

master document for generating PDF

= GOBii API

# Table of Contents

Overview	1
Version information	1
License information	1
URI scheme	1
System Architecture	1
DTO Models	1
Data Access	1
DTO Mapping	1
Services	1
The Web Controller	1
Web Service Operations	2
Request/Response Structure	2
Authentication	3
Error Handling	3
Documentation Tools	4
Paths	4
/analyses	4
/auth	5
/configsettings	5
Definitions	6
AnalysisDTO	6
ConfigSettingsDTO	7
EntityPropertyDTO	7
Header	7
HeaderAuth	8
HeaderStatusMessage	8
Link	9
LinkCollection	9
Pagination	9
PayloadAnalysisDTO	9
PayloadConfigSettingsDTO	9
AnalysisDTO	10
ConfigSettingsDTO	10
ServerConfig	10
Status	10

# 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>

## URI scheme

*Host* : localhost:8081

*BasePath* : /api

*Schemes* : HTTP

# System Architecture

## DTO Models

Content TBD

## Data Access

Content TBD

## DTO Mapping

Content TBD

## Services

Content TBD

## The Web Controller

Content TBD

# Web Service Operations

## Request/Response Structure

The http body structure is the same for all responses as well as for the body of POST and PUT requests. Here is a high level schematic of the body structure:

```
{
  "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 the type of the call. For example, for a POST to the the /project resource, it will contain a project object.

After a successful POST to the /auth resource, the "token" key in "dtoHeaderAuth" contains the token value, which should be used to set the X-Auth-Token header for subsequent requests.

## Authentication

A valid token may be acquired by doing a POST to the /auth resource with an empty body and the following request headers set:

X-Username X-Password

The username and password depend on how the destination GOBii server is configured (per the gobii-web.xml file):

If gobiiAuthenticationType of the server is TEST, the username and password are hard-coded to USER\_READER and reader, respectively. If the server is configured for LDAP authentication: The username/password must be provisioned in the configured LDAP server (if it is not, the error from the server will be 401: Unauthorized) There must be an entry for the user in contact table, where the username value matches the ldap user name; NOTE: if no such record is found, the error will be 403: Forbidden: 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 a 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. In this case, the token can also be retrieved from the meta data portion of the response body (see below).

## 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 currentStatusMessage :
header.getStatus().getStatusMessages()) {
        System.out.println(currentStatusMessage.getMessage());
    }
}

```

## Documentation Tools

The toolchain used to produce this documentation is swagger compatible. It is

## 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	Description	Schema	Default
Body	body		<a href="#">AnalysisDTO</a>	

#### Responses

HT TP Co de	Description	Schema
200	successful operation	<a href="#">AnalysisDTO</a>

## /auth

POST /gobii/v1/auth

### Description

The user credentials are specified in the request headers; the response and the response headers include the token, which can which the client will include in the request headers for subsequent requests.

### Parameters

Type	Name	Description	Schema	Default
Body	body		string	

### Responses

HT TP Co de	Description	Schema
200	successful operation	string

## /configsettings

GET /gobii/v1/configsettings

### Description

Provides generic configuration information about the GOBii instance

### Responses

HT TP Co de	Description	Schema
200	successful operation	<a href="#">ConfigSettingsDTO</a>

# Definitions

## AnalysisDTO

Name	Description	Schema
algorithm		string
allowedProcessesTypes		< enum (CREATE, READ, UPDATE, DELETE, NONE) > array
analysisDescription		string
analysisId		integer(int32)
analysisName		string
analysisTypeId		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)
id		integer(int32)
modifiedBy		integer(int32)
modifiedDate		string(date-time)
parameters		< <a href="#">EntityPropertyDTO</a> > array
program		string



Name	Description	Schema
<b>programVersion</b>		string
<b>referenceId</b>		integer(int32)
<b>sourceName</b>		string
<b>sourceUri</b>		string
<b>sourceVersion</b>		string
<b>statusId</b>		integer(int32)
<b>timeExecuted</b>		string(date-time)

## ConfigSettingsDTO

Name	Description	Schema
<b>allowedProcessesTypes</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="#">ServerConfig</a> > map

## EntityPropertyDTO

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

## Header

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

Name	Description	Schema
<b>gobiiProcessType</b>		enum (CREATE, READ, UPDATE, DELETE, NONE)
<b>gobiiVersion</b>		string
<b>pagination</b>		<a href="#">Pagination</a>
<b>status</b>		<a href="#">Status</a>

## HeaderAuth

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

## HeaderStatusMessage

Name	Description	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)
<b>message</b>		string

# Link

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

# LinkCollection

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

# Pagination

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

# PayloadAnalysisDTO

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

# PayloadConfigSettingsDTO

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

## AnalysisDTO

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

## ConfigSettingsDTO

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

## ServerConfig

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

## Status

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