This section describes general conventions used by the NetCloud API.

**Format of Examples**

The NetCloud API does not require the use of any particular software tools or libraries. Any software tools that support the use of HTTP request methods will work. However, the syntax of how to express the request varies with the tool or library. The examples focus on the HTTP method, the path, and the request body (when required). These examples don't show authentication headers. The following example shows a GET request to the routers endpoint:

GET https://www.cradlepointecm.com/api/v2/routers/

It's assumed that every request sends an HTTP header with the necessary API keys.

Please note that URLs for endpoints MUST end with a forward slash (trailing slash). If a trailing slash is not present the call to the endpoint is automatically redirected, resulting in two calls attributed to your account.  
**Correct format**:    https://www.cradlepointecm.com/api/v2/accounts/  
**Incorrect format**: https://www.cradlepointecm.com/api/v2/accounts

In cases of PUT or POST the examples show the data being sent as a JSON document below the PUT or POST, like this:

POST https://www.cradlepointecm.com/api/v2/groups/

{"name": "Test Group", "product":"https://www.cradlepointecm.com/api/v2/products/1/"}

PUT and POST must send a Content-Type header. If using JSON like the examples in this doc, the Content-Type must be 'application/json.'

**Note:** The term "device" is used in this documentation to mean either a router or access point. It is important to note that endpoint parameters and methods named, or containing, the word “router(s)” might also apply to access points.

**CRUD & HTTP Methods**

The API is RESTful, and uses the following HTTP methods to implement the four basic CRUD (Create, Read, Update, and Delete) operations:

Quick reference:

|  |  |  |  |
| --- | --- | --- | --- |
| Create | POST | https://www.cradlepointecm.com/api/v2/{resource-type}/ | The request body should contain a resource descriptor. |
| Read | GET | https://www.cradlepointecm.com/api/v2/{resource-type}/{id}/ |  |
| Update | PUT | https://www.cradlepointecm.com/api/v2/{resource-type}/{id}/ | The request body should contain a partial resource descriptor containing whatever attributes need to be replaced. |
| Delete | DELETE | https://www.cradlepointecm.com/api/v2/{resource-type}/{id}/ | This process is not reversible. |

Create a Resource

Create a resource by sending a POST to the root path for that resource type. The following example creates a new group, specifying the account, a group name, type of product, and the firmware for the devices in the group:

POST https://www.cradlepointecm.com/api/v2/groups/

{"account": "https://www.cradlepointecm.com/api/v2/accounts/23/", "name": "Group 2", "product": "https://www.cradlepointecm.com/api/v2/products/1/", "target\_firmware": "https://www.cradlepointecm.com/api/v2/firmwares/3771/"}

By default, the resource is added to the requestor's account. If the requestor wants to add to a different account, the request should include an 'account' field in the POST data.

Read Resources

Reading is simple, just issue a GET:

GET https://www.cradlepointecm.com/api/v2/routers/

Or to get a specific object, append the object's ID to the request:

GET https://www.cradlepointecm.com/api/v2/routers/1234/

Update a Resource

Like creating, updating resources requires you to provide a request body. The request body needs only list the attributes that are changing.

PUT https://www.cradlepointecm.com/api/v2/groups/23243/

{"name": "Renamed Group"}

Delete a Resource

Deleting a resource removes that specific resource and all the data associated with it. In the case of a router deletion, the router is unregistered from NCM and all of its associated data (e.g. alerts, state samples, signal samples, etc) is removed from the system. **The delete process is not reversible**.

DELETE https://www.cradlepointecm.com/api/v2/routers/1224/

**Request Parameters**

Parameters are appended to the path after a '?'. The examples uses a short-hand notation in '{' brackets '}' to show common parameters that are supported.

Quick reference:

{paging} := offset=<int>&limit=<int>

{fields} := fields=<val>.<val>,<val>

Examples:

GET https://www.cradlepointecm.com/api/v2/routers/?offset=100&limit=50

GET https://www.cradlepointecm.com/api/v2/router\_alerts/?type=wan\_unplugged&time\_\_gte=today

GET https://www.cradlepointecm.com/api/v2/accounts/?fields=name

Paging Parameters {paging}

Endpoints that return result sets with multiple records (lists) can be "paged" through by specifying 'offset' and 'limit' parameters.

GET https://www.cradlepointecm.com/api/v2/routers/?offset=50&limit=25

The offset parameter is supported for the following endpoints:

* routers
* net\_devices
* groups
* accounts
* products
* firmwares
* net\_device\_metrics
* configuration\_manager

When a list of records is returned from an endpoint call, the values used for offset and limit are returned in the meta portion of the response. Note that most resources have a max limit of 500. To page the whole list you need to increment the offset. So to get the next 25 devices:

GET https://www.cradlepointecm.com/api/v2/routers/?offset=75&limit=25

{fields} aka Partial Returns

If the requestor does not want to get the full response body, they can use the \_fields\_ param to get a partial result containing only the specified fields. Example:

GET https://www.cradlepointecm.com/api/v2/accounts/308/?fields=name,is\_disabled

**Format & Extensions**

The only officially supported format is JSON, which is the default.

**Reply Format**

The NetCloud API returns the following HTTP response codes:

GET : 200 OK

POST : 201 Created

PUT : 202 Accepted

DELETE: 204 No Content

307 Found (Redirected)

400 Bad Request

401 Unauthorized

403 Forbidden

404 Not Found

405 Method Not Allowed

409 Conflict

429 Too Many Requests

500 Internal Server Error

502 Bad Gateway

Normal response format

{

"data": [

{

"account": "https://www.cradlepointecm.com/api/v2/accounts/1/",

"id": "3",

"is\_disabled": false,

"name": "blue",

"resource\_url": "https://www.cradlepointecm.com/api/v2/accounts/3/"

},

{

"account": "https://www.cradlepointecm.com/api/v2/accounts/3/",

"id": "4",

"is\_disabled": false,

"name": "blue\_sub",

"resource\_url": "https://www.cradlepointecm.com/api/v2/accounts/4/"

}

],

"meta": {

"limit": 20,

"next": null,

"offset": 0,

"previous": null

}

}

Error response format

{

"errors": [

{

"path": "https://www.cradlepointecm.com/api/v2/groups/"

},

{

"method": "POST"

}

],

"exception": {

"message": "method not allowed",

"type": "error"

}

}

The exception codes are listed below, as well as the fields which appear with them in the error response.

unauthorized

{

"status\_code": 401,

"exception": "unauthorized",

"message": " Invalid Credentials i.e. missing or invalid keys.",

"success": false

}

forbidden

{

"exception": {

"message": "forbidden",

"type": "error"

},

"errors": []

}

server\_error

{

"exception": {

"message": "Expecting ',' delimiter or '}': line 3 column 2 (char 70)",

"type": "server\_error"

},

"errors": []

}

validation

{

"exception": {

"type": "validation",

"message": "bad or missing data"

},

"errors": [

{

"foo": "no such field"

}

]

}