

BUGtrack REST API

Common conventions

Base URL

All API calls are best described by the following template:

`https://domain.bugtrack.net/api/v2`

where **domain** is a second level domain name specific to your license;

In this documentation we'll refer to <https://api.bugtrack.net/api/v2> - this is the live license we've created to demonstrate API calls. All sample URL links are clickable; the output format is typically set to XML as almost every browser is capable to render XML inside.

API Output

For methods that return the information you should specify method's output format by appending [JSON](#) or [XML](#) extension to method name, such as `login.xml` or `user`. List method supports additional output formats: [HTML](#), [CSV](#) and [XLSX](#).

API utilizes HTTP code semantic to indicate success or failure.

HTTP Status Code	Meaning
200 (OK)	Operation is performed successfully; the data is returned in the response.
204 (No Content)	Operation is performed successfully; the method does not return any data.
304 (Not Modified)	If client-side caching is utilized this code is returned to indicate the cached copy is still valid.
400 (Bad Parameter)	Required parameter is missing or parameter's value is invalid.
401 (Authorization required)	No authorization information is passed or session established by the call to Login is expired.
403 (Access Forbidden)	Invalid user name or password or user does not have access to BUGtrack or requested object.
404 (Not Found)	Invalid method name
500 (Internal Error)	Unexpected error occurred while performing the operation.

In case of 4XX or 500 codes the response body contains error information in either JSON or XML format.

Client Side Caching

Response caching is encouraged as it reduces server load and saves the traffic between server and client. Attachments, addressed by their ID, never change and can be cached forever. Describe, User, List and Bug methods respond with [ETag](#) header – unique hash value of the content. You can cache output of these methods together with this hash and on second call pass hash value back with [If-](#)

[None-Match](#) header. Based on this value the API will check whether content has changed and will respond either with code 200 and updated content or with code 304 and no content to indicate cached information is still valid.

Authorization Methods

All API calls except for Login require authorization. You can authorize the user using either of methods described below.

Login/Logout Methods

Login method verifies user login and password are valid and returns random authentication token that is valid for 5 minutes since the last API call.

<https://api.bugtrack.net/api/v2/login.xml?login=test&password=test>

The method responds with the 32 character authorization token that should be inserted for all subsequent calls between `/v2/` and method name. Logout method can be used to invalidate the token.

<https://api.bugtrack.net/api/v2/12345678901234567890123456789012/logout>

Basic access authentication

In addition to Login method we support [Basic Access Authentication](#) (BA). This method might be useful when API call's result is imported to an application or tool with limited programming capabilities such as Excel's data import wizard. In this case the tool (assuming it supports BA) will prompt you for login and password.

<https://api.bugtrack.net/api/v2/list.html?output=Brief>

Embed user name and password

If none of the above methods are suitable you can specify login and password separated by the colon right in the URL, e.g.:

<https://api.bugtrack.net/api/v2/test:test/list.html?output=Brief>

API Reference

Login Method

Login method creates an authorization token to use with other API calls.

Parameters (all are required)

login	User's login
password	User's password

Sample URL

<https://api.bugtrack.net/api/v2/login.xml?login=test&password=test>

Sample code for JavaScript/jQuery

```
var api = "https://api.bugtrack.net/api/v2/";
// perform login
$.get(api + "login.json", { login: "test", password: "test" }, function(data) {
    // add auth token to the end of the url
    api = api + data.auth + "/";
    // get user information
    $.get(api + "user.json", function(data) { /*...*/ });
});
```

Logout Method

Logout method invalidates authorization token obtained via API call.

Parameters

None

Sample URL

<https://api.bugtrack.net/api/v2/fe255f9f636d4fb2b83993e528026413/logout>

Response

None

User Method

User method returns the information about the user currently logged on.

Parameters

None

Sample URL

<https://api.bugtrack.net/api/v2/test:test/user.xml>

Describe Method

Describe method returns the information about BUGtrack license from the user's standpoint. The information reported depends on user's ability to see various aspects of BUGtrack depending on a role assigned to a user in various projects. For example, if none of the projects available to the user have visible type field, types are not reported at all.

Parameters

None

Sample URL

<https://api.bugtrack.net/api/v2/test:test/describe.xml>

List Method

List method performs query over bug database. It can take either of two forms:

Sample URLs

<https://api.bugtrack.net/api/v2/test:test/list.xml/{id}?parameters>

<https://api.bugtrack.net/api/v2/test:test/list.xml?parameters>

Parameters (all are optional)

{id}	Either <i>my</i> for my stuff predefined filter, or <i>fav</i> to filter favorite bugs or id of the filter.
project¹	The name or id of the project to filter bugs for.
maxbugs¹	the number of bugs to display
output¹	Comma separated list of the field names to return in the result. Valid field names are: <i>Id, Title, Project, Area, OpenedBy, AssignedTo, Status, Priority, Version, Opened, Closed, Modified, Type, Category, Phase, Estimate, Uplink, DueDate, Created, CreatedBy, Resolved, ResolvedBy, CF1..CFN</i> . The order names are specified does not matter. You may also use two shorthand names: <i>Brief</i> and <i>Complete</i> to specify the list of output fields.
status	The status to filter bugs for. Valid values are <i>Any, Open, Closed, Active, Resolved</i> or id or name of resolution status.
query	Text string to search for. Supports couple of logical operators described here . If supplied, only parameters described above are applicable, with status values restricted to <i>Any</i> and <i>Open</i>
area	The name or id of the area to filter bugs for. If supplied, should be used together with project parameter.
openedBy	The name or id of the user to filter bugs for.
assignedTo	The name or id of the user to filter bugs for.
priority	The name or id of the priority to filter bugs for.
version	The name or id of the version to filter bugs for.
type	The name or id of the type to filter bugs for.
category	The name or id of the category to filter bugs for.
phase	The name or id of the phase to filter bugs for.
favorite	If supplied with any value, filter only for bugs marked as favorite.
estimate	Optional, either <i>Specified</i> or <i>NotSpecified</i> , filter bugs with specified/not specified estimates.
field	One of the following: <i>Opened, Resolved, Closed, Modified, DueDate, Created</i>
period	One of the following: <i>AnyTime, Today, Yesterday, YesterdayOrToday, ThisWeek, LastWeek, ThisMonth, LastMonth, Last7days, Last30days, Next7days, Next30days</i>
from	Start of the custom date range.
to	End of the custom date range.
cf[1]...cf[N]	The value of corresponding custom field to search for.
sort1...sort3	The field to use to sort the result. One of the following: <i>Project, Area, AssignedTo, OpenedBy, Priority, Status, Version, Type, Category, Phase, Estimate, DueDate, Recent, Created, Favorite, Newest, Oldest, CF1...CFN</i>

¹ these parameters are relevant for both forms, other parameters are relevant only for the second form

Attachment Method - GET

When using with GET method retrieves the content of the attachment, identified by its ID.

Sample URL

<https://api.bugtrack.net/api/v2/test:test/attachment/635509?parameters>

Parameters

w	Optional. If attachment is image, specified maximum output image width.
h	Optional, If attachment is image, specified maximum output image height.

These parameters have no effect for non-image file types. You can specify either of these parameters or both of them. If both parameters are specified the image is downscaled to fit into bounding box while preserving aspect ratio. If one parameter is specified, the image is downscaled by specified dimension, while preserving aspect ratio.

Attachment Method - POST

When using with POST method, stores the files on a server prior to making changes to a bug. It responds with the list of file ids that can be later passed to bug manipulation methods. You can post either single file content with Content-Type and Content-Disposition headers or multiple files via multipart/form data Content-Type. The server holds the file(s) till the next request to bug manipulation method within 5 minutes interval.

Sample URL

<https://api.bugtrack.net/api/v2/test:test/attachment.json>

<https://api.bugtrack.net/api/v2/test:test/attachment.xml>

Sample Request

```
POST /api/v2/test:test/attachment.json HTTP/1.1
Content-Type: text/plain
Content-Disposition: attachment;filename=hello.txt
Host: localhost
Content-Length: 12
```

Hello World!

Sample Response (JSON)

```
[
  ":37e2e485-d634-49af-b99a-9e858b09ece2:",
]
```

Sample Response (XML)

```
<Files xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <File>:37e2e485-d634-49af-b99a-9e858b09ece2:</File>
</Files>
```

Bug/{id} Method

Retrieves bug information given its id.

Sample URL

<https://api.bugtrack.net/api/v2/test:test/bug/1.xml>

Parameters:

None

Bug/{action} Methods

Set of methods to perform various actions on a single bug or multiple bugs. **{action}** is one of following: *new, edit, note, move, resolve, reactivate, reassign, close, reopen, link, delete*.

These methods can be invoked via either GET or POST methods. While GET being simpler to invoke the amount of data you can pass through is limited to about 50K. POST method's limit is about 20Mb.

Sample URL

<https://api.bugtrack.net/api/v2/test:test/bug/edit?id=1&body=Edited+via+API>

Parameters

id	Required, can appear multiple times, the id of a bug(s) to modify
title	The text of a title
assignto	The id or the name of the user to assign bug to
project	The id or the name of the project to create/move bug to
area	The id or the name of the area to create/move bug to
priority	The id or the name of the priority
version	The id or the name of the version
status	The id or the name of the resolution status
type	The id or the name of the type
category	The id or the name of the category
phase	The id or the name of the phase
format	Either <i>Text</i> or <i>HTML</i> , default is <i>Text</i>
body	The text of the comment
estimate	The value of estimate field expressed in seconds
elapsed	The value of elapsed field expressed in seconds
dueDate	The value of dueDate field. Pass empty string (e.g. dueDate= to clear due date)
cf[1]..cf[N]	The value of corresponding custom field
file	Can appear multiple times, the id of the file obtained from <i>Attachment</i> method call to attach to the comment.
notify	The flag (<i>true</i> or <i>false</i>) to indicate whether notifications should be sent, default is <i>true</i>
link	The id of the bug to link bugs to

Not all parameter are applicable to all methods. Below is a matrix of parameters and methods they are applicable to; with x to indicate required parameter, o – optional. In case of *New* method parameters omitted will be initialized with project defaults, for other methods omitted parameter means no modifications should be made to corresponding field.

new	edit	move	resolve	reactivate	reassign	close	reopen	link	delete
------------	-------------	-------------	----------------	-------------------	-----------------	--------------	---------------	-------------	---------------

id		X	X	X	X	X	X	X	X	X
title	X									
assignto	X					X		X		
project	X		X							
area	0		0							
priority	0	0	0							
version	0	0	0							
status					X					
type	0	0	0							
category	0	0	0							
phase	0	0	0							
format	0	0	0	0	0	0	0	0		
body	0	0	0	0	0	0	0	0		
estimate	0	0	0	0	0	0	0	0		
elapsed	0	0	0	0	0	0	0	0		
dueDate	0	0	0	0	0	0	0	0		
cf[1]..cf[N]	0	0	0							
file	0	0	0	0	0	0	0	0		
notify	0	0	0	0	0	0	0	0		
link										X

Return Values

New method returns the id of newly create bug. Other methods return no value and respond with status 204 (no content).