

# Nescloud API

## Detalles de las llamadas a la Nescloud API:

Nescloud API tiene la siguiente estructura de recursos y métodos:

- /auth
  - /validate (POST)
- /networks (GET)
- /sensordata (GET)

A continuación, se describen los tres recursos:

### **/auth/validate**

Este recurso tiene un único método POST que permite obtener a un usuario el token asociado a su cuenta.

Este token es imprescindible para el resto de llamadas a la API. El token determina si el usuario puede acceder a la información que se pide en la llamada. Si las credenciales suministradas son correctas, el parámetro `login_status` del response body será True. De lo contrario este parámetro será False.

URL: <https://data.nescloud.net/auth/validate>

El request body debe tener el siguiente formato:

```
{
  "username": "user@mail.com",
  "password": "foo"
}
```

Siendo username y password las credenciales de usuario de la plataforma Nescloud.

La llamada necesita de un request header: “x-api-key“. Este header debe contener una API Key de AWS válida.

Si las credenciales son válidas, esta llamada devuelve el ID del usuario, su nombre y apellidos, las organizaciones a las que tiene acceso y sus IDs, el token para poder realizar el resto de llamadas a la API y cuando expira el token dado:

```
[
  {
    "login_status": true,
    "user_id": 65,
    "first_name": "app",
```

```
    "last_name": "movil",
    "organizations": [
      {
        "id": 2,
        "name": "Nespra Demo"
      }
    ],
    "api_token": "L5S6FzpgslWuBhRsKQfRx4rTXKXwQst-XV6oOKvTVGs",
    "expiration_time": 1596105471
  }
]
```

## /Networks

Este recurso tiene un único método GET que permite obtener la información de los Nesgates, Nesmotes, alarmas y alertas de una organización.

URL: <https://data.nescloud.net/networks>

Este método requiere de un solo query parameter:

- **organisation\_id**: Solo se cogerá la información de las redes de la organización especificada. Nótese que el token suministrado a la llamada debe pertenecer a un usuario de la organización, de lo contrario la llamada devolverá un error conforme no se puede acceder a los datos.

De cara a identificar y autorizar esta llamada, también son necesarios dos request headers:

- **“x-api-key“** : Este header debe contener una API Key de AWS válida.
- **“Authorization“** : En este header se ha de poner el **“api\_token”** devuelto por el método **/auth/validate**

La respuesta a esta llamada tiene el formato:

```
[
  {
    "name": "Sevilla",
    "id": 7,
    "nesgates": [
      {
        "id": 5,
        "device_status": 1,
        "device_name": "Santiago",
        "deveui": "nespra_demo",
        "name_model": "NG-ENTERPRISE-LTE-868-EU-GB",
        "public_ip": "",
        "connectivity": "ETHERNET",
        "tx_total": 0,
        "timezone": "Europe/Madrid"
      }
    ]
  },
]
```

## Documentación de las API de datos de la plataforma Nescloud. Versión 1.1

```
"nesmotes": [
  {
    "device_status": 1,
    "id": 6,
    "device_name": "Ascensor 0003",
    "deveui": "34-35-31-31-74-37-78-11",
    "name_model": "NM-11-HW",
    "parent_id": 5,
    "parent_name": "Santiago",
    "sensor_labels": [
      "AI1",
      "AI2",
      "AI3",
      "AI4"
    ],
    "sensor_types": [
      "Conductividad",
      "Nivel",
      "pH",
      "Caudal"
    ],
    "units": [
      "uS/cm",
      "m",
      "pH",
      "m3/h"
    ]
  },
  {
    "device_status": 1,
    "id": 7,
    "device_name": "Ascensor 0004",
    "deveui": "70-b3-d5-7b-a0-00-17-ab",
    "name_model": "NM-WS11",
    "parent_id": 5,
    "parent_name": "Santiago",
    "sensor_labels": [
      "precipitation",
      "wind_speed",
      "temperature",
      "pressure",
      "humidity"
    ],
    "sensor_types": [
      "precipitation",
      "wind speed",
      "Temperature",
      "Pressure",
      "Humidity"
    ]
  }
],
"sensors": 9,
"number_of_alerts": 1,
"number_of_alarms": 2,
>alerts": [],
```

```
    "alarms": []  
  }  
]
```

## **/sensordata**

Este recurso tiene un único método GET que permite obtener la información de los sensores de un Nesmote.

Los parámetros de la query son:

- “deveui” es el DevEUI del Nesmote, y ha de pertenecer a la organización del usuario. Dependiendo del rol del usuario, también será necesario que el Nesmote pertenezca a una de las networks que el usuario puede ver. El formato de este dato se puede ver en la foto de arriba, no hace falta poner comillas.
- “t0” define el instante inicial de adquisición de datos. Debe ser dado como UNIX timestamp tal como se ve en la foto de arriba. No puede ser mayor que “t1”.
- “t1” define el instante final de adquisición de datos. Debe ser dado como UNIX timestamp tal como se ve en la foto de arriba. No puede ser menor que “t0”. El intervalo de tiempo entre “t0” y “t1” no puede ser mayor a 180 días.
- “sensor\_labels“ define los sensores de los cuales queremos obtener la información. Debe darse con el formato que se puede observar en la foto de arriba, cada nombre de sensor ha de estar separado por una coma (e.g DO1, AI1).
- “hourly“ define si queremos buscar en la base de datos horaria o no. Este parámetro es opcional, si no se define, por defecto se pillan los datos de la base de datos normal. Para obtener datos de la base de datos horaria, “hourly” debe ser True.

La llamada necesita de dos request headers:

- “x-api-key“. Este header debe contener una API Key de AWS válida
- “Authorization“ : En este header se ha de poner el “api\_token” devuelto por el método **/auth/validate**

Si el usuario asociado al token tiene visibilidad del Nesmote porque está en su organización y su rol lo permite ver, entonces se devolverán los datos de los sensores, de lo contrario se notificará al usuario en la respuesta que ese token no permite ver los sensores del Nesmote dado.

La respuesta de esta llamada tiene el siguiente formato con “hourly a False, donde el TimeStamp está en formato UNIX en segundos:

```
[  
  {  
    "TimeStamp": 1601480532,  
    "AI2": 36.274,  
    "AI1": 11.004  
  },  
  {  
    "TimeStamp": 1601480592,
```

## Documentación de las API de datos de la plataforma Nescloud. Versión 1.1

```
    "AI2": 37.852,  
    "AI1": 10.993  
  }  
]
```

En caso de que “hourly esté a True:

```
[  
  {  
    "yymmddhh": 20051908,  
    "AI2": 36.274,  
    "AI1": 11.004  
  },  
  {  
    "yymmddhh": 20051909,  
    "AI2": 37.852,  
    "AI1": 10.993  
  }  
]
```

En este último caso “yymmddhh” hace referencia al año, mes, día y hora.