

API Reference

Introduction

The up.time API provides you with an easy way to access your up.time inventory and active status information from third-party tools. The API is designed in a general RESTful API format, allowing you to access it using any programming language that can consume URL request output. The API is provided by the up.time Controller, which is installed with the up.time Monitoring Station.

In addition to the reference below, the following resources are available to help you get started with the API.

API Examples

Each API example is available as an open source project on github as well as posted as an install ready plug-ins on The Grid.

- Mobile UI - [github](#) - [The Grid](#)
- Drag & Drop Map Dashboard - [github](#) - [The Grid](#)
- Incident Console - [github](#) - [The Grid](#)
- jQuery Example Library - [github](#)

Helper Classes

Helper classes that simplify access to and management of API functions are available here:

- [.Net Helper Class](#)
- [PHP Helper Class](#)
- [JavaScript Helper Class](#)

API Overview

All API commands can be accessed using this general format:

```
https://<hostname>:<port>/api/<api_version>/<end_point>/<id>/<task>
```

- **hostname**: the URI for the up.time Controller installation
- **port**: the up.time Controller listener port, typically 9997
- **api_version**: the version of the API to run commands against (see [Version Control](#) below)
- **end_point**: the type of object you want to work with in up.time (see [End Points](#) below)
- **id**: the numerical ID of the object you are interested in
- **task**: where supported, specific tasks to be executed against the provided object

Methods

Accessing the standard format URL using different HTTP methods produces different results on the target object:

- **POST**: add an object or trigger an action
- **PUT**: edit an object
- **DELETE**: delete an object
- **GET**: view the details of an object

Currently only **GET** operations are supported.

Version Control

The version of the API you wish to access is embedded directly into the URL. Bug fixes and non-breaking feature changes will be made without changing the version number. Major feature changes or breaking changes will introduce a new version number. Backwards support for previous API version is currently not defined.

Non-breaking changes include:

- adding fields or functions to input requests
- adding fields to returned data

Breaking changes include:

- changing or removing fields or functions on input requests
- changing or removing fields in returned data
- changing the URI of any existing function

Authentication

Authentication to the API is based on the basic HTTP basic authentication template: each request to the API must provide a username and password pair. To safeguard this information, all requests to the up.time API must use SSL communication.

Each user in up.time has access to the API. User-visibility and role-based permissions are applied to each API call, ensuring users are only able to access and modify the same information they would be able to access from the up.time UI.

Endpoints

Endpoints define the different types of up.time objects that you can work with using the API. The currently supported endpoints include the following:

- elements: manage up.time Elements
- monitors: manage up.time service monitors
- groups: manage Element groups in up.time

For more information, see the following sections:

- [Working with Elements](#)
- [Working with Groups](#)
- [Working with Service Monitors](#)

Returned Results

All returned results are provided in JSON format. Successfully completed requests will return an HTTP code in the 200 range.

Error Handling

If the provided command produces an error, an HTTP status in the 400 range will be returned including a specific status code and a message with further details about the error. All error messages are returned in JSON format and look like the following:

```
{
  "error": "Required field missing",
  "errorDescription": null
}
```

Here is a quick reference of supported error codes.

Error Code	Error	HTTP Code	Details
UT-0400	Bad Request	400	The request could not be processed by the server due to incorrect syntax. API commands can be accessed with this format: <code>https://<hostname>:<port>/api/<api_version>/<end_point>/<id>/<task></code> If you encounter this error, ensure the referencing URL is correct.
UT-0404	Resource Not Found	404	The request could not be processed because an object is missing. Verify the endpoint has not been omitted from the command, and otherwise ensure it has been spelled correctly.
UT-0405	Method Not Allowed	405	The request method used for an object is not supported by that resource. This mismatch may cause an exception. As a starting point, look for this exception in the <code>uptime_controller.log</code> file, but it is likely you should verify the method you used is correct. See Methods for more information.
UT-0500	Unknown	500	The corresponding HTTP error code (500) is a catch-all error generated by the Web server where an unexpected condition prevented the request from being fulfilled. View the <code>uptime_controller.log</code> file for possible issues.
UT-0555	Unknown Exception	500	The corresponding HTTP error code (500) is a catch-all error generated by the Web server where an unexpected condition prevented the request from being fulfilled. In this case, an exception has occurred; as a starting point, look for this exception in the <code>uptime_controller.log</code> file.
UT-0560	Internal Server Error	500	The corresponding HTTP error code (500) is a catch-all error generated by the Web server where an unexpected condition prevented the request from being fulfilled. In this case, an exception caused a stack trace; as a starting point, look for this stack trace in the <code>uptime_controller.log</code> file.
UT-1000	Element Does Not Exist	404	A specifically referenced Element ID does not exist. In such a case, referencing <code>https://youruptime:9997/api/v1/elements/123/</code> would return the following: The element id '12345' does not exist.
UT-1001	Monitor Does Not Exist	404	A specifically referenced service monitor ID does not exist. In such a case, referencing <code>https://youruptime:9997/api/v1/monitors/456/</code> would return the following: The service monitor id '456' does not exist.

UT-1002	Element Group Does Not Exist	404	<p>A specifically referenced Element group does not exist.</p> <p>In such a case, referencing <code>https://youruptime:9997/api/v1/groups/789/</code> would return the following:</p> <p>The element group id '789' does not exist.</p>
---------	------------------------------	-----	---

In rare circumstances a 503 - `Service Unavailable` error may be returned. This indicates that the API is currently under heavy load and additional requests are being temporarily ignored in order to service existing requests.

Programming Guidance

When integrating with the API, the following general programming rules should be noted:

- fields will not always be returned in the same order; reference fields by name
- fields will not always be returned; validate a field is available before using it
- array results are not guaranteed to be sorted
- date and time fields will always be returned in ISO 8601 format

`An exception has occurred: null. Check uptime_controller.log for details.`