NOBIL API

Version 3.0

rev. 03.06.2024

Client API version 3.0

NOBIL is a comprehensive and trustworthy database covering most of the public charging stations in Norway and Sweden. Based on admission to the data, providers and clients can develop new and innovative services for promoting use of electric vehicles.

NOBIL is owned by the Norwegian governmental agency Enova. Maintenance is done by the Norwegian EV Association through a public tender. The Swedish governmental agency Energimyndigheten is maintaining the database in Sweden. The database is designed for the Nordics, and we are looking for country administrators in Denmark, Finland and Iceland.

What is an API?

API stands for Application Programming Interface and is a piece of software which enables the use of existing systems for a 3rd Party.

How to use the client API

By using http call with different parameters, you can ask the service to return charging station data.

What is needed?

Clients who want to yse the API call service have to apply on NOBIL.no. The data required for registration are as follows:

- First name, last name
- Email address
- Mobile Number
- Website Address (where service is to be run)

After the registration is approved by a NOBIL admin you will be sent an API key by email within two working days. Please check your spam-folder if you have not received a key. This key must be attached to each request against NOBIL API. This is to identify who uses the service and to ensure NOBIL's data and service. API key is a string. For those who use Google's APIs, this is familiar material.

Ver 3.0. has a larger dataset both for charging stations and points. Availability data (status) are included where they are reported to NOBIL. An important change from Ver 1.1 to 3.0 is that all data types now are displayed in English. If a user has written data in his og her native language in the database, those data will get back in that native language.

Another new feature is that JSONP is a possible method of retrieving API data for all the AJAX functions. This makes it possible to get data via AJAX from different domains than NOBIL, without having to write a server side proxy functionality. Read more on **JSONP.**

Server side proxy functionalities in services consuming the NOBIL API, could implement better caching mechanisms, which can result in better response and a better user experience.

NOBIL's API supports both HTTP GET and HTTP POST requests for getting data.

Search for charger stations via map references

An example how to do this with AJAX and jQuery:

```
jQuery.ajax({
            type: 'POST',
            url: 'https://nobil.no/api/server/search.php',
           data: { 'apikey': nobilApiKey, 'apiversion': '3',
                      'action': "search",
                       'type': 'rectangle',
                       'northeast': '(59.943921193288915, 10.826683044433594)',
'southwest': '(59.883683240905256, 10.650901794433594)',
                      'existingids': '189,195,199,89,48',
                      ‘limit’:’10’},
            dataType: 'json'
           });
```

Parameter	**Example values**
action	search
apikey	APIkey_that_You_have_received_from_Nobil.no
apiversion	3
existingids	189,195,199,89,48,190,58,77,83,75,63,64,86,181,207,182,180 A list of charger station id's that you don't want back from the server, cause you already have received those and stored them in client store. This is done for saving bandwidth and increase the speed of showing and navigating in online maps like Google Maps.
northeast	(59.943921193288915, 10.826683044433594)
southwest	(59.883683240905256, 10.650901794433594)
type	Rectangle
limit (optional)	10. Default value is set to 1000
format (optional)	json or xml

API service response from map area search

The response content type from this type of search is json object.

Example of a JSON object from the server:

[{"Provider":"NOBIL.no","Rights":"Creative Commons 3.0 Unported License","apiver":"3", "chargerstations": [{"csmd": {"id":41,"name":"IKEA Slependen","url":"http:\/\/www.ladestasjoner.no\/2010\/06\/ikea-slependen_08.html","blogpostid":"3599822862656171588","Active":true,"Street":"Nesbruveien ","House_number":"40","Zipcode":"1396","City":"BILLINGSTAD","Municipality_ID":"0220","Municipality":"ASKER","County_ID":"02","County":"Akershus","Description_of_location":"Tre plasser i parkeringshuset P1, rett til h\u00f8yre etter innkj\u00f8ring. Slependen. Det finnes ogs\u00e5 3 ladeplasser p\u00e5 omr\u00e5de L, n\u00e6rmest \"utgang\" i P-kjelleren.","Owned_by":"IKEA","Number_charging_points":6,"Position":"(59.87447,10.49982)","Image":"41.jpg","Available_charging_points":6,"User_comment":"","Contact_info":"48050000 - kar fra IKEA som ringte om videre utbygging.","Created":"2010-03-24 10:24:13","Created_by":4,"Created_by_Forename":"nobilAdmin","Created_by_Surname":"nobilAdmin","Updated":"2011-04-15 12:19:20","Updated_by":5,"Updated_by_Forename":"Hans H.","Updated_by_Surname":"Kvisle","Station_status":1,"Land_code":"nor","International_id":"NOR_00041"}, "attr": {"st":{"24":{"attrtypeid":"24","attrname":"Open 24h","attrvalid":"1","trans":"Yes","attrval":"true"},"21":{"attrtypeid":"21","attrname":"Real-time information","attrvalid":"2","trans":"No","attrval":""},"22":{"attrtypeid":"22","attrname":"Public funding","attrvalid":"4","trans":"None","attrval":""},"7":{"attrtypeid":"7","attrname":"Parking fee","attrvalid":"2","trans":"No","attrval":false},"6":{"attrtypeid":"6","attrname":"Time limit","attrvalid":"2","trans":"No","attrval":""},"3":{"attrtypeid":"3","attrname":"Location","attrvalid":"4","trans":"Shopping center","attrval":""},"2":{"attrtypeid":"2","attrname":"Availability","attrvalid":"1","trans":"Public","attrval":""}}, "conn":{"1":{"attrtypeid":"1","attrname":"Accessability","attrvalid":"1","trans":"Open","attrval":""},"4":{"attrtypeid":"4","attrname":"Connector","attrvalid":"14","trans":"Schuko CEE 7\/4","attrval":""},"5":{"attrtypeid":"5","attrname":"Charging capacity","attrvalid":"7","trans":"230V 1-phase max 16A","attrval":""},"17":{"attrtypeid":"17","attrname":"Vehicle type","attrvalid":"1","trans":"All vehicles","attrval":""},"20":{"attrtypeid":"20","attrname":"Charge mode","attrvalid":"1","trans":"Mode 1","attrval":""},"18":{"attrtypeid":"18","attrname":"Reservable","attrvalid":"2","trans":"No","attrval":""},"25":{"attrtypeid":"25","attrname":"Fixed cable","attrvalid":"2","trans":"No","attrval":""}}}}]

Cross domain Ajax - JSONP

Example of how this can be implemented across different domains using Ajax and JSONP and jQuery:

```
jQuery.ajax({
            url: 'https://nobil.no/api/server/search.php',
            data: {'apikey': nobilApiKey,
                         'apiversion': '3',
                  'action': "search",
                         'type': 'rectangle',
                         'northeast': bound.getNorthEast(),
                         'southwest': bound.getSouthWest(),
                         'existingids': idList},
            success: parseJsonResponse,
            dataType: 'jsonp'
});
```

Parameter	**Example values**
action	search
apikey	APIkey_that_You_have_received_from_Nobil.no
apiversion	3
existingids	189,195,199,89,48,190,58,77,83,75,63,64,86,181,207,182,180 A list of charger station id's that you don't want back from the server, cause you already have received those and stored them in client store. This is done for saving bandwidth and increase the speed of showing and navigating in online maps like Google Maps.
northeast	(59.943921193288915, 10.826683044433594)
southwest	(59.883683240905256, 10.650901794433594)
type	Rectangle
limit (optional)	10. Default value is set to 1000
format (optional)	json or xml

Service response from rectangle map search

The response from the search service is of the type jsonp object. This object is encapsulated in a callback method that is given by the client to the search method as a parameter. In the search request seen above, this function is called «parseJsonResponse».

Response example for the above search request:

jsonp1300629903865([{"id":228,"active":true,"name":"Statens arbeidsmilj\u00f8institutt, Oslo","address":"Gydasvei 8","zip":"0383","city":"OSLO","municipalityid":"0301","municipality":"OSLO","countyid":"03","county":"Oslo","description":"","placetype":"Street","accessibility":"Visitors","owner":"Statens arbeidsmilj\u00f8institutt","chargerpointnumber":4,"parkingfee":false,"timelimit":9,"chargerspeed":"16A","geolocation":"(59.93255,10.71514)","constructionsupportname":"Oslo kommune","image":"228.jpg","usercomment":"Tilgjengelig 07-16.00","contactinfo":"Statens arbeidsmilj\u00f8institutt\r\nSteinar Messel\r\nsteinarm@stami.no\r\n23195153","created":"2010-04-14 12:55:02","createdbygivenname":"Hans H.","createdbyfamilyname":"Kvisle","updated":"2010-06-04 09:56:44","updatedbygivenname":"Hans H.","updatedbyfamilyname":"Kvisle","accesstype":"Standard key","url":"http:\/\/www.ladestasjoner.no\/2010\/07\/statens-arbeidsmiljinstitutt-oslo_06.html","countrycode":"NO"}]);

Get all data on a given charger station

Example implementation done with AJAX and jQuery:

This example gets all data on charger station with id: NOR_00171

```
jQuery.ajax({
type: 'POST',
url: 'https://nobil.no/api/server/search.php',
data: {
        'apikey': nobilApiKey, 'apiversion': '3', 'action': "search",
        'type':'id', 'id': 'NOR_00171'
        },
success: printJsonResponse,
dataType: 'json'
});
```

Parameters	Example values
action	search
apikey	API key
apiversion	3
type	id
id	NOR_00171
Callback function on success	printJsonResponse
format (optional)	json or xml

The function below gets called if the request returns data of type JSON. The function iterates over all object values an adds these to a string. This string then gets displayed in a html div with ID="jsonOutput".

```
function printJsonResponse(data, textStatus, XMLHttpRequest){
        jQuery("#jsonOutput").html(dump(data));
}

function dump(arr,level) {
  var dumped_text = "";
  if(!level) level = 0;

  //The padding given at the beginning of the line.
  var level_padding = "";
  for(var j=0;j<level+1;j++) level_padding += "    ";

  if(typeof(arr) == 'object') { //Array/Hashes/Objects
    for(var item in arr) {
      var value = arr[item];

      if(typeof(value) == 'object') { //If it is an array,
        dumped_text += level_padding + "'" + item + "' ...\n";
        dumped_text += dump(value,level+1);
      } else {
        dumped_text += level_padding + "'" + item + "' => \"" + value + "\"\n";
      }
    }
  } else { //Stings/Chars/Numbers etc.
    dumped_text = "===>"+arr+"<===("+typeof(arr)+")";
  }
  return dumped_text;
}
```

Retrieving statistics for charger stations

The following statistics search method can be used:

stats_TotalsAllCounties

Gets the total count of active charger stations grouped by counties.

```
jQuery.ajax({
type: 'POST',
url: 'https://nobil.no/api/server/search.php',
data: {
        'apikey': nobilApiKey, 'apiversion': '3',
        'action': "search",
        'type':'stats_TotalsAllCounties',
        'countrycode':'NOR'
        },
success: printJsonResponse,
dataType: 'json'
});
```

Parameters	**Example values**
apikey	API key
apiversion	3
action	search
type	stats_TotalsAllCounties
countrycode	NOR (ISO 639-2)
format (optional)	json or xml (default json)
Callback function on success	printJsonResponse

stats_TotalsByCountyId

Gets the total count of active charger stations for a given county specified by county id. (11=Rogaland)

```
jQuery.ajax({
type: 'POST',
url: 'https://nobil.no/api/server/search.php',
data: {
        'apikey': nobilApiKey, 'apiversion': '3',
        'action': "search",
        'type':'stats_TotalsByCountyId',
        'id': '11',
        'countrycode':'NOR'
        },
success: printJsonResponse,
dataType: 'json'
});
```

Parameters	**Example values**
apikey	API key
apiversion	3
action	search
type	stats_TotalsByCountyId
id	11
countrycode	NOR (ISO 639-2)
format (optional)	json or xml (default json)
Callback function on success	printJsonResponse

stats_DetailedTotalsByCountyId

Gets the total count of active charger stations for a given county grouped by municipality specified by county id. (11=Rogaland)

```
jQuery.ajax({
type: 'POST',
url: 'https://nobil.no/api/server/search.php',
data: {
            'apikey': nobilApiKey, 'apiversion': '3',
            'action': "search",
            'type':'stats_DetailedTotalsByCountyId',
            'id': '11',
            'countrycode':'NOR'
            },
success: printJsonResponse,
dataType: 'json'
});
```

Parameters	**Example values**
apikey	API key
apiversion	3
action	search
type	stats_DetailedTotalsByCountyId
id	11
countrycode	NOR (ISO 639-2)
format (optional)	json or xml (default json)
Callback function on success	printJsonResponse

stats_TotalsByMunicipalId

Get the total count of active charger stations for a given municipality specified by municipality id. (1103=Stavanger)

```
jQuery.ajax({
type: 'POST',
url: 'https://nobil.no/api/server/search.php',
data: {
        'apikey': nobilApiKey, 'apiversion': '3',
        'action': "search",
        'type':'stats_TotalsByMunicipalId',
        'id': '1103',
        'countrycode':'NOR'
        },
success: printJsonResponse,
dataType: 'json'
});
```

Parameters	**Example values**
apikey	API key
apiversion	3
action	search
type	stats_TotalsByMunicipalId
id	11
countrycode	NOR (ISO 639-2)
format (optional)	json or xml (default json)
Callback function on success	printJsonResponse

stats_DetailTotalsByMunicipalId

Get the total count of active charger stations for all zip codes for a given municipality specified by municipality id. (1103=Stavanger)

```
jQuery.ajax({
type: 'POST',
url: 'https://nobil.no/api/server/search.php',
data: {
            'apikey': nobilApiKey, 'apiversion': '3',
            'action': "search",
            'type':'stats_DetailTotalsByMunicipalId',
            'id': '1103',
            'countrycode':'NOR'
            },
success: printJsonResponse,
dataType: 'json'
});
```

Parameters	Example values
apikey	API key
apiversion	3
action	search
type	stats_DetailTotalsByMunicipalId
id	11
countrycode	NOR (ISO 639-2)
format (optional)	json or xml (default json)
Callback function on success	printJsonResponse

The nearest charging station

Get the 10 closest charging stations which are within 2000 meters, ordered by meters, nearest first.

```
jQuery.ajax({
                type: 'POST',
                url: 'https://nobil.no/api/server/search.php',
                data: {
                        'apikey': nobilApiKey,
                        'apiversion': '3',
                        'action': "search",
                        'type':'near',
                        'lat':'59.91673',
                        'long': '10.74782',
                        'distance': '2000',
                        'limit': '10'
                        },
                success: printJsonResponse,
                dataType: 'json'
                });
```

Parameters	**Example values**
apikey	API key
apiversion	3
action	search
type	near
lat	59.91673
long	10.74782
distance	2000 (metre)
format (optional)	json or xml (default json)
Callback function on success	printJsonResponse

NOBIL Data Export

URL: https://nobil.no/api/server/datadump.php

Parameters	**Example values**
apikey	API key
countrycode (optional)	NOR (if no parameter is submitted to the server, all data from all countries is returned)
fromdate (optional)	2012-06-02
format (optional)	json (default format is XML)
file	false (default value is 'true', which returns a file, rather than the text stream)
callback (function on success)	PrintJsonResponse (wrapping data via jsonp)
norealtime	true|false (Filter out realtime stations from resultset)
nonimupdate	true|false (Filter out station that have been updated by realtime data)

URL examples:

https://nobil.no/api/server/datadump.php?apikey=2048b60b804ac019155675421c0ddb13&countrycode=NOR&fromdate=2012-06-02&format=xml&file=false

https://nobil.no/api/server/datadump.php?apikey=2048b60b804ac019155675421c0ddb13

https://nobil.no/api/server/datadump.php?apikey=2048b60b804ac019155675421c0ddb13&fromdate=2012-06-02&file=false&callback=mycallback&format=json

NOBIL station statistics

URL: https://nobil.no/api/server/search.php

type	params
stats_GetSumChargerstations	countrycode (ISO639-2) ex.: &countrycode=NOR (Norway)
stats_GetCountChargerConnectors	countrycode (ISO639-2) ex.: &countrycode=SWE (Sweden)
stats_GetCountPublicChargerConnectors	countrycode (ISO639-2) ex.: &countrycode=FIN (Finland)
stats_GetCountChargerConnectorsTypes	countrycode (ISO639-2) ex.: &countrycode=DAN (Denmark)
stats_GetCountChargersByCapasity	countrycode, capacityids ex.: countrycode=FIN&capacityids =capacityid,capacityid
stats_GetCountChargersModeOneTwo	countrycode, ids ex.: &countrycode=FIN&ids=1,2
stats_CountChargeModeThreeAndCapasity	countrycode, capacityids
stats_GetCountChargersByConnectorTypeAndCapasity	countrycode, capacityid, connectorid
stats_GetCountChargerConnectorsByConnectorTypeAndCapasity	countrycode, capacityid, connectorid
stats_GetCountRealtimeChargers	countrycode, daysold
stats_GetStatisticsTotalConnectorsPRMunicipalityPrCountyPrCountry	countrycode, countycode, municipalitycode
stats_GetStatisticsCountChargerConnectorsByCountryByMunicipalityByCountyByChargerTypes	countrycode, countycode, municipalitycode

Gets the total count of active charger stations for all zip codes for a given municipality specified by municipality id.

https://nobil.no/api/server/search.php?apiversion=3&apikey=KEY&action=search&type=stats_GetCountChargerConnectorsByConnectorTypeAndCapasity&countrycode=NOR&countycode=11&municipalitycode=1103&format=json

(countycode:11=Rogaland, municipalitycode:1103=Stavanger)

Capacity ids and connector ids can be found here:

https://nobil.no/admin/attributes.php

NOBILDBs changelog

Gets a changelog from the NobilDB generated by human interactions in the NobilAdmin CRUD style.

https://nobil.no/api/server/changelog.php?apikey=KEY&timestamp=2017-12-01%2017:30:00

timestamp	Max 7 days < todays date. Ex: &timestamp=2017-12-01%2017:30:00

Example implementation

Demonstration of these examples can be found here:

https://www.nobil.no/api/client/search_apiVer3.php

Attribute reference

An overview of all attributes and valid values can be found here:
Production server: https://nobil.no/admin/attributes.php

NOBIL Real-time data stream service

Nobil Real-time data streams is a service that provides status updates on charging stations. It is closely connected to data stored in NOBIL.

Technology

NOBIL Real-time data stream service is built on Enova's platform solution. The underlying technology is WebSockets, and there are several potential ways to integrate using preferred languages.

Here are examples of how to easily connect to a stream with various languages:

Microsoft Learn

The examples include retrieving a URL from an Azure Web PubSub Service Client, but this will not be relevant for NOBIL Real-time, as the WebSocket client's URL is retrieved via the API on the "NOBIL Real-time" product at data.enova.no.

Data Stream

Format

The contract for the data stream is defined as follows:

```
{
   "nobilId": "string",
   "evseUId": "string",
   "status": "string"
}
```

Definition of the attributes in the real-time data stream are as follow:

nobilId	is the international version of the ID in NOBIL,

	for example, **NOR_23314**
evseUid	is the unique identifier of the EVSE the update pertains to, for example, **7381cc78-a032-4f23-a46f-3e0ebe4590ff**
status	is one of the statuses listed below: "**AVAILABLE**" - Evse is able to start a new charging session "**BLOCKED**" - Evse is not accessible because of a physical barrier, i.e. a car "**CHARGING**" - EVSE is in use "**INOPERATIVE**" - EVSE is not yet active or it is no longer available (deleted) "**OUTOFORDER**" - EVSE is currently out of order "**PLANNED**" - EVSE is planned, will be operating soon "**REMOVED**" - EVSE is discontinued/removed. "**RESERVED**" - EVSE is reserved for a particular EV driver and is unavailable for other drivers "**UNKNOWN**" - No status information available. (Also used when offline)

Getting Started with NOBIL Real-time data stream service

To get started with NOBIL Real-time data stream service, you follow the steps below. Bear in mind that test environment and production environment are not connected in regards to account details and api-keys. You need to sign up for both. Aslo, there are currently no data flowing in Enova’s test environment. Use this test environment to ensure understanding of the procedure to connect to the data stream.

1. New User

Register a new user at https://test.data.enova.no/signup. Here you also approve the general terms for data shared via Enova.

2. Subscribe to Product

Go to https://test.data.enova.no/products and find "NOBIL Real-time". Create a subscription for the product. This will give you an API key that must be used every time you want to retrieve a connection URL for the data stream.

3. Get Connection URL

Once you have a subscription, you can use the API key to retrieve an access token to connect to the stream of real-time updates.

4. Connect to Stream

Use the access token from the API response to connect to the stream of real-time updates via standard WebSockets.

5. Go into Production

Currently, there is no test data on the data stream in the test environment, but it is good to use to ensure that the connection to the API and stream works.

Using real production data follows exactly the same steps as for testing, but then you refer to https://data.enova.no. For production data - signup to a production account at https://data.enova.no/signup

Postman

After registering a user and subscribing to the product, it's easy to test using, for example, Postman from your PC.

1. Create a new GET request to https://test.data.enova.no/real-time/v1/Realtime
2. Add an Authorization header "x-api-key" and your API key
3. Press "Send," and you'll receive a URL back starting with "wss".
4. Click New -> WebSocket in Postman, paste the link, and press "Connect"
5. You are connected to the stream, but as stated earlier, there are no data in the TEST environment at this point, so this is just for testing the flow. Doing the same procedure against the production environment will show actual data.