



# API Integration

## with the enCloud™/enSite™ Enterprise Management System

**Y**ou can track your EN-400™, EN-1000™, EN-2000™, EN-3000™, and EN-4000™ routers (EN™ devices) via a direct connection to your computer, or over a LAN, or in several other ways, including through Encore Network's enCloud™ management system.

The enCloud™ system lets you track your network from anywhere with a connection to the internet. You can use a computer, a tablet, or a smartphone to track your devices. Development and use of enCloud™ are discussed in the documents listed below.

The enSite™ premises-based enterprise management system, virtually identical to enCloud™, is available to manage Encore devices in closed, private networks.

**Note:** The use of the term "enCloud™" includes "enSite™" and is applicable in all instances, unless specific instructions are indicated for enCloud™ or for enSite™. In cases where specific enCloud™ or enSite™ instructions are indicated, those instructions apply only to the topic under discussion.

### List of Documents

Information follows for integration of the application programming interface with the enCloud™/enSite™ management system.

---

**Notice:** Encore Networks, Inc., provides technical support only to authorized Encore resellers and to end customers that have purchased a support contract.



# Encloud – API Integration

The purpose of this API Document is to explain the processes and the REST API in detail for the sake of an integration of a third party application to enCloud. Before getting started, the following items should be examined:

- The user(s) that is going to use the API should be added to enCloud with the user type 'API User'. Other users do not have permission to use the web services provided in this documentation.
- The API-KEY of the distributor accounts can be found in the **Distributors** tab in enCloud, from the **Actions** button on a specific **Distributor**.

Description	Creation Date	Last Update Date	Total Customers	Total Users	Total Devices	
-	06/16/2017 11:19:23 AM	06/16/2017 1:27:46 PM	0	0	0	Actions
-	03/13/2017 11:04:12 PM	06/16/2017 1:27:43 PM	0	1	0	Actions
Danielle's add	03/13/2017 9:37:52 PM	06/16/2017 1:27:44 PM	3	2	6	Actions
-	12/13/2016 10:41:29 PM	06/16/2017 1:27:42 PM	0	0		<ul style="list-style-type: none"> <li>Overview</li> <li>API Key</li> <li>Edit</li> <li>Delete</li> </ul>
Holder for Orbcomm devices	12/09/2016 11:23:13 PM	06/16/2017 1:27:42 PM	0	0		

- **Customer Unique ID:** The customer unique IDs in enCloud should be the same as the customer ids in the 3rd party system. The IDs are used to match the items between the two applications. Customer unique ID can be set from **Customers** tab in enCloud, from the **Actions** button in a specific **Customer**.

Filter: Distributor

Name	Domain	Total Devices	Total Users	Total Alerts	Assigned Distributor	Creation Date	Last Update Date	Total Use Cases	
Brad's company	bradco	12	8	0	ABC (Encore Engineering)	01/15/2016 5:25:06 PM	06/16/2017 7:47:09 PM	2	Actions

entries (filtered from 62 total entries)

« First < Previous

- Enter
- Overview
- Unique ID
- Edit
- Delete

**Table Of Contents:**

- Rest API
  - Authorise API User
    - Method Type & URL
    - Headers
    - URL Parameters
    - Body
    - Examples
    - Responses
  - Device Pull Information
    - Method Type & URL
    - Headers
    - URL Parameters
    - Body
    - Examples

- Responses
- Customer Users
  - Create User
    - Method Type & URL
    - Headers
    - URL Parameters
    - Body
    - Example
    - Responses
  - Update User
    - Method Type & URL
    - Headers
    - URL Parameters
    - Example
    - Responses
  - Delete User
    - Method Type & URL
    - Headers
    - URL Parameters
    - Body
    - Example
    - Responses
- Hot link to enCloud (2 Steps)
  - Step 1: Token Generation
    - Headers
    - URL Parameters
    - Body
    - Example
    - Responses
  - Step 2: Redirection to enCloud
  - Diagram: Hot Link to enCloud

## Rest API

### Authorise API User

Basic Auth system is used for authorising the api user. All of the API methods provided in this document can be used by authorising this user.

#### Method Type & URL

Method Type	URL
GET	<a href="https://myenCloud.com/integration/users/authorise">https://myenCloud.com/integration/users/authorise</a>

#### Headers

For Basic Auth's **Authorization** header, set:

**Username:** The email of the user.

**Password:** The md5 hashed version of user's password.

(Ex: Original Password: Pass1234, MD5 Hashed Password: **823da4223e46ec671a10ea13d7823534** use this one as password)

Field	Type	Value
Authorization	String	<b>Basic YmVya2JheWR1ckBnbWFpbC5jb206Zjc0MzgxOGNmZDAyOTgwYTdjMDQzMGVmNnNQ0OGE4YTc</b>

#### URL Parameters

None

## Body

No request body.

## Examples

Type	Example
cURL	curl -X GET -H "Authorization: Basic YmVya2JheWR1ckBnbWFpbC5jb206Zjc0MzgxOGNmZDAyOTgwYTdjMDQzMzVhbnQ0OGE4YTc=" "https://myenCloud.com/integration/users/authorise"
HTTP	GET integration/users/authorise HTTP/1.1 Host: myenCloud.com Authorization: Basic YmVya2JheWR1ckBnbWFpbC5jb206Zjc0MzgxOGNmZDAyOTgwYTdjMDQzMzVhbnQ0OGE4YTc=

## Responses

You must have the auth\_token in order to use the API.

HTTP Code	Message	Description
200	<pre>{   "user": {     "auth_token":     "N2CDnVtoSuSfQ4I2XWo6Aw",     "name": "API User",     "email": "api.user@iven.io"   } }</pre>	<p>The authorised user that is going to use the API.</p> <p><b>auth_token:</b> The auth_token specific to this API user.</p> <p><b>**The auth token gets updated automatically. So it is crucial to use this service when the API User's auth token is expired.</b></p>
404	<pre>{   "status": 404,   "description": "The user is not found.",   "ivenCode": 9002 }</pre>	The user is not found.
422	<pre>{   "status": 422,   "description": "Bad credentials.",   "ivenCode": 9007 }</pre>	Bad credentials.

## Device Pull Information(Pagination)

Device Pull Information is used to provide the requested devices' meta data or sensor information. It is either used to get current or historical information depending on the timestamp provided.

### Method Type & URL

Method Type	URL
GET	https://myenCloud.com/integration/devices

### Headers

Field	Type	Description
Authorization	String	The <b>auth_token</b> of the API user.
API-KEY	String	The <b>api_key</b> of the distributor account.

### URL Parameters

Field	Description
devices	Optional parameter. If not given, the information of all devices will be returned in the response.If given, the items should be separated by commas. Ex: devices=00A0EB801B98,00A0EB8000F8,00A0EB030E93
timestamp	Required parameter. If given 0, the system will give devices information for all time. Otherwise, the system is expecting Unix Timestamp( <i>seconds since Jan 01 1970(UTC)</i> ) Ex: 1496752350 = 06/06/2017 @ 12:32pm (UTC)
keys	Required parameter. The sensor information like RSSI, RSRQ etc. The available keys can be found in the profiles section. The items should be separated by commas. Ex: keys=cell.rssi,cell.uptime,cell.rsrq
customer_id	Optional parameter. The <b>customer unique id</b> that has been set before the integration. The devices under the given customer will be returned.
state	Optional parameter. The current state of the device. Permitted values are 'Registered', 'Waiting For Response', 'Passive', 'Expired'. The

	devices with the given <b>state</b> will be returned.
is_online	Optional parameter. Permitted values are 'true', 'false'. Online or offline devices will be returned.
search	Optional parameter. The input searches for the device name or the device UID.
limit	Optional parameter. The number of results that will be returned per page. The default value is set on the API User's settings. Permitted values are '10', '25', '50', '100', '-1'(all records)
page	Optional parameter. The page number of the returned records. The default value is 1.

## Body

No request body.

## Examples

Type	Example
cURL	<pre>curl -X GET \   'https://myenclud.com/integration/devices?timestamp=0&amp;keys=system.uptime%2Ccell2.status&amp;devices=&amp;state=&amp;is_online=false&amp;search=&amp;limit=100&amp;page='\   -H 'api-key: 30e087a1db41aca5942f6ae72f10db3facc409ed' \   -H 'authorization: SSryCZZzDTH5FyQvpIRBAg' \</pre>
HTTP	<pre>GET /integration/devices?timestamp=0&amp;keys=cell.uptime&amp; devices=00A0EB801B98,00A0EB030E13 HTTP/1.1 Host: localhost:3000 Authorization: SSryCZZzDTH5FyQvpIRBAg API-KEY: 30e087a1db41aca5942f6ae72f10db3facc409ed</pre>

## Responses

HTTP Code	Message	Description
200	See example_result.json	<p><b>totalResults:</b> Total number of devices</p> <p><b>totalFilteredResults:</b> The number of devices according to the filter applied.</p> <p><b>limit:</b> Current limit (number of devices returned per page)</p>

		<p><b>page:</b> Current page</p> <p><b>sort:</b> Default and only value is 'created_at' (results are returned in order of the devices creation. This value was chosen because it is static and can't be changed thus making the list of results more consistent)</p> <p><b>sort_type:</b> 'ASC' or 'DESC'</p> <p><b>devices[]:</b>An array of devices is returned, filled with the devices and key information requested. The result may be very long or very short, depending on how much information is requested.</p> <p><b>device_uid:</b> The unique ID of the device.</p> <p><b>sim_id:</b> The SIM information of the devices.</p> <p><b>location:</b> Location as [lat,lon]</p> <p><b>frequency:</b> The time interval that the device pushes the information to enCloud(as seconds)</p> <p><b>created_at:</b> Creation date (Note: the Creation date is the date the device was added to inventory on the enCloud by the Encore factory)</p> <p><b>updated_at:</b> Last Edit.</p> <p><b>location_string:</b> Location as address</p> <p><b>state:</b> The device state (passive, active etc.)</p> <p><b>last_seen:</b> The last time the device "checked in" (sent data) to enCloud.</p> <p><b>last_data:</b> Data contains all of the data contained in the key values requested in the form of a JSON object.</p>
500	Internal Server Error	enCloud related error.

## Customer Users

### Create User

Whenever a user is created in the 3rd party application, **Create User** can be used to create a duplicate item on enCloud.

### Method Type & URL

Method Type	URL
POST	<a href="https://myenCloud.com/integration/users">https://myenCloud.com/integration/users</a>

### Headers

Field	Type	Description
Authorization	String	The <b>auth_token</b> of the API User.



## Customer Users

### Create User

Whenever a user is created in the 3rd party application, **Create User** can be used to create a duplicate item on enCloud.

### Method Type & URL

Method Type	URL
POST	<a href="https://myenCloud.com/integration/users">https://myenCloud.com/integration/users</a>

### Headers

Field	Type	Description
Authorization	String	The <b>auth_token</b> of the API User.

--	--	--

API-KEY	String	The <b>api_key</b> of the distributor.
Content-Type	String	Content Type should be application/json

## URL Parameters

No URL Parameters.

## Body

The request body should be a JSON Object with the key **user**. The **user** object should include the following keys.

Field	Type	Description
user	Object { }	The value should be another key-value object including the information of the user that is going to be created.
user_type	String	The user type of this user on enCloud. The options are: ADMIN, VIEWER
email	String	Email address of the user. This has to be unique because users are selected via their email addresses in enCloud.
name	String	The name of the user.
customer_id	String	The <b>customer unique id</b> that has been set before the integration. The customer that the user is going to be added to.

## Example

Type	Example
cURL	<pre>curl -X POST \ http://myenCloud.com/integration/users \ -H 'api-key: 30e087a1db41aca5942f6ae72f10db3facc409ed' \ -H 'authorization: SSryCZZzDTH5FyQvplRBAg' \ -d '{   "user": {     "user_type": "ADMIN",     "customer_id": "SDK15",     "email": "sedetgungor@gmail.com",     "name": "Sedet"   } }'</pre>
HTTP	<pre>POST /integration/users HTTP/1.1 Host: myenCloud.com Authorization: SSryCZZzDTH5FyQvplRBAg API-KEY: 30e087a1db41aca5942f6ae72f10db3facc409ed Content-Type: application/json  {   "user": {     "user_type": "ADMIN",     "customer_id": "SDT15",     "email": "sedetgungor@gmail.com",     "name": "Sedet"   } }</pre>

## Responses

HTTP Code	Message	Description
-----------	---------	-------------

200	{ "status": 200, "description": "User is created successfully." }	User is created successfully.
400	{ "status": 400, "description": "**** is not present", "ivenCode": 0 }	Invalid Customer ID, or User Type.
400	{ "status": 400, "description": "User with this email already exists.", "ivenCode": 0 }	User already exists.
500	Internal Server Error	enCloud related error

## Update User

Whenever a user is updated in the 3rd party application, **Update User** can be used to update the duplicate item on enCloud.

## Method Type & URL

Method Type	URL
PUT	<a href="https://myenCloud.com/integration/users">https://myenCloud.com/integration/users</a>

## Headers

Field	Type	Description
Authorization	String	The <b>auth_token</b> of the API User.
API-KEY	String	The <b>api_key</b> of the distributor.
Content-Type	String	Content Type should be application/json

## URL Parameters

Field	Type	Description
email	String	The email of the updated user.

## Body

The request body should be a JSON Object with the key **user**. The **user** object should include the following keys.

Field	Type	Description
user_type	String	The changed user type of this user on enCloud. The options are: ADMIN, VIEWER
changed_email	String	Updated email of the user.
name	String	The changed name of the user.

## Example

Type	Example
------	---------

cURL	<pre>curl -X PUT \ 'http://myenCloud.com/integration/users?email=sedetgungor%40gmail.com' \ -H 'api-key: 30e087a1db41aca5942f6ae72f10db3facc409ed' \ -H 'authorization: SSryCZZzDTH5FyQvplRBAg' \ -d '{ "user": { "user_type": "ADMIN", "name": "Sedet", "changed_email": "xx@mail.com" } }'</pre>
HTTP	<pre>PUT /integration/users?email=sedetgungor@gmail.com HTTP/1.1 Host: myenCloud.com Authorization: SSryCZZzDTH5FyQvplRBAg API-KEY: 30e087a1db41aca5942f6ae72f10db3facc409ed Content-Type: application/json  { "user": { "user_type": "ADMIN", "name": "Sedet", "changed_email": "xx@mail.com" } }</pre>

## Responses

HTTP Code	Message	Description
200	<pre>{ "status": 200, "description": "User is updated successfully." }</pre>	User is created successfully.
404	<pre>{ "status": 404, "description": "Given User not found", "ivenCode": 0 }</pre>	A user with the given email address is not found in enCloud.
400	<pre>{ "status": 400, "description": "User Type is not present", "ivenCode": 0 }</pre>	Invalid User Type.
400	<pre>{ "status": 400, "description": "User with this email already exists.", "ivenCode": 0 }</pre>	The changed email already exists in enCloud.
500	Internal Server Error	enCloud related error.

## Delete User

Whenever a user is deleted in the 3rd party application, **Delete User** can be used to delete the duplicate item on enCloud.

## Method Type & URL

Method Type	URL
DELETE	<a href="https://myenCloud.com/integration/users">https://myenCloud.com/integration/users</a>

## Headers

Field	Type	Description
Authorization	String	The <b>auth_token</b> of the API User.
API-KEY	String	The <b>api_key</b> of the distributor.
Content-Type	String	Content Type should be application/json

## URL Parameters

Field	Type	Description
email	String	The email of the user to be deleted.

## Body

No request body.

## Example

Type	Example
cURL	<pre>curl -X DELETE \ 'http://myenCloud.com/integration/users?email=sedetgungor%40gmail.com' \ -H 'api-key: 30e087a1db41aca5942f6ae72f10db3facc409ed' \ -H 'authorization: SSryCZZzDTH5FyQvplRBAg' \ -d '{   "user": {     "user_type": "ADMIN",     "name": "Sedet",     "changed_email": "xx@mail.com"   } }'</pre>
HTTP	<pre>DELETE /integration/users?email=sedosgungor1@iven.io HTTP/1.1 Host: myenCloud.com Authorization: SSryCZZzDTH5FyQvplRBAg API-KEY: 30e087a1db41aca5942f6ae72f10db3facc409ed Content-Type: application/json</pre>

## Responses

HTTP Code	Message	Description
200	<pre>{   "status": 200,   "description": "User is deleted successfully." }</pre>	User is deleted successfully.
404	<pre>{   "status": 404,   "description": "Given User not found",   "ivenCode": 0 }</pre>	A user with the given email address is not found in enCloud.
500	Internal Server Error	enCloud related error.

## Hot link to enCloud (2 Steps)

In order to securely send users to enCloud via hot link, a two-step login process should be implemented:

### Step 1: Token Generation

The token request must include the URL parameters below. enCloud after receiving the request will generate a unique token that identifies the user's request and will return it. If the user does not exist in the system, enCloud will create a user from the email that was sent in the token generation request (in this case the Cust ID must also be sent).

The generated token can then be used in step 2.

#### Method Type & URL

Method Type	URL
GET	<a href="https://myenCloud.com/integration/hot link">https://myenCloud.com/integration/hot link</a>

#### Headers

Field	Type	Description
Authorization	String	The <b>auth_token</b> of the user which is returned by the authorize service.
API-KEY	String	The <b>api_key</b> of the user which is returned by the authorize service.

#### URL Parameters

Field	Description
email	Email address of the user. This has to be unique because users are selected via their email addresses in enCloud.
customer_id	Required field. The <b>customer unique id</b> that has been set before the integration. In case the user is not present, if the customer id is provided, a new user is created with the given email address.
device_id	Adding this field will cause the redirection to go to the device specific page. Else redirection will be sent to the default login page (usually the devices list). The device ID is the UID. It is put in the same format as the devices field in the Device Pull Information Method. Ex: 00A0EB802700

#### Body

No request body.

#### Example

Type	Example
cURL	<pre>curl -X GET \ 'http://myenCloud.com/integration/hot link?email=sedosgungor%40hotmail.com&amp;customer_id=YRK15' \ -H 'api-key: 30e087a1db41aca5942f6ae72f10db3facc409ed' \ -H 'authorization: SSryCZZzDTH5FyQvplRBAg' \</pre>
HTTP	<pre>GET /integration/hot link?email=sedosgungor@hotmail.com&amp;customer_id=YRK15 HTTP/1.1 Host: myenCloud.com Authorization: SSryCZZzDTH5FyQvplRBAg API-KEY: 30e087a1db41aca5942f6ae72f10db3facc409ed</pre>

#### Responses

HTTP Code	Message	Description
-----------	---------	-------------

200	<pre>{ "token" :   "a112390masn_aww2adjckfj" }</pre>	The token that is going to be passed to the redirection link.
400	<pre>{   "status": 400,   "description": "Customer is not present",   "ivenCode": 0 }</pre>	Invalid Customer ID.
500	Internal Server Error	enCloud related error.

### Step 2: Redirection to enCloud

After getting the **token** from **Token Generation** service, the redirection to enCloud should take place. Please note that this is not an API call. This is a direct link that looks for the generated token. If the token is wrong, the user will be redirected to a '404 Page Not Found' page.

If the link and token are correct, enCloud will handle the authentication via the token provided in the link.

Method Type	URL	Parameter	Example
GET	myenCloud.com/redirect	token	myenCloud.com/redirect?token=6fd83bede4544943696875ee1d5c3945

### Diagram: Hot Link to enCloud

