimetype": "text/plain",
"name": "Breast_cancer_N1_vs_N0.txt"
"type": "bioset"
},
{
"id": "d61",
"mimetype": "text/plain",
"name": "GSE4779/GPL1352_Samples.txt",
"type": "data"
}
],
"authors": "Landis MD, Seachrist DD, Montanez-Wiscovich ME, ...",
"biosets": [
{
"bioset": {
"id": 13736,
"title": "MMTV-neu tumors _vs_ preneoplastic MMTV-neu mammary gland"
},
"featureCount": 2432
},
{
"bioset": {
"id": 13734,
"title": "MMTV-neu tumors _vs_ wild-type control"
},
"featureCount": 3797
},
{
"bioset": {
"id": 13735,
"title": "Preneoplastic MMTV-neu mammary gland _vs_ wild-type control"
},
"featureCount": 1010
}
],
"dataType": {
"id": 1,
"name": "RNA expression"
},
"description": "Series GSE2528
r
r
\nStatus Public on May 03, 2005 \r\n\r
Title Comparative Analysis of MMTV-neu tumors, preneoplastic MMTV-neu mammary gland, and Wild-type controls...",
"id": 1498,
"institution": "Case Western Reserve School of Medicine...",
"organism": {
Correlation Engine Query API

"id": 2,
"name": "Mus musculus"
},
"shortDescription": "To identify early events of erbB2-induced mammary...",
"tags": [
{
"id": 142027,
"name": "Breast cancer",
"type": "disease"
},
{
"id": 215437,
"name": "Mammary Glands, Human",
"type": "tissue"
},...
],
"title": "ErbB2-Neu mammary tumor progression"
}

Correlating the Biosets in the Study

The study command above provides general information about the study, but does not provide information about correlations to the query term, in this case "breast cancer". To retrieve information about how the biosets in the study correlate to the query term, use the correlationbiosets command:

https://<company>.ussc.informatics.illumina.com/c/nbapi/correlationbiosets.api?v=0&apikey=...q=breast+cancer&studyid=1498&context=cT1icmVhc3QlMjBjYW5jZXImdHlwZT1kaXNlYXNlJmlkPTE0MjAyNw==

You must pass in the original query term, the study ID, and the context from the correlation query response.

The results include the list of biosets, and various statistics for how those biosets correlate to the original query. (For a disease query, such as breast cancer, all of the statistics are null. Other queries such as a gene would include actual statistics.)

{
"command": "correlationbiosets",
"parameters": { 
"v": "0",
"q": "breast cancer",
"context": "cT1icmVhc3QlMjBjYW5jZXImdHlwZT1kaXNlYXNlJmlkPTE0MjAyNw==",
"studyid": "1498",
"apikey": "."
},
"result": {
"biosets": [
{
"bioset": {
"id": 13736,
"title": "MMTV-neu tumors _vs_ preneoplastic MMTV-neu mammary gland"
},
"correlation": null,
"direction": null,
"pvalue": null,
"score": null,
"..."
Obtaining Bioset Details

You can obtain more information about each of the biosets in the study. There are two commands to obtain information about the bioset. The `bioset` command retrieves all information about the bioset except for the features:

```
https://<company>.ussc.informatics.illumina.com/c/nbapi/bioset.api?v=0&apikey=ke
y&biosetid=13736
```

The biosetid was obtained from the `correlationbiosets` command or the `study` command.

The response includes information about the bioset:

```
{
    "command": "bioset",
    "parameters": {
        "v": "0",
        "biosetid": "13736",
        "apikey": "..."
    },
    "result": {
        "dataType": {
            "id": 1,
            "name": "RNA expression"
        },
        "description": "GSE2528 MMTV-neu mammary tumors...",
        "featureCount": 2432,
        "geneCount": 2244,
```
"id": 13736,
"organism": {  
   "id": 2,
   "name": "Mus musculus"
},
"platform": "Affymetrix GeneChip Human X3P",
"studyId": 1498,
"studyName": "ErbB2-Neu mammary tumor progression",
"tags": [
   {   
      "id": 142027,
      "name": "Breast cancer",
      "type": "disease"
   },
   {   
      "id": 215437,
      "name": "Mammary Glands, Human",
      "type": "tissue"
   },
   {   
      "id": 195777,
      "name": "Primary tissue - fresh or fresh frozen",
      "type": "biosource"
   },
   {   
      "id": 195786,
      "name": "disease vs. disease",
      "type": "biodesign"
   },
   {   
      "id": 195795,
      "name": "Gene overexpression",
      "type": "genemode"
   },
   {   
      "id": 284764,
      "name": "Erbb2",
      "type": "gene"
   }
],
"title": "MMTV-neu tumors _vs_ preneoplastic MMTV-neu mammary gland"
}

**Obtaining Bioset Features**

The `biosetfeatures` command returns the list of features in the bioset:

https://<company>.ussc.informatics.illumina.com/c/nbapi/biosetfeatures.api?v=0&apikey=key&biosetid=13736

Here is the result:

{   
   "command": "biosetfeatures",
   "parameters": {

"v": "0",
"nr": "2",
"biosetid": "13736",
"apikey": "...
",
"result": {
"features": [ 
{
  "id": 320494,
  "importedid": "95714_at",
  "name": "0610009D07Rik",
  "relatedGenes": null,
  "sequence": null,
  "statistics": [
    {
      "name": "test expression",
      "value": "7.78"
    },
    {
      "name": "control expression",
      "value": "4.74"
    },
    {
      "name": "fold change",
      "value": "1.64"
    },
    {
      "name": "p-value",
      "value": "0.00917"
    }
  ]
},
{
  "id": 320512,
  "importedid": "97242_at",
  "name": "0610010O12Rik",
  "statistics": [ 
    {
      "name": "control expression",
      "value": "4.28"
    },
    {
      "name": "fold change",
      "value": "2.65"
    },
    {
      "name": "test expression",
      "value": "11.4"
    },
    {
      "name": "p-value",
      "value": "0.0183"
    }
  ]
},
...
],
"totalFeatures": 2432
Obtaining Projects

The allProjects command returns the list of projects for the login name given as a parameter, else if the login is empty, the command returns all projects of current user:

https://<company>.ussc.informatics.illumina.com/c/nbapi/allProjects.es.api?v=0&apikey=key&fmt=prettyjsontext&loginName=abc@xyz.com

Here is the result:

```json
{
    "command": "allProjects",
    "parameters": {
        "v": "0",
        "fmt": "prettyjsontext",
        "apikey": "...",
        "loginName": "abc@xyz.com"
    },
    "result": {
        "apiProjectList": [
            {
                "desc": null,
                "id": 515,
                "title": "ABC's Private Project"
            }
        ]
    }
}
```

Obtaining Studies

The allstudies command returns the list of studies for the login name given as a parameter, else if the login is empty, the command returns all studies of current user:

https://<company>.ussc.informatics.illumina.com/c/nbapi/allstudies.api?v=0&apikey=key&fmt=prettyjsontext&loginName=abc@xyz.com

Here is the result:

```json
{
    "command": "allstudies",
    "parameters": {
        "v": "0",
        "fmt": "prettyjsontext",
        "apikey": "...
    },
    "result": {
        "apiAllStudyList": [
            {
                "desc": null,
                "id": 3317,
                "speciesid": 2,
                "studydatatypeid": 1,
                "title": "ABC's Study"
            }
        ]
    }
}
```
"summary": null,
"title": "Mouse"
},

{
"desc": null,
"id": 4083,
"speciesid": 1,
"studydatatypeid": 1,
"summary": null,
"title": "Mini"
},

{
"desc": null,
"id": 4646,
"speciesid": 2,
"studydatatypeid": 1,
"summary": null,
"title": "cart"
},

{
"desc": null,
"id": 6719,
"speciesid": 1,
"studydatatypeid": 1,
"summary": null,
"title": "minitest 1205"
},

{
"desc": null,
"id": 7358,
"speciesid": 2,
"studydatatypeid": 1,
"summary": null,
"title": "DE1969"
},

{
"desc": null,
"id": 7360,
"speciesid": 2,
"studydatatypeid": 1,
"summary": null,
"title": "DE1969a"
},

{
"desc": null,
"id": 16822,
"speciesid": 1,
"studydatatypeid": 25,
"summary": null,
"title": "DNA1210"
},

{
"desc": null,
"id": 26536,
"speciesid": 2,
"studydatatypeid": 1,
"summary": null,
"title": "DE5236"
Downloading Data

In additional to accessing the data using JSON or XML calls, it is also possible to access raw data. In Obtaining Study Details, we obtained information about a study, including a list of AssociatedFiles. These associated files come in three types:

- related: files related to the original data, e.g. an image of a graph.
- bioset: a bioset file that was uploaded, when importing the study.
- data: raw data from the original study.

In this example, we will download one of the biosets.

In the previous study, one associated file was "Breast cancer N1 _vs_ N0.txt". This file has type "bioset", and id "b257". To download the file, use the URL:

https://<company>.ussc.informatics.illumina.com/c/nbapi/associatedfile.api?v=0&apikey=key&id=b257

Unlike other API calls, the result is not a JSON or XML string, but raw data. Since this file had a MIME type of text/plain, the result downloaded will be pure text, however, other associated files may be Excel spread sheets, images, or any other type of file.
General Concepts

This section talks about some general concepts of the Correlation Engine Query API.

API Key

To use the Correlation Engine API, you must obtain an API key from Illumina. This key is passed to all requests using the `apikey` parameter, and is the equivalent to signing in to Correlation Engine. Do not share your key with other people. Treat your key as you would a password.

The `autocomplete` API uses a separate API key from other commands. Autocomplete is typically run from the browser, using JavaScript. Keys embedded in the JavaScript can be observed by users of the application. To avoid giving access to other data, a separate key is provided. The `version` command will work with either API key.

Enterprise Customers may obtain both of these keys using the Illumina Labs link at the bottom of the Correlation Engine application.

URLs

Access to the Correlation Engine Query API is via URLs. While the URLs would normally be invoked by your application, you can also open them in a browser to see how the API behaves. See JSON Formats for the best way to display JSON results in your browser.

A typical URL for an API request is of the form


The first part of the URL will vary depending on your company or academic institution. Normally it is of the form https://<company>.ussc.informatics.illumina.com/c, but it may be different at your company or institution. Generally, it will be the same as the URL that you use to access the Correlation Engine application. Different requests are specified using a different command in the URL. Throughout this document, the term `command` will refer to a corresponding URL. For example, the `correlation` command refers to the `correlation.api` URL. Following the `?` in the URL, are the parameters. The parameters may be in any order, and are separated by &. Two parameters are required for all requests: `v`, and `apikey`. The `v` parameter specifies the version. This is the major version of the API in use, and is required to guarantee that changes in the Correlation Engine Query API will not break your application. See API Version Numbers for more information on version numbers. The `apikey` parameter specifies your API key.

Parameters in the URL containing spaces and non-ASCII or non-alphanumeric characters, should be encoded using UTF-8 encoding. For example, a query for "lung cancer" should be encoded as `q=lung+cancer` or `q=lung%20cancer`, and a query for "Elés" should be encoded as `q=E1%C3%A9s`.

API Version Numbers

API version numbers represent the version of the API. An API version number consists of a major and minor number, separated by a period, e.g. 1.2. The latest version number can be obtained using the `version` command of the API. That version number can be matched to the version number in the title of this document. The minor version number will be updated with every new release of the API. The major version number will change only when an incompatible change is made to the API. All requests must include the major version number should be passed in as the `v` parameter. For example, if you are using version 1.2, pass in `v=1`. When a new major version is released, Illumina may provide a compatible version of the API if an older API version is requested. For example, if the current API version is 2.3, and
the version of 1 is passed in, the response will be compatible with version 1. If it is no longer possible to provide a compatible response, an error will be returned instead.

Because JSON and XML are flexible formats, new attributes can be added to the responses, without breaking compatibility. The following are examples of changes to the API that will not change the major version number:

- Adding a new parameter to a URL, if omitting the parameter results in the same behavior as the previous version.
- Adding new attributes to the results of a request.

The following are examples of incompatible changes that will result in a new major version number:

- Removing or changing a parameter in the URL, unless the old version will operate correctly.
- Removing or changing an attribute in a response.

Be aware that a request for a specific major version number does not guarantee that the results are the same as the results documented for that version, only that they are compatible. For example, a request for version 1, may return some results only available in version 2, however, those results will be compatible with version 1.

During the beta, the major version is 0. During this beta, Illumina may make incompatible changes without updating the major version number, and may not support older versions. Information about such changes will be made available to customers using the beta. Once version 1 is released, Illumina will maintain compatibility.

**Output Formats**

The Correlation Engine Query API supports two primary output formats: JSON and XML. The output format chosen is based on the fmt parameter of the URL. Possible values of fmt are json, prettyjson, prettyjsontext, and xml. The first three all provide JSON output, however, the options affect the formatting. See [JSON Formats](#) for more details. If the fmt parameter is omitted or unrecognized, the output format will be json. Most of the sample responses in this document will use prettyjsontext formatting, however, the fmt parameter will be omitted from URLs shown in the examples.

**JSON Formats**


There are three values of the fmt parameter that produce JSON: json, prettyjson, and prettyjsontext. json produces raw, unformatted JSON, while the other two format it for readability. The default value if fmt is omitted is json.

The json format produces raw JSON, with no additional whitespace. It is the most compact format available, and is intended for production use of the Correlation Engine Query API. The MIME type of the result is application/json. (The mime type will be discussed in the explanation of prettyjsontext.)

This request:

https://<company>.ussc.informatics.illumina.com/c/nbapi/version.api?v=0&apikey=...&fmt=json

produces the following output:

```
{"command":"version","parameters":{"v":0,"fmt":json,"apikey":"..."},"result":{"major":0,"minor":1}}
```

The result displayed here is shown on two lines to fit in this document; the actual result is a single line.
The `prettyjson` format adds white space and line breaks to make the JSON output more readable. The result will be interpreted the same by any JSON reading software, but can be easier to debug during development. The MIME type is `application/json`.

This request:

```
https://<company>.ussc.informatics.illumina.com/c/nbapi/version.api?v=0&apikey=KEY&fmt=prettyjson
```

produces the following output:

```
{
  "command": "version",
  "parameters": {
    "v": "0",
    "fmt": "prettyjson",
    "apikey": "...
  },
  "result": {
    "major": 0,
    "minor": 1
  }
}
```

(Note: Windows users who open this result in notepad may not see the line breaks. This is because the `prettyjson` output includes the line feed character, but not the carriage return. Use an alternate editor such as WordPad to view these results.)

A MIME type is the mechanism that a web server uses to describe what type of file it is providing. While `application/json` is the correct MIME type for JSON results, some browsers do not know how to display `application/json` natively, and require the use of an external program. To make it easier to view the results in a browser, you can set `fmt=prettyjsontext`. This will produce the same result as `prettyjson`, except that the MIME type will be `text/plain` and the browser will render the results without an external program. `prettyjsontext` may not work correctly in Internet Explorer; the page may open up in an external application regardless of this setting.

**XML format**

The `xml` value of `fmt` provides XML representation the results. The contents of the results are the same as the JSON representation, only the layout is different. No DTD or schema is available for the XML.

This request:

```
https://<company>.ussc.informatics.illumina.com/c/nbapi/version.api?v=0&apikey=KEY&fmt=xml
```

produces the following output:

```
<?xml version="1.0" encoding="UTF-8"?>
<nextbioResults><command>version</command><parameters><apikey>...</apikey><fmt xml"/fmt"></v></parameters><result><major>0</major><minor>1</minor></result>
</nextbioResults>
```

In this document, the output is shown on multiple lines due to space limitations; the actual results include just two lines. Although there is no prettyxml format available, some browsers, including Firefox and Internet Explorer, can format the XML output for easier reading. Here is how the above results look in Firefox:
Translating JSON results into XML

The XML is produced by translating the JSON output into XML. Most of the examples in this document will show JSON results; the corresponding XML is based on these translations. The following rules are used to translate from JSON to XML:

- The outermost XML tag is always `<nextbioResults>`.
- An object in JSON is represented as "name": "value". The corresponding XML is `<name>value</name>`. For example, "major": 0 in JSON translates to `<major>0</major>` in XML.
- Nested objects result in nested tags. For example, "dataType":{"id":12,"name":"therapeutic"} translates to `<dataType><id>12</id><name>therapeutic</name></dataType>`.
- Array elements are placed in an element tag. For example: "ids": [123, 532] translates to `<ids><element>123</element><element>532</element></ids>`.
- Null elements are represented by a null attribute set to "true". For example "studies":null translates to `<studies null="true"/>`.

Raw Results

Some API commands return raw data. For example, the associatedfile command downloads a raw file, which may be an image, spreadsheet, or any other type of result. Such commands ignore the fmt parameter and do not return JSON or XML, but instead return the raw data. The exception is if an error occurs, in which case a standard error response will be returned based on the fmt parameter.
The Correlation Engine API Testbed

The Correlation Engine API Testbed allows you to experiment with the API. Using a form you can choose a command, and select parameters. The resulting URL and the results of the query are displayed.

To access the testbed, Click on the "Correlation Engine APIs" link at the bottom of the home page, select "Learn more about the Correlation Engine Query API" (near the bottom of the page), and then select "try the API here" (near the bottom of the page).

When you start the testbed, it will look similar to the following:

**Entering your keys**

The first step is to fill in your keys in the upper left section:

If you know your keys, or are using keys from a different account, you may fill them in directly. Only the keys that you are actually using need to be filled in. If you are testing autocomplete, fill in the Autocomplete key. If you are testing anything else, fill in the API key. You may fill in both.
If you wish to use the keys associated with your own account, it is easier to click "Fill in my keys". You may be requested to fill in your password.

Enter your password, and both keys will be filled in (if they are available).

If you are giving a demo and do not wish others to see your keys, click "Hide my keys".

When keys are hidden, they will also be masked in the URL displayed for your command

https://enterprise.nextbio.com/c/nbapi/literature.api?apikey=*****v=0&fmt=pretty)sontext&q=heart

Note that hiding your keys will block their display, however, another user with access to your web browser, may still be able to extract them from the page.

**Selecting the Format**

Select the format from the "fmt [format]" dropdown.
See Output Formats for a discussion of the different formats. Within the testbed, it is recommended that "prettyjsoncontext" or "xml" be used. Those formats should be displayed correctly within the box on the right. If you use "json" or "prettyjson", the results are up to the browser, but generally it will attempt to display them in an external program.

**Selecting a Command**

Select a command from the "command" drop down. All commands from this document should be available in the dropdown. However, optional extensions to the API may not be available.

**Choosing Parameters**

After choosing a command, the parameters available to that command will be displayed below the command selector. Enter the values for any parameters that you wish to set.
Some parameters may be repeated multiple times in the URL. When you enter a value for one of these fields, another copy of the field will be generated.

**Seeing the URL and Running the Command**

As you adjust the parameters, the URL in the upper right will be updated to reflect the command. This URL may be pasted into your application. (If you have chosen "Hide my keys" you will need to replace the "***" with your actual API key. Click the "Run" button to run the command.)
For XML output, the output is formatted by the browser. Some browsers will add additional formatting. For example, the following appears in Firefox:
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```xml
<nextbioResults>
  <command>literature</command>
  <parameters>
    <apikey>...</apikey>
    <fmt>xml</fmt>
    <q>heart</q>
    <v>0</v>
  </parameters>
  <result>
    <items>
      <element>
        <affiliations>
          <element>Ohio State University</element>
        </affiliations>
        <articleAbstract>
          Clinical sequelae of hypertension include heart failure, arrhythmias, and ischemic events, especially myocardial infarction and stroke. Recognizing the hypertensive heart has diagnostic as well...
        </articleAbstract>
      </element>
    </items>
  </result>
</nextbioResults>
```
Common URL Parameters

There are several parameters that are common to many commands. Rather than explaining them with each command, they will be discussed once here.

Three parameters have already been discussed:

- **apikey** specifies your key for accessing the Correlation Engine Query API. It must be provided for all commands. See API Key.
- **v** specifies the API version number. It must be provided for all commands. See API Version Numbers.
- **fmt** specifies the format for the results. It is optional on all commands. See Output Formats.

Since these three parameters are available for every command, they will not be listed as parameters for the commands, and will not be shown in sample URLs.

The following parameters are available on some commands. Those commands will list these parameters, but the explanation will be provided here:

**Query Parameter: q**

Commands that take a query parameter use the **q** parameter to specify the query term.

**Paging Parameters: fr and nr**

Commands that support pagination use two parameters: **fr** and **nr**. The first result to be returned is specified in the **fr** parameter (which stands for first result). Result numbering starts at 0, so **fr=0** will return results starting from the first. The default value for **fr** is zero, so if **fr** is omitted, results start from the first result.

The number of results is specified with **nr**. If omitted, a default value of 10 results is returned.

As an example, to obtain the third page of results, with 20 results per page, you would specify **fr=40&nr=20**.

**Correlation context**

When using the correlation family of commands, you will frequently perform several queries based on the same query term. After performing an initial query, you may page through the results, apply filters, or dig deeper using commands like **correlationbiosets**. The initial query performs some calculations that would be inefficient to recalculate on each subsequent query. The **context** parameter is used to save the results of these calculations and pass them to additional queries.

When the original correlation command is run, the context is calculated, and returned in a field called **context**. The value of this field should be saved, and passed in as the **context** to subsequent correlation requests.

For multiple calls to the correlation command, e.g. with different filters, the context is optional; if not provided, it will be recalculated. However, it is more efficient if the context is passed in if there are multiple calls with the same query term. For secondary commands, such as **correlationbiosets**, the context is required. The documentation for each command will specify if the context is require or optional.

**Context groups**

The context from some correlation commands cannot be used with certain other commands. Commands can be divided into groups, referred to as a "context group". Within the same context group, the context can and should be reused. However, context should not be passed to commands in other groups. For example, after using the correlation command, the context should be passed to subsequent correlation commands and to correlationbiosets commands, since they are both in the "correlation" context group. However, if you use a correlationbiogroups command, you should not pass the initial context from the correlation command. Instead, the context from the first correlationbiogroups command should be used in subsequent correlationbiogroups commands. The following are the context groups:
<table>
<thead>
<tr>
<th>Context Group</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>correlation</td>
<td>correlation, correlationbiosets</td>
</tr>
<tr>
<td>correlationbiogroups</td>
<td>correlationbiogroups</td>
</tr>
</tbody>
</table>
**Data Types**

While JSON is a compact and efficient form for transmitting data, there is no standard way of documenting the format of the results for a specific query—for example, that the results may consist of an integer "a", and a string "b". XML formats can be specified using DTDs or schemas, but those are designed for XML, while this API returns results in either JSON or XML. For clarity, all data types will be defined using tables with field name, field type, and a description.

Here is an example of a definition for a data type. Suppose a command was documented as returning the following results:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>integer</td>
<td>Description would go here</td>
</tr>
<tr>
<td>s</td>
<td>string</td>
<td>...</td>
</tr>
<tr>
<td>w</td>
<td>Widget</td>
<td>...</td>
</tr>
</tbody>
</table>

Lower case names indicate a primitive type. In this example, the results would contain an integer named "i" and a string named "s". Names that begin with uppercase indicate an object defined in the Result Objects chapter. So you would look up Widget to find its definition. Suppose Widget was defined as:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>Description would go here</td>
</tr>
<tr>
<td>f</td>
<td>float</td>
<td>...</td>
</tr>
</tbody>
</table>

Widget contains a string named "name" and a float named "f".

The documentation defines the fields in the result, but it does not define the order. For example, it is possible that the s field will appear before the i field in the results. In JSON the order of the results is not significant. In XML the order can be significant, but in this case it is not.

Here is a sample return value in JSON:

```json
{
    "i": "5",
    "w": {
        "name": "my widget",
        "f": 3.14159
    },
    "s": "a string"
}
```

Note that the w field was output before the s field, even though the s field is shown before w in the table above.

Here is the same output in XML:

```xml
<tag>
    <i>5</i>
    <w>
        <name>my widget</name>
        <f>3.14159</f>
    </w>
    <s>a string</s>
</tag>
```

The `<tag>` tag was used arbitrarily in this example; the actual tag would depend on where this XML appeared.
**Basic Data Types**

The following sections describe the basic types, and how they are represented in JSON and XML.

**integer**

An integer represent a positive or negative integer. In JSON the number is output without quotes. The following are some sample integers in JSON:

```json
  "id":1234
  "score":-3
  "n":null
```

Here are the same examples in XML:

```xml
  <id>1234</id>
  <score>-3</score>
  <n null="true"/>
```

**float**

A float is a floating point number. The number can be output in exponential notation. In JSON the number is output without quotes. Here are some sample floats in JSON:

```json
  "pvalue": 1.23E-7
  "r":-23
  "score":null
```

Here are the same examples in XML:

```xml
  <pvalue>1.23E-7</pvalue>
  <r>-23</r>
  <score null="true"/>
```

**string**

A string represents text data. In JSON, the string will be encoded with quotes. The string uses standard escape sequences, including \" for a double quote, and \n for a line break. See [http://www.json.org/](http://www.json.org/) for a complete list of string escape sequences. Note that strings may contain numeric values. Here are some standard strings in JSON:

```json
  "s":"my string"
  "empty":"
  "v":"123"
  "quoted":"A string including \"quotes\"
  "whitespace":"A string with blanks and\n line break"
  "entities":"&gt;&lt;\&"
  "nullstring":null
```

In XML, white space in a string will be included directly in the string. Standard XML entities will be used. Be aware that when viewing the XML in a browser, the white space will be compressed. Here are the same examples in XML:

```xml
  <s>my string</s>
  <empty/>
  <v>123</v>
  <quoted>A string including "quotes"</quoted>
  <whitespace>A string with blanks and a line break</whitespace>
  <entities>&lt;&gt;&amp;</entities>
  <nullstring null="true"/>
```

**array**

An array represents multiple values. An array is defined by adding [] after a type, for example:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>intarray</td>
<td>integer[]</td>
<td>An array of integers</td>
</tr>
<tr>
<td>stringarray</td>
<td>string[]</td>
<td>An array of strings</td>
</tr>
<tr>
<td>widgetarray</td>
<td>Widget[]</td>
<td>An array of widgets</td>
</tr>
</tbody>
</table>

Illumina Confidential. Do Not Distribute.  Page 32
Unlike for other objects, the order of an array is significant. In JSON, arrays are represented using square brackets to surround the array, and separating the array elements with commas, for example:

```
"intarray": [5, 3]
"stringarray": ["ESR1", "GATA2", "HER2"]
"widgetarray": [
  {"name": "pi", "f": 3.14159},
  {"name": "e", "f": 2.71828}
]
"emptyarray": []
"nullarray": null
```

In XML, each array element is placed inside an `<element>` tag. Here are the same examples in XML:

```
<intarray>
  <element>5</element>
  <element>3</element>
</intarray>
<stringarray>
  <element>ESR1</element>
  <element>GATA2</element>
  <element>HER2</element>
</stringarray>
<widgetarray>
  <element><name>pi</name><f>3.14159</f></element>
  <element><name>e</name><f>2.71828</f></element>
</widgetarray>
<emptyarray/>
<nullarray null="true"/>
```

map

A map is a set of key/value pairs. The keys and values are always strings. In JSON, the map is represented as a JSON object. For example:

```
"parameters":{
  "q":"ESR1",
  "filter":"chemotherapy"
}
```

In xml, the keys become tags, for example:

```
<parameters>
  <q>ESR1</q>
  <filter>chemotherapy</filter>
<parameters>
```

object

When the type "object" is explicitly specified, that means that the type of the value will not always be the same type. See the documentation for that particular field for more explanation.

null fields

As you can see in the above examples, any type of field may be null. A null value indicates that the value is unavailable, or that it is not applicable for the particular request.
Response Formats

There are three types of responses returned by the Correlation Engine Query API. A "Success response" is returned when a query is successful. It contains information about the query, and the results of that query. An "Error response" is returned when the query fails, for example if a required parameter is missing, or if you don't have permission to execute your request. It includes information about the query and an error message. A "Raw response" is returned for certain queries that return raw data.

**Success Response**

A success response is returned for most queries when the query is successful. Here are the fields in a success response:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
<td>object</td>
<td>This is the result object. The actual format will vary based on the query.</td>
</tr>
<tr>
<td>command</td>
<td>string</td>
<td>The name of the command requested. This is the last component of the URL without the .api.</td>
</tr>
<tr>
<td>parameters</td>
<td>map</td>
<td>A map of all of the parameters passed into the request. The apikey parameter is replaced with &quot;...&quot;.</td>
</tr>
</tbody>
</table>

In the documentation of the commands, only the result portion of the response will be documented.

Here is a sample success response to the request

https://<company>.ussc.informatics.illumina.com/c/nbapi/version.api?v=0&apikey=key&fmt=prettyjson

produces the following output:

```
{
  "command": "version",
  "parameters": {
    "v": "0",
    "fmt": "prettyjson",
    "apikey": "..."
  },
  "result": {
    "major": 0,
    "minor": 1
  }
}
```

In documentation for the version command, only the following results would be shown:

```
"result": {
  "major": 0,
  "minor": 1
}
```

**The Error Response**

An error response is returned when there is an error is servicing the request. The error response can be distinguished from a success response by the presence of an error field, and the absence of a result field. Here are the fields in an error response:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>string</td>
<td>A string describing the error</td>
</tr>
<tr>
<td>command</td>
<td>string</td>
<td>The name of the command requested. This is the last component of the URL without the .api.</td>
</tr>
<tr>
<td>parameters</td>
<td>map</td>
<td>A map of all of the parameters passed into the request. The apikey parameter is replaced with &quot;...&quot;.</td>
</tr>
</tbody>
</table>

Here is a sample error response. In this case, the required parameter q was missing.
The Raw Response

Certain commands return raw data rather than structured data in JSON and XML format. For example, the associatedfile command will download a raw file. These commands will return raw data rather than a success response. If there is an error, an error response is returned.
Correlation Engine Query API Commands

This section lists all of the commands available in the Correlation Engine Query API. The first part is an overview of the commands, organized by function. This is followed by details of each command, organized alphabetically.

Command Overview

This section provides an overview of the commands, organized by function.

Correlation commands

The correlation commands allow you to find studies and biosets correlated to query terms.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>correlation</td>
<td>Returns all studies correlated with a specific query term or bioset. Various filters may be applied.</td>
</tr>
<tr>
<td>correlationbiosets</td>
<td>Once a correlation is run, correlationbiosets can provide the biosets in a specific study, and how they correlate with the query term.</td>
</tr>
<tr>
<td>correlationbiogroups</td>
<td>Returns all biogroups correlated with a specific bioset.</td>
</tr>
<tr>
<td>referencingstudies</td>
<td>Returns all studies whose tags or descriptive text refers to a search term.</td>
</tr>
<tr>
<td>referencingbiosets</td>
<td>Once referencingstudies is run, referencingbiosets can provide the biosets in a specific study that refer to the search term</td>
</tr>
</tbody>
</table>

Data access commands

The data access commands allow you to access the details of various items.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>study</td>
<td>Provides all information about a study, including the description, the list of biosets, the list of associated files, and the list of tags.</td>
</tr>
<tr>
<td>bioset</td>
<td>Provides all information about a bioset except for the list of features. The information includes the description and the list of tags.</td>
</tr>
<tr>
<td>biosetfeatures</td>
<td>Provides the list of features in a bioset.</td>
</tr>
<tr>
<td>associatedfile</td>
<td>Download one or more associated files.</td>
</tr>
<tr>
<td>allstudies</td>
<td>Finds and returns all studies for the current user. Based on the permission, the user can view studies of other users filling the additional parameter.</td>
</tr>
<tr>
<td>biogroups</td>
<td>Provides detailed info about biogroups</td>
</tr>
<tr>
<td>bodyatlas</td>
<td>Provides detailed body atlas information</td>
</tr>
<tr>
<td>tagcloud</td>
<td>Provides tag cloud information</td>
</tr>
<tr>
<td>ontologyinfo</td>
<td>Provides information about the ontology</td>
</tr>
<tr>
<td>pharmacoatlas</td>
<td>Provides pharmaco atlas information</td>
</tr>
<tr>
<td>snpgwasdata</td>
<td>Provides SNP GWAS data</td>
</tr>
</tbody>
</table>

Other query commands

These commands allow querying of data types other than studies and biosets (e.g. literature).

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>literature</td>
<td>Finds PubMed literature items matching a keyword. This command can also return a tag cloud of related terms.</td>
</tr>
<tr>
<td>trials</td>
<td>Finds clinical trials matching a keyword. This command can also return a tag cloud of related terms.</td>
</tr>
<tr>
<td>trial</td>
<td>Find complete information on a single clinical trial.</td>
</tr>
<tr>
<td>autocomplete</td>
<td>Finds terms that start with a query string.</td>
</tr>
<tr>
<td>allProjects</td>
<td>Finds and returns all project for current user. Based on the permission, the user can view projects of other users filling the additional parameter.</td>
</tr>
</tbody>
</table>
Filter IDs

These commands provide IDs of various Correlation Engine objects that can be used in filters. The information changes very infrequently, so you should read the information once when starting your application.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>allorganisms</td>
<td>Returns the list of all organisms recognized by Correlation Engine, and their IDs.</td>
</tr>
<tr>
<td>alldatatypes</td>
<td>Returns a list of all bioset data types (e.g. &quot;RNA expression&quot;) and their IDs.</td>
</tr>
<tr>
<td>allexperimentdesigns</td>
<td>Returns a list of all experiment designs and their IDs.</td>
</tr>
<tr>
<td>allsamplesources</td>
<td>Returns a list of all sample sources and their IDs.</td>
</tr>
<tr>
<td>allplatforms</td>
<td>Returns the list of all platforms.</td>
</tr>
<tr>
<td>libraries</td>
<td>Returns a list of all libraries for that domain.</td>
</tr>
<tr>
<td>studyids</td>
<td>Return a list of study ids.</td>
</tr>
</tbody>
</table>

Miscellaneous commands

These are commands that don't fit into any group.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>Returns the current version number of the Correlation Engine Query API.</td>
</tr>
</tbody>
</table>

Details of the Commands

This section describes the commands available in the Correlation Engine Query API. For each command we provide the following:

- A description of the command
- A list of all parameters accepted by that command (except for apikey, v, and fmt).
- One or more sample queries. Generally a simple query will be provided, followed by a more complex query if appropriate. In this section, a shortened version of URLs will be used. Everything before the /nbapi will be omitted, as will the apikey and v parameters. For example, a correlation request would be shown as: /nbapi/correlation.api?q=queryterm instead of https://<company>.ussc.informatics.illumina.com/c/nbapi/correlation.api?q=queryterm
- A description of the response. If the response consists of a single field described in the Result Objects section, then a link to that object will be provided. Otherwise a list of all the fields will be provided.
- A sample response. Only the result section of the success response will be shown.

alldatatypes

The alldatatypes command returns a list of all bioset data types (e.g. RNA expression) recognized by Correlation Engine. Each data type has a name and ID. The ID can be used to filter by data type. This information changes very infrequently, so you should cache it.

Parameters: none

Sample request: /nbapi/alldatatypes.api?

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>DataType[]</td>
<td>The list of all data types and their IDs</td>
</tr>
</tbody>
</table>

Sample Response:
"result": {"list": [  
  {
    "id": 1,  
    "name": "RNA expression"
  },
  {
    "id": 24,  
    "name": "miRNA expression"
  },
  ...  
]

allexperimentdesigns

The allexperimentdesigns command returns a list of all experiment designs (e.g. circadian time course) recognized by Correlation Engine. Each experiment design has a name and ID. The ID can be used to filter by experiment design. This information changes very infrequently, so you should cache it.

Parameters: none

Sample request: /nbapi/allexperimentdesigns.api?

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>IdAndName[]</td>
<td>The list of all experiment designs and their IDs</td>
</tr>
</tbody>
</table>

Sample Response:
"result": {"list": [  
  {
    "id": 204656,  
    "name": "circadian time course"
  },
  {
    "id": 218748,  
    "name": "development/differentiation"
  },
  ...
]

allorganisms

The allorganisms command returns a list of all organisms (e.g. Homo sapiens) recognized by Correlation Engine. Organisms are also frequently referred to as species in Correlation Engine. Each organism has a name and ID. The ID can be used to filter by organism. This information changes very infrequently, so you should cache it.

Parameters: none

Sample request: /nbapi/allorganisms.api?

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>IdAndName[]</td>
<td>The list of all sample sources and their IDs</td>
</tr>
</tbody>
</table>

Sample Response:
"result": {"list": [  
  {

Illumina Confidential. Do Not Distribute.  Page 38
The allplatforms command returns a list of all platforms (e.g. Affymetrix GeneChip Human 35k). Each platform has a name and ID. The ID can be used to filter by platform.

Parameters: none

Sample request: /nbapi/allplatforms.api?

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>IdAndName[]</td>
<td>The list of all platforms and their IDs</td>
</tr>
</tbody>
</table>

Sample Response:

```
{
  "command": "allplatforms",
  "parameters": {
    "v": "0",
    "fmt": "prettyjsontext",
    "apikey": "...
  },
  "result": {
    "list": [
      {
        "id": 1,
        "name": "Affymetrix GeneChip Human 35k"
      },
      {
        "id": 2,
        "name": "Affymetrix GeneChip Human Cancer HC-G110"
      },
      {
        "id": 3,
        "name": "Affymetrix GeneChip Human Muscle Chip"
      },
      {
        "id": 4,
        "name": "Affymetrix GeneChip Human Full Length HuGeneFL"
      },
      {
        "id": 5,
        "name": "Affymetrix GeneChip Human HG-Focus Target Array"
      },
      {
        "id": 6,
        "name": "Affymetrix GeneChip Human HG_U133A version [1 or 2]"
      }
    ]
  }
}
allsamplesources

The allsamplesources command returns a list of all sample sources (e.g. blood fraction) recognized by Correlation Engine. Each experiment design has a name and ID. The ID can be used to filter by sample source. This information changes very infrequently, so you should cache it.

Parameters: none

Sample request: /nbapi/allsamplesources.api?

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>IdAndName[]</td>
<td>The list of all sample sources and their IDs</td>
</tr>
</tbody>
</table>

Sample Response:

```
"result": {"list": [
    {
      "id": 195783,
      "name": "blood fraction"
    },
    {
      "id": 204645,
      "name": "bone marrow fraction"
    },
    ...
]
```

associatedfile

The associatedfile command downloads an associated file. The response is in raw format, e.g. if it is an image, an image file will be downloaded. The compress=zip parameter allows multiple associated files to be downloaded in zip format.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>yes</td>
<td>The ID of the file. This ID can be obtained from the study command. If compress=zip parameter is set, multiple id parameters may be specified.</td>
</tr>
<tr>
<td>compress</td>
<td>no</td>
<td>If set to zip, a zip file is downloaded. In this case, multiple id parameters may be specified, all of the files will be downloaded in a single zip file. The maximum uncompressed size of all of the files is 4GB.</td>
</tr>
</tbody>
</table>

Sample Requests:

```
/nbapi/associatedfile.api?id=b42
This downloads the file with id "b42" in its native format.
```

```
/nbapi/associatedfile.api?id=b42&id=d53&compress=zip
This downloads the files with ids "b42" and "d53" as a single zip file.
```
autocomplete
The autocomplete command returns a list of terms that start with a given query. Also included are synonyms for those terms.

IMPORTANT: The autocomplete command uses a different API Key from other commands. Attempting to access autocomplete using the general API key will result in an error.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term. All the results will have either terms or synonyms that start with this letter.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results to return. The default is 10, the maximum is 100.</td>
</tr>
<tr>
<td>type</td>
<td>no</td>
<td>An optional filter to return only items of this type. If this parameter is omitted, terms of all types are returned. Valid types are: GENE, BIOGROUP, TISSUE, DISEASE, COMPOUND, SNP, AUTHOR, BIOSOURCE, BIODESIGN, GENEMODE. Multiple type parameters may be included, in which case results of all specified types may be returned.</td>
</tr>
<tr>
<td>cv</td>
<td>no</td>
<td>Command version. This is an optional parameter. If it is omitted or set to 0 no action is taken compared to the previous version. If it is set to 1 (or higher) than type ‘DISEASE’ is replaced with ‘PHENOTYPE’.</td>
</tr>
<tr>
<td>alttype</td>
<td>no</td>
<td>An optional parameter, if true a value is returned for type that matches the XML import TagType. Example types are: Gene, Biogroup, Tissue, Disease, Compound, Snp, Author, Biosource, Biodesign, Genemode.</td>
</tr>
</tbody>
</table>

Sample Requests:

/nbapi/autocomplete.api?id=gata
This returns 10 results that begin with the letters "gata"

/nbapi/autocomplete.api?q=hea&nr=50&type=GENE&type=BIOGROUP
This returns 50 genes and biogroups that begin with the letters "hea".

Response: The response is a list of AutocompleteItems

Sample response:
"result": {"items": [  
  "id": 195186,  
  "name": "Dmel_CG6007",  
  "synonyms": [  
    "GATA",  
    "GatA",  
    "gatA",  
    "gata-PA",  
    "bene",  
    ...  
  ],  
  "type": "GENE"  
},  

"id": 195864,  
"name": "GATA5",  
"synonyms": [  
  "GATA binding factor-5",  
  "GATA binding protein 5",  
  "GATA-binding protein 5",  
  "bB379O24.1"  
],
"OTTHUMP00000031490",
"OTTMUSP00000017494",
"RP23-16P3.7",
"transcription factor GATA-5",
"Transcription factor GATA-5 (GATA binding factor-5)"
],
"type": "GENE"
},

allstudies

The allstudies command returns all the studies for the current user. Studies include information such as the description, speciesid, studydatatypeid, summary, url and title.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>loginName</td>
<td>no</td>
<td>An optional parameter that returns another user’s study information if the APIKEY provided has the correct credentials. Otherwise, returns all studies for the APIKEY user.</td>
</tr>
<tr>
<td>includeDomain</td>
<td>no</td>
<td>If set to on, the response includes all studies from that domain.</td>
</tr>
</tbody>
</table>

Sample request:

```
/nbapi/allstudies.api?
/nbapi/allstudies.api? loginName=abc@xyz.com
```

Response: The response is a list of ApiAllStudy objects.

Sample Response:
```
"result": {"apiAllStudyList": [  
   {  
      "desc": Integration study,  
      "id": 39978,  
      "projectlist": [640],  
      "samplespeciesid": 3,  
      "speciesid": 1,  
      "studydatatypeid": 21,  
      "summary": null,  
      "title": "Integration 5-18"  
      "url": "https://florida.nextbiosystem.com/c/study/ov.nb?id=39978"
   },  
   {  
      "desc": test study for DB,  
      "id": 39979,  
      "projectlist": [  
         9,  
         769  
      ],  
      "samplespeciesid": 1,  
      "speciesid": 1,  
      "studydatatypeid": 21,  
      "summary": summary testing,  
      "title": "Integrationtest MYSQL"  
      "url": "https://florida.nextbiosystem.com/c/study/ov.nb?id=39979"
   }
]}
```
allprojects
The allprojects command returns all the projects for the current user. Projects include information such as the description, libraryid, title and writable flag.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>loginName</td>
<td>no</td>
<td>An optional parameter that returns another user’s project information if the APIKEY provided has the special credentials. Otherwise, returns all projects for the APIKEY user.</td>
</tr>
<tr>
<td>createDefaultIfEmpty</td>
<td>no</td>
<td>If set to on and the response has no projects, create a default new project for the user. Allows a user with special permissions to create project for another user (loginName) when the user has no projects.</td>
</tr>
<tr>
<td>includeDomain</td>
<td>no</td>
<td>If set to on, the response includes all projects in that domain</td>
</tr>
</tbody>
</table>

Sample request:

 /nbapi/allprojects.api?
 /nbapi/allprojects.api? loginName=abc@xyz.com

Response: The response is a list of ApiProject objects. The project information includes description, libraryid to which the project belongs, title of the project and a writable flag. This writable flag is true, if the user has permission to write to this project. The response also includes the total number of projects.

Sample Response:

```
"result": {  
   "apiProjectList": [  
   {     
      "desc": "Immune response project",
      "id": 1,
      "libraryid": 1,
      "studylist": [       
       8627,
       8631,
       8637,
       8818
      ],
      "title": "Immune response and infection",
      "writable": false
    },
    {     
      "desc": null,
      "id": 949,
      "libraryid": 100,
      "studylist": [15670],
      "title": "0223 staging import",
      "writable": true
    },
    {     
      "desc": "0504 test project",
      "id": 4327,
```
biogroups

The biogroups command allows the user to find biogroups for which your queried phenotype, compound or bioset is highly enriched.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term.</td>
</tr>
<tr>
<td>type</td>
<td>only when type = bioset</td>
<td>If the biogroup is against a bioset, type must be set to bioset. Otherwise, type is optional, and if provided should be the type of the query term.</td>
</tr>
<tr>
<td>id</td>
<td>only when type = bioset</td>
<td>If the type is a bioset, this should be the id of the bioset, which can be obtained from other queries. Otherwise, this is optional, and if provided should be the id of the query term.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested for ‘bioset’ query. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>Number of results requested for ‘bioset’ query. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>tf</td>
<td>no</td>
<td>A text filter. If the type is a bioset, Only those biogroups that contain text matching this value will be returned.</td>
</tr>
<tr>
<td>taxf</td>
<td>no</td>
<td>If the type is a bioset, Limit the results to biogroups in the specific taxonomy. Multiple taxf parameters can be provided, in which case biogroups belonging to any of the taxonomies will be included. The following are the values supported for taxf:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>taxf</th>
<th>Biogroup Type</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>ontologies</td>
<td>Gene Ontology</td>
<td>GO</td>
</tr>
<tr>
<td>targets</td>
<td>miRNA targets</td>
<td>TargetScan</td>
</tr>
<tr>
<td>pathways</td>
<td>Canonical pathways</td>
<td>Broad MSigDB</td>
</tr>
<tr>
<td>positional</td>
<td>Positional gene sets</td>
<td>Broad MSigDB</td>
</tr>
<tr>
<td>families</td>
<td>Protein families</td>
<td>InterPro, PMAP</td>
</tr>
<tr>
<td>motifs</td>
<td>Regulatory motifs</td>
<td>Broad MSigDB</td>
</tr>
</tbody>
</table>

Sample request:

/nbapi/biogroups.api?q=cell
/nbapi/biogroups.api?type=bioset&id=802981&fr=0&nr=10&taxf=ontologies

Response: The response is a list of CategoryObject objects.

Sample Response:

{...}
"command": "biogroups",
"parameters": {
  "v": "0",
  "fmt": "prettyjsoncontext",
  "g": "cell",
  "apikey": "...
},
"result": [
  {
    "conceptLabel": "SP1 binding site geneset 6",
    "datatypes": [
      "DNA Methylation",
      "RNA expression",
      "protein-DNA binding",
      "DNA Copy Number",
      "Histone methylation",
      "Histone acetylation"
    ],
    "id": "217960",
    "normalizedScore": 100,
    "regulation": 67.2503712152756,
    "studyNum": 179,
    "subcategories": null,
    "topLevelConcept": "no top level concept"
  },
  {
    "conceptLabel": "mitochondrion",
    "datatypes": [
      "RNA expression",
      "protein-DNA binding",
      "DNA Copy Number",
      "Histone methylation",
      "Histone acetylation"
    ],
    "id": "33959",
    "normalizedScore": 100,
    "regulation": -197.4751624458896,
    "studyNum": 172,
    "subcategories": null,
    "topLevelConcept": "no top level concept"
  }
]
}

bioset

The bioset command returns information about a bioset. Included are information such as the description, the tags, and information about the study containing the bioset. Not included is the list of features, which may be obtained from the biosetfeatures command.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biosetid</td>
<td>yes</td>
<td>The ID of the bioset. This ID must be obtained from another command, such as the study or correlationbiosets command</td>
</tr>
</tbody>
</table>

Sample request: /nbapi/bioset.api?biosetid=64636

Response: The response is a single Bioset object. See that object for details of the format.
Sample response:
"result": {  
  "dataType": {  
    "id": 1,  
    "name": "RNA expression"  
  },  
  "description": "Bioset summary = GSE4779 - Clinical Data\nComparison ...",  
  "featureCount": 1959,  
  "geneCount": 1612,  
  "id": 64636,  
  "organism": {  
    "id": 1,  
    "name": "Homo sapiens"  
  },  
  "platform": "Affymetrix GeneChip Human X3P",  
  "studyId": 9253,  
  "studyName": "Gene expression signature of breast cancer ...",  
  "tags": [],  
  "title": "Breast cancer N2 _vs_ N0" 
}

biosetfeatures
The biosetfeatures command returns a list of features in a bioset. The results are paginated. For sequence-centric biosets, the result lists sequences in the bioset.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biosetid</td>
<td>yes</td>
<td>The ID of the bioset. This ID must be obtained from another command, such as the study or correlationbiosets command</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>relatedgenes</td>
<td>no</td>
<td>If set to true, then the response will include information about related genes. For Sequence Centric biosets, this includes genes overlapping or near the sequences in the biosets. For miRNA biosets, this includes genes targeted by the genes in the biosets. Setting this flag to true may result in slower queries with significantly larger results, so it is off by default. See RelatedGene for more information.</td>
</tr>
</tbody>
</table>

Sample requests:

```
/nbapi/biosetfeatures.api?biosetid=64636
```
This returns the first 10 features in the specified bioset.

```
/nbapi/biosetfeatures.api?biosetid=64636&fr=40&nr=20
```
This returns the third page of results in the bioset, using a page size of 20.

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalFeatures</td>
<td>integer</td>
<td>The total number of features in the bioset. Unless the nr parameter is larger than the number of features in the bioset, totalFeatures will be larger than the number of results returned.</td>
</tr>
<tr>
<td>Features</td>
<td>BiosetFeature[]</td>
<td>A list of features and associated statistics. No more than ( nr ) features will be returned.</td>
</tr>
</tbody>
</table>

Sample response:
```
"result": {
  "features": [ 
    { 
      "entrezId": "1571",
      "id": 6461,
      "importedid": "1431_at",
      "name": "CYP2E1",
      "relatedGenes": null,
      "sequence": null,
      "statistics": [ 
        { 
          "name": "fold change",
          "value": "-11.1"
        },
        { 
          "name": "p-value",
          "value": "0.0134"
        },
        { 
          "name": "test expression",
          "value": "472.8"
        },
        { 
          "name": "control expression",
          "value": "5263.5"
        },
        { 
          "name": "rank",
          "value": "1"
        }
      ]
    },
    { 
      "entrezId": "2168",
      "id": 8755,
      "importedid": "38586_at",
      "name": "FABP1",
      "relatedGenes": null,
      "sequence": null,
      "statistics": [ 
        { 
          "name": "fold change",
          "value": "8.67"
        },
        { 
          "name": "p-value",
          "value": "0.0172"
        },
        { 
          "name": "test expression",
          "value": "3350.9"
        },
        { 
          "name": "control expression",
```
bodyatlas

A Body Atlas command returns a list of tissues, cell types and cell lines ranked by relevance for the queried gene, biogroup, or bioset. Results can also be sorted by body system or by ranks (across all body systems).

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes, unless type=bioset</td>
<td>The query term. If the type is a bioset, q should be omitted.</td>
</tr>
<tr>
<td>type</td>
<td>only when type = bioset</td>
<td>If the bodyatlas is against a bioset, type must be set to bioset. Otherwise, type is optional, and if provided should be the type of the query term.</td>
</tr>
<tr>
<td>id</td>
<td>Only when type =bioset</td>
<td>If the type is a bioset, this should be the id of the bioset, which can be obtained from other queries. Otherwise, this is optional, and if provided should be the id of the query term.</td>
</tr>
<tr>
<td>bodyatlastype</td>
<td>yes</td>
<td>Option to view normalized expression across tissues, cell types, cell lines or stem cells. Default is set to TISSUE.</td>
</tr>
<tr>
<td>bodyatlasview</td>
<td>yes</td>
<td>Option to sort your results by absolute gene expression(ranks) or body system. Default is set to SYSTEM.</td>
</tr>
<tr>
<td>source</td>
<td>No</td>
<td>Option to select source type (currently only available for bodyatlastype Tissue). Available source types are 1 (Array Based) and 2 (RNA-Seq based). Default is 1</td>
</tr>
</tbody>
</table>

Sample request:

```
/nbapi/bodyatlas.api?
/nbapi/bodyatlas.api? bodyatlastype=TISSUE&bodyatlasview=SYSTEM
/nbapi/bodyatlas.api? type=bioset&id=234&bodyatlastype=TISSUE&bodyatlasview=SYSTEM&source=1
```

Response:

1) When searched by gene type query, the response is a list of BodyAtlasObjectImpl objects. This object includes the Body System and the list of concepts (ConceptBodyAtlasObject) for that body system.

2) When searched by biosetid or biogroup type query, the response is a list of BiosetBiogroupObject objects. This object includes the Body System and the list of concepts (BiosetBiogroupBodyAtlasObject) for that body system.

Sample Response 1: [search by gene type query]

```
{
  "command": "bodyatlas",
  "parameters": {
    "v": "0",
    "bodyatlasview": "SYSTEM",
    "fmt": "prettyjsontext",
    "bodyatlastype": "TISSUE",
```
Sample Response 2: [search by gene type query]
{
    "command": "bodyatlas",
    "parameters": {
        "id": "516066",
        "v": "0",
        "bodyatlasview": "SYSTEM",
        "fmt": "prettyjsontext",
        "bodyatlastype": "TISSUE",
        "type": "bioset",
        "apikey": "...
    },
    "result": [
        {
            "bodySystem": "Immune System",
            "concepts": [
                {
                    "bodySystems": ["Immune System"],
                    "conceptId": 200167,
                    "conceptLabel": "Leukocyte (polymorphonuclear) of peripheral blood",
                    "direction": 1,
                    "pvalue": 0.021220067230533712,
                }
            ]
        }
    ]
}
The **correlation** command returns a list of studies correlated to a query term or a sequence. It can also be used to correlate studies against a specific bioset, by setting `type=bioset&id=`biosetid and leaving off the query term.

The **correlation** command may be confused with the referencingstudies command. Indeed, the correlation command may return the results from the referencingstudies command as well. The following table describes the differences between the two types of results. If the column marked with "*" has a "Yes" in it, then the referencingstudies command results will also be returned if the referencingstudies parameter is included in the request.

<table>
<thead>
<tr>
<th>Type</th>
<th>Correlation results</th>
<th>Referencing studies results</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene</td>
<td>The study contains biosets for which the gene is up- or down-regulated.</td>
<td>The study is tagged with the gene, or the title or description of the study refers to the gene.</td>
<td>Yes</td>
</tr>
<tr>
<td>Biogroup</td>
<td>The study contains biosets that are correlated with the biogroup.</td>
<td>The study is tagged with the biogroup, or the title or description of the study refers to the biogroup.</td>
<td>Yes</td>
</tr>
<tr>
<td>Tissue, Disease, or Compound</td>
<td>A bioset in the study is tagged with the tissue, disease, or compound.</td>
<td>The study is tagged with the biogroup, or the title or description of the study refers to the biogroup.</td>
<td>Yes</td>
</tr>
<tr>
<td>Bioset</td>
<td>The study contains biosets that are correlated with the specified bioset.</td>
<td>Not available</td>
<td>No</td>
</tr>
<tr>
<td>Sequence region</td>
<td>The study contains biosets that are active in the region.</td>
<td>Not available</td>
<td>No</td>
</tr>
<tr>
<td>Author, Other, or Unknown</td>
<td>No results returned</td>
<td>The title or description of the study refers to the search term</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes, unless type=bioset or chr is specified</td>
<td>The query term. It should be URL encoded. See <a href="#">Query Parameter: q</a> for more information. If the type is a bioset, q should be omitted.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested. See <a href="#">Paging Parameters: fr and nr</a> for information</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results requested. See <a href="#">Paging Parameters: fr and nr</a> for information about pagination.</td>
</tr>
<tr>
<td>context</td>
<td>recommended if available</td>
<td>The context information. It should be provided on all but the first correlation request for a query term or bioset. See <a href="#">Correlation context</a> for information about the context. The correlation command is part of the &quot;correlation&quot; Context group.</td>
</tr>
<tr>
<td>id</td>
<td>only if type=bioset</td>
<td>If the type is a bioset, this should be the id of the bioset, which can be obtained from other queries. Otherwise, this is optional, and if provided should be the id of the query term. May be included along with the type if you have this information about the query term. The id may be obtained from other queries. Use of an id and type can ensure that the correct version of the query term is used if the same term has multiple meanings.</td>
</tr>
<tr>
<td>type</td>
<td>only if type=bioset</td>
<td>If the correlation is against a bioset, this must be set to bioset. Otherwise, this is optional, and if provided should be the type of the query term. May be included along with the id if you have this information about the query term. The type may be obtained from other queries. Use of an id and type can ensure that the correct version of the query term is used if the same term has multiple meanings.</td>
</tr>
<tr>
<td>tf</td>
<td>no</td>
<td>A text filter. Only those studies that contain text matching this value will be returned.</td>
</tr>
<tr>
<td>tag</td>
<td>no</td>
<td>A tag filter. Only studies that are tagged with this term will be returned. Multiple tag parameters may be included, in which case studies matching any of the tags will be returned.</td>
</tr>
<tr>
<td>organism</td>
<td>no</td>
<td>An organism filter. The value must be an id returned from the allorganisms command. Only studies matching the specified organism will be returned. Multiple organism parameters may be included, in which case studies matching any of the organisms will be returned.</td>
</tr>
<tr>
<td>datatype</td>
<td>no</td>
<td>A data type filter. The value must be an id returned from the alldatatypes command. Only studies matching the specified data type will be returned. Multiple data type parameters may be included, in which case studies matching any of the data types will be returned.</td>
</tr>
<tr>
<td>experimentdesign</td>
<td>no</td>
<td>An experiment design filter. The value must be an id returned from the allexperimentdesigns command. Only studies matching the specified experiment design will be returned. Multiple experimentdesign parameters may be included, in which case studies matching any of the experiment designs will be returned.</td>
</tr>
<tr>
<td>samplesource</td>
<td>no</td>
<td>A sample source filter. The value must be an id returned from the allsamplesources command. Only studies matching the specified sample source will be returned. Multiple samplesource parameters may be included, in which case studies matching any of the sample sources will be returned.</td>
</tr>
<tr>
<td>origin</td>
<td>no</td>
<td>One of the values enterprise or public. Only studies matching the specified origin will be returned. This parameter is ignored when using the API on the public site, or if any other value of origin is specified.</td>
</tr>
<tr>
<td>chr</td>
<td>no</td>
<td>For sequence-centric queries, the name of the chromosome. The name should be the same name as the Correlation Engine &quot;Chromosomes&quot; drop down uses when searching sequence regions, e.g. &quot;1&quot; or &quot;X&quot; for Homo Sapiens, or &quot;2L&quot; for D. Melanogaster. This option is only available for accounts that subscribe to the sequence-centric module.</td>
</tr>
<tr>
<td>chorg</td>
<td>only if chr is specified</td>
<td>The organism to which the chromosome belongs. The value must be an id returned from the allorganisms command.</td>
</tr>
<tr>
<td>chrstart</td>
<td>only if chr is specified</td>
<td>The start of the sequence.</td>
</tr>
<tr>
<td>chrstop</td>
<td>no</td>
<td>The end of the sequence. If not specified, chrstart is used. The maximum range of chrstart to chrstop is 10 million.</td>
</tr>
<tr>
<td>referencingstudies</td>
<td>no</td>
<td>If true, the referencingstudies are also returned in the</td>
</tr>
</tbody>
</table>
referencingStudies field in the result. This parameter is ignored unless both of the following are true:

- The fr parameter is not set or is zero.
- The type of the search not a bioset or sequence region.

The tf (text) filter parameter will be passed to the referencing studies queries. Since other filters are not yet available for referencing studies, they will not be passed through.

Sample requests:

/nbapi/correlation.api?q=ESR1
This returns the first 10 studies correlated with ESR1.

/nbapi/correlation.api?type=bioset&id=1234
This returns the first 10 studies correlated with the bioset with id 1234.

/nbapi/correlation.api?chr=X&chrorg=1&chrstart=1&chrstop=10000
This returns the first 10 studies correlated with the X chromosome in Homo Sapiens in the range from 1 to 10000.

/nbapi/correlation.api?q=ESR1&fr=40&nr=20&organism=2&organism=3&tf=transgenic&context=cTllc3IxJnhzPTk5LjvkzODAzODQxNjE4MiZ0eXB1PWd1bmUmaWQ9ODQ5MQ==
This returns correlated results for ESR1 with the following criteria:

- page 3 with a page size of 20 (fr=40&nr=20)
- only results for mouse and rat (organism=2&organism=3)
- containing the term "transgenic" in the description (tf=transgenic)
- using the context from the original query (context=cTllc3IxJnhzPTk5LjvkzODAzODQxNjE4MiZ0eXB1PWd1bmUmaWQ9ODQ5MQ==)

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>context</td>
<td>string</td>
<td>The context to be passed to subsequent correlation commands. See <a href="#">Correlation context</a> for more details</td>
</tr>
<tr>
<td>totalResults</td>
<td>integer</td>
<td>The total number of correlated studies. Unless the nr parameter is larger than the number of studies, totalResults will be larger than the number of results returned.</td>
</tr>
<tr>
<td>studies</td>
<td>CorrelationStudy[]</td>
<td>A list of studies and associated score information.</td>
</tr>
<tr>
<td>referencingStudies</td>
<td>see referencingstudies</td>
<td>If the referencingstudies parameter is set, and all conditions mentioned for that parameter are met, this will include the same results that calling referencingstudies would return.</td>
</tr>
</tbody>
</table>

Sample Response:

```
"result": {
    "context": "q cTllc3IxJnhzPTk5LjvkzODAzODQxNjE4MiZ0eXB1PWd1bmUmaWQ9ODQ5MQ==",
    "referencingStudies": null,
    "studies": [
        {
            "score": {
                "normalizedScore": 22,
                "score": 22.108828497396
            },
            "study": {
                "dataType": {
                    "id": 1,
                    "name": "RNA expression"
                }
            }
        }
    ]
}
```
The correlationstudies command returns a list of studies correlated to filters, text or id.

The correlationstudies command may be confused with the correlation command which requires a query term.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>tf</td>
<td>no</td>
<td>A text filter. Only those studies that contain text matching this value will be returned.</td>
</tr>
<tr>
<td>tag</td>
<td>no</td>
<td>A tag filter. Only studies that are tagged with this term will be returned. Multiple tag parameters may be included, in which case studies matching any of the tags will be returned.</td>
</tr>
<tr>
<td>organism</td>
<td>no</td>
<td>An organism filter. The value must be an id returned from the allorganisms command. Only studies matching the specified organism will be returned. Multiple organism parameters may be included, in which case studies matching any of the organisms will be returned.</td>
</tr>
</tbody>
</table>
### Correlation Engine Query API

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>datatype</strong></td>
<td>no</td>
<td>A data type filter. The value must be an id returned from the <code>alldatatypes</code> command. Only studies matching the specified data type will be returned. Multiple data type parameters may be included, in which case studies matching any of the data types will be returned.</td>
</tr>
<tr>
<td><strong>experimentdesign</strong></td>
<td>no</td>
<td>An experiment design filter. The value must be an id returned from the <code>allexperimentdesigns</code> command. Only studies matching the specified experiment design will be returned. Multiple experimentdesign parameters may be included, in which case studies matching any of the experiment designs will be returned.</td>
</tr>
<tr>
<td><strong>samplesource</strong></td>
<td>no</td>
<td>A sample source filter. The value must be an id returned from the <code>allsamplesources</code> command. Only studies matching the specified sample source will be returned. Multiple samplesource parameters may be included, in which case studies matching any of the sample sources will be returned.</td>
</tr>
<tr>
<td><strong>origin</strong></td>
<td>no</td>
<td>One of the values <strong>enterprise</strong> or <strong>public</strong>. Only studies matching the specified origin will be returned. This parameter is ignored when using the API on the public site, or if any other value of <strong>origin</strong> is specified.</td>
</tr>
</tbody>
</table>

Sample requests:

```
/nbapi/correlationstudies.api?tf=esr1
```

This returns the first 10 studies correlated with the search term ESR1.

```
/nbapi/correlation.api?&datatype=1&datatype=24
```

This returns the first 10 studies with datatypes 1 or 24.

### Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>context</td>
<td>string</td>
<td>Not used</td>
</tr>
<tr>
<td><strong>totalResults</strong></td>
<td>integer</td>
<td>The total number of correlated studies. Unless the <code>nr</code> parameter is larger than the number of studies, <code>totalResults</code> will be larger than the number of results returned.</td>
</tr>
<tr>
<td><strong>studies</strong></td>
<td>CorrelationStudy[]</td>
<td>A list of studies and associated score information.</td>
</tr>
<tr>
<td>referencingStud</td>
<td>string</td>
<td>Not used</td>
</tr>
</tbody>
</table>

Sample Response:

```
"result": {
  "context": null,
  "referencingStudies": null,
  "studies": [
    {
      "score": {
        "normalizedScore": 22,
        "score": 22.108828497396
      },
      "study": {
        "dataType": {
          "id": 1,
          "name": "RNA expression"
        },
        "id": 6181,
        "institution": "Thomas Jefferson University",
        "organism": {
          "id": 2,
          "name": "Mus musculus"
        }
      }
    }
  ]
}
```
correlationbiogroups
The correlationbiogroups command correlates a single bioset against biogroups.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>yes</td>
<td>This must be the string &quot;bioset&quot;. It is required to allow for future extension of this command.</td>
</tr>
<tr>
<td>id</td>
<td>yes</td>
<td>The id of the bioset.</td>
</tr>
<tr>
<td>context</td>
<td>recommended if available</td>
<td>The context information. It should be provided on all but the first correlation request for a query term or bioset. See Correlation context for information about the context. The correlationbiogroups command is part of the &quot;correlationbiogroups&quot; Context group.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>tf</td>
<td>no</td>
<td>A text filter. Only those biogroups that contain text matching this value will be returned.</td>
</tr>
<tr>
<td>taxf</td>
<td>no</td>
<td>Limit the results to biogroups in the specific taxonomy. Multiple taxf parameters can be provided, in which case biogroups belonging to any of the taxonomies will be included. The following are the values supported for taxf:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>taxf</th>
<th>Biogroup Type</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>ontologies</td>
<td>Gene Ontology</td>
<td>GO</td>
</tr>
</tbody>
</table>
Sample Requests:

/\nbapi/correlationbiogroups.api?type=bioset&id=64636

This returns the first 10 biogroups correlated with bioset 64636.

/\nbapi/correlationbiogroups.api?type=bioset&id=64636&fr=20&nr=20&taxf=pathways&taxf=positional&tf=metabolism&context=dHlwZT1iaW9zZXQmaWQ9NjQ2MzYmeHM9MzAuNDc5NTcyMTIzMTY4MjM=

This returns correlated results for bioset 64636 with the following criteria:

- page 2 with a page size of 20 (fr=20&nr=20)
- only results for biogroup types "Canonical pathways" and "Positional gene sets" (taxf=pathways&taxf=positional)
- containing the term "transgenic" in the metabolism (tf=metabolism)
- using the context from the original query (context=dHlwZT1iaW9zZXQmaWQ9NjQ2MzYmeHM9MzAuNDc5NTcyMTIzMTY4MjM=)

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>context</td>
<td>string</td>
<td>The context to be passed to subsequent correlation commands. See Correlation context for more details.</td>
</tr>
<tr>
<td>totalResults</td>
<td>integer</td>
<td>The total number of correlated biogroups. Unless the nr parameter is larger than the number of biogroups, totalResults will be larger than the number of results returned.</td>
</tr>
<tr>
<td>biogroups</td>
<td>CorrelationBiogroup[]</td>
<td>A list of biogroups and associated score information.</td>
</tr>
</tbody>
</table>

Sample Response:

"result": {
   "biogroups": [
   {
      "biogroup": {
         "description": "Catalysis of an oxidation-reduction (redox) ...",
         "id": 39152,
         "taxonomy": "GO",
         "title": "oxidoreductase activity, acting on the CH-CH group ...",
         "commonGenes": 3,
         "direction": -1,
         "pvalue": 5.79281976307422E-14,
         "score": {
            "normalizedScore": 100,
            "score": 30.47957212316823
         }
      },
      "biogroup": {
         "description": "Genes with promoter regions [-2kb,2kb] ...",
         "id": 12345,
         "taxonomy": "UCSC",
         "title": "Transgenic promoters ...",
         "commonGenes": 5,
         "direction": 1,
         "pvalue": 1.23456789E-05,
         "score": {
            "normalizedScore": 80,
            "score": 20.123456789
         }
      }
   ]
}
correlationbiosets
The correlationbiosets command is used after the correlation command. It provides details about the biosets in a specific study returned by the correlation command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes, unless the original correlation had type=bioset or chr</td>
<td>The query term. It must be the same as the q parameter in the correlation command.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>context</td>
<td>yes</td>
<td>The context information returned by the correlation command. See Correlation context for information about the context. The correlation biosets command is part of the &quot;correlation&quot; Context group.</td>
</tr>
<tr>
<td>studyid</td>
<td>yes</td>
<td>The ID of the study being queried. This should have been returned by one of the entries in the correlation command.</td>
</tr>
<tr>
<td>type</td>
<td>only if the original query had type=bioset</td>
<td>If the original query had type=bioset, this type must also be bioset.</td>
</tr>
<tr>
<td>id</td>
<td>only if the original query had type=bioset</td>
<td>If the original query had type=bioset, the id of the bioset must also be set here.</td>
</tr>
<tr>
<td>chr, chrorg, chrstop, chrstart</td>
<td>only if the original query specified a sequence-centric query</td>
<td>If the original query specified a sequence-centric query, they should also be specified here.</td>
</tr>
</tbody>
</table>

Sample Requests:
/nbapi/correlationbiosets.api?q=ESR1&studyid=9253&context=cT1lc3IxJnhzPTk5LjkzODAzODQxNjE4MiZ0eXBlPWdlbmUmaWQ9ODQ5MQ==
Returns the first 10 biosets for the correlation of ESR1 with study 9253.
/nbapi/correlationbiosets.api?q=ESR1&studyid=9253&fr=40&nr=20&context=cT1lc3IxJnhzPTk5LjkzODAzODQxNjE4MiZ0eXBlPWd1bmUmaWQ9ODQ5MQ==

Returns page 3 of the results with a page size of 20 from the same study.

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>context</td>
<td>string</td>
<td>The context. See <a href="#">Correlation context</a> for more details</td>
</tr>
<tr>
<td>totalResults</td>
<td>integer</td>
<td>The total number of biosets. Unless the nr parameter is larger than the number of biosets, totalResults will be larger than the number of results returned.</td>
</tr>
<tr>
<td>biosets</td>
<td>CorrelationBioset[]</td>
<td>A list of biosets and associated score information. No more than nr biosets will be returned.</td>
</tr>
</tbody>
</table>

Sample Response:

```
"result": {
  "biosets": [
  {
    "bioset": {
      "id": 64636,
      "title": "Breast cancer N2 _vs_ N0"
    },
    "commonGenes": null,
    "correlation": null,
    "direction": -1,
    "pvalue": 8.04E-4,
    "score": {
      "normalizedScore": 100,
      "score": 99.9260547195076
    },
    "statistics": [ {
      "name": "fold change",
      "value": "-5.29"
    }]
  },
  {
    "bioset": {
      "id": 64651,
      "title": "Breast cancer N2 _vs_ N1"
    },
    "commonGenes": null,
    "correlation": null,
    "direction": -1,
    "pvalue": 4.05E-4,
    "score": {
      "normalizedScore": 100,
      "score": 99.8542128492401
    },
    "statistics": [ {
      "name": "fold change",
      "value": "-5.68"
    }]
  }
  ],
  "context": "q=esr1&xs=99.938038416182&type=gene&id=8491",
  "totalResults": 9
}
```
createproject
The createproject command allows the user with special permission to create a project to the user’s account as well as for other user’s account.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>login</td>
<td>No</td>
<td>An optional parameter that creates another user’s project if the APIKEY provided has the special credentials.</td>
</tr>
<tr>
<td>name</td>
<td>Yes</td>
<td>Name of the new project.</td>
</tr>
<tr>
<td>description</td>
<td>No</td>
<td>Description of the project.</td>
</tr>
</tbody>
</table>

Sample request:
```
/nbapi/createproject.api?
/nbapi/createproject.api? name=proj name sample&description=sample desc
```

Response: The response is a message showing the id of the project created, if successful. Otherwise it would be an error message.

Sample Response 1:
```
{
   "command": "createproject",
   "parameters": {
      "v": "0",
      "fmt": "prettyjsontext",
      "description": "sample desc",
      "name": "sample proj name",
      "apikey": "..."
   },
   "result": "project with id 50097 created successfully"
}
```

deletebioset
The deletebioset command allows the user to delete a bioset if the user has permission. The bioset is queued for deletion.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biosetid</td>
<td>Yes</td>
<td>Id of the bioset to be deleted.</td>
</tr>
</tbody>
</table>

Sample request:
```
/nbapi/deletebioset.api?
/nbapi/deletebioset.api? biosetid=5001
```

Response: The response is always a message showing the status of the delete.

Sample Response 1:
```
{
   "command": "deletebioset",
   "result": "bioset with id 5001 deleted"
}
```
"error": "please enter valid bioset id",
"parameters": 
  { "v": "0",
    "fmt": "prettyjsontext",
    "biosetid": "2323",
    "apikey": "...
  }
}

Sample Response 2:
{
  "command": "deletebioset",
  "parameters": 
    { "v": "0",
      "fmt": "prettyjsontext",
      "biosetid": "50018",
      "apikey": "...
    },
  "result": "bioset queued for deletion"
}

deletestudy
The deletestudy command allows the user to delete a study if the user has permission. The study is queued for deletion.

Parameters:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>studyid</td>
<td>yes</td>
<td>Id of the study to be deleted.</td>
</tr>
</tbody>
</table>

Sample request:
/nbapi/deletestudy.api?
/nbapi/deletestudy.api?studyid=104

Response: The response is always a message showing the status of the delete.

Sample Response 1:
{
  "command": "deletestudy",
  "error": "studyid not specified or studyid not valid",
  "parameters": 
    { "v": "0",
      "fmt": "prettyjsontext",
      "apikey": "...
    }
}

Sample Response 2:
{
  "command": "deletestudy",
  "parameters": 
    { "v": "0",
      "fmt": "prettyjsontext",
      "studyid": "254",
      "apikey": "...
    },
}
"result": "study has been queued for removal"
}

diseaseatlas
The diseaseatlas command allows the user to find diseases, traits, conditions, and surrogate endpoints associated with a gene, SNP, biogroup or bioset. Results are grouped by disease and ranked according to statistical significance.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term.</td>
</tr>
<tr>
<td>type</td>
<td>only when type = bioset</td>
<td>If the diseaseatlas is against a bioset, type must be set to bioset. Otherwise, type is optional, and if provided should be the type of the query term.</td>
</tr>
<tr>
<td>id</td>
<td>only when type = bioset</td>
<td>If the type is a bioset, this should be the id of the bioset, which can be obtained from other queries. Otherwise, this is optional, and if provided should be the id of the query term.</td>
</tr>
<tr>
<td>sectiontype</td>
<td>yes</td>
<td>Option to select response content to contain Meta-Categories or correlated studies or both. Default is set to ‘meta categories’. Note: For correlated studies, it returns statistically significant results correlated to individual studies filtered by “Disease vs. normal” biodesign filter.</td>
</tr>
<tr>
<td>diseaseatlasview</td>
<td>yes</td>
<td>Option to sort results by ranks or categories. Default is set to ‘categories’.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested for ‘meta categories’ in ‘ranks’ view. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>Number of results requested for ‘meta categories’ in ‘ranks’ view. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
</tbody>
</table>

Sample request:

/nbapi/diseaseatlas.api? q=esr1&diseaseatlasview=CATEGORY&sectiontype=METACATEGORY
/nbapi/diseaseatlas.api? q=esr1&diseaseatlasview=RANK&sectiontype=METACATEGORY
/nbapi/diseaseatlas.api? q=esr1&diseaseatlasview=RANK&fr=3&nr=5&sectiontype=METACATEGORY
/nbapi/diseaseatlas.api? type=bioset&id=802981&diseaseatlasview=RANK&fr=3&nr=5&sectiontype=METACATEGORY

Response: The response is a list of CategoryObject objects and/or CategoryStudy objects and count of result items.

<table>
<thead>
<tr>
<th>parameter in response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numOfCategories</td>
<td>If sectiontype parameter has ‘METACATEGORY’ or ‘ALL’ value selected, then it returns number of meta-category results into response, else returns 0.</td>
</tr>
<tr>
<td>numOfCorrelations</td>
<td>If sectiontype parameter has ‘STUDY’ or ‘ALL’ value selected, then it returns numbers of correlation results into response, else returns 0.</td>
</tr>
</tbody>
</table>

Sample Response:

```json
{
  "command": "diseaseatlas",
  "parameters": {
    "v": "0",
    "fmt": "prettyjsontext",
    "diseaseatlasview": "CATEGORY",
    "sectiontype": "METACATEGORY",
    ...
  }
}```
"q": "esr1",
    "apikey": "...
},
"result":
{
    "conceptLabel": "Cancer",
    "datatypes": [
        "DNA Methylation",
        "RNA expression",
        "somatic mutation",
        "SNP GWAS",
        "DNA Copy Number",
        "mutations/phenotypes"
    ],
    "id": "4",
    "normalizedScore": 99,
    "regulation": -0.012943795541760254,
    "studyNum": 555,
    "subcategories": [
        {
            "conceptLabel": "Breast cancer",
            "datatypes": [
                "RNA expression",
                "SNP GWAS",
                "DNA Copy Number",
                "mutations/phenotypes"
            ],
            "id": "142027",
            "normalizedScore": 100,
            "regulation": -0.0036586763515754264,
            "studyNum": 121,
            "subcategories": null,
            "topLevelConcept": "Cancer"
        },
        {
            "conceptLabel": "Metastases",
            "datatypes": [
                "RNA expression",
                "DNA Copy Number"
            ],
            "id": "130425",
            "normalizedScore": 56,
            "regulation": -4.7434234446515247E-4,
            "studyNum": 42,
            "subcategories": null,
            "topLevelConcept": "Cancer"
        }
    ],
    "numOfCategories": 195,
    "numOfCorrelations": 0,
    "studyList": null
}
editbioset
The editbioset command allows the user to edit bioset data such as name and description. Which bioset is editable depends upon the permission level of user.

Parameters:
Note: Both name and description cannot be empty.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biosetid</td>
<td>yes</td>
<td>ID of the bioset to be edited.</td>
</tr>
<tr>
<td>name</td>
<td>no</td>
<td>New name of the bioset. Leaving this field empty will not update the field.</td>
</tr>
<tr>
<td>description</td>
<td>no</td>
<td>New description of the bioset. Leaving this field empty will not update the field.</td>
</tr>
</tbody>
</table>

Sample request:

/nbapi/editbioset.api?
/nbapi/editbioset.api? biosetid=50018

Response: The response is always a message showing the status of the update.

Sample Response 1:

```
{
  "command": "editbioset",
  "error": "bioset not updated -> please enter atleast one field",
  "parameters":
  {
    "v": "0",
    "fmt": "prettyjsonprettytext",
    "studyid": "50118",
    "apikey": "...
  }
}
```

Sample Response 2:

```
{
  "command": "editbioset",
  "parameters":
  {
    "v": "0",
    "fmt": "prettyjsonprettytext",
    "name": "bioset-name-api-update",
    "biosetid": "50018",
    "apikey": "...
  },
  "result": "bioset was edited successfully"
}
```

deitstudy
The editstudy command allows the user to edit study data such as name, description and summary. Which study is editable depends upon the permission level of user.

Parameters:
Note: All fields – name, description and summary cannot be empty.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biosetid</td>
<td>Yes</td>
<td>ID of the study to be edited.</td>
</tr>
<tr>
<td>name</td>
<td>No</td>
<td>New name of the study. Leaving this field empty will not update the field.</td>
</tr>
<tr>
<td>description</td>
<td>No</td>
<td>New description of the study. Leaving this field empty will not update the</td>
</tr>
</tbody>
</table>
Sample request:

/nbapi/editstudy.api?
/nbapi/editstudy.api? studyid=5018&name=new name&description=description-update&summary=update summary

Response: The response is always a message showing the status of the update.

Sample Response 1:

```
{
    "command": "editstudy",
    "error": "study not updated -> please enter atleast one field",
    "parameters": {
        "v": "0",
        "fmt": "prettyjsontext",
        "studyid": "50118",
        "apikey": "...
    }
}
```

Sample Response 2:

```
{
    "command": "editstudy",
    "parameters": {
        "v": "0",
        "fmt": "prettyjsontext",
        "name": "new study name API",
        "studyid": "508",
        "apikey": "...
    },
    "result": "study updated successfully"
}
```

geneticmarkers

The geneticmarkers command finds genes significantly correlated to a phenotype or compound. Results are ranked in order of statistical significance.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>Yes</td>
<td>The query term.</td>
</tr>
</tbody>
</table>

Sample request:

/nbapi/geneticmarkers.api?
/nbapi/geneticmarkers.api? q=esr1

Response: The response is a list of CategoryGeneObject objects.

Sample Response:

```
{
    "command": "geneticmarkers",
    "parameters": {
        "v": "0",
        "fmt": "prettyjsontext",
```
"q": "cell",
"apikey": "...
",
"result": [
{
  "datatypes": ["RNA expression"],
  "entrezId": "5728",
  "gene": "S100A8",
  "id": "0",
  "normalizedScore": 100,
  "regulation": -0.002044989775051125,
  "studyNum": 34
},
{
  "datatypes": ["RNA expression"],
  "entrezId": "2328",
  "gene": "COL3A1",
  "id": "0",
  "normalizedScore": 96,
  "regulation": -4.9965024482862E-5,
  "studyNum": 25
},
{
  "datatypes": ["RNA expression"],
  "entrezId": "5728",
  "gene": "VCAN",
  "id": "0",
  "normalizedScore": 94,
  "regulation": 0,
  "studyNum": 24
}
]

genomebuild
The genomebuild command displays genome build information.
Parameters: none

Sample request:

/nbapi/genomebuild.api?
Response: The response is a list of genomeBuild objects.

Sample Response:

"result": [
{
  "current": false,
  "genomeBuildId": 25,
  "name": "NCBI build 37 | UCSC hg19 ",
  "ncbiBuild": "build 37",
  "speciesName": "Homo sapiens",
  "ucscAssembly": "hg19"
},
{
knockdownatlas

The knockdownatlas command finds genes whose perturbation affects your query term.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>Yes</td>
<td>The query term.</td>
</tr>
<tr>
<td>type</td>
<td>only when type = bioset</td>
<td>If the knockdownatlas is against a bioset, type must be set to bioset. Otherwise, type is optional, and if provided should be the type of the query term.</td>
</tr>
<tr>
<td>id</td>
<td>only when type = bioset</td>
<td>If the type is a bioset, this should be the id of the bioset, which can be obtained from other queries. Otherwise, this is optional, and if provided should be the id of the query term.</td>
</tr>
<tr>
<td>geneperturbationclass</td>
<td>Yes</td>
<td>Class to be categorized by. Default is set to ‘Gene knockout’.</td>
</tr>
<tr>
<td>includeBiosetID</td>
<td>No</td>
<td>This section applies only for gene and biogroup queries. If set to on, the response includes biosets with significance in terms of perturbation for the gene of interest.</td>
</tr>
<tr>
<td>experimentdesign</td>
<td>No</td>
<td>This selection applies only when includeBiosetID flag is ON. BiosetIds are filtered based on selection. Default is set to ‘Mutant vs. wildtype studies’.</td>
</tr>
</tbody>
</table>

Sample request:

```
/nbapi/knockdownatlas.api?
/nbapi/knockdownatlas.api? q=esr1
/nbapi/knockdownatlas.api? q=esr1&geneperturbationclass=195793&experimentdesign=204652
/nbapi/knockdownatlas.api? q=esr1&q=esr1&geneperturbationclass=195793&includeBiosetID=true&experimentdesign=204652
/nbapi/knockdownatlas.api? type=bioset&id=802981&geneperturbationclass=195793&experimentdesign=204652
```

Response: The response is a list of `CategoryGeneObject` objects.

Sample Response:

```
{
    "command": "knockdownatlas",
    "parameters": {
        "v": "0",
        "fmt": "prettyjsontext",
        "q": "esr1",
    }
}
```
"includeBiosetID": "true",
"experimentdesign": "204652",
"geneperturbationclass": "195793",
"apikey": "..."
],
"result": [
{
"biosetIds": [16639],
"datatypes": ["RNA expression"],
"entrezId": "192119",
"gene": "Dicer1",
"id": "371117",
"normalizedScore": 100,
"regulation": 0.01020408163265306,
"studyNum": 1,
"topLevelConcept": "Gene knockout"
},
{
"biosetIds": [241738,
241747
],
"datatypes": ["RNA expression"],
"entrezId": "9856",
"gene": "NR1I2",
"id": "33249",
"normalizedScore": 94,
"regulation": 0.001570813881407063,
"studyNum": 1,
"topLevelConcept": "Gene knockout"
},
{
"biosetIds": [424189],
"datatypes": ["RNA expression"],
"entrezId": "10681",
"gene": "GNB5",
"id": "40605",
"normalizedScore": 92,
"regulation": 4.3122035360068997E-4,
"studyNum": 1,
"topLevelConcept": "Gene knockout"
}
]
}

libraries
The libraries command returns user libraries available for that domain. Included are information such as the library id and title.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>includeDomain</td>
<td>no</td>
<td>If set to on, the response includes all libraries in current domain.</td>
</tr>
</tbody>
</table>

Sample request: /nbapi/libraries.api?
Response: The response is a ApiLibraryList which contains a list of ApiLibrary objects.

Sample response:
"result": {"apiLibraryList": [ 
  { "id": 100, "title": "Public User Library" }, 
  { "id": 10, "title": "Company User Library" }
]}

literature
The literature command returns a list of PUBMED articles related to a query term.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term. It should be URL encoded. See Query Parameter: q for more information.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>tf</td>
<td>no</td>
<td>A text filter. The string may be enclosed in double quotes (&quot;) for an exact match. This filter may be used to filter to a specific tag in a tag cloud.</td>
</tr>
<tr>
<td>tagCloud</td>
<td>no</td>
<td>If true, a tag cloud is returned in addition to the list of publications. To get only a tag cloud, set nr=0 to skip the publications.</td>
</tr>
<tr>
<td>tagType</td>
<td>no</td>
<td>Only used if tagCloud=true. Specifies the type of tag cloud (e.g. AUTHOR, DISEASE). This parameter should normally be omitted on the initial tag cloud query, in which case the default tag cloud is returned. The allowable values are returned in the tagCloudTypes field in the result.</td>
</tr>
<tr>
<td>tagLookupSize</td>
<td>no</td>
<td>Only used if tagCloud=true. Specifies the number of articles to be considered in computing the tag cloud. The value should be between 50 and 1000. Other values will be adjusted. The default is 50. Larger values will take longer.</td>
</tr>
<tr>
<td>tagCloudSize</td>
<td>no</td>
<td>Only used if tagCloud=true. Specifies the maximum number of terms to return in the tag cloud. The maximum value is 70; larger values will be adjusted. If omitted, the default is to return the same number of terms as the UI would return, which depends on tagLookupSize.</td>
</tr>
<tr>
<td>sort</td>
<td>no</td>
<td>The sort criteria. Valid values are date and relevance. Default is relevance.</td>
</tr>
<tr>
<td>bold</td>
<td>no</td>
<td>If set to true, the title and abstractItems fields in the results will have the search term and appropriate synonyms surrounded by &lt;b&gt; and &lt;/b&gt; tags. (In JSON, the &lt;b&gt; may appear as &lt;\b&gt;.)</td>
</tr>
<tr>
<td>abstractsection</td>
<td>no</td>
<td>If set to ON, search results are filtered by abstract name.</td>
</tr>
<tr>
<td>titlesection</td>
<td>no</td>
<td>If set to ON, search results are filtered by title.</td>
</tr>
<tr>
<td>meshtagssection</td>
<td>no</td>
<td>If set to ON, search results are filtered by mesh tags.</td>
</tr>
<tr>
<td>facetview</td>
<td>yes</td>
<td>This option allows the user to select the facet to be returned. By default, facetview is set to 'All' which returns all facets. Each facet has capacity to hold 20 elements at maximum.</td>
</tr>
</tbody>
</table>

Sample requests:
/nbapi/literature.api?q=cancer
This returns the first 10 PUBMED articles containing cancer.

/nbapi/literature.api?q=cancer&tagCloud=true&tagType=AUTHOR&nr=0

This returns the author tag cloud for articles containing cancer.

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalResults</td>
<td>integer</td>
<td>The total number of literature items. Unless the nr parameter is larger than</td>
</tr>
<tr>
<td>items</td>
<td>LiteratureItem[]</td>
<td>A list of literature items.</td>
</tr>
<tr>
<td>tagCloudItems</td>
<td>TagCloudItem[]</td>
<td>A list of tag cloud items. This is only returned if tagCloud=true.</td>
</tr>
<tr>
<td>tagCloudTypes</td>
<td>string[]</td>
<td>A list of valid types for the tagType attribute. This is only returned if</td>
</tr>
<tr>
<td>facetGroup</td>
<td>ApiPMFacetGroup[]</td>
<td>Each facet contains many facet elements. All facets are returned for default</td>
</tr>
</tbody>
</table>

Sample Response:

```json
"result": {
  "items": [
    {
      "affiliations": [
        "University of British Columbia",
        "University of Calgary"
      ],
      "articleAbstract": "BACKGROUND: The purpose of this study was ...",
      "authors": [
        "Marc W Deyell",
        "William A Ghali",
        ...
      ],
      "id": 20211314,
      "journal": "American heart journal",
      "publishDate": "2010 Mar",
      "title": "Timing of nonemergent coronary artery bypass ...",
      "type": "PUBMED"
    },
    {
      "affiliations": [
        "Centre Hospitalier Affilié Universitaire de Québec"
      ],
      "articleAbstract": "BACKGROUND: Chronic obstructive pulmonary ...",
      "authors": [
        "Jonathan De Blois",
        "Serge Simard",
        ...
      ],
      "id": 20206897,
      "journal": "Journal of cardiac failure",
      "publishDate": "2010 Mar",
      "title": "COPD Predicts Mortality in HF: The Norwegian Heart ...",
      "type": "PUBMED"
    },
    ...
  ],
  "tagCloudItems": [
    {
```
ontologyinfo
The ontologyinfo command allows the user to find ontology information. Information includes description, id, synonyms and type of the ontology term.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term.</td>
</tr>
</tbody>
</table>

Sample request:
/nbapi/ ontologyinfo.api?
/nbapi/ ontologyinfo.api? q=shock

Response:
The response is an OntologyInfoObject.

Sample Response:

```
"result":  {
   "description": "a profound hemodynamic and metabolic disturbance characterized by the failure of the circulatory system to maintain adequate perfusion of vital organs; causes inadequate delivery of nutrients and insufficient removal of cellular waste products to and from the tissues.",
   "id": "147714",
   "synonyms": null,
   "type": "disease"
}
```
OntologyHierarchy

The OntologyHierarchy command allows a user to find structured ontology information. The user provides a query term as an input and if it matches with Illumina Ontology term then OntologyHierarchy results will be returned. The information returned includes name, id, synonyms, description, type of the ontology term, whether the term was user added, and the query’s immediate children. Information returned in association with children includes name, id, synonyms, description, type of the ontology term, and whether the term was user added. API will not return result for gene search as it doesn’t contain hierarchical structure.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term.</td>
</tr>
</tbody>
</table>

Sample request:

```
/nbapi/ontologyhierarchy.api?q=lung cancer
```

Response:

The Response is an OntologyHierarchyObject.

Sample Response:

```
{   
  "result": {   
    "id": 1335491, 
    "name": "Lung cancer", 
    "numberOfChildren": 2, 
    "description": "Malignant tumor of lung", 
    "synonymList": [   
      "Malignant tumour of lung", 
      "CA - Lung cancer", 
      "Malignant tumor of lung (disorder)"
    ],
    "type": "DISEASE",
    "useradded": false,
    "childrenList": [{
      "description": null,
      "id": 1308992,
      "name": "Carcinoma of lung",
      "synonymList": [   
        "Malignant epithelial neoplasm of lung",
        "Malignant epithelial neoplasm of lung (disorder)"
      ],
      "type": "DISEASE",
      "useradded": false
    }, {
      "description": "A malignant tumor of the lung that is a mixture of carcinoma and sarcoma.",
      "id": 1530172,
      "name": "Carcinosarcoma of lung",
      "synonymList": [],
      "type": "DISEASE",
      "useradded": false
    }]
  }
}
```
pharmacoatlas
The pharmacoatlas command allows the user to find compounds and treatments significantly correlated to a gene, biogroup or bioset. Results are ranked in order of statistical significance.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term.</td>
</tr>
<tr>
<td>type</td>
<td>only when type = bioset</td>
<td>If the pharmacoatlas is against a bioset, type must be set to bioset. Otherwise, type is optional, and if provided should be the type of the query term.</td>
</tr>
<tr>
<td>id</td>
<td>only when type = bioset</td>
<td>If the type is a bioset, this should be the id of the bioset, which can be obtained from other queries. Otherwise, this is optional, and if provided should be the id of the query term.</td>
</tr>
<tr>
<td>categoryclass</td>
<td>yes</td>
<td>Class to be categorized by. Default is set to ‘Mechanisms of Action’.</td>
</tr>
<tr>
<td>pharmacoatlasview</td>
<td>yes</td>
<td>Option to sort your results by ranks or categories. Default is set to ‘categories’.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested for ‘ranks’ view. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>Number of results requested for ‘ranks’ view. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
</tbody>
</table>

Sample request:

```
/nbapi/pharmacoatlas.api?q=esr1&categoryclass=26&pharmacoatlasview=CATEGORY
/nbapi/pharmacoatlas.api?q=esr1&categoryclass=26&pharmacoatlasview=RANK&fr=3&nr=45
/nbapi/pharmacoatlas.api?type=bioset&id=802981&categoryclass=26&pharmacoatlasview=RANK&fr=3&nr=5
```

Response:

The response is a list of `CategoryObject` objects.

Sample Response:

```
{
   "command": "pharmacoatlas",
   "parameters": {
      "v": "0",
      "categoryclass": "26",
      "fmt": "prettyjsontext",
      "q": "esr1",
      "pharmacoatlasview": "CATEGORY",
      "apikey": "...
   },
   "result": [
      {
         "conceptLabel": "Unclassified Mechanisms of Action",
         "datatypes": [
            "RNA expression",
            "SNP GWAS",
            "DNA Copy Number",
            "therapeutic"
         ],
         "id": "705380",
         "normalizedScore": 98,
```
Correlation Engine Query API

"regulation": -0.018596861796737835,
"studyNum": 189,
"subcategories": [
  {
    "conceptLabel": "Estrogens, Conjugated (USP),",
    "datatypes": ["therapeutic"],
    "id": "259692",
    "normalizedScore": 100,
    "regulation": 0,
    "studyNum": 1,
    "subcategories": null,
    "topLevelConcept": "Unclassified Mechanisms of Action"
  },
  {
    "conceptLabel": "Toremifene",
    "datatypes": ["therapeutic"],
    "id": "255646",
    "normalizedScore": 100,
    "regulation": 0,
    "studyNum": 1,
    "subcategories": null,
    "topLevelConcept": "Unclassified Mechanisms of Action"
  }
]

referencingbiosets

The referencingbiosets command is used after the referencingstudies command. It provides the list of biosets in a specific study that refer to the specified search term. Only biosets that refer to the search term are returned, to see all of the biosets in the study, use the study command instead.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term. It should be URL encoded. See Query Parameter: q for more information.</td>
</tr>
<tr>
<td>studyid</td>
<td>id</td>
<td>The ID of the study being queried. This should have been returned by one of the entries in the referencingstudies command.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
</tbody>
</table>

Sample Request:

```
/nbapi/referencingbiosets.api?q=ESR1&studyid=9253
```

Returns the first 10 biosets that refer to ESR1 in study 9253.

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalResults</td>
<td>integer</td>
<td>The total number of referencing biosets. Unless the nr parameter is larger than the number of biosets, totalResults will be larger than the number of results returned.</td>
</tr>
<tr>
<td>Biosets</td>
<td>ShortBioset[]</td>
<td>A list of biosets.</td>
</tr>
</tbody>
</table>

Sample Response:

```
"result": {
  "biosets": [
```
Correlation Engine Query API

```
{
  "id": 65119,
  "title": "Breast cancer prior to chemotherapy from no response patient..."
},
{
  "id": 65107,
  "title": "Breast cancer prior to chemotherapy from partial remission..."
},
"totalResults": 2
}
```

**referencingstudies**

Returns a list of studies whose tags or text reference the search term. See the correlation command for a comparison of correlations to referencing studies. The correlation command can also return the referencing studies results if the referencingstudies parameter is set to true.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>q</strong></td>
<td>yes</td>
<td>The query term. It should be URL encoded. See Query Parameter: q for more information.</td>
</tr>
<tr>
<td><strong>fr</strong></td>
<td>no</td>
<td>The first result requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td><strong>nr</strong></td>
<td>no</td>
<td>The number of results requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td><strong>tf</strong></td>
<td>no</td>
<td>A text filter. Only those studies that contain text matching this value will be returned.</td>
</tr>
</tbody>
</table>

Sample Requests:

```
/nbapi/referencingstudies.api?q=ESR1
This returns all studies that reference the gene ESR1.
```

```
/nbapi/correlation.api?q=ESR1&referencingstudies=true
This includes the referencing studies in the correlation command results.
```

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>totalResults</strong></td>
<td>integer</td>
<td>The total number of referencing studies. Unless the nr parameter is larger than the number of studies, totalResults will be larger than the number of results returned.</td>
</tr>
<tr>
<td><strong>studies</strong></td>
<td>ShortStudy</td>
<td>A list of studies.</td>
</tr>
</tbody>
</table>

Sample Response:

```
"result": {
  "studies": [
    {
      "authors": "Yu K, Tan P",
      "dataType": {
        "id": 1,
        "name": "RNA Expression"
      },
      "id": 8411,
      "institution": "National Cancer Centre...Singapore"
    }
  ]
}
```
The snpgwasdata command provides GWAS data for a gene in Q and it includes: list of biosets from correlated studies and related diseases (studies tagged with).

### Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term. It should be URL encoded. See <a href="#">Query Parameter: q</a> for more information.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result of study requested. See <a href="#">Paging Parameters: fr and nr</a> for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of study results requested. See <a href="#">Paging Parameters: fr and nr</a> for information about pagination.</td>
</tr>
<tr>
<td>nr_bioset</td>
<td>no</td>
<td>The number of biosets for each study requested. Default is set to 5.</td>
</tr>
</tbody>
</table>

### Sample Requests:

```
/nbapi/snpgwasdata.api? q=cacnalc
/nbapi/snpgwasdata.api? q=cacnalc&fr=2&nr=3&fr_bioset=1&nr_bioset=3
```

### Response:

The response is list of `GeneGWASBiosetObject` object. See that object for details of the format.

### Sample Response:

```
"result": [ 
  { 
    "biosetTitle": "Bipolar Disorder associated SNPs- in combined study", 
    "biosetid": 143947, 
    "cohortSize": "Sample Number = 10596", 
    "diseases": ["Bipolar disorder"], 
  } 
] 
```
"ethnic": "Reference population: Utah residents with Northern and Western European ancestry from the CEPH collection. (CEU)",
"paperTitle": "paper title not found",
"parentStudyid": 24229,
"snpGWASBiosetFeatureList": [
  {
    "biosetid": 143947,
    "oddRatio": "1.181",
    "pvalue": "7.0E-8",
    "rank": "1",
    "rsName": "rs1006737"
  }
],

"biosetTitle": "Response to antidepressants associated SNPs in GENDEP discovery cohort",
"biosetid": 183874,
"cohortSize": "Sample Number = 706 European descent individuals",
"diseases": ["Response to antidepressants"],
"ethnic": "Reference population: Utah residents with Northern and Western European ancestry from the CEPH collection. (CEU)",
"paperTitle": "Genome-wide pharmacogenetics of antidepressant response in the GENDEP project.",
"parentStudyid": 28063,
"snpGWASBiosetFeatureList": [
  {
    "biosetid": 183874,
    "oddRatio": null,
    "pvalue": "1.17815E-7",
    "rank": "1",
    "rsName": "rs10848645"
  }
],

"biosetTitle": "Heart rate variability - Low to High freq power ratio associated SNPs in FHS discovery cohort",
"biosetid": 202411,
"cohortSize": "cohort size not found",
"diseases": ["Heart rate variability - Low to High freq power ratio"],
"ethnic": "Reference population = Utah residents with Northern and Western European ancestry from the CEPH collection. (CEU)",
"paperTitle": "paper title not found",
"parentStudyid": 30940,
"snpGWASBiosetFeatureList": [
  {
    "biosetid": 202411,
    "oddRatio": null,
    "pvalue": "2.106E-7",
    "rank": "1",
    "rsName": "rs11062066"
  }
],

"biosetTitle": "Inflammatory Bowel Disease associated SNPs in Jewish and non-Jewish discovery cohort",
"biosetid": 193897,
"cohortSize": "Sample Number = 939 cases, 973 controls",
"diseases": ["Inflammatory bowel disease"],
"ethnic": "Reference population: Utah residents with Northern and Western European ancestry from the CEPH collection. (CEU)",
"paperTitle": "paper title not found"
studyannotation

The studyannotation command provides sample level annotation for publically available GSE studies resident in the Correlation Engine database.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>studyid</td>
<td>yes</td>
<td>The id of the Correlation Engine study or GEO Id.</td>
</tr>
</tbody>
</table>

Sample Request:

```
nbapi/studyannotation.api?studyid=GSE25624
```

Response: The response is a single StudyAnnotation object. See that object for details of the format.

Sample Response:

```
"result": [  
  {   
    "NBstudyId": 64864,  
    "biosetId": 526651,  
    "biosetName": "Bone marrow of patients with metastatic neuroblastoma _vs_ healthy controls",  
    "comparisonId": "1",  
    "description": "bone marrow; Gene expression",  
    "group": "Bone_marrow_of_healthy_young_donor",  
    "gseId": "GSE25624",  
    "gsmId": "GSM629613",  
    "lorR": "R",  
    "platform": "GPL4133",  
    "sampleTitle": "32-H_BM",  
    "studyTitle": "Bone marrow microenvironment in patients with neuroblastoma",  
    "tags": {  
      "biodesign": [  
        {  
          "id": "195785",  
          "name": "Disease vs. normal"  
        }  
      ]  
    }  
  }  
]  
```
"biosource": {  "id": "195777",  "name": "Primary tissue - fresh or fresh frozen" },  "tissue": {  "id": "217757",  "name": "Bone Marrow" },  "disease": {  "id": "1405775",  "name": "Neuroblastoma" }},  }],  }
  {  "NBstudyId": 64864,  "biosetId": 526648,  "biosetName": "Bone marrow of patients with localized neuroblastoma _vs_ healthy controls",  "comparisonId": "2",  "description": "bone marrow; Gene expression",  "group": "Bone marrow of healthy young donor",  "gseId": "GSE25624",  "gsmId": "GSM629613",  "lorR": "R",  "platform": "GPL4133",  "sampleTitle": "32-H_BM",  "studyTitle": "Bone marrow microenvironment in patients with neuroblastoma",  "tags": {  "biodesign": {  "id": "195785",  "name": "Disease vs. normal" },  "biosource": {  "id": "195777",  "name": "Primary tissue - fresh or fresh frozen" },  "tissue": {  "id": "217757",  "name": "Bone Marrow" },  "disease": {  "id": "1405775",  "name": "Neuroblastoma" }  }  }
]}

study

The study command provides information about a study, including the description, biosets, associated files, and tags.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>studyid</td>
<td>yes</td>
<td>The id of the study. This is normally obtained from another query.</td>
</tr>
</tbody>
</table>
Sample Request:
/nbapi/study.api?studyid=9253

Response: The response is a single Study object. See that object for details of the format.

Sample Response:

"result": {
  "associatedFiles": [
    {
      "id": 24091,
      "mimetype": "image/png",
      "name": "Stat_Report_GSE4779_Boxplot_logbase2.png",
      "type": "related"
    },
    {
      "id": 24100,
      "mimetype": "image/png",
      "name": "Stat_Report_GSE4779_GPL1352_comparison_1_histogram.png",
      "type": "related"
    },
    {
      "id": b257,
      "mimetype": "text/plain",
      "name": "Breast cancer N1 _vs_ N0.txt",
      "type": "bioset"
    },
    {
      "id": "d61",
      "mimetype": "text/plain",
      "name": "GSE4779_GPL1352_Samples.txt",
      "type": "data"
    }
  ],
  "authors": "Farmer P, Bonnefoi H, Iggo R",
  "biosets": [
    {
      "bioset": {
        "id": 64645,
        "title": "Breast cancer estrogen receptor ...
      },
      "featureCount": 6247
    },
    {
      "bioset": {
        "id": 64654,
        "title": "Breast cancer N1 _vs_ N0"
      },
      "featureCount": 377
    }
  ],
  "dataType": {
    "id": 1,
    "name": "RNA expression"
  },
  "description": "Series GSE4779 \r\n\r\nStatus ...",
  "id": 9253,
"institution": "Swiss Institute of Bioinformatics",
"organism": {
    "id": 1,
    "name": "Homo sapiens"
},
"shortDescription": "Gene expression signature ...",
"tags": [
    {
        "id": 142027,
        "name": "Breast cancer",
        "type": "disease"
    },
    {
        "id": 215436,
        "name": "Breast",
        "type": "tissue"
    }
],
"title": "Gene expression signature of breast cancer..."
}

studyids
The studyids command returns a list of internal study ID’s for a given external study ID.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gseid</td>
<td>yes</td>
<td>External study Id</td>
</tr>
</tbody>
</table>

Sample Request:

/ nbapi/studyids.api?gseid=gse9253

Response: The response is a list of study Id’s.

Sample Response:

    "result":  [
        13192,
        13354,
        13390,
        13465,
        13468
    ]


tagcloud
The tagcloud command returns a set of ApiTagCloudElement items for a given pubmed article id.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pmIds</td>
<td>Yes</td>
<td>Id’s of pubmed articles separated by commas.</td>
</tr>
</tbody>
</table>

Sample Requests:

/ nbapi/tagcloud.api?
/ nbapi/tagcloud.api?pmIds=12,200
Sample Response:
"result": [ 
  { 
    "freq": 0.1543033499620919,
    "name": "adipose tissue",
    "occurence": 1,
    "types": ["tissue"]
  },
  { 
    "freq": 1.0157134811107313,
    "name": "atp",
    "occurence": 4,
    "types": ["treatment"]
  }
]

trial
The trial command returns information about a single clinical trial. Unlike other commands in the Correlation Engine API, there is no Correlation Engine structure to represent the trial object. Instead, the XML object from clinicaltrials.gov is returned directly. If you don't wish to instantiate a special XML parser to parse this XML, the XML can optionally be expanded inline in the JSON or XML of the response.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>yes</td>
<td>The trial ID. This ID should be obtained from the trials command.</td>
</tr>
<tr>
<td>expand</td>
<td>no</td>
<td>If true, the XML is expanded inline in the result. See the expanded field in the response, discussed below.</td>
</tr>
</tbody>
</table>

Sample Requests:

/nbapi/trial.api?id=NCT00389649
Return the XML for the specified trial.

/nbapi/trial.api?id=NCT00389649&expand=true
Return the XML for the trial, and expand it into JSON or inline XML

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xml</td>
<td>string</td>
<td>The xml from clinicaltrials.gov</td>
</tr>
<tr>
<td>expanded</td>
<td>special</td>
<td>If the expand parameter is set, the XML is expanded into a JSON or XML object inline. While the data is the same as the original XML, it may not be in order.</td>
</tr>
</tbody>
</table>

Sample Response without expand parameter:

"result": { 
  "expanded": null,
  "xml": "<?xml version="1.0" encoding="UTF-8"?>
  \n  <clinical_study>
  ...
  </clinical_study>"
}

Sample Response with expand parameter:
"result": {  
"expanded": {  
  "lastchanged_date": "April 16, 2008",  
  "firstreceived_date": "October 17, 2006",  
  "location": [ {  
    "address": {  
      "zip": "21201",  
      "state": "Maryland",  
      "country": "United States",  
      "city": "Baltimore"  
    },  
    "name": "University of Maryland"  
  }],  
  "condition_browse": ["Heart Failure"],  
  ...  
  "xml": "<?xml version="1.0" encoding="UTF-8"?>
  <clinical_study>
    ...  
  </clinical_study>"
}

trials
The trials command returns a list of clinical trials related to a query term. Brief information is provided about each matching trial; to obtain full details about a trial use the trial command.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term. It should be URL encoded. See Query Parameter: q for more information.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of results requested. See Paging Parameters: fr and nr for information about pagination. A value over 1000 will be treated as 1000.</td>
</tr>
<tr>
<td>tf</td>
<td>no</td>
<td>A text filter. The string may be enclosed in double quotes (&quot;) for an exact match. This filter may be used to filter to a specific tag in a tag cloud.</td>
</tr>
<tr>
<td>tagCloud</td>
<td>no</td>
<td>If true, a tag cloud is returned in addition to the list of publications. To get only a tag cloud, set nr=0 to skip the publications.</td>
</tr>
<tr>
<td>tagType</td>
<td>no</td>
<td>Only used if tagCloud=true. Specifies the type of tag cloud (e.g. PUBLICATION, DISEASE). This parameter should normally be omitted on the initial tag cloud query, in which case the default tag cloud is returned. The allowable values are returned in the tagCloudTypes field in the result.</td>
</tr>
<tr>
<td>tagLookupSize</td>
<td>no</td>
<td>Only used if tagCloud=true. Specifies the number of articles to be considered in computing the tag cloud. The value should be between 50 and 1000. Other values will be adjusted. The default is 50. Larger values will take longer.</td>
</tr>
<tr>
<td>tagCloudSize</td>
<td>no</td>
<td>Only used if tagCloud=true. Specifies the maximum number of terms to return in the tag cloud. The maximum value is 70; larger values will be adjusted. If omitted, the default is to return the same number of terms as the UI would return, which depends on tagLookupSize.</td>
</tr>
<tr>
<td>sort</td>
<td>no</td>
<td>The sort criteria. Valid values are date and relevance. Default is relevance.</td>
</tr>
<tr>
<td>recruiting</td>
<td>no</td>
<td>If set to true, only clinical trials currently accepting volunteers are listed.</td>
</tr>
<tr>
<td>section</td>
<td>no</td>
<td>Only return results where the query term appears in the specified section(s). This parameter may appear multiple times. Valid values are &quot;title&quot;, &quot;summary&quot;, &quot;condition&quot;, &quot;intervention&quot;, &quot;description&quot;, &quot;outcome&quot;, &quot;group&quot;, and &quot;location&quot;.</td>
</tr>
</tbody>
</table>
Sample requests:

/\nbapi/trials.api?q=cancer
This returns the first 10 clinical trials articles containing cancer.

/\nbapi/trials.api?q=cancer&tagCloud=true&tagType=PUBLICATION&nr=0
This returns the publication tag cloud for articles containing cancer.

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalResults</td>
<td>integer</td>
<td>The total number of clinical trials. Unless the nr parameter is larger than the number of trials, totalResults will be larger than the number of results returned.</td>
</tr>
<tr>
<td>trials</td>
<td>ShortTrial[]</td>
<td>A list of trials</td>
</tr>
<tr>
<td>tagCloudItems</td>
<td>TagCloudItem[]</td>
<td>A list of tag cloud items. This is only returned if tagCloud=true.</td>
</tr>
<tr>
<td>tagCloudTypes</td>
<td>string[]</td>
<td>A list of valid types for the tagType attribute. This is only returned if tagCloud=true and tagType is omitted.</td>
</tr>
</tbody>
</table>

Sample Response:

"result": {
  "trials": [
    {
      "briefTitle": "AbioCor Implantable Replacement Heart",
      "conditions": ["Severe Biventricular End Stage Heart Disease"],
      "id": "NCT00669357",
      "interventions": ["Abiocor Implantable Replacement Heart"],
      "leadSponsor": "Abiomed Inc.",
      "status": "Available",
      "updated": "2010 May 26"
    },
    {
      "briefTitle": "Jarvik 2000 Heart as a Bridge to Cardiac...",
      "conditions": ["End-stage Heart Failure"],
      "id": "NCT00591799",
      "interventions": ["The Jarvik 2000 Heart"],
      "leadSponsor": "Jarvik Heart, Inc.",
      "status": "Recruiting",
      "updated": "2010 Dec 20"
    },
    ...
  ],
  "tagCloudItems": [
    {
      "name": "ace inhibitors",
      "rank": 0,
      "score": 0.6041850022919153,
      "types": ["treatment"]
    },
    {
      "name": "aortic valve",
      "rank": 1,
      "score": 0.9177588460407088,
      "types": ["tissue"]
    },
    ...
  ]
}
"tagCloudTypes": [  "BIOGROUP",  "COMPOUND",  "DISEASE",  "GENE",  "GEOGRAPHICAL",  "HEALTHCARE",  "HUMANITIES",  "INFORMATIONSCIENCE",  "OCCUPATION",  "ORGANISM",  "PROCESSES",  "PSYCHIATRY",  "PUBLICATION",  "SOCIOLOGY",  "SPONSOR",  "TECHNIQUE",  "TECHNOLOGY",  "TISSUE"  ],  "totalResults": 26704  
}

Genequickview

The genequickview command returns general info contents of a gene. The output includes information such as associated SNPs, gene ontology pathways, transcript proteins, biogroups etc. See GeneGeneralInfo for more information.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>yes</td>
<td>The query term. It should be URL encoded. See Query Parameter: q for more information.</td>
</tr>
<tr>
<td>fr</td>
<td>no</td>
<td>The first result of SNPs requested. See Paging Parameters: fr and nr for information about pagination.</td>
</tr>
<tr>
<td>nr</td>
<td>no</td>
<td>The number of SNPs results requested. See Paging Parameters: fr and nr for information about pagination. A value over 1000 will be treated as 1000.</td>
</tr>
<tr>
<td>includeAssociatedSNPs</td>
<td>no</td>
<td>If set to ON, result includes SNPsList. See GeneGeneralInfo for more information.</td>
</tr>
<tr>
<td>includeBiogroups</td>
<td>no</td>
<td>If set to ON, result includes featureAttribute. See GeneGeneralInfo for more information.</td>
</tr>
<tr>
<td>includeTranscriptsProteins</td>
<td>no</td>
<td>If set to ON, result includes proteinsList. See GeneGeneralInfo for more information.</td>
</tr>
</tbody>
</table>

Sample requests:

/nbapi/genequickview.api? q=esr1
/nbapi/genequickview.api? q=esr1&
includeAssociatedSNPs=true&includeBiogroups=true&includeTranscriptsProteins=true

Response: The response is a single GeneGeneralInfo object. See that object for details of the format.

Sample Response:

```json
{
  "command": "genequickview",
```
"parameters": { "includeTranscriptsProteins": "true", "v": "0", "fmt": "prettyjson:text", "includeAssociatedSNPs": "true", "q": "esr1", "apikey": "...
},
"result": { "GENEAnnotations": { "annotations": { "summary": "This gene encodes an estrogen receptor, a ligand-activated transcription factor composed of several domains important for hormone binding, DNA binding, and activation of transcription. The protein localizes to the nucleus where it may form a homodimer or a heterodimer with estrogen receptor 2. Estrogen and its receptors are essential for sexual development and reproductive function, but also play a role in other tissues such as bone. Estrogen receptors are also involved in pathological processes including breast cancer, endometrial cancer, and osteoporosis. Alternative splicing results in several transcript variants, which differ in their 5' UTRs and use different promoters. [provided by RefSeq]", "proteinnames": "estrogen receptor alpha",
"featureid": "8491",
"geneid": "2099",
"genesymbol": "ESR1",
"species": "Homo sapiens",
"inDictionary": "true",
"alias": "DKFZp686N23123, ER, Era, ESR, ESRA, estrogen receptor, NR3A1, OTTHUMP00000017718, OTTHUMP00000017719, RP1-130E4.1",
"speciesid": "1",
"synonyms": "DKFZp686N23123::ER::Era::ESR::ESRA::estrogen receptor::NR3A1::OTTHUMP00000017718::OTTHUMP00000017719::RP1-130E4.1",
"identifiersource": "NCBI Gene",
"phenotypes": "Atherosclerosis, susceptibility to::Breast cancer::Estrogen resistance::HDL response to hormone replacement, augmented::Migraine, susceptibility to::Myocardial infarction, susceptibility to::New sequence variants associated with bone mineral density::Multiple Genetic Loci for Bone Mineral Density and Fractures",
"description": "estrogen receptor 1",
"map": "6q25.1",
"featuretype": "gene" }
},
"GENEInfo": { "speciesList": [ { "featureid": 8491, "speciesid": 1, "speciesname": "Homo sapiens" }
, { "featureid": 284854, "speciesid": 2, "speciesname": "Mus musculus" }
, { "featureid": 604756, "speciesid": 3, "speciesname": "Rattus norvegicus" } ]}
{  
"featureid": 19964235,
"speciesid": 16,
"speciesname": "Macaca mulatta"
}

{  
"featureid": 20804710,
"speciesid": 47,
"speciesname": "Danio rerio"
}

{  
"featureid": 20816157,
"speciesid": 28,
"speciesname": "Gallus gallus"
}

{  
"featureid": 20817145,
"speciesid": 116,
"speciesname": "Canis lupus familiaris"
}

{  
"featureid": 20818487,
"speciesid": 44,
"speciesname": "Bos taurus"
}

"summaryLength": 704,
"entrezGeneLabel": "Entrez Gene",
"featureLocation": "Chr 6: 152170379-152466099",

"SNPsList":  
{  
"pageArgs": "id=8491",
"total#SNPs": 2226,
"pageSize": 10,
"snps": 
[  
  
  "function": "missense",
  "id": 5625319,
  "name": "rs17847065"
  
  },
  
  "function": "missense",
  "id": 5625322,
  "name": "rs17847076"
  
  },
  
  "function": "missense",
  "id": 5356199,
  "name": "rs9340773"
  
  },
  
  "function": "exon",
  "id": 5944431,
  "name": "rs61760169"
  
  },

```json
{
    "function": "exon",
    "id": 5850218,
    "name": "rs57015982"
},
{
    "function": "exon",
    "id": 5944354,
    "name": "rs61757379"
},
{
    "function": "coding-synonymous",
    "id": 5625321,
    "name": "rs17847067"
},
{
    "function": "coding-synonymous",
    "id": 5356227,
    "name": "rs9340802"
},
{
    "function": "coding-synonymous",
    "id": 5081458,
    "name": "rs1141584"
},
{
    "function": "coding-synonymous",
    "id": 5356481,
    "name": "rs9341069"
}
"featureAttribute": {},
"proteinsList": {
    "transcripts": [
        {
            "id": 8492,
            "label": "NM_000125"
        },
        {
            "id": 21858355,
            "label": "NM_001122740"
        },
        {
            "id": 21858356,
            "label": "NM_001122741"
        },
        {
            "id": 21858357,
            "label": "NM_001122742"
        }
    ],
    "uniprot": [
        "P03372",
        "Q5T5H7",
        "Q9UBT1"
    ]
}```
version
The version command returns the version number of the Correlation Engine API. Note that the current version is always returned, even if the value of the v parameter is for an older version.

The version command may be used with both the general and autocomplete API Keys.

Parameters: none

Sample request: /nbapi/version.api?

Response fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major</td>
<td>Integer</td>
<td>The major version number</td>
</tr>
<tr>
<td>Minor</td>
<td>Integer</td>
<td>The minor version number</td>
</tr>
</tbody>
</table>

Sample Response:
"result": {
  "major": 1,
  "minor": 2
}
Result Objects

This section describes the objects that are returned embedded in query results. These are the objects referred to in the Correlation Engine Query API Commands chapter. For each object, we provide a brief description, a list of the fields, and sample JSON results. This section is intended primarily as a reference, we recommend that you look at each object when it is included in a query result that you are interested in.

**AutocompleteItem**

An AutocompleteItem provides information on a single autocomplete result.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>ID of the item</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the item</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>The type of the item. One of: GENE, BIOGROUP, TISSUE, DISEASE, COMPOUND, SNP, AUTHOR, BIOSOURCE, Biodesign, GENEMODE. These types may be passed as the type parameter to the autocomplete command.</td>
</tr>
<tr>
<td>synonyms</td>
<td>string[]</td>
<td>A list of synonyms to the term. Synonyms that start with the query term will be listed before other synonyms.</td>
</tr>
</tbody>
</table>

Example:

```
{
    "id": 12345,
    "name": "GATA4",
    "synonyms": [
        "GATA binding protein 4",
        "Gata-4",
        "GATA-binding protein 4",
        "MGC126629",
        "OTTHUMP00000116010"
    ],
    "type": "GENE"
}
```

**BodyAtlasObjectImpl**

An BodyAtlasObjectImpl contains the name of the body system and a list of `ConceptBodyAtlasObject`

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bodySystem</td>
<td>string</td>
<td>The name of the body system.</td>
</tr>
<tr>
<td>concepts</td>
<td>List</td>
<td>List of <code>ConceptBodyAtlasObject</code> objects.</td>
</tr>
</tbody>
</table>

Example:

```
{
    "bodySystem": "Urogenital System",
    "concepts": [
        {
            "bodySystems": ["Urogenital System"],
            "conceptId": 196615,
            "conceptLabel": "Endometrium",
            "controlExpression": 252,
            "direction": 1,
```
"standardDeviation": 9910,
"tissueExpression": 18300
},

{
"bodySystems": ["Urogenital System"],
"conceptId": 196591,
"conceptLabel": "Cervix",
"controlExpression": 252,
"direction": 1,
"standardDeviation": 4640,
"tissueExpression": 10700
}
]

**BiosetBiogroupObject**

A BiosetBiogroupObject contains the name of the body system and a list of BiosetBiogroupBodyAtlasObject.

### Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bodySystem</td>
<td>string</td>
<td>The name of the body system.</td>
</tr>
<tr>
<td>concepts</td>
<td>List</td>
<td>List of BiosetBiogroupBodyAtlasObject objects.</td>
</tr>
</tbody>
</table>

Example:

```json
"bodySystem": "Immune System",
"concepts": [
  {
    "bodySystems": ["Immune System"],
    "conceptId": 200167,
    "conceptLabel": "Leukocyte (polymorphonuclear) of peripheral blood",
    "direction": 1,
    "pvalue": 0.021220067230533712,
    "score": 3.8528079775386557
  },
  {
    "bodySystems": ["Immune System"],
    "conceptId": 196780,
    "conceptLabel": "Spleen",
    "direction": 1,
    "pvalue": 0.11241447014924862,
    "score": 2.1855626118259304
  }
]
```

**ConceptBodyAtlasObject**

A ConceptBodyAtlasObject contains information about which body system the concept belongs to and overview information for a concept such as concept label, concept id, control expression, tissue expression etc.

### Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bodySystems</td>
<td>string[]</td>
<td>List of body systems to which this concept belongs to.</td>
</tr>
<tr>
<td>conceptId</td>
<td>integer</td>
<td>Id of the concept.</td>
</tr>
<tr>
<td>conceptLabel</td>
<td>string</td>
<td>Name of the concept.</td>
</tr>
<tr>
<td>controlExpression</td>
<td>double</td>
<td>Control expression is the median expression across all tissues.</td>
</tr>
<tr>
<td>stdDeviation</td>
<td>double</td>
<td>Gene expression level = tissueExpression +/- standardExpression.</td>
</tr>
<tr>
<td>tissueExpression</td>
<td>double</td>
<td>Gene expression level = tissueExpression +/- standardExpression.</td>
</tr>
<tr>
<td>direction</td>
<td>integer</td>
<td>If positive, the gene is overexpressed and if negative, the gene is</td>
</tr>
</tbody>
</table>
Example:
{
    "bodySystems": ["Urogenital System"],
    "conceptId": 196615,
    "conceptLabel": "Endometrium",
    "controlExpression": 252,
    "direction": 1,
    "standardDeviation": 9910,
    "tissueExpression": 18300
}

**BiosetBiogroupBodyAtlasObject**

A BiosetBiogroupBodyAtlasObject contains information about which body system the concept belongs to and overview information for a concept such as concept label, concept id, score and direction.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bodySystems</td>
<td>string[]</td>
<td>List of body systems to which this concept belongs to.</td>
</tr>
<tr>
<td>conceptId</td>
<td>integer</td>
<td>Id of the concept.</td>
</tr>
<tr>
<td>conceptLabel</td>
<td>string</td>
<td>Name of the concept.</td>
</tr>
<tr>
<td>pvalue</td>
<td>double</td>
<td>P-Value of the concept.</td>
</tr>
<tr>
<td>score</td>
<td>double</td>
<td>Gene expression enrichment score.</td>
</tr>
<tr>
<td>direction</td>
<td>integer</td>
<td>If positive, the bioset is overexpressed and if negative, the bioset is underexpressed.</td>
</tr>
</tbody>
</table>

Example:
{
    "bodySystems": ["Immune System"],
    "conceptId": 200197,
    "conceptLabel": "Natural killer cell of peripheral blood",
    "direction": 1,
    "pvalue": 0.5200029477317862,
    "score": 0.6539207987077575
}

**CategoryObject**

A CategoryObject contains information for a category such as label, id, regulation, normalized score, top level category, datatypes of the studies contributed to this category, number of studies and subcategories.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>datatypes</td>
<td>string[]</td>
<td>List of supporting data types of the study contributed to categorization.</td>
</tr>
<tr>
<td>id</td>
<td>integer</td>
<td>Id of the concept.</td>
</tr>
<tr>
<td>conceptLabel</td>
<td>string</td>
<td>Name of the concept.</td>
</tr>
<tr>
<td>regulation</td>
<td>double</td>
<td>Effect on query. If positive, the gene is up-regulated and if negative, the gene is down-regulated.</td>
</tr>
<tr>
<td>normalizedScore</td>
<td>double</td>
<td>Normalized score.</td>
</tr>
<tr>
<td>studyNum</td>
<td>integer</td>
<td>Number of studies.</td>
</tr>
<tr>
<td>topLevelConcept</td>
<td>String</td>
<td>Top level category of this concept.</td>
</tr>
<tr>
<td>subcategories</td>
<td>List</td>
<td>List of CategoryObject, which are subcategories of this category.</td>
</tr>
</tbody>
</table>

Example:

CategoryGeneObject
A CategoryGeneObject contains information for a category such as gene name, id, normalized score, regulation, number of studies and data types.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>datatypes</td>
<td>string[]</td>
<td>List of supporting data types of the study contributed to categorization.</td>
</tr>
<tr>
<td>entrezId</td>
<td>String</td>
<td>Entrez Id of the gene.</td>
</tr>
<tr>
<td>id</td>
<td>integer</td>
<td>Id of the gene.</td>
</tr>
<tr>
<td>gene</td>
<td>string</td>
<td>Name of the gene.</td>
</tr>
<tr>
<td>regulation</td>
<td>double</td>
<td>Effect on query. If positive, the gene is up-regulated and if negative, the gene is down-regulated.</td>
</tr>
<tr>
<td>normalizedScore</td>
<td>integer</td>
<td>Normalized score of this gene.</td>
</tr>
<tr>
<td>studyNum</td>
<td>integer</td>
<td>Number of studies.</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "command": "geneticmarkers",
  "parameters": {
    "v": "0",
    "fmt": "prettyjsontext",
    "q": "cell",
    "apikey": "...
```
},
"result": [

  {
    "datatypes": ["RNA expression"],
    "gene": "S100A8",
    "id": "0",
    "normalizedScore": 100,
    "regulation": -0.002044989775051125,
    "studyNum": 34
  },

  {
    "datatypes": ["RNA expression"],
    "gene": "COL3A1",
    "id": "0",
    "normalizedScore": 96,
    "regulation": -4.9965024482862E-5,
    "studyNum": 25
  },

  {
    "datatypes": ["RNA expression"],
    "gene": "VCAN",
    "id": "0",
    "normalizedScore": 94,
    "regulation": 0,
    "studyNum": 24
  }

]

**AssociatedFile**

An AssociatedFile provides the information needed to download a study's associated file. Associated files included related information (e.g. graphical images), bioset files (the files that were uploaded to import the bioset), and raw data.

### Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the file</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>The id of the file. This id must be passed to the associatedfile command to download the associated file.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>One of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• bioset: The file used to create a bioset.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• data: The raw data. This may be very large.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• related: Other related data, e.g. a graph.</td>
</tr>
<tr>
<td>mimetype</td>
<td>string</td>
<td>The mime type that will be returned for this file when it is downloaded.</td>
</tr>
<tr>
<td>length</td>
<td>integer</td>
<td>The length of the file in bytes. Because different computers handle end of lines differently, the size of a text file may vary slightly when downloaded.</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "id": 12345,
  "mimetype": "image/png",
  "name": "Stat_Report_GSE4779_Boxplot_logbase2.png",
  "type": "related"
}
```
Biogroup
A bioset provides overview information about a biogroup.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>string</td>
<td>The name of the biogroup</td>
</tr>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the biogroup. Commands that refer to this biogroup need this id.</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>The complete description of the biogroup</td>
</tr>
<tr>
<td>taxonomy</td>
<td>string</td>
<td>The taxonomy of the biogroup</td>
</tr>
</tbody>
</table>

Example:
```json
{
  "description": "Catalysis of an oxidation-reduction (redox) reaction...",
  "id": 39152,
  "taxonomy": "GO",
  "title": "oxidoreductase activity, acting on the CH-CH group of donors..."
}
```

Bioset
A bioset provides overview information about a bioset

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>string</td>
<td>The name of the bioset</td>
</tr>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the bioset. Commands that refer to this bioset need this id.</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>The complete description of the bioset</td>
</tr>
<tr>
<td>studyName</td>
<td>string</td>
<td>The name of the study containing this bioset</td>
</tr>
<tr>
<td>studyId</td>
<td>integer</td>
<td>The ID of the study containing this bioset</td>
</tr>
<tr>
<td>organism</td>
<td>Organism</td>
<td>The organism of the data in this bioset</td>
</tr>
<tr>
<td>dataType</td>
<td>DataType</td>
<td>The type of data in this bioset, e.g. RNA Expression</td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td>The tags on this bioset</td>
</tr>
<tr>
<td>featureCount</td>
<td>integer</td>
<td>The number of features in this bioset. For non gene-centric biosets, it may represent the number of sequences, SNPs, or miRNAs.</td>
</tr>
<tr>
<td>geneCount</td>
<td>integer</td>
<td>The number of genes in this bioset</td>
</tr>
<tr>
<td>platform</td>
<td>string</td>
<td>The platform</td>
</tr>
</tbody>
</table>

Example:
```json
{
  "dataType": {
    "id": 1,
    "name": "RNA expression"
  },
  "description": "Bioset summary = GSE4779 - Clinical Data\nComparison ...",
  "featureCount": 1959,
  "geneCount": 1612,
  "id": 64636,
  "organism": {
    "id": 1,
    "name": "Homo sapiens"
  },
  "platform": "Affymetrix GeneChip Human X3P",
  "studyId": 9253,
  "studyName": "Gene expression signature of breast cancer ..."
}
```
"tags": [],
"title": "Breast cancer N2 _vs_ N0"
}

**BiosetFeature**

A BiosetFeature provides information about one feature in a bioset, including statistics for that feature. For sequence-centric biosets, the BiosetFeature may provide information about a sequence instead.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the feature. This may be null for sequence centric biosets.</td>
</tr>
<tr>
<td>entrezId</td>
<td>integer</td>
<td>The id of the feature. This may be required for some queries. This may be</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>The imported ID of the feature. This may be null for sequence centric biosets.</td>
</tr>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the feature. This may be required for some queries. This may be</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>The statistics for this feature</td>
</tr>
<tr>
<td>sequence</td>
<td>Sequence</td>
<td>For sequence centric biosets, information about the sequence.</td>
</tr>
<tr>
<td>relatedGenes</td>
<td>RelatedGene[]</td>
<td>For sequence centric and miRNA biosets, information about genes related to</td>
</tr>
</tbody>
</table>

Example:

```
{
    "entrezId": "1571",
    "id": 40296,
    "importedId": "g5454037_3p_at",
    "name": "10609",
    "relatedGenes": null,
    "sequence": null,
    "statistics": [
        {
            "name": "p-value",
            "value": "0.0321"
        },
        {
            "name": "fold change",
            "value": "-1.37"
        },
        {
            "name": "Test expression",
            "value": "2086.1"
        },
        {
            "name": "Control expression",
            "value": "2850.7"
        }
    ]
}
```

**CorrelationBiogroup**

The CorrelationBiogroup provides information about a single biogroup when running the `correlationbiogroups` command.

Fields:
### Correlation Engine Query API

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biogroup</td>
<td>Biogroup</td>
<td>The biogroup summary information</td>
</tr>
<tr>
<td>direction</td>
<td>integer</td>
<td>The direction of the correlation. 1 is up, -1 is down, 0 is no directional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>information, and null means both up and down correlation.</td>
</tr>
<tr>
<td>pvalue</td>
<td>float</td>
<td>The p-value of the correlation</td>
</tr>
<tr>
<td>score</td>
<td>Score</td>
<td>The score for this biogroup</td>
</tr>
<tr>
<td>commonGenes</td>
<td>integer</td>
<td>The number of genes in common with the bioset</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "biogroup": {
    "description": "Catalysis of an oxidation-reduction (redox) ...",
    "id": 39152,
    "taxonomy": "GO",
    "title": "oxidoreductase activity, acting on the CH-CH group of donors..."
  },
  "commonGenes": 3,
  "direction": -1,
  "pvalue": 5.79281976307422E-14,
  "score": {
    "normalizedScore": 100,
    "score": 30.47957212316823
  }
}
```

**CorrelationBioset**

The CorrelationBioset provides information about a single bioset when running the `correlationbiosets` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bioset</td>
<td>ShortBioset</td>
<td>The bioset summary information</td>
</tr>
<tr>
<td>direction</td>
<td>integer</td>
<td>The direction of the correlation. 1 is up, -1 is down, 0 is no directional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>information, and null means both up and down correlation.</td>
</tr>
<tr>
<td>correlation</td>
<td>integer</td>
<td>When correlating against another bioset, provides the correlation between the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>two biosets, provide the correlation between them. 1 is a positive correlation,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1 is a negative correlation. Only one of direction and correlation will be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>provided.</td>
</tr>
<tr>
<td>pvalue</td>
<td>float</td>
<td>The p-value of the correlation</td>
</tr>
<tr>
<td>score</td>
<td>Score</td>
<td>The score for this bioset</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>The statistics for this feature</td>
</tr>
<tr>
<td>commonGenes</td>
<td>integer</td>
<td>When the original query was for a bioset, this contains the number of genes in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>common between the original bioset and the correlated bioset. Otherwise, it is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>null</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "bioset": {
    "id": 64636,
    "title": "Breast cancer N2 _vs_ N0"
  },
  "correlation": null,
```
"commonGenes": null,
"direction": -1,
"pvalue": 8.04E-4,
"score": {
  "normalizedScore": 100,
  "score": 99.9260547195076
},
"statistics": [ {
  "name": "fold change",
  "value": "-5.29"
}]
}

**CorrelationStudy**

A CorrelationStudy provides information about a study, when running a correlation command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>study</td>
<td>ShortStudy</td>
<td>The study summary information</td>
</tr>
<tr>
<td>score</td>
<td>Score</td>
<td>The score for this study</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "score": {
    "normalizedScore": 22,
    "score": 22.108828497396
  },
  "study": {
    "dataType": {
      "id": 1,
      "name": "RNA expression"
    },
    "id": 6181,
    "institution": "Thomas Jefferson University",
    "organism": {
      "id": 2,
      "name": "Mus musculus"
    },
    "shortDescription": "Cardiac-specific TNF-alpha transgenic mice ...",
    "title": "Heart failure in TNF-alpha transgenics"
  }
}
```

**DataType**

A DataType provides the ID and name of a bioset data type, e.g. RNA Expression.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the data type. This ID must be used when filtering by data type.</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the data type</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "id": 1,
```
"name": "RNA expression"
}

**GeneGWASBiosetFeature**

A GeneGWASBiosetFeature object provides information about a bioset feature. It contains information such as rsName, biosetid, pvalue, odds ratio and rank.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biosetid</td>
<td>integer</td>
<td>The id of the bioset.</td>
</tr>
<tr>
<td>rsName</td>
<td>string</td>
<td>rs name for this bioset feature.</td>
</tr>
<tr>
<td>pvalue</td>
<td>string</td>
<td>pvalue of this bioset feature.</td>
</tr>
<tr>
<td>oddRatio</td>
<td>string</td>
<td>Odds ratio for this bioset feature.</td>
</tr>
<tr>
<td>rank</td>
<td>string</td>
<td>Rank information.</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "biosetid": 143947,
    "oddRatio": "1.181",
    "pvalue": "7.0E-8",
    "rank": "1",
    "rsName": "rs1006737"
}
```

**GeneGWASBiosetObject**

A GeneGWASBiosetObject provides information about a bioset. It also contains list of GeneGWASBiosetFeature objects.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biosetid</td>
<td>integer</td>
<td>The id of the bioset.</td>
</tr>
<tr>
<td>biosetTitle</td>
<td>string</td>
<td>The title of the bioset.</td>
</tr>
<tr>
<td>parentStudyid</td>
<td>integer</td>
<td>The id of the study which this bioset belongs to.</td>
</tr>
<tr>
<td>diseases</td>
<td>list</td>
<td>List of diseases the bioset is tagged with.</td>
</tr>
<tr>
<td>cohortSize</td>
<td>string</td>
<td>The cohortSize fo this bioset, which is the number of cases.</td>
</tr>
<tr>
<td>ethnic</td>
<td>string</td>
<td>The population information for this bioset.</td>
</tr>
<tr>
<td>paperTitle</td>
<td>string</td>
<td>The title of the paper where the GWAS data comes from.</td>
</tr>
<tr>
<td>snpGWASBiosetFeatureList</td>
<td>list</td>
<td>The list of bioset features of this bioset.</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "biosetTitle": "Bipolar Disorder associated SNPs- in combined study",
    "biosetid": 143947,
    "cohortSize": "Sample Number = 10596",
    "diseases": ["Bipolar disorder"],
    "ethnic": "Reference population: Utah residents with Northern and Western European ancestry from the CEPH collection. (CEU)",
    "paperTitle": "paper title not found",
    "parentStudyid": 24229,
    "snpGWASBiosetFeatureList": [

```
"biosetid": 143947,
"oddRatio": "1.181",
"pvalue": "7.0E-8",
"rank": "1",
"rsName": "rs1006737"
},
{
"biosetid": 143949,
"oddRatio": "3.181",
"pvalue": "7.0E-8",
"rank": "1",
"rsName": "rs106588"
}

**GenomeBuild**

A GenomeBuild object provides information about all genome builds:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>genomeBuildId</td>
<td>integer</td>
<td>The id of the genome build.</td>
</tr>
<tr>
<td>speciesName</td>
<td>string</td>
<td>The name of the species.</td>
</tr>
<tr>
<td>name</td>
<td>integer</td>
<td>The name of the build.</td>
</tr>
<tr>
<td>ncbiBuild</td>
<td>list</td>
<td>Name of ncbi build.</td>
</tr>
<tr>
<td>ucscAssembly</td>
<td>string</td>
<td>Name of ucsc assembly.</td>
</tr>
<tr>
<td>isCurrent</td>
<td>boolean</td>
<td>State of the build weather it is current or not.</td>
</tr>
</tbody>
</table>

Example:
```
"result": [
  {
    "current": false,
    "genomeBuildId": 25,
    "name": "NCBI build 37 | UCSC hg19",
    "ncbiBuild": "build 37",
    "speciesName": "Homo sapiens",
    "ucscAssembly": "hg19"
  }
]
```

**IdAndName**

An IdAndName provides a mapping from an id to a name

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name</td>
</tr>
</tbody>
</table>

Example:
```
{
  "id": 195783,
  "name": "blood fraction"
}
```
**LiteratureItem**

A LiteratureItem object provides complete information about an individual literature result.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The PUBMED id of the article.</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>The name of the article.</td>
</tr>
<tr>
<td>articleAbstract</td>
<td>string</td>
<td>A summary of the article.</td>
</tr>
<tr>
<td>authors</td>
<td>String</td>
<td>The article authors.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>The article data type, e.g. PUBMED.</td>
</tr>
<tr>
<td>journal</td>
<td>string</td>
<td>The source journal of the article.</td>
</tr>
<tr>
<td>publishDate</td>
<td>string</td>
<td>The date the article was published.</td>
</tr>
<tr>
<td>affiliations</td>
<td>String[]</td>
<td>The list of article affiliations.</td>
</tr>
</tbody>
</table>

**OntologyInfoObject**

An OntologyInfoObject provides the id, synonyms, description and type of the ontology term.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the ontology term.</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>The description of this ontology term.</td>
</tr>
<tr>
<td>synonyms</td>
<td>string</td>
<td>Synonyms of the ontology as a string.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>The type of the ontology.</td>
</tr>
</tbody>
</table>

Example:
```
{
  "description": "a profound hemodynamic and metabolic disturbance characterized by the failure of the circulatory system to maintain adequate perfusion of vital organs; causes inadequate delivery of nutrients and insufficient removal of cellular waste products to and from the tissues.",
  "id": "147714",
  "synonyms": null,
  "type": "disease"
}
```

**OntologyHierarchyObject**

An OntologyHierarchy object provides ontology term information as well its children information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Integer</td>
<td>The id of the ontology term</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the ontology term</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>The description of this ontology term</td>
</tr>
<tr>
<td>synonymList</td>
<td>String[]</td>
<td>Synonyms of the ontology term as a list of string</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>The type of the ontology</td>
</tr>
<tr>
<td>numberOfChildren</td>
<td>Integer</td>
<td>The number of immediate children of a ontology term</td>
</tr>
<tr>
<td>useradded</td>
<td>Boolean</td>
<td>The term is added by user or not. Custom/Non-Illumina Ontology term is treated as useradded term</td>
</tr>
<tr>
<td>childrenList</td>
<td>OntologyChildNode</td>
<td>Returns immediate children of an ontology term</td>
</tr>
<tr>
<td></td>
<td>[]</td>
<td>Returns immediate children of an ontology term</td>
</tr>
</tbody>
</table>
## OntologyChildNode

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the ontology term</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the ontology term</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>The description of this ontology term</td>
</tr>
<tr>
<td>synonymList</td>
<td>String[]</td>
<td>Synonyms of the ontology term as a list of string</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>The type of the ontology</td>
</tr>
<tr>
<td>useradded</td>
<td>Boolean</td>
<td>The term is added by user or not. Custom/Non-Illumina Ontology term is treated as useradded term</td>
</tr>
</tbody>
</table>

## Organism

An Organism provides the ID and name of an organism, e.g. Homo Sapiens.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the organism. This ID must be used when filtering by organism.</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the organism</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "id": 1,
  "name": "Homo sapiens"
}
```

## RelatedGene

This provides information about a gene related to a gene or sequence in a bioset. For sequence-centric biosets, a RelatedGene represents a gene that overlaps or is near a sequence in the bioset. For miRNA biosets, a RelatedGene represents a gene targeted by a gene in the bioset. Because calculation of related genes may be slow, and because it may significantly increase the result size, related genes are only returned if the relatedgenes=true parameter is passed to the code.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>geneName</td>
<td>string</td>
<td>The name of the gene</td>
</tr>
<tr>
<td>geneId</td>
<td>integer</td>
<td>The id of the gene. This may be passed to other queries.</td>
</tr>
<tr>
<td>relation</td>
<td>string</td>
<td>The relationship between the sequence or feature in the bioset and this gene. The following values may be returned:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>targeted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>within</td>
</tr>
<tr>
<td></td>
<td></td>
<td>covering</td>
</tr>
<tr>
<td></td>
<td></td>
<td>left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>left overlapping</td>
</tr>
<tr>
<td></td>
<td></td>
<td>right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>right overlapping</td>
</tr>
</tbody>
</table>

Example:

```json
```
{  
  "geneId": 36706,  
  "geneName": "PCDHA9",  
  "relation": "left overlapping"  
}

**Score**

A score provides information about a score in a correlation. Both the absolute score as well as a normalized score are provided.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>score</td>
<td>float</td>
<td>The actual score</td>
</tr>
<tr>
<td>normalizedScore</td>
<td>integer</td>
<td>The score normalized to a range of 1-100. The highest study object gets a score of 100, and others are adjusted accordingly. The top study is calculated from the full query with no filters applied.</td>
</tr>
</tbody>
</table>

Example:

```
"score": {  
  "normalizedScore": 22,  
  "score": 22.108828497396  
}
```

**Sequence**

A Sequence provides information about a sequence of nucleotides.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chromosome</td>
<td>string</td>
<td>The chromosome containing the sequence.</td>
</tr>
<tr>
<td>start</td>
<td>integer</td>
<td>The start of the sequence.</td>
</tr>
<tr>
<td>stop</td>
<td>integer</td>
<td>The end of the sequence.</td>
</tr>
</tbody>
</table>

Example:

```
{  
  "chromosome": "3",  
  "start": 60452453,  
  "stop": 60521519  
}
```

**ShortBioset**

A ShortBioset provides basic information about a bioset. It is used when a list of biosets is returned in a command, e.g. the list of biosets in a study.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the bioset. This can be used with the bioset and biosetfeatures commands to obtain full information about the bioset.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>The name of the bioset.</td>
</tr>
</tbody>
</table>

Example:

```
{
}
```
"id": 64636,
"title": "Breast cancer N2 _vs_ N0"
}

**ShortStudy**

A ShortStudy provides basic information about a study. It is used when a list of studies is returned in a command, e.g. the list of studies in a correlation.

**Fields:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the study. This can be used with the study command to obtain full information about the study, and with the correlationbiosets command to obtain detailed correlations related to the study.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>The name of the study.</td>
</tr>
<tr>
<td>shortDescription</td>
<td>string</td>
<td>A summary of the study.</td>
</tr>
<tr>
<td>authors</td>
<td>string</td>
<td>The study authors.</td>
</tr>
<tr>
<td>institution</td>
<td>string</td>
<td>The study institution.</td>
</tr>
<tr>
<td>organism</td>
<td>Organism</td>
<td>The study organism.</td>
</tr>
<tr>
<td>dataType</td>
<td>DataType</td>
<td>The study data type.</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "dataType": {
        "id": 1,
        "name": "RNA expression"
    },
    "id": 6181,
    "institution": "Thomas Jefferson University",
    "organism": {
        "id": 2,
        "name": "Mus musculus"
    },
    "shortDescription": "Cardiac-specific TNF-alpha transgenic mice ...",
    "title": "Heart failure in TNF-alpha transgenics"
}
```

**ShortTrial**

A ShortTrial provides brief information about a clinical trial. It is used in search results. For complete information about a trial use the trial command.

**Fields:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>The identifier of the trial from clinicaltrials.gov. This ID may be passed into the trial command.</td>
</tr>
<tr>
<td>briefTitle</td>
<td>string</td>
<td>A short title</td>
</tr>
<tr>
<td>conditions</td>
<td>string[]</td>
<td>A list of conditions</td>
</tr>
<tr>
<td>interventions</td>
<td>string[]</td>
<td>A list of interventions</td>
</tr>
<tr>
<td>updated</td>
<td>String</td>
<td>The date this was last updated</td>
</tr>
<tr>
<td>status</td>
<td>String</td>
<td>The current status</td>
</tr>
<tr>
<td>leadSponsor</td>
<td>String</td>
<td>The lead sponsor</td>
</tr>
</tbody>
</table>

Example:

```json
{
}
```
"briefTitle": "AbioCor Implantable Replacement Heart",
"conditions": ["Severe Biventricular End Stage Heart Disease"],
"id": "NCT00669357",
"interventions": ["Abiocor Implantable Replacement Heart"],
"leadSponsor": "Abiomed Inc.",
"status": "Available",
"updated": "2010 May 26"
}

**Statistic**

A Statistic provides one a single piece of information, e.g. fold change.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the statistic.</td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>The value of a statistic. While many statistics are floating point numbers, some are not. Also, even statistics that are normally floating point numbers, may have an empty value, or a string like &quot;n/a&quot;.</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "name": "fold change",
  "value": "-1.37"
}
```

**Study**

A Study object provides complete information about a study. It includes all of the information provided by a ShortStudy, as well as additional information.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the study.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>The name of the study.</td>
</tr>
<tr>
<td>shortDescription</td>
<td>string</td>
<td>A summary of the study.</td>
</tr>
<tr>
<td>authors</td>
<td>string</td>
<td>The study authors.</td>
</tr>
<tr>
<td>institution</td>
<td>string</td>
<td>The study institution.</td>
</tr>
<tr>
<td>organism</td>
<td>Organism</td>
<td>The study organism.</td>
</tr>
<tr>
<td>dataType</td>
<td>DataType</td>
<td>The study data type.</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>The full description of the study</td>
</tr>
<tr>
<td>biosets</td>
<td>StudyBioset[]</td>
<td>The list of biosets, and information about them</td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td>The list of tags for this study</td>
</tr>
<tr>
<td>associatedFiles</td>
<td>AssociatedFile</td>
<td>The list of associated files for this study</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "associatedFiles": [
    {
      "id": 24091,
      "mimetype": "image/png",
      "name": "Stat_Report_GSE4779_Boxplot_logbase2.png"
    },
    {
      "id": 24100,
      "type": "related"
    }
  ]
}
```
"mimetype": "image/png",
"name": "Stat_Report_GSE4779_GPL1352_comparison_1_histogram.png",
"type": "related"
],
[
"id": b257,
"mimetype": "text/plain",
"name": "Breast cancer N1_vs_N0.txt",
"type": "bioset"
],
[
"id": "d61",
"mimetype": "text/plain",
"name": "GSE4779_GPL1352_Samples.txt",
"type": "data"
]
],
"authors": "Farmer P, Bonnefoi H, Iggo R",
"biosets": [
[
"bioset": {
"id": 64645,
"title": "Breast cancer estrogen receptor ...
"featureCount": 6247
}
],
[
"bioset": {
"id": 64654,
"title": "Breast cancer N1_vs_N0"
"featureCount": 377
}
]
],
"dataType": {
"id": 1,
"name": "RNA expression"
},
"description": "Series GSE4779 \n\n\nStatus ...",
"id": 9253,
"institution": "Swiss Institute of Bioinformatics",
"organism": {
"id": 1,
"name": "Homo sapiens"
},
"shortDescription": "Gene expression signature ...",
"tags": [
[
"id": 142027,
"name": "Breast cancer",
"type": "disease"
]
],
[
"id": 215436,
"name": "Breast",
"type": "tissue"
]
StudyAnnotation

A StudyAnnotation object provides information about a study annotation at sample level.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBstudyId</td>
<td>integer</td>
<td>The id of the study.</td>
</tr>
<tr>
<td>studyTitle</td>
<td>string</td>
<td>The name of the study.</td>
</tr>
<tr>
<td>gseId</td>
<td>string</td>
<td>GEO accession number-series of the study.</td>
</tr>
<tr>
<td>gsmId</td>
<td>string</td>
<td>GEO accession number</td>
</tr>
<tr>
<td>sampleTitle</td>
<td>string</td>
<td>The sample name.</td>
</tr>
<tr>
<td>platform</td>
<td>string</td>
<td>GEO platform number</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Sample description</td>
</tr>
<tr>
<td>group</td>
<td>string</td>
<td>Group name, defined in study annotation file</td>
</tr>
<tr>
<td>LorR</td>
<td>string</td>
<td>Comparison position – Left or Right side of comparison</td>
</tr>
<tr>
<td>comparisonId</td>
<td>string</td>
<td>Comparison number, defined in study annotation file</td>
</tr>
<tr>
<td>biosetName</td>
<td>string</td>
<td>The name of the bioset</td>
</tr>
<tr>
<td>biosetId</td>
<td>integer</td>
<td>The name of the bioset</td>
</tr>
<tr>
<td>tags</td>
<td>map</td>
<td>The list of tags associated with a bioset</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "NBstudyId": 64864,
  "biosetId": 526651,
  "biosetName": "Bone marrow of patients with metastatic neuroblastoma _vs_ healthy controls",
  "comparisonId": "1",
  "description": "bone marrow; Gene expression",
  "group": "Bone marrow of healthy young donor",
  "gseId": "GSE25624",
  "gsmId": "GSM629613",
  "lorR": "R",
  "platform": "GPL4133",
  "sampleTitle": "32-H_BM",
  "studyTitle": "Bone marrow microenvironment in patients with neuroblastoma",
  "tags": {
    "biodesign": {
      "id": "195785",
      "name": "Disease vs. normal"
    },
    "biosource": {
      "id": "195777",
      "name": "Primary tissue - fresh or fresh frozen"
    },
    "tissue": {
      "id": "217757",
      "name": "Bone Marrow"
    },
    "disease": {
      "id": "1405775",
      "name": "Neuroblastoma"
    }
  }
}
```
**StudyBioset**

A StudyBioset provides information about a bioset, within a Study object.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bioset</td>
<td>ShortBioset</td>
<td>The bioset</td>
</tr>
<tr>
<td>featureCount</td>
<td>integer</td>
<td>The number of features in the bioset</td>
</tr>
</tbody>
</table>

Example:
```
{
  "bioset": {
    "id": 64645,
    "title": "Breast cancer estrogen receptor ..."
  },
  "featureCount": 6247
}
```

**Tag**

A Tag object represents a tag.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>The id of the tag object</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the tag object</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>The type of the tag object</td>
</tr>
<tr>
<td>subTags</td>
<td>Tag</td>
<td>Few gene types will have genemode type subtags</td>
</tr>
</tbody>
</table>

Example:
```
[
  {
    "id": 142027,
    "name": "Breast cancer",
    "type": "disease"
  },
  {
    "id": 290349,
    "name": "Hivep3",
    "subTags": [
      {
        "id": 195793,
        "name": "Gene knockout",
        "subTags": null,
        "type": "genemode"
      }
    ],
    "type": "gene"
  }
]
```

**TagCloudItem**

A TagCloudItem represents one term in a tag cloud.
**Field** | **Type** | **Description**
--- | --- | ---
**name** | string | The string for the term. This may be passed in as a "tf" parameter to filter a search.
**rank** | integer | A value from 0 to 5 indicating the relative importance of this term. Larger values represent more important terms. These values correspond to the font sizes in the Correlation Engine UI.
**score** | float | A floating point number representing the importance of this term. The actual value is not significant, however, the ordering can be used for more exact ordering than can be obtained from rank.
**types** | string[] | An array of strings representing the type of term, e.g. "disease", or "author". Usually only a single term will be included, but for terms with multiple meanings, a larger array is returned.

Example:
```
{
    "name": "aortic stenosis",
    "rank": 1,
    "score": 0.5380305967850928,
    "types": ["disease"]
}
```

**ApiPMFacetGroup**
An ApiPMFacetGroup represents one facet.

**Fields:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>facet</strong></td>
<td>string</td>
<td>The name of the facet.</td>
</tr>
<tr>
<td><strong>facetElements</strong></td>
<td>ApiPMFacetElement[]</td>
<td>List of facet elements for this facet.</td>
</tr>
</tbody>
</table>

Example:
```
"facetGroup": [
{
    "facet": "Publication Year",
    "facetElements": [
    {
        "count": 49,
        "displayName": "2011"
    },
    {
        "count": 38,
        "displayName": "1975"
    }
    
},
{
    "facet": "Publication Type",
    "facetElements": [
    {
        "count": 257,
        "displayName": "Journal Article"
    },
    {
        "count": 48,
        "displayName": "Research Support, Non-U.S. Gov't"
    }
    
}]
```
## ApiPMFacetElement
An ApiPMFacetElement represents each element in a facet.

**Fields:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayName</td>
<td>string</td>
<td>The name of the element in a facet.</td>
</tr>
<tr>
<td>count</td>
<td>long</td>
<td>Value of the element.</td>
</tr>
</tbody>
</table>

**Example:**

```json
{
    "count": 49,
    "displayName": "2011"
}
```

## ApiTagCloudElement
A TagCloudElement represents one term in a tag cloud.

**Fields:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the element.</td>
</tr>
<tr>
<td>occurrence</td>
<td>integer</td>
<td>Number of times this term occurs in the related text.</td>
</tr>
<tr>
<td>freq</td>
<td>double</td>
<td>A double point number representing the importance of this term. The actual value is not significant, however, the ordering can be used for more exact ordering than can be obtained from rank</td>
</tr>
<tr>
<td>types</td>
<td>string[]</td>
<td>An array of strings representing the type of term, e.g. &quot;phenotype&quot;, or &quot;compound&quot;. Usually only a single term will be included, but for terms with multiple meanings, a larger array is returned</td>
</tr>
</tbody>
</table>

**Example:**

```json
{
    "freq": 0.1543033499620919,
    "name": "adipose tissue",
    "occurrence": 1,
    "types": ["tissue"]
}
```

## allstudy
The allstudy API provides information about a single study when running the `allstudies` command.

**Fields:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>desc</td>
<td>string</td>
<td>Description of this study.</td>
</tr>
<tr>
<td>id</td>
<td>integer</td>
<td>Id of the study.</td>
</tr>
<tr>
<td>projectlist</td>
<td>list</td>
<td>List of projects, which this study belongs to.</td>
</tr>
<tr>
<td>samplespeciesid</td>
<td>integer</td>
<td>Id of the samplespecies for this study.</td>
</tr>
</tbody>
</table>
**Correlation Engine Query API**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>speciesid</td>
<td>integer</td>
<td>Species to which this study is mapped to.</td>
</tr>
<tr>
<td>studydatatypeid</td>
<td>integer</td>
<td>Data type of this study.</td>
</tr>
<tr>
<td>summary</td>
<td>string</td>
<td>Summary of the study.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title of the study.</td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>URL link to the Study.</td>
</tr>
</tbody>
</table>

Example:

```
"result": {"apiAllStudy": [

  {
    "desc": "Mouse study description",
    "id": 3317,
    "projectlist": [
      4,
      1003
    ],
    "samplespeciesid": 1,
    "speciesid": 2,
    "studydatatypeid": 1,
    "summary": null,
    "title": "Mouse"
    "url": "https://florida.nextbiosystem.com/c/study/ov.nb?id=3317"
  },

  {
    "desc": "Mini study desc",
    "id": 4083,
    "projectlist": [746],
    "samplespeciesid": 3,
    "speciesid": 1,
    "studydatatypeid": 1,
    "summary": null,
    "title": "Mini"
    "url": "https://florida.nextbiosystem.com/c/study/ov.nb?id=4083"
  },

  {
    "desc": null,
    "id": 4646,
    "projectlist": [640],
    "samplespeciesid": 3,
    "speciesid": 2,
    "studydatatypeid": 1,
    "summary": null,
    "title": "cart"
    "url": "https://florida.nextbiosystem.com/c/study/ov.nb?id=4646"
  }
]
```

**Project**

The Project provides information about a single project when running the `allProjects` command.

Fields:
GeneGeneralInfo

The GeneGeneralInfo object provides information about general info of a gene when running the genequickview command.

Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>featureAttribute</td>
<td>map</td>
<td>Includes transcriptSites, miRNAsites, miRecords, molecular functions, bioprocesses, cellular Comps, pathways and proteins.</td>
</tr>
<tr>
<td>GENEAnnotations</td>
<td>map</td>
<td>Includes information like geneid, genesymbol, species, synonyms, summary etc.</td>
</tr>
<tr>
<td>GENEInfo</td>
<td>map</td>
<td>Returns default data. This includes species list, Entrez Gene Url etc.</td>
</tr>
<tr>
<td>SNPsList</td>
<td>map</td>
<td>Includes SNPs list, page size, total number of SNPs and pagination information.</td>
</tr>
<tr>
<td>proteinsList</td>
<td>map</td>
<td>Includes protein and transcript list.</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "command": "genequickview",
    "parameters": {
        "includeTranscriptsProteins": "true",
        "v": "0",
        "fmt": "prettyjsontext",
        "includeAssociatedSNPs": "true",
        "g": "esr1",
        "includeBiogroups": "true",
        "apikey": "..."
    },
    "result": {
        "GENEAnnotations": {
            "annotations": {
                "summary": "This gene encodes an estrogen receptor, a ligand-activated transcription factor composed of several domains important for hormone binding,"
            }
        }
    }
}
```
DNA binding, and activation of transcription. The protein localizes to the nucleus where it may form a homodimer or a heterodimer with estrogen receptor 2. Estrogen and its receptors are essential for sexual development and reproductive function, but also play a role in other tissues such as bone. Estrogen receptors are also involved in pathological processes including breast cancer, endometrial cancer, and osteoporosis. Alternative splicing results in several transcript variants, which differ in their 5' UTRs and use different promoters. [provided by RefSeq]

```
  "proteinnames": "estrogen receptor alpha",
  "featureid": "8491",
  "geneid": "2099",
  "genesymbol": "ESR1",
  "species": "Homo sapiens",
  "inDictionary": "true",
  "alias": "DKFZp686N23123, ER, Era, ESR, ESRA, estrogen receptor, NR3A1, OTTHUMP00000017718, OTTHUMP00000017719, RP1-130E4.1",
  "speciesid": "1",
  "synonyms": "DKFZp686N23123::ER::Era::ESR::ESRA::estrogen receptor::NR3A1::OTTHUMP00000017718::OTTHUMP00000017719::RP1-130E4.1",
  "identifiersource": "NCBI Gene",
  "phenotypes": "Atherosclerosis, susceptibility to::Breast cancer::Estrogen resistance::HDL response to hormone replacement, augmented::Migraine, susceptibility to::Myocardial infarction, susceptibility to::New sequence variants associated with bone mineral density::Multiple Genetic Loci for Bone Mineral Density and Fractures",
  "description": "estrogen receptor 1",
  "map": "6q25.1",
  "featuretype": "gene"
```
},
"SNPsList": {
  "pageArgs": "id=8491",
  "total#SNPs": 2226,
  "pageSize": 10,
  "snps": [
    {
      "function": "missense",
      "id": 5625319,
      "name": "rs17847065"
    },
    {
      "function": "missense",
      "id": 5625322,
      "name": "rs17847076"
    },
    {
      "function": "missense",
      "id": 5356199,
      "name": "rs9340773"
    },
    {
      "function": "coding-synonymous",
      "id": 5356227,
      "name": "rs9340802"
    },
    {
      "function": "coding-synonymous",
      "id": 5081458,
      "name": "rs1141584"
    },
    {
      "function": "coding-synonymous",
      "id": 5356481,
      "name": "rs9341069"
    }
  ]
},
"featureAttribute": {
  "miRecords": [{
    "id": 218847,
    "label": "hsa-miR-22 Targeted Genes"
  }],
  "bioProcesses": [
    {
      "id": 31160,
      "label": "antral ovarian follicle growth"
    },
    {
      "id": 52266,
      "label": "epithelial cell proliferation"
    },
    {
      "id": 241202,
      "label": "branch elongation involved in mammary gland duct branching"
    }
  ]}
"transcriptSites": [ 
  { 
    "id": 217915, 
    "label": "AP4 binding site geneset 5" 
  }, 

  { 
    "id": 218029, 
    "label": "SREBP1 binding site geneset 3" 
  }, 

  { 
    "id": 218251, 
    "label": "MMEF2 binding site geneset 1" 
  }, 

  { 
    "id": 218409, 
    "label": "GATA3 binding site geneset 1" 
  }
],

"cellComps": [ 
  { 
    "id": 33886, 
    "label": "nucleus" 
  }, 

  { 
    "id": 33958, 
    "label": "cytoplasm" 
  }, 

  { 
    "id": 34078, 
    "label": "plasma membrane" 
  }, 

  { 
    "id": 38668, 
    "label": "membrane" 
  }, 

  { 
    "id": 39111, 
    "label": "chromatin remodeling complex" 
  }, 

  { 
    "id": 47882, 
    "label": "plasma membrane part" 
  }, 

  { 
    "id": 46930, 
    "label": "neuronal cell body" 
  }, 

  { 
    "id": 703573, 
    "label": "cell body" 
  }
],

"miRNASites": [ 

{  
  "id": 54854,  
  "label": "Predicted Gene Targets for miR-203"
},  
{  
  "id": 54881,  
  "label": "Predicted Gene Targets for miR-496"
}
"pathways": [  
{  
  "id": 199331,  
  "label": "CARM1 and Regulation of The Estrogen Receptor"
},  
{  
  "id": 199370,  
  "label": "Diacylglycerol Signaling"
}
],  
"molecularFuns": [  
{  
  "id": 32437,  
  "label": "transcription factor activity"
},  
{  
  "id": 32443,  
  "label": "steroid hormone receptor activity"
},  
{  
  "id": 42324,  
  "label": "nitric-oxide synthase regulator activity"
},  
{  
  "id": 35798,  
  "label": "lipid binding"
}
],  
"proteins": [  
{  
  "id": 203672,  
  "label": "Zinc finger, nuclear hormone receptor-type"
},  
{  
  "id": 211850,  
  "label": "Nuclear hormone receptor, ligand-binding, core"
},  
{  
  "id": 212098,  
  "label": "Oestrogen receptor, AF1 modulatory"
},  
{  
  "id": 213982,  
  "label": "Nuclear hormone receptor, ligand-binding"
}
],  
"proteinsList": {  
}
"transcripts": [ 
  { 
    "id": 8492,
    "label": "NM_000125"
  }, 
  { 
    "id": 21858355,
    "label": "NM_001122740"
  }, 
],
"uniprot": [ 
  "P03372",
  "Q5T5H7",
  "Q9UBT1"
]
}

**CategoryStudy**

The CategoryStudy object provides information about study that contains a significant correlation with your query term.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activityType</td>
<td>string</td>
<td>Statistics from the bioset. For example, a public RNA expression bioset will contain fold change and p-value, while a copy number bioset will show either copy number change or Z-score.</td>
</tr>
<tr>
<td>activityValue</td>
<td>string</td>
<td>Statistics derived from the bioset.</td>
</tr>
<tr>
<td>biosetName</td>
<td>string</td>
<td>The name of the bioset.</td>
</tr>
<tr>
<td>biosetid</td>
<td>integer</td>
<td>The Id of the bioset associated with the study.</td>
</tr>
<tr>
<td>commonGenes</td>
<td>string</td>
<td>The number of genes in common with the bioset.</td>
</tr>
<tr>
<td>datatype</td>
<td>string</td>
<td>The study data type.</td>
</tr>
<tr>
<td>direction</td>
<td>string</td>
<td>The direction of the correlation. 1 is up, -1 is down, 0 is no directional information, and null means both up and down correlation. If positive, the bioset is overexpressed and if negative, the bioset is underexpressed.</td>
</tr>
<tr>
<td>institution</td>
<td>string</td>
<td>The study institution.</td>
</tr>
<tr>
<td>Platform</td>
<td>string</td>
<td>The name of the platform.</td>
</tr>
<tr>
<td>publicId</td>
<td>string</td>
<td>GEO Id.</td>
</tr>
<tr>
<td>Pvalue</td>
<td>string</td>
<td>P-value of the correlation.</td>
</tr>
<tr>
<td>score</td>
<td>string</td>
<td>The actual score. See <a href="#">Score</a> for more information.</td>
</tr>
<tr>
<td>species</td>
<td>string</td>
<td>Species to which this study is mapped to.</td>
</tr>
<tr>
<td>studyName</td>
<td>string</td>
<td>The name of the study.</td>
</tr>
<tr>
<td>studyid</td>
<td>integer</td>
<td>The Id of the study.</td>
</tr>
</tbody>
</table>

Example:

```json
{
}```
"activityType": "fold change",
"activityValue": "-1.4",
"biosetName": "Biopsy at diagnosis of diffuse large B-cell lymphoma patients vs follicular lymphoma patients",
"BioSetId": 340514,
"commonGenes": "N/A",
"dataType": "RNA Expression",
"direction": "N/A",
"institution": "",
"platform": "Affymetrix GeneChip Human Full Length HuGeneFL",
"publicId": "",
"pvalue": "0.0179",
"score": "31.50519205565077",
"species": "Homo sapiens",
"studyName": "Diffuse large B-cell lymphoma outcome prediction by gene expression profiling",
"studyId": 47495
}

Change History

This chapter lists changes in the API by version. Any changes incompatible with previous versions will be noted.

Version 0.30
- Added ontologyhierarchy command.

Version 0.29
- Added source to bodyatlas command

Version 0.28
- Added sectiontype to diseaseatlas command.

Version 0.27
- Added studyannotation command.
- Added bioset query type to diseaseatlas, pharmacoatlas, knockdownatlas and biogroups command.
- Added text filter and biogroup(taxonomy) filter to biogroups command.

Version 0.26
- Extend the knockdownatlas command to include biostIds and gene perturbation type for each CategoryGeneObject.

Version 0.25
- Added note to Documentation Conventions.

Version 0.24
- Added correlationstudies command.

Version 0.23
- Added optional Autocomplete alttype parameter.
Version 0.22
  • Documentation update and cleanup. The allstudies, allprojects and allplatforms APIs were changes to lowercase. Note: the uppercase versions still function the same for backward compatibility.

Version 0.21
  • Added studyids command.
  • Added subtags field to the Tag result object.

Version 0.20
  • Added genomebuild command.
  • Extend the biotsetfeatures command to include entrezId for each BiosetFeature.
  • Extend the knockdownatlas command to include entrezId for each CategoryGeneObject.
  • Extend the geneticmarkers command to include entrezId for each CategoryGeneObject.

Version 0.19
  • Added createproject command.
  • Add tagcloud command.
  • Extend the literature command to return facet results.

Version 0.18
  • Added ‘cv’ parameter to the autocomplete command.

Version 0.17
  • Add ontologyinfo command.
  • Add snpgwasdata command.

Version 0.16
  • Extend the bodyatlas command to include p-value to result objects (biogroups query).
  • Added ‘biosetid’ parameter to the bodyatlas command.
  • Add diseaseatlas command.
  • Add pharmacoatlas command.
  • Add knockdownatlas command.
  • Add geneticmarkers command.
  • Add biogroups command.

Version 0.15
  • Add bodyatlas command to find normalized gene expression across all tissues, cell types, cell lines, and stem cells. Also used to find bioset and biogroup information.
  • Added abstractsection, titlesection, meshtagssection parameters to the literature command.

Version 0.14
  • Add editbioset command to update name and description of a bioset.
  • Add deletebioset command to delete a bioset.
  • Add editstudy command to update name, description and summary of a study.
  • Add deletestudy command to delete a study.
  • Add genequickview command to view general information of a gene.
  • Extend the allProjects command to include shared projects.
Correlation Engine Query API

Version 0.13
- Add studylist to ApiProject result object.
- Add projectlist to ApiAllStudy result object.
- Add samplespeciesid to ApiAllStudy result object.

Version 0.12
- Add writable flag to ApiProject result object.
- Add libraryid to ApiProject result object.
- Extend the allProjects command to add total number of results.
- Add includeDomain parameter to the allProjects command.
- Add includeDomain parameter to the allStudies command.
- Add includeDomain parameter to the libraries command.
- Extend the allStudies command to add total number of results.
- Add Term ID to AutocompleteItem result object.

Version 0.11
- Extend the allStudies command to add a URL link to each ApiAllStudy object.

Version 0.10
- Add allProjects command to view the project details.
- Add allStudies command to view study details.
- Add libraries command, to view libraries based on the domain.
- Add allPlatforms command to obtain information of all the platforms.

Version 0.9
- Add trials and trial commands to obtain information about clinical trials.
- Add The Correlation Engine API Testbed.

Version 0.8
- Add sort and bold parameters to literature command.
- Add the referencingstudies and referencingbiosets commands.
- Extend the correlation command, so it can also return the referencingstudies results.
- The correlation and literature commands no longer return errors if there are no results. Instead they return regular results with totalResults=0.

Version 0.7.1
- Autocomplete uses a separate API Key.

Version 0.7
- Add autocomplete command.

Version 0.6
- Add text filter (tf) and tag cloud attributes to literature command

Version 0.5
- Add literature command
- Add literatureItem result object
Version 0.4

- Add Sequence to BiosetFeature result object.
- The correlation and correlationbiosets commands now support sequence-centric queries through the chr, chrorg, chrstart, and chrstop parameters.
- Add relatedGenes to BiosetFeature result object. Add relatedgenes=true parameter to the biosetfeatures command.
- Add platform to Bioset result object.
- Add length to AssociatedFile result object.
- Add zip compression to the associatedfile command, with support for multifile download.
- Add correlationbiogroups command to correlate bioset against biogroups. Add Biogroup and CorrelationBiogroup result objects.
- Add the concept of Context groups.
- Add commonGenes field to CorrelationBioset object.
- The internal format of the Correlation context has changed. A context from version 0.3 will no longer work in version 0.4.