**Nominatim**

**Nominatim** (from the Latin, 'by name') is a tool to search OSM data by name and address and to generate synthetic addresses of OSM points (reverse geocoding). It can be found at [nominatim.openstreetmap.org](https://nominatim.openstreetmap.org).

Nominatim is also used as one of the sources for the search box on the [OpenStreetMap home page](https://www.openstreetmap.org). Several companies provide hosted instances of Nominatim that you can query via an API, see section Alternatives below.

This page provides usage instructions. For details of how Nominatim works please see the [Development Overview](https://wiki.openstreetmap.org/wiki/Nominatim/Development_overview) and there is a short [FAQ](https://wiki.openstreetmap.org/wiki/Nominatim/FAQ). There is also a list of [language mappings per country](https://wiki.openstreetmap.org/wiki/Nominatim/Country_Codes), some experimental [address formats per country](https://wiki.openstreetmap.org/wiki/Nominatim/Country_Address_Format) and a [list of abbreviations](https://wiki.openstreetmap.org/wiki/Name_finder:Abbreviations).

**Содержание**

* [1 Search](https://wiki.openstreetmap.org/wiki/Nominatim#Search)
  + [1.1 Special Keywords](https://wiki.openstreetmap.org/wiki/Nominatim#Special_Keywords)
  + [1.2 Parameters](https://wiki.openstreetmap.org/wiki/Nominatim#Parameters)
  + [1.3 Examples](https://wiki.openstreetmap.org/wiki/Nominatim#Examples)
* [2 Reverse Geocoding](https://wiki.openstreetmap.org/wiki/Nominatim#Reverse_Geocoding)
  + [2.1 Parameters](https://wiki.openstreetmap.org/wiki/Nominatim#Parameters_2)
  + [2.2 Example](https://wiki.openstreetmap.org/wiki/Nominatim#Example)
  + [2.3 Example with format=jsonv2](https://wiki.openstreetmap.org/wiki/Nominatim#Example_with_format.3Djsonv2)
  + [2.4 Hierarchy](https://wiki.openstreetmap.org/wiki/Nominatim#Hierarchy)
* [3 Address lookup](https://wiki.openstreetmap.org/wiki/Nominatim#Address_lookup)
  + [3.1 Parameters](https://wiki.openstreetmap.org/wiki/Nominatim#Parameters_3)
  + [3.2 Example](https://wiki.openstreetmap.org/wiki/Nominatim#Example_2)
* [4 Details / Gazetteer](https://wiki.openstreetmap.org/wiki/Nominatim#Details_.2F_Gazetteer)
* [5 Usage Policy](https://wiki.openstreetmap.org/wiki/Nominatim#Usage_Policy)
* [6 Source Code](https://wiki.openstreetmap.org/wiki/Nominatim#Source_Code)
* [7 Bugs / Error reporting](https://wiki.openstreetmap.org/wiki/Nominatim#Bugs_.2F_Error_reporting)
* [8 Alternatives / Third-party providers](https://wiki.openstreetmap.org/wiki/Nominatim#Alternatives_.2F_Third-party_providers)

**Search**

Nominatim indexes named (or numbered) features with the OSM data set and a subset of other unnamed features (pubs, hotels, churches, etc)

Search terms are processed first left to right and then right to left if that fails.

Both searches will work: [pilkington avenue, birmingham](https://nominatim.openstreetmap.org/search?q=pilkington+avenue,birmingham) [birmingham, pilkington avenue](https://nominatim.openstreetmap.org/search?q=birmingham,+pilkington+avenue)

(Commas are optional, but improve performance by reducing the complexity of the search.)

Where house numbers have been defined for an area they should be used: [135 pilkington avenue, birmingham](https://nominatim.openstreetmap.org/search?q=135+pilkington+avenue,+birmingham)

**Special Keywords**

Various keywords are translated into searches for specific osm tags (e.g. Pub => amenity=pub). A current list of [special phrases](https://wiki.openstreetmap.org/wiki/Nominatim/Special_Phrases) processed is available.

**Parameters**

<https://nominatim.openstreetmap.org/search>?<params>

<https://nominatim.openstreetmap.org/search/><query>?<params>

format=[**html**|xml|json|jsonv2]

Output format

json\_callback=<string>

Wrap json output in a callback function (JSONP) i.e. <string>(<json>)

accept-language=<browser language string>

Preferred language order for showing search results, overrides the value specified in the "Accept-Language" HTTP header.

Either uses standard rfc2616 accept-language string or a simple comma separated list of language codes.

q=<query>

Query string to search for. Alternatively can be entered as:

street=<housenumber> <streetname>  
city=<city>  
county=<county>  
state=<state>  
country=<country>  
postalcode=<postalcode>

**(experimental)** Alternative query string format for structured requests.

Structured requests are faster and require fewer server resources.

For North American addresses, use the full name for state to improve the results.

DO NOT COMBINE WITH q=<query> PARAMETER.

countrycodes=<countrycode>[,<countrycode>][,<countrycode>]...

Limit search results to a specific country (or a list of countries).

<countrycode> should be the ISO 3166-1alpha2 code, e.g. gb for the United Kingdom, de for Germany, etc.

viewbox=<x1>,<y1>,<x2>,<y2>

The preferred area to find search results. Any two corner points of the box are accepted in any order as long as they span a real box.

bounded=[**0**|1]

Restrict the results to only items contained with the viewbox (see above).

Restricting the results to the bounding box also enables searching by amenity only.

For example a search query of just "[pub]" would normally be rejected but with bounded=1 will result in a list of items matching within the bounding box.

polygon=[**0**|1]

Output polygon outlines for items found

**(deprecated, use one of the polygon\_\* parameters instead)**

addressdetails=[**0**|1]

Include a breakdown of the address into elements

email=<valid email address>

If you are making large numbers of request please include a valid email address or alternatively include your email address as part of the User-Agent string.

This information will be kept confidential and only used to contact you in the event of a problem, see [Usage Policy](https://operations.osmfoundation.org/policies/nominatim/) for more details.

exclude\_place\_ids=<place\_id,[place\_id],[place\_id]>

If you do not want certain openstreetmap objects to appear in the search result, give a comma separated list of the place\_id's you want to skip. This can be used to broaden search results. For example, if a previous query only returned a few results, then including those here would cause the search to return other, less accurate, matches (if possible)

limit=<integer>

Limit the number of returned results. Default is 10.

dedupe=[0|**1**]

Sometimes you have several objects in OSM identifying the same place or object in reality. The simplest case is a street being split in many different OSM ways due to different characteristics.

Nominatim will attempt to detect such duplicates and only return one match; this is controlled by the dedupe parameter which defaults to 1. Since the limit is, for reasons of efficiency, enforced before and not after de-duplicating, it is possible that de-duplicating leaves you with less results than requested.

debug=[**0**|1]

Output assorted developer debug information. Data on internals of nominatim "Search Loop" logic, and SQL queries. The output is (rough) HTML format. This overrides the specified machine readable format.

polygon\_geojson=1

Output geometry of results in geojson format.

polygon\_kml=1

Output geometry of results in kml format.

polygon\_svg=1

Output geometry of results in svg format.

polygon\_text=1

Output geometry of results as a WKT.

extratags=1

Include additional information in the result if available, e.g. wikipedia link, opening hours.

namedetails=1

Include a list of alternative names in the results.

These may include language variants, references, operator and brand.

**Examples**

<https://nominatim.openstreetmap.org/search?q=135+pilkington+avenue,+birmingham&format=xml&polygon=1&addressdetails=1>

<https://nominatim.openstreetmap.org/search/135%20pilkington%20avenue,%20birmingham?format=xml&polygon=1&addressdetails=1>

<https://nominatim.openstreetmap.org/search/gb/birmingham/pilkington%20avenue/135?format=xml&polygon=1&addressdetails=1>

<searchresults timestamp="Sat, 07 Nov 09 14:42:10 +0000" querystring="135 pilkington, avenue birmingham" polygon="true">

<place

place\_id="1620612" osm\_type="node" osm\_id="452010817"

boundingbox="52.548641204834,52.5488433837891,-1.81612110137939,-1.81592094898224"

polygonpoints="[['-1.81592098644987','52.5487429714954'],['-1.81592290792183','52.5487234624632'],...]"

lat="52.5487429714954" lon="-1.81602098644987"

display\_name="135, Pilkington Avenue, Wylde Green, City of Birmingham, West Midlands (county), B72, United Kingdom"

class="place" type="house">

<house\_number>135</house\_number>

<road>Pilkington Avenue</road>

<village>Wylde Green</village>

<town>Sutton Coldfield</town>

<city>City of Birmingham</city>

<county>West Midlands (county)</county>

<postcode>B72</postcode>

<country>United Kingdom</country>

<country\_code>gb</country\_code>

</place>

</searchresults>

<https://nominatim.openstreetmap.org/search/Unter%20den%20Linden%201%20Berlin?format=json&addressdetails=1&limit=1&polygon_svg=1>

[

{

"address": {

"city": "Berlin",

"city\_district": "Mitte",

"construction": "Unter den Linden",

"continent": "European Union",

"country": "Deutschland",

"country\_code": "de",

"house\_number": "1",

"neighbourhood": "Scheunenviertel",

"postcode": "10117",

"public\_building": "Kommandantenhaus",

"state": "Berlin",

"suburb": "Mitte"

},

"boundingbox": [

"52.5170783996582",

"52.5173187255859",

"13.3975105285645",

"13.3981599807739"

],

"class": "amenity",

"display\_name": "Kommandantenhaus, 1, Unter den Linden, Scheunenviertel, Mitte, Berlin, 10117, Deutschland, European Union",

"importance": 0.73606775332943,

"lat": "52.51719785",

"licence": "Data \u00a9 OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright",

"lon": "13.3978352028938",

"osm\_id": "15976890",

"osm\_type": "way",

"place\_id": "30848715",

"svg": "M 13.397511 -52.517283599999999 L 13.397829400000001 -52.517299800000004 13.398131599999999 -52.517315099999998 13.398159400000001 -52.517112099999999 13.3975388 -52.517080700000001 Z",

"type": "public\_building"

}

]

<https://nominatim.openstreetmap.org/?format=json&addressdetails=1&q=bakery+in+berlin+wedding&format=json&limit=1>

[

{

"address": {

"bakery": "B\u00e4cker Kamps",

"city\_district": "Mitte",

"continent": "European Union",

"country": "Deutschland",

"country\_code": "de",

"footway": "Bahnsteig U6",

"neighbourhood": "Sprengelkiez",

"postcode": "13353",

"state": "Berlin",

"suburb": "Wedding"

},

"boundingbox": [

"52.5460929870605",

"52.5460968017578",

"13.3591794967651",

"13.3591804504395"

],

"class": "shop",

"display\_name": "B\u00e4cker Kamps, Bahnsteig U6, Sprengelkiez, Wedding, Mitte, Berlin, 13353, Deutschland, European Union",

"icon": "https://nominatim.openstreetmap.org/images/mapicons/shopping\_bakery.p.20.png",

"importance": 0.201,

"lat": "52.5460941",

"licence": "Data \u00a9 OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright",

"lon": "13.35918",

"osm\_id": "317179427",

"osm\_type": "node",

"place\_id": "1453068",

"type": "bakery"

}

]

**Reverse Geocoding**

Reverse geocoding generates an address from a latitude and longitude. The optional zoom parameter specifies the level of detail required in terms of something suitable for a Leaflet.js/OpenLayers/etc. zoom level.

**Parameters**

<https://nominatim.openstreetmap.org/reverse>?<query>

format=[**xml**|html|json|jsonv2]

Output format, default = xml.

jsonv2 adds the next fields to response:

* place\_rank

The *place\_rank* is based on a rather complex algorithm taking the place type and various other attributes into account. For example it seems checks whether this object is a village, a city, a country, a continent, a highway, a lake and similar other properties. [[1]](https://wiki.openstreetmap.org/wiki/Nominatim#cite_note-1)

* category
* type
* importance

The *importance* is used for ordering search results according to their relevance. The importance value is calculated/estimated using various attributes including the place's popularity on Wikipedia and its *place\_rank*.[[2]](https://wiki.openstreetmap.org/wiki/Nominatim#cite_note-2)

* addresstype

json\_callback=<string>

Wrap json output in a callback function (JSONP) i.e. <string>(<json>)

accept-language=<browser language string>

Preferred language order for showing search results, overrides the value specified in the "Accept-Language" HTTP header.

Either uses standard rfc2616 accept-language string or a simple comma separated list of language codes.

osm\_type=[N|W|R]  
osm\_id=<value>

A specific osm node / way / relation to return an address for

**Please use this in preference to lat/lon where possible**

lat=<value> lon=<value>

The location to generate an address for

zoom=[0-18]

Level of detail required where 0 is country and 18 is house/building

addressdetails=[0|1]

Include a breakdown of the address into elements

email=<valid email address>

If you are making large numbers of request please include a valid email address

or alternatively include your email address as part of the User-Agent string.

This information will be kept confidential and only used to contact you in the

event of a problem, see Usage Policy for more details.

polygon\_geojson=1

Output geometry of result in geojson format.

polygon\_kml=1

Output geometry of result in kml format.

polygon\_svg=1

Output geometry of result in svg format.

polygon\_text=1

Output geometry of result as a WKT.

extratags=1

Include additional information in the result if available, e.g. wikipedia link, opening hours.

namedetails=1

Include a list of alternative names in the results.

These may include language variants, references, operator and brand.

**Example**

<https://nominatim.openstreetmap.org/reverse?format=xml&lat=52.5487429714954&lon=-1.81602098644987&zoom=18&addressdetails=1>

<reversegeocode timestamp="Fri, 06 Nov 09 16:33:54 +0000" querystring="...">

<result place\_id="1620612" osm\_type="node" osm\_id="452010817">

135, Pilkington Avenue, Wylde Green, City of Birmingham, West Midlands (county), B72, United Kingdom

</result>

<addressparts>

<house\_number>135</house\_number>

<road>Pilkington Avenue</road>

<village>Wylde Green</village>

<town>Sutton Coldfield</town>

<city>City of Birmingham</city>

<county>West Midlands (county)</county>

<postcode>B72</postcode>

<country>United Kingdom</country>

<country\_code>gb</country\_code>

</addressparts>

</reversegeocode>

**Example with format=jsonv2**

<https://nominatim.openstreetmap.org/reverse?format=jsonv2&lat=-34.44076&lon=-58.70521>

{

"place\_id":"134140761",

"licence":"Data © OpenStreetMap contributors, ODbL 1.0. http:\/\/www.openstreetmap.org\/copyright",

"osm\_type":"way",

"osm\_id":"280940520",

"lat":"-34.4391708",

"lon":"-58.7064573",

"place\_rank":"26",

"category":"highway",

"type":"motorway",

"importance":"0.1",

"addresstype":"road",

"display\_name":"Autopista Pedro Eugenio Aramburu, El Triángulo, Partido de Malvinas Argentinas, Buenos Aires, 1.619, Argentina",

"name":"Autopista Pedro Eugenio Aramburu",

"address":{

"road":"Autopista Pedro Eugenio Aramburu",

"village":"El Triángulo",

"state\_district":"Partido de Malvinas Argentinas",

"state":"Buenos Aires",

"postcode":"1.619",

"country":"Argentina",

"country\_code":"ar"

},

"boundingbox":["-34.44159","-34.4370994","-58.7086067","-58.7044712"]

}

**Hierarchy**

|  |  |
| --- | --- |
| **Admin level** | **XML entity** |
| 2 | <country> |
| 4 | <state> |
| 5 | <state\_district> |
| 6 |  |
| 7 | <county> |
| 8 | <village> |
| 9 | <city\_district> |
| 10 | <suburb> |

**Address lookup**

Lookup the address of one or multiple OSM objects like node, way or relation.

**Parameters**

<https://nominatim.openstreetmap.org/lookup>?<query>

format=[xml|json]

Output format

json\_callback=<string>

Wrap json output in a callback function (JSONP) i.e. <string>(<json>)

accept-language=<browser language string>

Preferred language order for showing search results, overrides the value specified in the "Accept-Language" HTTP header.

Either uses standard rfc2616 accept-language string or a simple comma separated list of language codes.

osm\_ids=[N|W|R]<value>,…,[N|W|R]<value>

A list of up to 50 specific osm node, way or relations ids to return the addresses for

addressdetails=[0|1]

Include a breakdown of the address into elements

email=<valid email address>

If you are making large numbers of request please include a valid email address or alternatively include your email address as part of the User-Agent string.

This information will be kept confidential and only used to contact you in the event of a problem, see Usage Policy for more details.

extratags=1

Include additional information in the result if available, e.g. wikipedia link, opening hours.

namedetails=1

Include a list of alternative names in the results.

These may include language variants, references, operator and brand.

**Example**

<https://nominatim.openstreetmap.org/lookup?osm_ids=R146656,W104393803,N240109189>

<lookupresults timestamp="Mon, 29 Jun 15 18:01:33 +0000" attribution="Data © OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright" querystring="R146656,W104393803,N240109189" polygon="false">

<place place\_id="127761056" osm\_type="relation" osm\_id="146656" place\_rank="16" lat="53.4791466" lon="-2.2447445" display\_name="Manchester, Greater Manchester, North West England, England, United Kingdom" class="boundary" type="administrative" importance="0.704893333438333">

<city>Manchester</city>

<county>Greater Manchester</county>

<state\_district>North West England</state\_district>

<state>England</state>

<country>United Kingdom</country>

<country\_code>gb</country\_code>

</place>

<place place\_id="77769745" osm\_type="way" osm\_id="104393803" place\_rank="30" lat="52.5162024" lon="13.3777343363579" display\_name="Brandenburg Gate, 1, Pariser Platz, Mitte, Berlin, 10117, Germany" class="tourism" type="attraction" importance="0.443472858361592">

<attraction>Brandenburg Gate</attraction>

<house\_number>1</house\_number>

<pedestrian>Pariser Platz</pedestrian>

<suburb>Mitte</suburb>

<city\_district>Mitte</city\_district>

<city>Berlin</city>

<state>Berlin</state>

<postcode>10117</postcode>

<country>Germany</country>

<country\_code>de</country\_code>

</place>

<place place\_id="2570600569" osm\_type="node" osm\_id="240109189" place\_rank="15" lat="52.5170365" lon="13.3888599" display\_name="Berlin, Germany" class="place" type="city" importance="0.822149797630868">

<city>Berlin</city>

<state>Berlin</state>

<country>Germany</country>

<country\_code>de</country\_code>

</place>

</lookupresults>

**Details / Gazetteer**

Currently the details page (details.php) creates a high load on the server and is only intended as a debugging aide, as a result no xml api is currently provided. Scraping or heavy use is likely to result in an IP ban.

**Usage Policy**

For the usage policy of nominatim.openstreetmap.org, please see [Nominatim usage policy](https://operations.osmfoundation.org/policies/nominatim/).

**Source Code**

Nominatim is based around the postgresql import utility [osm2pgsql](https://wiki.openstreetmap.org/wiki/Osm2pgsql) using the alternative gazetteer output option. Indexing and search are performed using a combination of C, plpgsql and php. The source can be found here:

<https://github.com/openstreetmap/Nominatim>

Full installation instructions can be found in [on the nominatim.org website](http://nominatim.org/release-docs/latest/admin/Installation/).

**Bugs / Error reporting**

Please report bugs and problems in the [Nominatim issue tracker on Github](https://github.com/openstreetmap/Nominatim/issues) (there are also [reports on trac.openstreetmap.org](https://trac.openstreetmap.org/query?status=!closed&component=nominatim)).

When you report unexpected search results, please include the following in your bug report:

* the exact search term you were using
* links to the OSM objects you expect to find and, optionally, a link to the OSM object that was actually found

**Alternatives / Third-party providers**

For slightly larger requirements you may be able to use one the various third-party providers, though of course, you will need to agree to their terms of service.

* [MapQuest Open](http://developer.mapquest.com/web/products/open/nominatim)
* [OpenCage Geocoder](https://opencagedata.com/)
* [LocationIQ](https://locationiq.org)

If your requirements are even larger, you can [install your own instance](http://nominatim.org/release-docs/latest/admin/Installation/) of Nominatim.