master document for generating PDF

= GOBii API

Table of Contents

Overview

The GOBii 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 GOBii 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 : localhost:8081
BasePath : /api
Schemes : HTTP

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

Creates analysis entity

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

Description

Creates an analysis entity for GOBii system. Consult the Request/Response Section before attempting to use this call.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeAnalysisDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeAnalysisDTO

Gets all the analysis

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

Description

Retrieves an unfiltered list of all Analysis entities. Consult the Request/Response Section before attempting to use this call.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeAnalysisDTO

Gets analysis by Analysis ID

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

Description

Retrieves the Analysis entity having the specified ID. Consult the Request/Response Section before attempting to use this call.

Parameters

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

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeAnalysisDTO

Updates analysis by Analysis ID

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

Description

Updates the Analysis entity having the specified analysisId. Consult the Request/Response Section before attempting to use this call.

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](#)

Authentication service for accessing GOBii System

```
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. this header and value be included in the request headers for subsequent requests. The token value is also supplied in the dtoHeaderAuth object.

Parameters

Type	Name	Schema
Body	**body** —	string

Responses

HTTP Code	Description	Schema
200	successful operation	string

Gets the configuration settings for GOBii system

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

Description

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](#)

/contacts-search

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

Description

Gets contacts by searching 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

Adds new contact to GOBii system

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

Description

Adds new contact.

Parameters

Type	Name	Schema
Body	**body** —	PayloadEnvelopeContactDTO

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeContactDTO

Gets all the contacts in the GOBii system

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

Description

Gets all contacts.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeContactDTO

Gets contacts by contact ID

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

Description

Gets 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

Updates contacts by contact ID

PUT /gobii/v1/contacts/{contactId}

Description

Updates contacts for a given 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

Gets contacts by email id

GET /gobii/v1/contacts/{email}

Description

Gets contacts 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

/cvs

```
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

/cvs

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

Description

Gets all cvs in the system.

Responses

HTTP Code	Description	Schema
200	successful operation	PayloadEnvelopeCvDTO

/cvs

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

Description

Gets 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

/cvs

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

Description

Updates cv's 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

/cvs

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

Description

Deletes cv by the 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

/cvs/{groupName:[a-zA-Z_]+}

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

Description

Gets 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

Gets REST profiles of GOBii system

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

Description

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

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)

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)
id —	integer (int32)
modifiedBy —	integer (int32)
modifiedDate —	string (date-time)
parameters —	< EntityPropertyDTO > array
program —	string
programVersion —	string
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

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

EntityPropertyDTO

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

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
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)

Name	Schema
message —	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

Pagination

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

PayloadAnalysisDTO

Name	Schema
data —	< AnalysisDTO > array

Name	Schema
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

PayloadEnvelopeAnalysisDTO

Name	Schema
header —	Header
payload —	PayloadAnalysisDTO

PayloadEnvelopeConfigSettingsDTO

Name	Schema
header —	Header

Name	Schema
payload —	PayloadConfigSettingsDTO

PayloadEnvelopeContactDTO

Name	Schema
header —	Header
payload —	PayloadContactDTO

PayloadEnvelopeCvDTO

Name	Schema
header —	Header
payload —	PayloadCvDTO

PayloadEnvelopeRestProfileDTO

Name	Schema
header —	Header
payload —	PayloadRestProfileDTO

PayloadRestProfileDTO

Name	Schema
data —	< RestProfileDTO > array
linkCollection —	LinkCollection

RestProfileDTO

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

Name	Schema
id —	integer (int32)
max —	integer (int32)
restMethodType —	enum (GET, POST, PUT, PATCH, DELETE, OPTIONS)
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

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

Status

Name	Description	Schema
statusMessages —		< HeaderStatusMessage > array
statusMessagesByCode —		< string, string > map
succeeded —	**Default** : false	boolean