master document for generating PDF

= Genomic Data Management (GDM)

Table of Contents

Overview

The GDM web service API provides a comprehensive set of RESTful methods for accomplishing the following core tasks in the GOBii Genomic Data Management (GDM) system. This manual explains the GDM web architecture and its web service operations.

Version information

Version : 1.0

License information

License : Apache 2.0
License URL : http://www.apache.org/licenses/LICENSE-2.0.html
Terms of service : null

URI scheme

Host : cbsugobiixvm11.biohpc.cornell.edu:8081
BasePath : /gobii-dev
Schemes : HTTP

Tags

- Analyses

- Authentication

- ConfigSettings

- Contacts

- ControlledVocabularies

- Datasets

- Displays

- Entities

- Experiments

- Files

- Instructions

- Jobs

- Manifests

- Maps

- Markers

- Names

- Organizations
- Ping
- Platforms
- Projects
- Protocols
- Protocols.vendor
- References
- RestProfiles

Overview

GOBii's internal documentation explains in detail how the web service infrastructure is architected. It is not necessary for a consumer of the GOBii web API to understand this in great detail. However, knowing a few things about it will make it easier to understand the web method documentation.

Language and Framework

GOBii Web Services are implemented in Java with the Spring frameowrk. Virtually all calls conform to the same basic structure. For example, the /analyses POST method is as follows:

```
@RequestMapping(value = "/analyses", method = RequestMethod.POST)
@ResponseBody
public PayloadEnvelope<AnalysisDTO> createAnalysis(@ApiParam(required = true)
@RequestBody PayloadEnvelope<AnalysisDTO> analysisPostEnvelope,
                                          HttpServletRequest request,
                                          HttpServletResponse response) {

    PayloadEnvelope<AnalysisDTO> returnVal = new PayloadEnvelope<>();

    // populate returnVal . . .

    return returnVal;
}
```

Ther request and response are almost always wrapped in a type-parameterized PayloadEvenlope. In this case, the PayloadEnvelope is type-parameterized for the AnalysisDTO. Other methods are type parameterized according to the entity that they touch (e.g., the implementation of the /datasets POST method is type-parameterized for the DataSetDTO).

The DTO Payload

The DTO objects are lightweight classes that contain simple data properties. They correspond roughly but not exactly to entities in the database (for example, in some cases a DTO defines a child collection of foreign-keyed entitites).

For a client of this application, however, the strongly-typed nature of the implementation is irrelvant: all request responses and bodies are conveyed in plain JSON, and it is the application's job to serialize and deserialize accordingly. What a client implementation must take into account, however, is the structure of the DTO that pertains to a specific all. From the reference section for each web method in this document, youcan navigate to the DTO definition.

Request/Response Structure

The following illustrates the http body structure encompassed by the PayloadEnvelope. As mentioned above, it is the same for almost all responses as well as for the body of POST and PUT requests.

```
{
    "payload":{
        "linkCollection":{
            "linksPerDataItem":[

            ],
            "exploreLinksPerDataItem":[

            ]
        },
        "data":[

        ]
    },
    "header":{
        "gobiiProcessType":"",
        "dtoHeaderAuth":{
            "userName":null,
            "password":null,
            "token":"",
            "gobiiCropType":""
        },
        "status":{
            "succeeded":true,
            "statusMessages":[

            ],
            "statusMessagesByCode":{

            }
        },
        "pagination":null,
        "gobiiVersion":"2.0_2018-07-31",
        "cropType":""
    }
}
```

For requests, the following keys can be omitted:

- payload.linkCollection
- header

The content of the "data" array will vary depending on how the call is type-parameterized. For example, for a POST to the the /project resource, it will contain a project object, as defined by ProjectDTO.

Authentication

All calls except for /configsettings GET require a valid authentication token. The token can be acquired via a POST to the /auth resource with an empty body and the following request headers set:

- X-Username
- X-Password

The specified header values must, of course, be a valid GOBii username and password. Both values are expected to be unencrypted (https will provide encryption for the overall request stream).

GOBii users must be defined in the system's LDAP and in the GOBii contact table. Accordingly, there can be two types of authentication failures:

- 401 Unauthorized: the user does not authenticate to LDAP;
- 403 Forbidden: the contact record does not exist. In this case, the error will indicate "Missing contact info for user <user-name> in crop database <crop-db>; a contact record must have username = <user-name>";

When authentication succeeds, the response headers will contain an X-Auth-Token key whose value is a valid token. For subsequent requests, it is sufficient to supply X-Auth-Token in the request headers with the token that was received from the /auth POST. Some browsers prevent the client code from reading custom response headers. For such cases, the token value can also be retrieved from the "token" key in the "dtoHeaderAuth" object.

Error Handling

Errors internal to GOBii are reported in the "header" key accompanying the response. An example of how to retrieve error data is provided in the following code fragment.

```
if (!header.getStatus().isSucceeded() ||
        header
                .getStatus()
                .getStatusMessages()
                .stream()
                .filter(headerStatusMessage ->
headerStatusMessage.getGobiiStatusLevel().equals(GobiiStatusLevel.VALIDATION))
                .count() > 0) {
    returnVal = true;
    System.out.println();
    System.out.println("*** Header errors: ");
    for (HeaderStatusMessage currentStatusMesage :
header.getStatus().getStatusMessages()) {
        System.out.println(currentStatusMesage.getMessage());
    }
}
```

Paths

Create a new analyses.

```
POST /gobii/v1/analyses
```

Description

Create analysis entity for GDM system.

Parameters

Type	Name	Schema
Body	**body** __	PayloadEnvelopeAnalysisDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeAnalysisDTO

Tags

- Analyses

List all analyses

```
GET /gobii/v1/analyses
```

Description

List of all Analysis entities.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeAnalysisDTO

Tags

- Analyses

Get an analyses by analysesId

```
GET /gobii/v1/analyses/{analysisId}
```

Description

Get analyses by analyses Id.

Parameters

Type	Name	Description	Schema
Path	**analysisId** —	ID of Analysis to be extracted.	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeAnalysisDTO

Tags

- Analyses

Update the analyses by analysesId

> PUT /gobii/v1/analyses/{analysisId}

Description

Updates the Analysis entity having the specified analysisId.

Parameters

Type	Name	Description	Schema
Path	**analysisId** —	ID of Analysis to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeAnalysisDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeAnalysisDTO

Tags

- Analyses

authenticate

> POST /gobii/v1/auth

Description

The user credentials are specified in the request headers X-Username and X-Password; the response and the response headers include the token in the X-Auth-Token header. X-Auth-Token header and value obtained from /auth call will be used as an API-key for the rest of the GDM calls.

Parameters

Type	Name	Description	Schema
Header	**X-Password** —	User password	string

Type	Name	Description	Schema
Header	**X-Username** —	User Identifier	string
Body	**body** —		string

Responses

HTTP Code	Description	Schema
200	OK **Headers** : `X-Auth-Token ` (string) : API key to authenticate GDM api calls.	string

Tags

- Authentication

List all configuration settings.

```
GET /gobii/v1/configsettings/
```

Description

List all configuration settings. +Provides generic configuration information about the GOBii instances in a given deployment. This call does not require authentication

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeConfigSettingsDTO

Tags

- ConfigSettings

List all contacts from Contacts search

```
GET /gobii/v1/contact-search
```

Description

List contacts for emailid, lastname, firstname, username.

Parameters

Type	Name	Description	Schema
Query	**email** —	contact's email	string
Query	**firstName** —	contact's first name	string
Query	**lastName** —	contact's last name	string
Query	**userName** —	contact's user name	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeContactDTO

Tags

- Contacts

Create a new contact

```
POST /gobii/v1/contacts
```

Description

Create new contact.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeContactDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeContactDTO

Tags

- Contacts

List all contacts

```
GET /gobii/v1/contacts
```

Description

List all contacts.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeContactDTO

Tags

- Contacts

Get a contact by contact id

```
GET /gobii/v1/contacts/{contactId}
```

Description

Get contacts by Contact Id.

Parameters

Type	Name	Description	Schema
Path	**contactId** —	ID of contacts to be extracted.	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	[PayloadEnvelopeContactDTO](#)

Tags

- Contacts

Update a contact by contactId

```
PUT /gobii/v1/contacts/{contactId}
```

Description

Update contact by contact id.

Parameters

Type	Name	Description	Schema
Path	**contactId** __	ID of contacts to be updated.	integer (int32)
Body	**body** __		[PayloadEnvelopeContactDTO](#)

Responses

HTTP Code	Description	Schema
200	successful operation	[PayloadEnvelopeContactDTO](#)

Tags

- Contacts

Get contact by email id

```
GET /gobii/v1/contacts/{email}
```

Description

Get contact by email id.

Parameters

Type	Name	Description	Schema
Path	**email** —	email id of contacts to be extracted.	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeContactDTO

Tags

- Contacts

list

```
GET /gobii/v1/cvgroups/{cvGroupId}/cvs
```

Description

List CV terms in CV group with cvGroupId

Parameters

Type	Name	Description	Schema
Path	**cvGroupId** —	ID of the CV group.	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvDTO

Tags

- ControlledVocabularies

Get a CV group by cvGroupTypeId

```
GET /gobii/v1/cvgroups/{cvGroupTypeId}
```

Description

Get CV group by cv group type ID

Parameters

Type	Name	Description	Schema
Path	cvGroupTypeId __	ID of the cv group type	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvGroupDTO

Tags

- ControlledVocabularies

Get a CV group by groupName

```
GET /gobii/v1/cvgroups/{groupName}
```

Description

Get CV group by name

Parameters

Type	Name	Description	Schema
Path	groupName __	name of the cv group to be extracted	string
Query	cvGroupTypeId __		integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvGroupDTO

Tags

- ControlledVocabularies

Create a new Controlled Vocabulary

```
POST /gobii/v1/cvs
```

Description

Creates new cv's.

Parameters

Type	Name	Schema
Body	**body** __	PayloadEnvelopeCvDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvDTO

Tags

- ControlledVocabularies

List all CVs

```
GET /gobii/v1/cvs
```

Description

List all cvs in the GDM.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvDTO

Tags

- ControlledVocabularies

Get a CV by cvId

```
GET /gobii/v1/cvs/{cvId}
```

Description

Get cv by the id.

Parameters

Type	Name	Description	Schema
Path	**cvId** __	ID of the CV to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvDTO

Tags

- ControlledVocabularies

Update a CV by cvId

```
PUT /gobii/v1/cvs/{cvId}
```

Description

Update cv by cvId.

Parameters

Type	Name	Description	Schema
Path	**cvId** —	ID of the CV to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeCvDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvDTO

Tags

- ControlledVocabularies

Delete a CV by cvId

```
DELETE /gobii/v1/cvs/{cvId}
```

Description

Deletes cv by id.

Parameters

Type	Name	Description	Schema
Path	**cvId** —	ID of cv to be deleted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvDTO

Tags

- ControlledVocabularies

List all CVs in a given groupName

```
GET /gobii/v1/cvs/{groupName}
```

Description

List cvs by the group name.

Parameters

Type	Name	Description	Schema
Path	**groupName** —	name of cv group to be extracted	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvDTO

Tags

- ControlledVocabularies

Create a new dataset

```
POST /gobii/v1/datasets
```

Description

Creates a new dataset in the system.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeDataSetDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDataSetDTO

Tags

- Datasets

List all datasets

```
GET /gobii/v1/datasets
```

Description

List all the existing datasets in the system. The list can be retrieved by page and specific page size.

Parameters

Type	Name	Description	Schema
Query	**pageNo** —	Retrieve the specified page by number	integer (int32)
Query	**pageSize** —	Specify the custom page size	integer (int32)
Query	**queryId** —		string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDataSetDTO

Tags

- Datasets

List of types of datasets

```
GET /gobii/v1/datasets/types
```

Description

Lists dataset types in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeNameIdDTO

Tags

- Datasets

Get datasets by type id

```
GET /gobii/v1/datasets/types/{id}
```

Description

Gets the Dataset type by type ID.

Parameters

Type	Name	Description	Schema
Path	**id** __	ID of the dataset type	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDataSetDTO

Tags

- Datasets

Get dataset by datasetId

```
GET /gobii/v1/datasets/{dataSetId}
```

Description

Gets the Dataset entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**dataSetId** —	ID of the Dataset to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDataSetDTO

Tags

- Datasets

Update/Replace a dataset by datasetId

```
PUT /gobii/v1/datasets/{dataSetId}
```

Description

Updates the Dataset entity having the specified datasetId.

Parameters

Type	Name	Description	Schema
Path	**dataSetId** —	ID of the Dataset to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeDataSetDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDataSetDTO

- Datasets

List all analyses in a dataset with given datasetId

```
GET /gobii/v1/datasets/{dataSetId}/analyses
```

Description

Lists all the analysis in a dataset identified by dataset id.

Parameters

Type	Name	Description	Schema
Path	**dataSetId** __	ID of the dataset	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeAnalysisDTO

Tags

- Datasets

List all jobs for a dataset with datasetId

```
GET /gobii/v1/datasets/{datasetId}/jobs
```

Description

List information for active job for a given datasetId

Parameters

Type	Name	Description	Schema
Path	**datasetId** __	ID of the dataset	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeJobDTO

Tags

- Datasets

Create a new display

```
POST /gobii/v1/displays
```

Description

Creates displays in GDM.

Parameters

Type	Name	Schema
Body	**body** __	PayloadEnvelopeDisplayDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDisplayDTO

Tags

- Displays

List all displays

```
GET /gobii/v1/displays
```

Description

Lists all Displays in GDM

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDisplayDTO

Tags

- Displays

Get a display by displayId

```
GET /gobii/v1/displays/{displayId}
```

Description

Get the Display by display Id

Parameters

Type	Name	Description	Schema
Path	**displayId** __	ID of the Display to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDisplayDTO

Tags

- Displays

Update a display with displayId

```
PUT /gobii/v1/displays/{displayId}
```

Description

Updates the Display entity having the specified displayId.

Parameters

Type	Name	Description	Schema
Path	**displayId** —	ID of the Display to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeDisplayDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeDisplayDTO

Tags

- Displays

List all entities

```
GET /gobii/v1/entities
```

Description

Lists all the existing entities in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeEntityStatsDTO

Tags

- Entities

Get entity count for a given entity child

```
GET /gobii/v1/entities/{entityNameParent}/{parentId}/{entityNameChild}/count
```

Description

Retrieves the total count of the children for the given entity

Parameters

Type	Name	Description	Schema
Path	**entityNameCh ild** —	Name of the child entity	string
Path	**entityNamePa rent** —	Name of the parent entity	string
Path	**parentId** —	ID of the parent entity	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeE ntityStatsDTO

Tags

- Entities

Get count for a given entity

```
GET /gobii/v1/entities/{entityName}/count
```

Description

Gets the total Entity count for the given entityName

Parameters

Type	Name	Description	Schema
Path	**entityName** —	Name of the Entity	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeEntityStatsDTO

Tags

- Entities

Get last modified date of an entity

```
GET /gobii/v1/entities/{entityName}/lastmodified
```

Description

Gets last modified Entity for the given entityName

Parameters

Type	Name	Description	Schema
Path	**entityName** __	Name of the Entity	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeEntityStatsDTO

Tags

- Entities

Create a new experiment

```
POST /gobii/v1/experiments
```

Description

Creates a new experiment in the system.

Parameters

Type	Name	Schema
Body	**body** —	[PayloadEnvelopeExperimentDTO](#)

Responses

HTTP Code	Description	Schema
200	successful operation	[PayloadEnvelopeExperimentDTO](#)

Tags

- Experiments

List all experiments

```
GET /gobii/v1/experiments
```

Description

List all the existing experiments in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	[PayloadEnvelopeExperimentDTO](#)

Tags

- Experiments

Get a experiment by experimentId

```
GET /gobii/v1/experiments/{experimentId}
```

Description

Gets the Experiment entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**experimentId** __	ID of the Experiment to be updated	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeExperimentDTO

Tags

- Experiments

Update an experiment by experimentId

```
PUT /gobii/v1/experiments/{experimentId}
```

Description

Updates the Experiment entity having the specified experimentId.

Parameters

Type	Name	Description	Schema
Path	**experimentId** __	ID of the Experiment to be updated	integer (int32)
Body	**body** __		PayloadEnvelopeExperimentDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeExperimentDTO

Tags

- Experiments

List all protocols for a given experimentId

```
GET /gobii/v1/experiments/{experimentId}/protocols
```

Description

Retrieves all the protocols having the specified experimentId in the system.

Parameters

Type	Name	Description	Schema
Path	**experimentId**__	ID of the Experiment	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeProtocolDTO

Tags

- Experiments

List of loader files in a given directory

```
GET /gobii/v1/files/loader/{directoryName}
```

Description

Gets file preview for the specified directory name

Parameters

Type	Name	Description	Schema
Path	**directoryName**__	Name of the directory	string
Query	**fileFormat**__	Format/Extension of the file	string

Responses

HTTP Code	Description	Schema
200	successful operation	[PayloadEnvelopeLoaderFilePreviewDTO](#)

Tags

- Files

Add a loader file to given directory

```
PUT /gobii/v1/files/loader/{directoryName}
```

Description

Updates a directory in the system that will be used for storing the data files for loading

Parameters

Type	Name	Description	Schema
Path	**directoryName** __	Name of the directory/folder	string
Body	**body** __		[PayloadEnvelopeLoaderFilePreviewDTO](#)

Responses

HTTP Code	Description	Schema
200	successful operation	[PayloadEnvelopeLoaderFilePreviewDTO](#)

Tags

- Files

Upload a file to given destination

```
POST /gobii/v1/files/{destinationType}
```

Description

Uploads an arbitrary file to the specified destination

Parameters

Type	Name	Description	Schema
Path	**destinationTy pe** —	Destination type where the file will be uploaded to	string
Query	**file** —	The file to be uploaded	string
Query	**fileName** —	Name of the file to be uploaded	string

Responses

HTTP Code	Description	Schema
200	successful operation	string

Tags

- Files

Detele a file by destination type

```
DELETE /gobii/v1/files/{destinationType}
```

Description

Deletes an arbitrary file from the specified destination

Parameters

Type	Name	Description	Schema
Path	**destinationTy pe** —	Destination type where the file will be delete from	string
Query	**fileName** —	Name of the file to be deleted	string

Responses

HTTP Code	Description	Schema
200	successful operation	string

Produces

- `text/plain`

Tags

- Files

Upload a file for a given job and destination type

```
POST /gobii/v1/files/{gobiiJobId}/{destinationType}
```

Description

Uploads the specified file for a specific job to the specified directory

Parameters

Type	Name	Description	Schema
Path	**destinationType** —	Destination type where the file will be uploaded to	string
Path	**gobiiJobId** —	ID of the Job that the file will be associated to	string
Query	**file** —	The file to be uploaded	string
Query	**fileName** —	Name of the file	string

Responses

HTTP Code	Description	Schema
200	successful operation	string

Tags

- Files

Download a job file from a given destination type

```
GET /gobii/v1/files/{gobiiJobId}/{destinationType}
```

Description

Downloads the specified file for a specific job from the specified directory

Parameters

Type	Name	Description	Schema
Path	destinationType —	Destination type where the file will be downloaded from	string
Path	gobiiJobId —	ID of the Job	string
Query	fileName —	Name of the file to be downloaded	string

Responses

HTTP Code	Description	Schema
200	successful operation	InputStreamResource

Tags

- Files

Create an extractor instruction file

```
POST /gobii/v1/instructions/extractor
```

Description

Creates extractor instruction file and then submits a new Job.

Parameters

Type	Name	Schema
Body	body —	PayloadEnvelopeExtractorInstructionFilesDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeExtractorInstructionFilesDTO

Tags

- Instructions

Get extractor job status by jobName

```
GET /gobii/v1/instructions/extractor/jobs/{jobName}
```

Description

Retrieves the extract job status along with other job details having the specified Job Name.

Parameters

Type	Name	Description	Schema
Path	**jobName** __	Name of the job	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeJobDTO

Tags

- Instructions

Download an extractor instructor file

```
GET /gobii/v1/instructions/extractor/{instructionFileName}
```

Description

Retrieves the extractor instruction file entity having the specified instruction file name.

Parameters

Type	Name	Description	Schema
Path	**instructionFileName** —	Name of the instruction file to be retrieved	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeExtractorInstructionFilesDTO

Tags

- Instructions

Create loader instruction file

```
POST /gobii/v1/instructions/loader
```

Description

Creates loader instruction file and then submits a new Job.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeLoaderInstructionFilesDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeLoaderInstructionFilesDTO

Tags

- Instructions

Get loader job status by job name

```
GET /gobii/v1/instructions/loader/jobs/{jobName}
```

Description

Gets the loading job status along with other job details having the specified Job Name.

Parameters

Type	Name	Description	Schema
Path	**jobName** __	Name of the job	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeJobDTO

Tags

- Instructions

Get a loader instruction file

```
GET /gobii/v1/instructions/loader/{instructionFileName}
```

Description

Gets the loader instruction file entity having the specified instruction file name.

Parameters

Type	Name	Description	Schema
Path	**instructionFileName** __	Name of the instruction file to be retrieved.	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeLoaderInstructionFilesDTO

Tags

- Instructions

Create a new job

```
POST /gobii/v1/jobs
```

Description

Creates a new job in the system.

Parameters

Type	Name	Schema
Body	**body** __	PayloadEnvelopeJobDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeJobDTO

Tags

- Jobs

List all jobs

```
GET /gobii/v1/jobs
```

Description

Lists all the existing jobs in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeJobDTO

Tags

- Jobs

Create DNA samples for a given job

```
POST /gobii/v1/jobs/dnasamples/{jobName}
```

Description

Creates DNA samples for a given Job having the specified name in the system.

Parameters

Type	Name	Description	Schema
Path	**jobName** __	Name of the Job that the DNA samples will be added to	string
Body	**body** __		PayloadEnvelopeDnaSampleDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeJobDTO

Tags

- Jobs

Get job details by job name

```
GET /gobii/v1/jobs/{jobName}
```

Description

Gets the Job entity having the specified name.

Parameters

Type	Name	Description	Schema
Path	**jobName** __	Name of the Job to be extracted	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeJobDTO

Tags

- Jobs

Update a job by jobName

```
PUT /gobii/v1/jobs/{jobName}
```

Description

Updates the Job entity having the specified jobName.

Parameters

Type	Name	Description	Schema
Path	**jobName** __	Name of the Job to be updated	string
Body	**body** __		PayloadEnvelopeJobDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeJobDTO

Create a manifest

```
POST /gobii/v1/manifests
```

Description

Creates a Manifest entity for GOBii system.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeManifestDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope ManifestDTO

Tags

• Manifests

List all manifests

```
GET /gobii/v1/manifests
```

Description

Lists an unfiltered list of all Manifest entities.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope ManifestDTO

Get manifests by manifestId

```
GET /gobii/v1/manifests/{manifestId}
```

Description

Gets the Manifest entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**manifestId** __	ID of the Manifest to be retrieved	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope ManifestDTO

Tags

• Manifests

Update a manifest

```
PUT /gobii/v1/manifests/{manifestId}
```

Description

Updates the Manifest entity having the specified manifestId.

Parameters

Type	Name	Description	Schema
Path	**manifestId** __	ID of the Manifest to be updated	integer (int32)
Body	**body** __		PayloadEnvelopeMa nifestDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope ManifestDTO

Tags

- Manifests

List all mapsets

```
GET /gobii/v1/maps
```

Description

Lists all the existing Mapsets in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MapsetDTO

Tags

- Maps

Create a new Mapset

```
POST /gobii/v1/mapsets
```

Description

Creates a new mapset in the system.

Parameters

Type	Name	Schema
Body	**body** __	PayloadEnvelopeMapsetDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MapsetDTO

Tags

- Maps

List all mapsets

```
GET /gobii/v1/mapsets
```

Description

List all the existing mapsets in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MapsetDTO

Tags

- Maps

Get mapsets by mapsetId

```
GET /gobii/v1/mapsets/{mapsetId}
```

Description

Gets the Mapset entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**mapsetId** __	ID of the Mapset to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MapsetDTO

Tags

- Maps

Update a mapset by mapsetId

```
PUT /gobii/v1/mapsets/{mapsetId}
```

Description

Updates the Mapset entity having the specified mapsetId.

Parameters

Type	Name	Description	Schema
Path	**mapsetId** —	ID of the Mapset to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeMa psetDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MapsetDTO

Tags

- Maps

List all markers from markers search

```
GET /gobii/v1/marker-search
```

Description

List Marker search results.

Parameters

Type	Name	Description	Schema
Query	**name** —	Name of the marker	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerDTO

Tags

- Markers

Create a new Marker group

```
POST /gobii/v1/markergroups
```

Description

Creates a new marker group in the system.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeMarkerGroupDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerGroupDTO

Tags

- Markers

List all markergroups

```
GET /gobii/v1/markergroups
```

Description

Lists all the existing marker groups in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerGroupDTO

Tags

- Markers

Gets all markers in a given markergroup

```
GET /gobii/v1/markergroups/{markerGroupId}
```

Description

Retrieves the Marker Group entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**markerGroup Id** __	ID of the marker group to be updated	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerGroupDTO

Tags

- Markers

Update a marker by markergroupId

> PUT /gobii/v1/markergroups/{markerGroupId}

Description

Updates the Marker Group entity having the specified markerGroupId.

Parameters

Type	Name	Description	Schema
Path	**markerGroup Id** —	ID of the Marker Group to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeMarkerGroupDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerGroupDTO

Tags

- Markers

Create a new marker

> POST /gobii/v1/markers

Description

Creates a new marker in the system.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeMarkerDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerDTO

Tags

- Markers

List all markers

```
GET /gobii/v1/markers
```

Description

Lists all the existing markers in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerDTO

Tags

- Markers

Get a marker by markerId

```
GET /gobii/v1/markers/{markerId}
```

Description

Retrieves the Marker entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**markerId** __	ID of the Marker to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerDTO

Tags

- Markers

Update a marker by markerId

```
PUT /gobii/v1/markers/{markerId}
```

Description

Updates the Marker entity having the specified markerId.

Parameters

Type	Name	Description	Schema
Path	**markerId** __	ID of the Marker to be updated	integer (int32)
Body	**body** __		PayloadEnvelopeMarkerDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelope MarkerDTO

Tags

- Markers

Create a new NameID

```
POST /gobii/v1/names/{entity}
```

Description

Retrieves a list of name/ID combination for a given entity and name list. For the list of entities supported see class GobiiEntityNameType.This is service is specifically implemented for these filter types: NAMES_BY_NAME_LIST - given a list of names, return the same list with the corresponding ID in the database. If name doesn't exist, ID will be 0.NAMES_BY_NAME_LIST_RETURN_EXISTS - given a list of names, return the list of names with ID that exists in the database.NAMES_BY_NAME_LIST_RETURN_ABSENT - given a list of names, return the list of names that doesn't exist in the database with 0 as the IDFilter value varies per entity. This can be cv group name, project ID, platform ID, etc.Example use case: entity = CV; filterType = NAMES_BY_NAME_LIST; filterValue = germplasm_typeResult will be a list of CV terms with ID having germplasm_type as the cv group

Parameters

Type	Name	Description	Schema
Path	**entity** —	The entity to be retrieved	string
Query	**filterType** —	The filter type for the name list	string
Query	**filterValue** —	The value for the filter type	string
Body	**body** —		PayloadEnvelopeNameIdDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeNameIdDTO

Tags

- Names

List all NameIds

```
GET /gobii/v1/names/{entity}
```

Description

List of name/ID combination for a given entity. For the list of entities supported see class GobiiEntityNameType.List can further be filtered out by specifying the filter type and value.For the list of filter types supported see class GobiiFilterType.Example use case: entity = CV; filterType =

50

NAMES_BY_TYPE_NAME; filterValue = statusResult will be a list of CV terms having status as the cv group

Parameters

Type	Name	Description	Schema
Path	**entity** —	The entity to be retrieved	string
Query	**filterType** —	The filter type for the name list	string
Query	**filterValue** —	The value for the filter type	string

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeNameIdDTO

Tags

- Names

Create a new organization

```
POST /gobii/v1/organizations
```

Description

Creates a new organization in the system.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeOrganizationDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeOrganizationDTO

List all organizations

```
GET /gobii/v1/organizations
```

Description

List all the existing organizations in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeOrganizationDTO

Tags

- Organizations

Get organization by organizationId

```
GET /gobii/v1/organizations/{organizationId}
```

Description

Retrieves the Organization entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**organizationId** __	ID of the Organization to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeOrganizationDTO

- Organizations

Update an organization by organizationId

```
PUT /gobii/v1/organizations/{organizationId}
```

Description

Updates the Organization entity having the specified organizationId.

Parameters

Type	Name	Description	Schema
Path	**organizationId** —	ID of the Organization to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeOrganizationDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeOrganizationDTO

Tags

- Organizations

ping

```
POST /gobii/v1/ping
```

Description

Pings the GDB Web server.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopePingDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopePingDTO

Tags

- Ping

Create a new platform

```
POST /gobii/v1/platforms
```

Description

Creates a new Platform in the system.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopePlatformDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopePlatformDTO

Tags

- Platforms

List all platforms

```
GET /gobii/v1/platforms
```

Description

List all the existing platforms in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopePlatformDTO

Tags

- Platforms

GET /platforms/protocols/{vendorProtocolId}

```
GET /gobii/v1/platforms/protocols/{vendorProtocolId}
```

Description

Gets the Platform entity having the specified Vendor Protocol ID.

Parameters

Type	Name	Description	Schema
Path	**vendorProtocolId** ___	ID of the vendor protocol	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopePlatformDTO

Tags

- Platforms

Get a platform by platformId

```
GET /gobii/v1/platforms/{platformId}
```

Description

Retrieves the Platform entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**platformId** —	ID of the Platform to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopePlatformDTO

Tags

- Platforms

Update a platform by platformId

```
PUT /gobii/v1/platforms/{platformId}
```

Description

Updates the Platform entity having the specified platformId.

Parameters

Type	Name	Description	Schema
Path	**platformId** —	ID of the Platform to be updated	integer (int32)
Body	**body** —		PayloadEnvelopePlatformDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopePlatformDTO

- Platforms

Create a new project

```
POST /gobii/v1/projects
```

Description

Creates a new project in the system.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeProjectDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeProjectDTO

Tags

- Projects

List all projects

```
GET /gobii/v1/projects
```

Description

List all the existing projects in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeProjectDTO

Get a project by projectId

```
GET /gobii/v1/projects/{projectId}
```

Description

Retrieves the Project entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**projectId** —	ID of the Project to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeProjectDTO

Tags

- Projects

Update a project by projectId

```
PUT /gobii/v1/projects/{projectId}
```

Description

Updates the Project entity having the specified projectId.

Parameters

Type	Name	Description	Schema
Path	**projectId** —	ID of the Project to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeProjectDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeProjectDTO

Tags

- Projects

Create a new protocol

```
POST /gobii/v1/protocols
```

Description

Creates a new Protocol in the system.

Parameters

Type	Name	Schema
Body	**body** __	PayloadEnvelopeProtocolDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeProtocolDTO

Tags

- Protocols

List all protocols

```
GET /gobii/v1/protocols
```

Description

List all the existing protocols in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeProtocolDTO

Tags

- Protocols

Get a protocol by protocolId

```
GET /gobii/v1/protocols/{protocolId}
```

Description

Retrieves the Protocol entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**protocolId** __	ID of the Protocol to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeProtocolDTO

Tags

- Protocols

Update a protocol by protocolId

```
PUT /gobii/v1/protocols/{protocolId}
```

Description

Updates the Protocol entity having the specified protocolId.

Parameters

Type	Name	Description	Schema
Path	**protocolId** —	ID of the Protocol to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeProtocolDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopePprotocolDTO

Tags

- Protocols

Create new vendors for a given protocolId

```
POST /gobii/v1/protocols/{protocolId}/vendors
```

Description

Creates a new Vendor Protocol in the system for specified Protocol ID.

Parameters

Type	Name	Description	Schema
Path	**protocolId** —	ID of the Protocol	integer (int32)
Body	**body** —		PayloadEnvelopeOrganizationDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeOrganizationDTO

Tags

- Protocols.vendor

List all vendors for a given protocolId

```
GET /gobii/v1/protocols/{protocolId}/vendors
```

Description

List all the vendor protocols given protocolId in the system.

Parameters

Type	Name	Description	Schema
Path	protocolId —	ID of the Protocol ID	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeOrganizationDTO

Tags

- Protocols

Update vendors for a given protocolId

```
PUT /gobii/v1/protocols/{protocolId}/vendors
```

Description

Updates the Vendor Protocol entity having the specified protocolId.

Parameters

Type	Name	Description	Schema
Path	protocolId —	ID of the Protocol	integer (int32)
Body	body —		PayloadEnvelopeOrganizationDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeOrganizationDTO

Tags

- Protocols

Create a new reference

```
POST /gobii/v1/references
```

Description

Creates a new reference in the system.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeReferenceDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeReferenceDTO

Tags

- References

List all references

```
GET /gobii/v1/references
```

Description

Lists all the existing references in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeReferenceDTO

Tags

- References

GET /references/{referenceId}

```
GET /gobii/v1/references/{referenceId}
```

Description

Gets the Reference entity having the specified ID.

Parameters

Type	Name	Description	Schema
Path	**referenceId** __	ID of the Reference to be extracted	integer (int32)

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeReferenceDTO

Tags

- References

Update a reference by referenceId

```
PUT /gobii/v1/references/{referenceId}
```

Description

Updates the Reference entity having the specified referenceId.

Parameters

Type	Name	Description	Schema
Path	**referenceId** —	ID of the Reference to be updated	integer (int32)
Body	**body** —		PayloadEnvelopeReferenceDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeReferenceDTO

Tags

- References

/restprofiles/

```
PUT /gobii/v1/restprofiles
```

Description

Update REST profiles When the Header of the payload envelope for a resource contains maxGet, maxPost, and maxPut values, this resource provides a means to update the max for a given rest resource ID and for a given HTTP verb. The values are transient in the sense that they will be confined only to a specific web service deployment. They are stored in the web service configurationdocument

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeRestProfileDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeRestProfileDTO

Tags

- RestProfiles

Definitions

AnalysisDTO

Name	Schema
algorithm —	string
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
analysisDescription —	string
analysisId —	integer (int32)
analysisName —	string
anlaysisTypeId —	integer (int32)
createdBy —	integer (int32)
createdDate —	string (date-time)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
id —	integer (int32)
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
parameters —	< EntityPropertyDTO > array
program —	string
programVersion —	string

Name	Schema
referenceId —	integer (int32)
sourceName —	string
sourceUri —	string
sourceVersion —	string
statusId —	integer (int32)
timeExecuted —	string (date-time)

ConfigSettingsDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
id —	integer (int32)
maxUploadSizeMbytes —	integer (int32)
serverCapabilities —	< string, boolean > map
serverConfigs —	< string, ServerConfigItem > map

ContactDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
code —	string
contactId —	integer (int32)
createdBy —	integer (int32)
createdDate —	string (date-time)
email —	string

Name	Schema
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
firstName —	string
id —	integer (int32)
lastName —	string
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
organizationId —	integer (int32)
roles —	< integer (int32) > array
userName —	string

CvDTO

Name	Schema
abbreviation —	string
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
cvId —	integer (int32)
definition —	string
entityStatus —	integer (int32)
group —	integer (int32)
groupId —	integer (int32)
groupType —	integer (int32)

Name	Schema
id —	integer (int32)
rank —	integer (int32)
term —	string
xrefId —	integer (int32)

CvGroupDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
cvGroupId —	integer (int32)
definition —	string
groupType —	integer (int32)
id —	integer (int32)
name —	string

DataSetDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
analysesIds —	< integer (int32) > array
callingAnalysisId —	integer (int32)
callingAnalysisName —	string
createdBy —	integer (int32)
createdDate —	string (date-time)
dataFile —	string

Name	Schema
dataSetId —	integer (int32)
dataTable —	string
datasetName —	string
datatypeId —	integer (int32)
datatypeName —	string
experimentId —	integer (int32)
experimentName —	string
id —	integer (int32)
jobId —	integer (int32)
jobStatusId —	integer (int32)
jobStatusName —	string
jobSubmittedDate —	string (date-time)
jobTypeId —	integer (int32)
jobTypeName —	string
loaderFirstName —	string
loaderLastName —	string
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
piContactId —	integer (int32)
piEmail —	string
piFirstName —	string

Name	Schema
piLastName —	string
platformId —	integer (int32)
platformName —	string
projectId —	integer (int32)
projectName —	string
protocolId —	integer (int32)
protocolName —	string
qualityFile —	string
qualityTable —	string
scores —	< integer (int32) > array
statusId —	integer (int32)
totalMarkers —	integer (int32)
totalSamples —	integer (int32)

DisplayDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
columnName —	string
createdBy —	integer (int32)
createdDate —	string (date-time)
displayId —	integer (int32)
displayName —	string

Name	Schema
displayRank —	integer (int32)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
id —	integer (int32)
includeDetailsList —	boolean
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
tableName —	string
tableNamesWithColDisplay —	< string, < TableColDisplay > array > map

DnaSampleDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
analysisMethodName —	string
dnaRunId —	integer (int32)
dnaSampleId —	integer (int32)
dnaSampleName —	string
dnaSampleNum —	integer (int32)
dnarunName —	string
experimentId —	integer (int32)
externalCode —	string

Name	Schema
germplasmExternalCode —	string
germplasmId —	integer (int32)
germplasmName —	string
id —	integer (int32)
markerCount —	integer (int32)
projectId —	integer (int32)

EntityPropertyDTO

Name	Schema
entityIdId —	integer (int32)
propertyId —	integer (int32)
propertyName —	string
propertyValue —	string

EntityStatsDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
count —	integer (int32)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
entityStateDateType —	enum (INSERT_UPDATE, INSERT_ONLY)
id —	integer (int32)

Name	Schema
lastModified —	string (date-time)

ExperimentDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
createdBy —	integer (int32)
createdDate —	string (date-time)
datasets —	< DataSetDTO > array
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
experimentCode —	string
experimentDataFile —	string
experimentId —	integer (int32)
experimentName —	string
id —	integer (int32)
manifestId —	integer (int32)
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
projectId —	integer (int32)
statusId —	integer (int32)
vendorProtocolId —	integer (int32)

ExtractorInstructionFilesDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
gobiiExtractorInstructions —	< GobiiExtractorInstruction > array
id —	integer (int32)
instructionFileName —	string
jobId —	string

GobiiDataSetExtract

Name	Schema
accolate —	boolean
dataSet —	PropNameId
extractDestinationDirectory —	string
extractedFiles —	< file > array
gobiiDatasetType —	PropNameId
gobiiExtractFilterType —	enum (UNKNOWN, WHOLE_DATASET, BY_MARKER, BY_SAMPLE)
gobiiFileType —	enum (HAPMAP, FLAPJACK, GENERIC, VCF, META_DATA)
gobiiJobStatus —	enum (CV_PROGRESSSTATUS_PENDING, CV_PROGRESSSTATUS_INPROGRESS, CV_PROGRESSSTATUS_COMPLETED, CV_PROGRESSSTATUS_FAILED, CV_PROGRESSSTATUS_VALIDATION, CV_PROGRESSSTATUS_DIGEST, CV_PROGRESSSTATUS_TRANSFORMATION, CV_PROGRESSSTATUS_METADATALOAD, CV_PROGRESSSTATUS_MATRIXLOAD, CV_PROGRESSSTATUS_ABORTED, CV_PROGRESSSTATUS_METADATAEXTRACT, CV_PROGRESSSTATUS_FINALASSEMBLY, CV_PROGRESSSTATUS_QCPROCESSING, CV_PROGRESSSTATUS_NOSTATUS)

Name	Schema
gobiiSampleListType —	enum (GERMPLASM_NAME, EXTERNAL_CODE, DNA_SAMPLE)
listFileName —	string
logMessage —	string
markerGroups —	< PropNameId > array
markerList —	< string > array
platforms —	< PropNameId > array
principleInvestigator —	PropNameId
project —	PropNameId
sampleList —	< string > array

GobiiExtractorInstruction

Name	Schema
contactEmail —	string
contactId —	integer (int32)
dataSetExtracts —	< GobiiDataSetExtract > array
gobiiCropType —	string
mapsetIds —	< integer (int32) > array
qcCheck —	boolean

GobiiFile

Name	Schema
createSource —	boolean
delimiter —	string

Name	Schema
destination —	string
gobiiFileType —	enum (HAPMAP, FLAPJACK, GENERIC, VCF, META_DATA)
requireDirectoriesToExist —	boolean
source —	string

GobiiFileColumn

Name	Schema
constantValue —	string
dataSetOrientationType —	enum (SAMPLE_FAST, MARKER_FAST)
dataSetType —	enum (NUCLEOTIDE_2_LETTER, IUPAC, SSR_ALLELE_SIZE, DOMINANT_NON_NUCLEOTIDE, CO_DOMINANT_NON_NUCLEOTIDE, VCF)
filterFrom —	string
filterTo —	string
findText —	string
getcCoord —	integer (int32)
getrCoord —	integer (int32)
gobiiColumnType —	enum (CSV_COLUMN, CSV_ROW, CSV_BOTH, VCF_SAMPLE, VCF_MARKER, VCF_VARIANT, VCF_METADATA, VCF_INFO, CONSTANT, AUTOINCREMENT, VCF_MARKER_POS)
index —	string
metaDataId —	string
name —	string
replaceText —	string
subcolumn —	boolean

Name	Schema
subcolumnDelimiter —	string

GobiiLoaderInstruction

Name	Schema
columnsByTableName —	< string, < GobiiFileColumn > array > map
contactEmail —	string
contactId —	integer (int32)
dataSet —	PropNameId
dataSetId —	integer (int32)
datasetType —	PropNameId
experiment —	PropNameId
gobiiCropType —	string
gobiiFile —	GobiiFile
gobiiFileColumns —	< GobiiFileColumn > array
gobiiJobStatus —	enum (CV_PROGRESSSTATUS_PENDING, CV_PROGRESSSTATUS_INPROGRESS, CV_PROGRESSSTATUS_COMPLETED, CV_PROGRESSSTATUS_FAILED, CV_PROGRESSSTATUS_VALIDATION, CV_PROGRESSSTATUS_DIGEST, CV_PROGRESSSTATUS_TRANSFORMATION, CV_PROGRESSSTATUS_METADATALOAD, CV_PROGRESSSTATUS_MATRIXLOAD, CV_PROGRESSSTATUS_ABORTED, CV_PROGRESSSTATUS_METADATAEXTRACT, CV_PROGRESSSTATUS_FINALASSEMBLY, CV_PROGRESSSTATUS_QCPROCESSING, CV_PROGRESSSTATUS_NOSTATUS)
jobPayloadType —	enum (CV_PAYLOADTYPE_SAMPLES, CV_PAYLOADTYPE_MARKERS, CV_PAYLOADTYPE_MATRIX, CV_PAYLOADTYPE_MARKERSAMPLES, CV_PAYLOADTYPE_ALLMETA)

Name	Schema
logMessage —	string
mapset —	PropNameId
platform —	PropNameId
project —	PropNameId
qcCheck —	boolean
table —	string
vcfParameters —	VcfParameters

Header

Name	Schema
cropType —	string
dtoHeaderAuth —	HeaderAuth
gobiiProcessType —	enum (CREATE, READ, UPDATE, DELETE, NONE)
gobiiVersion —	string
maxGet —	integer (int32)
maxPost —	integer (int32)
maxPut —	integer (int32)
pagination —	Pagination
status —	Status

HeaderAuth

Name	Schema
gobiiCropType —	string

Name	Schema
password —	string
token —	string
userName —	string

HeaderStatusMessage

Name	Schema
gobiiStatusLevel —	enum (ERROR, VALIDATION, WARNING, INFO, OK)
gobiiValidationStatusType —	enum (NONE, UNKNOWN, VALIDATION_COMPOUND_UNIQUE, VALIDATION_NOT_UNIQUE, NONEXISTENT_FK_ENTITY, BAD_REQUEST, MISSING_REQUIRED_VALUE, ENTITY_DOES_NOT_EXIST, ENTITY_ALREADY_EXISTS, UNKNOWN_ENUM_VALUE, RESOURCE_LIMIT)
message —	string

InputStream

Type : object

InputStreamResource

Name	Schema
description —	string
file —	file
filename —	string
inputStream —	InputStream
open —	boolean
readable —	boolean
uri —	string (uri)

Name	Schema
url —	string (url)

JobDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
datasetIds —	< integer (int32) > array
id —	integer (int32)
jobId —	integer (int32)
jobName —	string
message —	string
payloadType —	string
status —	string
submittedBy —	integer (int32)
submittedDate —	string (date-time)
type —	string

Link

Name	Schema
description —	string
href —	string
methods —	< enum (GET, POST, PUT, PATCH, DELETE, OPTIONS) > array

LinkCollection

Name	Schema
exploreLinksPerDataItem —	< < [Link](#) > array > array
linksPerDataItem —	< [Link](#) > array

LoaderFilePreviewDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
directoryName —	string
fileList —	< string > array
filePreview —	< < string > array > array
id —	integer (int32)
previewFileName —	string

LoaderInstructionFilesDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
gobiiLoaderInstructions —	< [GobiiLoaderInstruction](#) > array
id —	integer (int32)
instructionFileName —	string

ManifestDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
code —	string
createdBy —	integer (int32)

Name	Schema
createdDate —	string (date-time)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
filePath —	string
id —	integer (int32)
manifestId —	integer (int32)
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
name —	string

MapsetDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
code —	string
createdBy —	integer (int32)
createdDate —	string (date-time)
description —	string
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
id —	integer (int32)

Name	Schema
mapType —	integer (int32)
mapsetId —	integer (int32)
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
name —	string
properties —	< EntityPropertyDTO > array
referenceId —	integer (int32)
statusId —	integer (int32)

MarkerDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
alts —	< integer (int32) > array
code —	string
id —	integer (int32)
markerId —	integer (int32)
markerName —	string
platformId —	integer (int32)
platformName —	string
ref —	string
referenceId —	integer (int32)
sequence —	string

Name	Schema
status —	integer (int32)
strand —	integer (int32)
strandId —	integer (int32)
variantId —	integer (int32)

MarkerGroupDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
code —	string
createdBy —	integer (int32)
createdDate —	string (date-time)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
germplasmGroup —	string
id —	integer (int32)
markerGroupId —	integer (int32)
markerMap —	< string, < string > array > map
markers —	< MarkerGroupMarkerDTO > array
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
name —	string

Name	Schema
statusId —	integer (int32)

MarkerGroupMarkerDTO

Name	Schema
favorableAllele —	string
gobiiProcessType —	enum (CREATE, READ, UPDATE, DELETE, NONE)
markerExists —	boolean
markerId —	integer (int32)
markerName —	string
platformId —	integer (int32)
platformName —	string

NameIdDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
entityLasetModified —	string (date-time)
fkId —	integer (int32)
gobiiEntityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
gobiiFkEntityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)

Name	Schema
id —	integer (int32)
name —	string
queryObject —	object

OrganizationDTO

Name	Schema
address —	string
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
createdBy —	integer (int32)
createdDate —	string (date-time)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
id —	integer (int32)
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
name —	string
organizationId —	integer (int32)
statusId —	integer (int32)
vendorProtocols —	< VendorProtocolDTO > array
website —	string

Pagination

Name	Schema
currentPage —	integer (int32)
pageSize —	integer (int32)
pagedQueryId —	string
queryTime —	string (date-time)
totalPages —	integer (int32)

Payload

Name	Schema
data —	< object > array
linkCollection —	LinkCollection

PayloadAnalysisDTO

Name	Schema
data —	< AnalysisDTO > array
linkCollection —	LinkCollection

PayloadConfigSettingsDTO

Name	Schema
data —	< ConfigSettingsDTO > array
linkCollection —	LinkCollection

PayloadContactDTO

Name	Schema
data —	< ContactDTO > array
linkCollection —	LinkCollection

PayloadCvDTO

Name	Schema
data —	< CvDTO > array
linkCollection —	LinkCollection

PayloadCvGroupDTO

Name	Schema
data —	< CvGroupDTO > array
linkCollection —	LinkCollection

PayloadDataSetDTO

Name	Schema
data —	< DataSetDTO > array
linkCollection —	LinkCollection

PayloadDisplayDTO

Name	Schema
data —	< DisplayDTO > array
linkCollection —	LinkCollection

PayloadDnaSampleDTO

Name	Schema
data —	< DnaSampleDTO > array
linkCollection —	LinkCollection

PayloadEntityStatsDTO

Name	Schema
data —	< EntityStatsDTO > array
linkCollection —	LinkCollection

PayloadEnvelope

Name	Schema
header —	Header
payload —	PayloadObject

PayloadEnvelopeAnalysisDTO

Name	Schema
header —	Header
payload —	PayloadAnalysisDTO

PayloadEnvelopeConfigSettingsDTO

Name	Schema
header —	Header
payload —	PayloadConfigSettingsDTO

PayloadEnvelopeContactDTO

Name	Schema
header —	Header
payload —	PayloadContactDTO

PayloadEnvelopeCvDTO

Name	Schema
header —	Header
payload —	PayloadCvDTO

PayloadEnvelopeCvGroupDTO

Name	Schema
header —	Header
payload —	PayloadCvGroupDTO

PayloadEnvelopeDataSetDTO

Name	Schema
header —	Header
payload —	PayloadDataSetDTO

PayloadEnvelopeDisplayDTO

Name	Schema
header —	Header
payload —	PayloadDisplayDTO

PayloadEnvelopeDnaSampleDTO

Name	Schema
header —	Header
payload —	PayloadDnaSampleDTO

PayloadEnvelopeEntityStatsDTO

Name	Schema
header —	Header
payload —	PayloadEntityStatsDTO

PayloadEnvelopeExperimentDTO

Name	Schema
header —	Header
payload —	PayloadExperimentDTO

PayloadEnvelopeExtractorInstructionFilesDTO

Name	Schema
header —	Header
payload —	PayloadExtractorInstructionFilesDTO

PayloadEnvelopeJobDTO

Name	Schema
header —	Header
payload —	PayloadJobDTO

PayloadEnvelopeLoaderFilePreviewDTO

Name	Schema
header —	Header
payload —	PayloadLoaderFilePreviewDTO

PayloadEnvelopeLoaderInstructionFilesDTO

Name	Schema
header —	Header
payload —	PayloadLoaderInstructionFilesDTO

PayloadEnvelopeManifestDTO

Name	Schema
header —	Header
payload —	PayloadManifestDTO

PayloadEnvelopeMapsetDTO

Name	Schema
header —	Header
payload —	PayloadMapsetDTO

PayloadEnvelopeMarkerDTO

Name	Schema
header —	Header
payload —	PayloadMarkerDTO

PayloadEnvelopeMarkerGroupDTO

Name	Schema
header —	Header
payload —	PayloadMarkerGroupDTO

PayloadEnvelopeNameIdDTO

Name	Schema
header —	Header
payload —	PayloadNameIdDTO

PayloadEnvelopeOrganizationDTO

Name	Schema
header —	Header
payload —	PayloadOrganizationDTO

PayloadEnvelopePingDTO

Name	Schema
header —	Header
payload —	PayloadPingDTO

PayloadEnvelopePlatformDTO

Name	Schema
header —	Header
payload —	PayloadPlatformDTO

PayloadEnvelopeProjectDTO

Name	Schema
header —	Header
payload —	PayloadProjectDTO

PayloadEnvelopeProtocolDTO

Name	Schema
header —	Header
payload —	PayloadProtocolDTO

PayloadEnvelopeReferenceDTO

Name	Schema
header —	Header
payload —	PayloadReferenceDTO

PayloadEnvelopeRestProfileDTO

Name	Schema
header —	Header
payload —	PayloadRestProfileDTO

PayloadExperimentDTO

Name	Schema
data —	< ExperimentDTO > array
linkCollection —	LinkCollection

PayloadExtractorInstructionFilesDTO

Name	Schema
data —	< ExtractorInstructionFilesDTO > array
linkCollection —	LinkCollection

PayloadJobDTO

Name	Schema
data —	< JobDTO > array
linkCollection —	LinkCollection

PayloadLoaderFilePreviewDTO

Name	Schema
data —	< LoaderFilePreviewDTO > array
linkCollection —	LinkCollection

PayloadLoaderInstructionFilesDTO

Name	Schema
data —	< LoaderInstructionFilesDTO > array
linkCollection —	LinkCollection

PayloadManifestDTO

Name	Schema
data —	< ManifestDTO > array
linkCollection —	LinkCollection

PayloadMapsetDTO

Name	Schema
data —	< [MapsetDTO](#) > array
linkCollection —	[LinkCollection](#)

PayloadMarkerDTO

Name	Schema
data —	< [MarkerDTO](#) > array
linkCollection —	[LinkCollection](#)

PayloadMarkerGroupDTO

Name	Schema
data —	< [MarkerGroupDTO](#) > array
linkCollection —	[LinkCollection](#)

PayloadNameIdDTO

Name	Schema
data —	< [NameIdDTO](#) > array
linkCollection —	[LinkCollection](#)

PayloadObject

Name	Schema
data —	< object > array
linkCollection —	[LinkCollection](#)

PayloadOrganizationDTO

Name	Schema
data —	< OrganizationDTO > array
linkCollection —	LinkCollection

PayloadPingDTO

Name	Schema
data —	< PingDTO > array
linkCollection —	LinkCollection

PayloadPlatformDTO

Name	Schema
data —	< PlatformDTO > array
linkCollection —	LinkCollection

PayloadProjectDTO

Name	Schema
data —	< ProjectDTO > array
linkCollection —	LinkCollection

PayloadProtocolDTO

Name	Schema
data —	< ProtocolDTO > array
linkCollection —	LinkCollection

PayloadReferenceDTO

Name	Schema
data —	< ReferenceDTO > array
linkCollection —	LinkCollection

PayloadRestProfileDTO

Name	Schema
data —	< RestProfileDTO > array
linkCollection —	LinkCollection

PingDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
dbMetaData —	< string > array
id —	integer (int32)
pingResponses —	< string > array

PlatformDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
createdBy —	integer (int32)
createdDate —	string (date-time)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
id —	integer (int32)

Name	Schema
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
platformCode —	string
platformDescription —	string
platformId —	integer (int32)
platformName —	string
properties —	< EntityPropertyDTO > array
statusId —	integer (int32)
typeId —	integer (int32)

ProjectDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
createdBy —	integer (int32)
createdDate —	string (date-time)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
id —	integer (int32)
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
piContact —	integer (int32)

Name	Schema
principleInvestigators —	< string, string > map
projectCode —	string
projectDescription —	string
projectId —	integer (int32)
projectName —	string
projectStatus —	integer (int32)
properties —	< EntityPropertyDTO > array

PropNameId

Name	Schema
id —	integer (int32)
name —	string

ProtocolDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
createdBy —	integer (int32)
createdDate —	string (date-time)
description —	string
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
id —	integer (int32)

Name	Schema
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
name —	string
platformId —	integer (int32)
props —	< EntityPropertyDTO > array
protocolId —	integer (int32)
status —	integer (int32)
typeId —	integer (int32)
vendorProtocols —	< VendorProtocolDTO > array

ReferenceDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
createdBy —	integer (int32)
createdDate —	string (date-time)
entityNameType —	enum (UNKNOWN, ANALYSIS, CONTACT, DATASET, CV, CVGROUP, PROJECT, ORGANIZATION, PLATFORM, MANIFEST, MAPSET, MARKER_GROUP, EXPERIMENT, REFERENCE, ROLE, DISPLAY, MARKER, PROTOCOL, VENDOR_PROTOCOL, DNASAMPLE, LINKAGE_GROUP, DNARUN, GERMPLASM, JOB)
filePath —	string
id —	integer (int32)
link —	string
modifiedBy —	integer (int32)

Name	Schema
modifiedDate —	string (date-time)
name —	string
referenceId —	integer (int32)
version —	string

RestProfileDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
id —	integer (int32)
max —	integer (int32)
restMethodType —	enum (GET, POST, PUT, PATCH, DELETE, OPTIONS)

Name	Schema
restResourceId —	enum (GOBII_AUTH, GOBII_PING, GOBII_PROJECTS, GOBII_NAME_ID_LIST_DEPRECATED, GOBII_NAMES, GOBII_FILE_LOAD, GOBII_FILE_LOAD_INSTRUCTIONS, GOBII_FILE_EXTRACTOR_INSTRUCTIONS, GOBII_FILE_EXTRACTOR_JOBS, GOBII_FILE_LOADER_JOBS, GOBII_DISPLAY, GOBII_CV, GOBII_CVGROUP, GOBII_CONTACTS, GOBII_CONTACT_SEARCH, GOBII_ORGANIZATION_SEARCH, GOBII_REFERENCE, GOBII_EXPERIMENTS, GOBII_DATASETS, GOBII_DATASETTYPES, GOBII_ANALYSIS, GOBII_MARKERGROUP, GOBII_MANIFEST, GOBII_ORGANIZATION, GOBII_PLATFORM, GOBII_MAPSET, GOBII_CONFIGSETTINGS, GOBII_ROLES, GOBII_MARKERS, GOBII_MARKER_SEARCH, GOBII_CALLS, GOBII_GERMPLASM, GOBII_OBSERVATION_VARIABLES, GOBII_PROTOCOL, GOBII_VENDORS, GOBII_FILES, GOBII_JOB, GOBII_ENTITIES, GOBII_COUNT, GOBII_LAST_MODIFIED, GOBII_VERTICES, GOBII_VALUES, GOBII_REST_PROFILES, GOBII_DNASAMPLES, GOBII_DNARUN, GOBII_LINKAGEGROUP, GOBII_JOB_DNASAMPLE, BRAPI_ALLELE_MATRIX_SEARCH, BRAPI_ALLELE_MATRIX_SEARCH_STATUS, BRAPI_ALLELE_MATRICES, KDC_START, KDC_STATUS, KDC_DOWNLOAD, KDC_PURGE, BRAPI_LOGIN, BRAPI_STUDIES_SEARCH, BRAPI_STUDIES, BRAPI_CALLS)
templateParameter —	string

ServerConfigItem

Name	Schema
confidentialityNotice —	string
contextRoot —	string
domain —	string
fileLocations —	< string, string > map
gobiiCropType —	string
port —	integer (int32)

Status

Name	Schema
statusMessages —	< HeaderStatusMessage > array
statusMessagesByCode —	< string, string > map
succeeded —	boolean

TableColDisplay

Name	Schema
columnName —	string
displayId —	integer (int32)
displayName —	string
rank —	integer (int32)

VcfParameters

Name	Schema
maf —	number (float)
minDp —	number (float)
minQ —	number (float)
removeIndels —	boolean
toIupac —	boolean

VendorProtocolDTO

Name	Schema
allowedProcessTypes —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
id —	integer (int32)

Name	Schema
name —	string
organizationId —	integer (int32)
protocolId —	integer (int32)
status —	integer (int32)
vendorProtocolId —	integer (int32)