

# Filtering Objects

- [Object IDs](#)
- [Monitors Filter Example](#)
- [Elements Filter Example](#)
- [Groups Filter Example](#)
- [Filter Persistence](#)

You can perform a filter on objects for use with the `GET` method to reduce API calls and improve performance.

To perform a filter, use the `POST` method to indicate the object type (the elements, groups, and monitors endpoints), and provide JSON-compliant object IDs:

```
POST https://youruptime/api/v1/<endPoint>/filter

{ ids : [a,b,c] }
```

The response includes a newly created filter ID for the collection of endpoints:

```
{ "id":n }
```

Use the ID with a `GET` method to return an array of elements in the [standard element listing](#) format:

```
GET https://youruptime/api/v1/<endPoint>/filter/<n>/
```

You can also return an array of [element statuses](#):

```
GET https://youruptime/api/v1/<endPoint>/filter/<n>/status
```

## Object IDs

For "monitor" and "group" object types, the IDs provided in the `POST` request correspond to the `id` field for the objects filtered. The object ID provided must match the object type declared in the request.

Similarly, element IDs can be provided for "element" object types:

```
{
  ids : [14,6,9],
}
```

However, for this object type, groups can also be provided. This can be helpful in cases where you want to output element listings, but are using groups as a way to declare the elements. In the request, the group `id` field is denoted as `groupIDs`:

```
{
  groupIDs : [2,3]
}
```

With this call, both elements and groups can also be filtered together:

```
{
  ids : [14,6,9],
  groupIDs : [2,3]
}
```

## Monitors Filter Example

Request:

```
POST https://youruptime:9997/api/v1/monitors/filter
```

```
{ ids : [21801,21802] }
```

Returned:

```
{"id":1}
```

Next request:

```
GET https://youruptime:9997/api/v1/monitors/filter/1/status
```

Returned:

```
[
  {
    "acknowledgedComment": null,
    "elementId": 556,
    "elementStatus": {
      "id": 556,
      "isMonitored": true,
      "lastCheckTime": "2016-01-15T10:36:40",
      "lastTransitionTime": "2016-10-01T08:35:16",
      "message": null,
      "name": "Aetos",
      "powerState": null,
      "status": "OK"
    },
    "id": 21801,
    "isAcknowledged": false,
    "isHidden": false,
    "isHostCheck": false,
    "isMonitored": true,
    "lastCheckTime": "2016-01-15T10:31:58",
    "lastTransitionTime": "2016-01-15T10:02:15",
    "message": "Unable to contact Agent (Aetos on port 3333)",
    "name": "UPTIME-Aetos",
    "status": "CRIT"
  },
  {
    "acknowledgedComment": null,
    "elementId": 556,
    "elementStatus": {
      "id": 556,
      "isMonitored": true,
      "lastCheckTime": "2016-01-15T10:36:40",
      "lastTransitionTime": "2016-10-01T08:35:16",
      "message": null,
      "name": "Aetos",
      "powerState": null,
      "status": "OK"
    },
    "id": 21802,
    "isAcknowledged": false,
    "isHidden": false,
    "isHostCheck": true,
    "isMonitored": true,
    "lastCheckTime": "2016-01-15T10:36:40",
    "lastTransitionTime": "2016-10-01T08:35:16",
    "message": "Ping completed: 5 sent, 0.0% loss, 0.6ms average round trip time",
    "name": "PING-Aetos",
    "status": "OK"
  }
]
```

## Elements Filter Example

Request:

```
POST https://youruptime:9997/api/v1/elements/filter

{ ids : [14,6,9] }
```

Returned:

```
{ "id":2 }
```

Next request:

```
GET https://youruptime:9997/api/v1/elements/filter/2/status
```

## Groups Filter Example

Request:

```
POST https://youruptime:9997/api/v1/groups/filter

{ ids : [2,3] }
```

Returned:

```
{ "id":3 }
```

Next request:

```
GET https://youruptime:9997/api/v1/groups/filter/3/status
```

## Filter Persistence

By default, a filter created from a `POST` request persists for 5 minutes, and are designed to be used immediately after creation. The `GET` method can use the filter until expiry, after which a `410 Gone` status code is returned.

Only the Uptime Infrastructure Monitor user who created the filter is able to see and use it.