

master document for generating PDF

= GOBii API

# Table of Contents

Overview.....	1
Version information .....	1
License information .....	1
URI scheme .....	1
Overview.....	1
Language and Framework.....	1
The DTO Payload .....	2
Request/Response Structure .....	2
Authentication .....	4
Error Handling .....	4
Paths .....	5
/analyses.....	5
/analyses.....	5
/analyses.....	5
/auth.....	6
/configsettings .....	7
/restprofiles .....	7
Definitions .....	8
AnalysisDTO .....	8
ConfigSettingsDTO .....	9
EntityPropertyDTO .....	9
Header .....	9
HeaderAuth .....	10
HeaderStatusMessage .....	10
Link .....	11
LinkCollection.....	11
Pagination .....	11
PayloadAnalysisDTO .....	11
PayloadConfigSettingsDTO .....	11
PayloadEnvelopeAnalysisDTO .....	12
PayloadEnvelopeConfigSettingsDTO .....	12
PayloadEnvelopeRestProfileDTO .....	12
PayloadRestProfileDTO .....	12
RestProfileDTO .....	12
ServerConfigItem.....	13
Status.....	14

# 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 framework. 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;
    }

```

The request and response are almost always wrapped in a type-parameterized `PayloadEnvelope`. 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 entities).

For a client of this application, however, the strongly-typed nature of the implementation is irrelevant: 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 call. From the reference section for each web method in this document, you can 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

## /analyses

POST /gobii/v1/analyses

### Description

Creates an analysis entity. Consult the [Request/Response Section](#) before attempting to use this call.

### Parameters

Type	Name	Schema
Body	body —	<a href="#">PayloadEnvelopeAnalysisDTO</a>

### Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">PayloadEnvelopeAnalysisDTO</a>

## /analyses

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	<a href="#">PayloadEnvelopeAnalysisDTO</a>

## /analyses

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	<b>analysisId</b> —	ID of Analysis to be updated	integer (int32)
Body	<b>body</b> —		<a href="#">PayloadEnvelopeAnalysisDTO</a>

## Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">PayloadEnvelopeAnalysisDTO</a>

## /auth

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	<b>body</b> —	string

## Responses



HTTP Code	Description	Schema
200	successful operation	string

## /configsettings

```
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	<a href="#">PayloadEnvelopeConfigSettingsDTO</a>

## /restprofiles

```
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 configuration document

### Parameters

Type	Name	Schema
Body	body —	<a href="#">PayloadEnvelopeRestProfileDTO</a>

### Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">PayloadEnvelopeRestProfileDTO</a>

# Definitions

## AnalysisDTO

Name	Schema
<b>algorithm</b> —	string
<b>allowedProcessTypes</b> —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
<b>analysisDescription</b> —	string
<b>analysisId</b> —	integer (int32)
<b>analysisName</b> —	string
<b>anlaysisTypeId</b> —	integer (int32)
<b>createdBy</b> —	integer (int32)
<b>createdDate</b> —	string (date-time)
<b>entityNameType</b> —	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)
<b>id</b> —	integer (int32)
<b>modifiedBy</b> —	integer (int32)
<b>modifiedDate</b> —	string (date-time)
<b>parameters</b> —	< <a href="#">EntityPropertyDTO</a> > array
<b>program</b> —	string
<b>programVersion</b> —	string
<b>referenceId</b> —	integer (int32)
<b>sourceName</b> —	string

Name	Schema
<b>sourceUri</b> —	string
<b>sourceVersion</b> —	string
<b>statusId</b> —	integer (int32)
<b>timeExecuted</b> —	string (date-time)

## ConfigSettingsDTO

Name	Schema
<b>allowedProcessTypes</b> —	< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
<b>id</b> —	integer (int32)
<b>maxUploadSizeMbytes</b> —	integer (int32)
<b>serverCapabilities</b> —	< string, boolean > map
<b>serverConfigs</b> —	< string, <a href="#">ServerConfigItem</a> > map

## EntityPropertyDTO

Name	Schema
<b>entityIdId</b> —	integer (int32)
<b>propertyId</b> —	integer (int32)
<b>propertyName</b> —	string
<b>propertyValue</b> —	string

## Header

Name	Schema
<b>cropType</b> —	string
<b>dtoHeaderAuth</b> —	<a href="#">HeaderAuth</a>

Name	Schema
<b>gobiiProcessType</b> —	enum (CREATE, READ, UPDATE, DELETE, NONE)
<b>gobiiVersion</b> —	string
<b>maxGet</b> —	integer (int32)
<b>maxPost</b> —	integer (int32)
<b>maxPut</b> —	integer (int32)
<b>pagination</b> —	<a href="#">Pagination</a>
<b>status</b> —	<a href="#">Status</a>

## HeaderAuth

Name	Schema
<b>gobiiCropType</b> —	string
<b>password</b> —	string
<b>token</b> —	string
<b>userName</b> —	string

## HeaderStatusMessage

Name	Schema
<b>gobiiStatusLevel</b> —	enum (ERROR, VALIDATION, WARNING, INFO, OK)
<b>gobiiValidationStatusType</b> —	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)
<b>message</b> —	string

## Link

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

## LinkCollection

Name	Schema
<b>exploreLinksPerDataItem</b> —	< < <a href="#">Link</a> > array > array
<b>linksPerDataItem</b> —	< <a href="#">Link</a> > array

## Pagination

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

## PayloadAnalysisDTO

Name	Schema
<b>data</b> —	< <a href="#">AnalysisDTO</a> > array
<b>linkCollection</b> —	<a href="#">LinkCollection</a>

## PayloadConfigSettingsDTO

Name	Schema
<b>data</b> —	< <a href="#">ConfigSettingsDTO</a> > array
<b>linkCollection</b> —	<a href="#">LinkCollection</a>

## PayloadEnvelopeAnalysisDTO

Name	Schema
<b>header</b> —	<a href="#">Header</a>
<b>payload</b> —	<a href="#">PayloadAnalysisDTO</a>

## PayloadEnvelopeConfigSettingsDTO

Name	Schema
<b>header</b> —	<a href="#">Header</a>
<b>payload</b> —	<a href="#">PayloadConfigSettingsDTO</a>

## PayloadEnvelopeRestProfileDTO

Name	Schema
<b>header</b> —	<a href="#">Header</a>
<b>payload</b> —	<a href="#">PayloadRestProfileDTO</a>

## PayloadRestProfileDTO

Name	Schema
<b>data</b> —	< <a href="#">RestProfileDTO</a> > array
<b>linkCollection</b> —	<a href="#">LinkCollection</a>

## RestProfileDTO

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

## ServerConfigItem

Name	Schema
<b>confidentialityNotice</b> —	string
<b>contextRoot</b> —	string

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

## Status

Name	Description	Schema
<b>statusMessages</b> —		< <a href="#">HeaderStatusMessage</a> > array
<b>statusMessagesByCode</b> —		< string, string > map
<b>succeeded</b> —	<b>Default :</b> <b>false</b>	boolean