

# 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 [Endpoints](#) 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

## Returned Results

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

## 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.

## 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

## 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 – up.time Elements  
/monitors – up.time service monitors  
/groups – up.time Element groups
```

The following is a summary of presently available resources:

### [Working with Elements](#)

list all Elements and attached service monitors; display an Element's information and attached service monitors; display an Element's information and its current status:

```
GET  
GET  
GET
```

```
/api/v1/elements
/api/v1/elements/{id}
/api/v1/elements/{id}/status
```

update Element configuration:

```
PUT
/api/v1/elements/{id}
```

add, delete Element:

```
POST
DELETE
/api/v1/elements
/api/v1/elements/{id}
```

### Working with service monitors

list all service monitors; display a specific server monitor; display status information for a service monitor:

```
GET
GET
GET
/api/v1/monitors
/api/v1/monitors/{id}
/api/v1/monitors/{id}/status
```

### Working with Element groups

list all Element groups, member Elements, and monitors; list an Element group's member Elements and monitors; display the status of an Element group's member Elements and monitors:

```
GET
GET
GET
/api/v1/groups
/api/v1/groups/{id}
/api/v1/groups/{id}/status
```

## 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
}
```