Geocoder Autocomplete API

Developer's Guide

Version 6.2.96

Contents

Legal Notices

Trademark Acknowledgements

HERE and Nokia are trademarks or registered trademarks of Nokia Corporation.

Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

This content is provided "as-is" and without warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, satisfactory quality and non-infringement. HERE does not warrant that the content is error free and HERE does not warrant or make any representations regarding the quality, correctness, accuracy, or reliability of the content. You should therefore verify any information contained in the content before acting on it.

To the furthest extent permitted by law, under no circumstances, including without limitation the negligence of HERE, shall HERE be liable for any damages, including, without limitation, direct, special, indirect, punitive, consequential, exemplary and/ or incidental damages that result from the use or application of this content, even if HERE or an authorized representative has been advised of the possibility of such damages.

Document Information

Product	
Name:	Geocoder Autocomplete API
Version:	Version 6.2.96
Document	
Name:	Geocoder Autocomplete API Developer's Guide
Id:	7093a97-1446049222
Status:	DRAFT
Date:	2015-Oct-28, 16:20 (GMT)

Chapter 1

Overview

Topics:

This document introduces the Geocoder Autocomplete API and:

- explains key concepts
- provides examples
- documents resources and query parameters
- documents response structures and data types

What is the Geocoder Autocomplete API?

Entering a full, correctly spelled address into a UI is difficult for consumer users and a barrier to conversion that the HERE Geocoder Autocomplete API can help reduce. The HERE Geocoder Autocomplete API allows end users to get better address search results with fewer keystrokes. Spatial filters can be used to return suggestions with greater relevance to users, e.g. results that are within specified countries, around a user's location, or within a map view a user looks at. The API returns complete addresses and an ID but not a geocode. The HERE Geocoder API can be used to subsequently geocode the address based on the ID. Additional map content attributes for an address or locality can be included in the geocoding response as well. The HERE Geocoder Autocomplete API is kept very simple and lightweight. It allows an easy integration of the service into web applications and to process the response quickly in Java Script.

Why use the Geocoder Autocomplete API?

Geocoder Autocomplete API provides solutions for the following high level use cases:

Table 1: Main features of the HERE Geocoder Autocomplete API

Feature	Description
Speed up typing	User starts typing and after a few characters the system provides a list of suggestions. The user sees the address he is looking for in the list and selects it to retrieve the address' location.
User is unsure about spelling of an address	A user receives immediate feedback relative to input in form of suggestions and can complete or correct input quickly based on suggestions
User is unsure about details of address	Context completion: • "520 Hayes" can be completed to "520 Hayes St". The 520 suggests an address context. But "Hayes" alone could also mean the area "Hayes Valley". • "Avenida Copacabana Rio" suggests a complete address with all its details and can be completed to "Avenida Nossa Senhora de Copacabana, Rio de Janeiro"

Key Features

- No minimum number of characters. Good results start showing after 2 or 3 characters

- Indexed elements:
  - Street names, house numbers
  - Areas (any combination of district, city, county, state, country name and country code)
  - Postal codes
- For house numbers both Point Addresses as well as Address Ranges are indexed
- The index contains the whole World
- Multi-language: Retrieval of language variant that matches the query best
- Street types match in abbreviated and spelled out forms
- Normalization of special characters, for example German umlauts, French acute accent, acute grave etc.
- Country filter limits suggestions to single country or list of countries
- Simple JSON response
- Formatted label
- Qualified address fields
- Support for match highlighting
- ID for location retrieval

Chapter 2

Quick Start

Topics:

This article helps you start using the Geocoder Autocomplete API.

Making your first request

Sometimes the easiest way to start using new software is to run simple working examples.

app_id and app_code are authentication credentials. This document uses {YOUR_APP_CODE} and {YOUR_APP_ID} as placeholders for access and authorization credentials. Please replace these placeholders with your own unique application-specific credentials to access the API resources. For more information about the access and authorization credentials credentials, see *Acquiring Credentials* on page 11.

The examples in this guide use the Customer Integration Testing (CIT) environment. This environment allows you to test your software. For production please use the production environment. See *Constructing a Request* on page 11 for the base URLs of both environments.

Note that most example URLs in this guide are broken up into multiple lines for better readability. Remove these line breaks and spaces when copying and pasting the examples to make sure URLs are still well formed.

Sample Request

```
http://autocomplete.geocoder.cit.api.here.com/6.2/suggest.json
?app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}
&query=Pariser+1+Berl
&beginHighlight=<b>
&endHighlight=</b>
```

Response Snippet

```
"suggestions": [
{
   "label": " Deutschland, <b>Berl</b>in, <b>Berl</b>in, 10117, <b>Berl</b>in, Mitte,
 <b>Pariser</b> Platz <b>1</b>",
   "label": "Deutschland, <b>Berl</b>in, <b>Berl</b>in, 12623, <b>Berl</b>in, Mahlsdorf,
 <b>Pariser</b> Straße <b>1</b>",
…}
]
```

Chapter 3

User Guide

Topics:

The articles in this section provide a guide to using the Geocoder Autocomplete API.

Acquiring Credentials

All users of HERE APIs must obtain authentication and authorization credentials and provide them as values for the parameters `app_id` and `app_code`. The credentials are assigned per application.

This document uses the placeholder text `{YOUR_APP_CODE}` and `{YOUR_APP_ID}` as placeholders for access and authorization credentials. Please replace these placeholders with your own unique application-specific credentials to access the API resources.

To obtain the credentials for an application, please visit *http://developer.here.com/get-started* for more details.

If you wish to explore the API, use the API Explorer at *https://developer.here.com/api-explorer*.

Constructing a Request

A request to the Geocoder Autocomplete API includes the basic elements shown in the following table and, in addition, it may contain resource-specific parameters or data.

Table 2: Basic request elements

Element	Value/Example	Description
Base URL	`http://autocomplete.geocoder.api.here.com` `http://autocomplete.geocoder.cit.api.here.com`	Production environment only Staging environment only [1]
Path	`/6.2`	
Resource	`/suggest.json`	GET only, specify request details via query parameters
Application Code	`&app_code={YOUR_APP_CODE}`	Substitute your own unique `app_code`
Application Id	`&app_id={YOUR_APP_ID}`	Substitute your own unique `app_id`

[1] The staging environment allows you to test your software against a newer version of the service before HERE brings that version into production. Note that the same application id can be used in both environments, but staging may require a dedicated application code. If this is the case, please contact us as described under *Service Support* on page 18.

The staging environment is not intended for production use and HERE SLAs do not apply for this environment.

Here is an example of a suggestions request, which uses the HTTP GET method:

```
/6.2/suggest.json
?app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}
&query=Berli
```

Key Concepts

This section provides insights into the key concepts used throughout the Geocoder Autocomplete API.

Examples

This section provides examples of requests along with the responding results.

Basic Example

The simplest use of the API is to retrieve suggestions for a query string and highlight the matching tokens in the response.

Request

```
http://autocomplete.geocoder.cit.api.here.com/6.2/suggest.json
?query=Pariser+1+Berl
&beginHighlight=<b>
&endHighlight=</b>
&app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}
```

Response

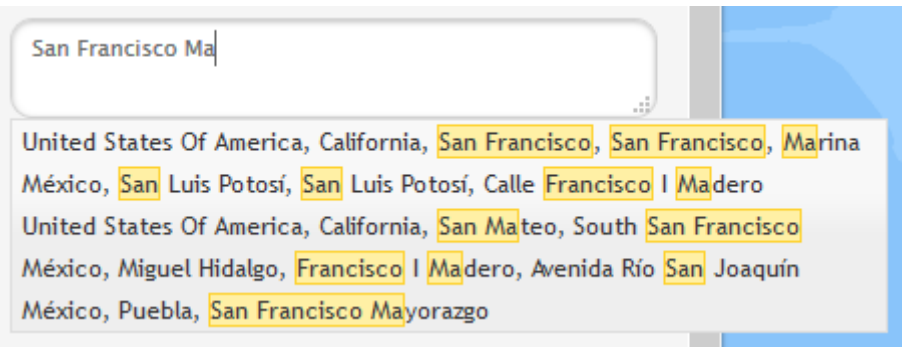
```
{
   "suggestions": [
      {
        "label": "Deutschland, <b>Berl</b>in, <b>Berl</b>in, 10117, <b>Berl</b>in, Mitte,
 <b>Pariser</b> Platz <b>1</b>",
        "language": "de",
        "countryCode": "DEU",
        "locationId": "NT_5mGkj3z90Fbj4abzMbUE4C_xA",
```

```
         "address": {
            "country": "Deutschland",
            "state": "<b>Berl</b>in",
            "county": "<b>Berl</b>in",
            "city": "<b>Berl</b>in",
            "district": "Mitte",
            "street": "<b>Pariser</b> Platz",
            "houseNumber": "<b>1</b>",
            "postalCode": "10117" },
         "matchLevel": "houseNumber" },
       {
         "label": "Deutschland, <b>Berl</b>in, <b>Berl</b>in, 10243, <b>Berl</b>in, Straße
 der <b>Pariser</b> Kommune <b>1</b>",
         "language": "de",
         "countryCode": "DEU",
         "locationId": "NT_bD2VXvX6WIWNBOM8VFvXbA_xA",
         "address": {
            "country": "Deutschland",
            "state": "<b>Berl</b>in",
            "county": "<b>Berl</b>in",
            "city": "<b>Berl</b>in",
            "district": "Friedrichshain",
            "street": "Straße der <b>Pariser</b> Kommune",
            "houseNumber": "<b>1</b>",
            "postalCode": "10243" },
         "matchLevel": "houseNumber" },
...
    ]
}
```


More Examples

Below are more examples in form of screen shots from an internal application. The screen shots show the matched elements and characters highlighted in yellow. This highlighting is based on the mark-up provided for each suggestion. Any client application can of course choose their own highlighting, do other things with the mark-up or ignore it.

Figure 1: Partial name input ...



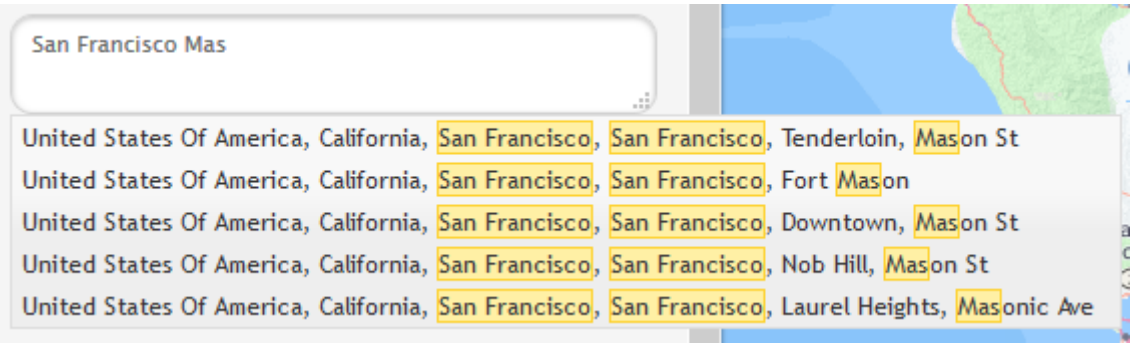


Figure 2: Street name + partial district name ...

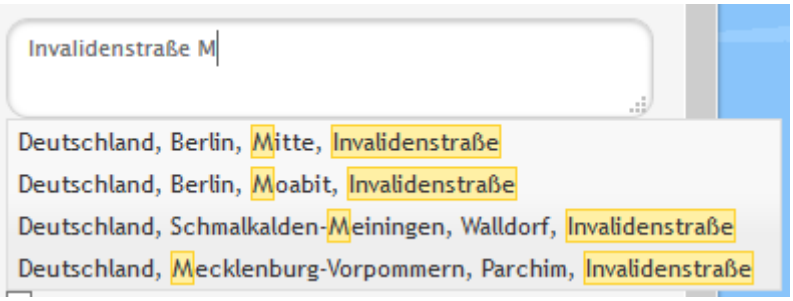


Figure 3: Omitted preposition in street name ...

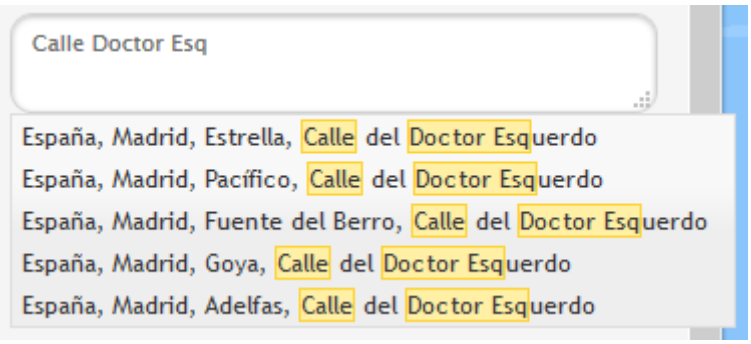


Figure 4: Special characters è vs e ...

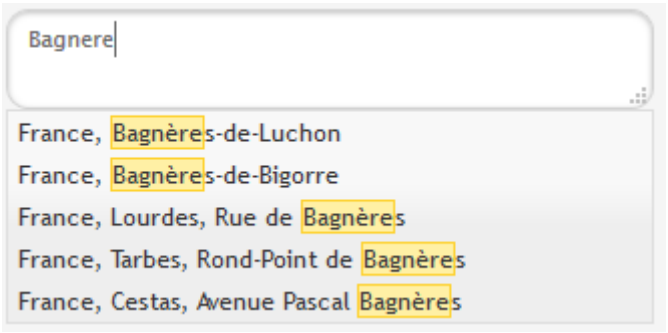


Figure 5: Special characters ß vs ss ...

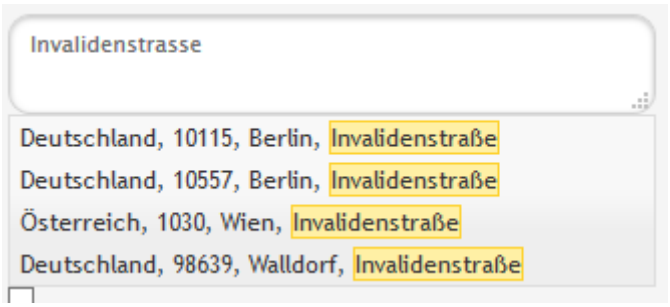


Figure 6: Country filter: Request without country filter ...

Figure 7: Country filter: With FRA country filter (`country=FRA`) ...

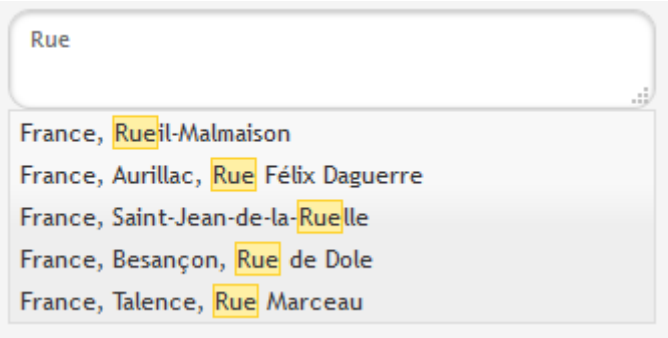


Figure 8: Location bias: Request without mapview parameter ...

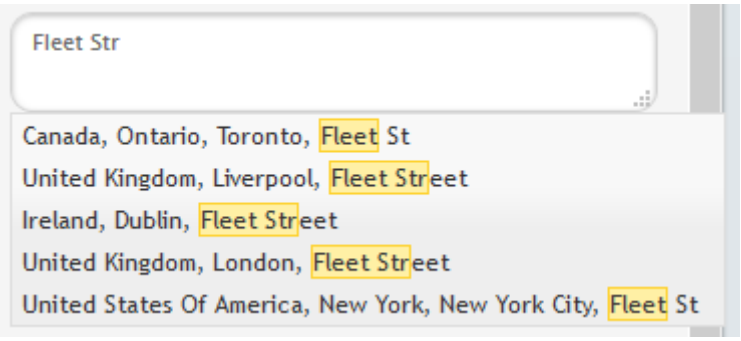
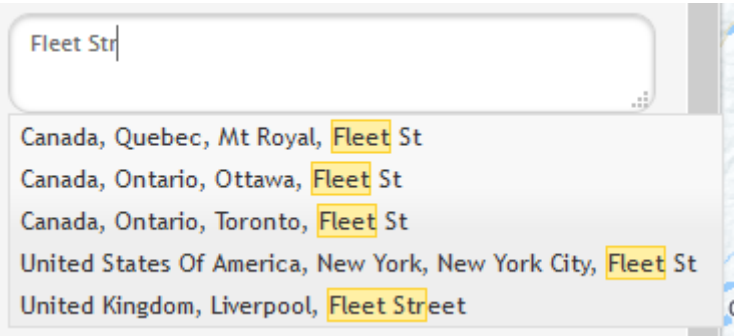


Figure 9: Location bias: With mapview over Montréal, Québec (`mapview=46.0346628,-74.3384855;45.2173875,-73.043472`) ...



Retrieve location details for a suggestion

Every suggested location holds a minimal set of attributes which provides sufficient information for choosing one of the suggestions. Detailed location information like coordinates are not included in the response.

These location details can be retrieved from the HERE Geocoder API using a lookup by locationId.

LocationId in Suggestions Response

```
{
   "suggestions": [
      {
        "label": "Deutschland, Berlin, Berlin, 10117, Berlin, Mitte, Pariser Platz 1",
        "language": "de",
        "countryCode": "DEU",
        "locationId": "NT_5mGkj3z90Fbj4abzMbUE4C_xA",
        "address": { ... },
        "matchLevel": "houseNumber"},
      ... ]
}
```

Here is a sample request to the Geocoder API which looks up the details for the above suggestion. Please check the Geocoder API documentation for the domain name and how to shape the response to your needs.

Geocoder Request

```
http://<domain name Geocoder API>/6.2/geocode.json
?locationid=NT_5mGkj3z90Fbj4abzMbUE4C_xA
&jsonattributes=1
&gen=9
&app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}
```

Geocoder Response

```
{
  "response":
...
      "view": [{
         "result": [{
            "matchLevel": "houseNumber",
            "matchType": "pointAddress",
            "location": {
               "locationId": "NT_5mGkj3z90Fbj4abzMbUE4C_xA",
               "locationType": "point",
               "displayPosition": {
                  "latitude": 52.51588,
                  "longitude": 13.37804 },
               "navigationPosition": [{
                  "latitude": 52.51591,
```

```
                  "longitude": 13.37833 }],
            "mapView": {
               "topLeft": {
                  "latitude": 52.5248732,
                  "longitude": 13.3632617 },
               "bottomRight": {
                  "latitude": 52.5068868,
                  "longitude": 13.3928183 }},
            "address": {
               "label": "Pariser Platz 1, 10117 Berlin, Deutschland",
               "country": "DEU",
               "state": "Berlin",
               "county": "Berlin",
               "city": "Berlin",
               "district": "Mitte",
               "street": "Pariser Platz",
               "houseNumber": "1",
               "postalCode": "10117",
               "additionalData": [{
                  "value": "Deutschland",
                  "key": "CountryName" },
                  {
                  "value": "Berlin",
                  "key": "StateName" },
                  {
                  "value": "Berlin",
                  "key": "CountyName" }]}}],
         "viewId": 0 }]
   }
}
```


Service Support

If you need assistance with this or other HERE products, please contact your HERE representative or Technical Customer Support.

Chapter 4

API Reference

Topics:

This part of the guide provides a comprehensive reference to Geocoder Autocomplete API.

Request Parameters

The Geocoder Autocomplete API is very simple. Only a few request parameters are available and besides the authorization (app_id, app_code) parameters only the “query” parameter is mandatory.

```
.../6.2/suggest.json?<parameter>=<value>...
```

Table 3: Request Parameters

Parameter	Type	Description	Examples
query	`string`	The search query	`query=Pariser Pl`
maxresults	`integer`	Limit the number of suggestions. Default is set to 5. Valid range: 1 to 20.	`query=Pariser Pl &maxresults=10`
country	comma separated list of strings (3-letter ISO country codes)	The country parameter is a spatial filter that limits suggestions to country or set of countries. country=`ISO3 country code` Can be combined with the `mapview` or `prox` spatial filters.	`query=Rue du Froma &country=FRA` `query=Rue du Froma &country=FRA,BEL`
mapview	two lat/lon pairs	The map view parameter is a spatial filter. The bounding box may be derived from the corners of the map displayed on a user’s device. When all results within a mapview are returned but the `maxresults` number of results is not yet reached relevant results from outside the `mapview` are added to results. The mapview is specified by two latitude / longitude pairs; the first pair defines the top left corner of the bounding box, the second set the lower right. `mapview= TopLeft.Latitude, TopLeft.Longitude; BottomRight.Latitude, BottomRight.Longitude`	`mapview=41.90852,-87.67629; 41.86827,-87.60419`

Parameter	Type	Description	Examples
		Can be combined with the `country` spatial filter.	

prox	lat/lon pair and (optional) radius	The prox parameter is a spatial filter. A single geo-coordinate pair and optionally a radius (in meters) is used to define the filter criterion. This may be derived from a positioning of the user's device. When all results from within the filter criteria are returned but the `maxresults` number of results is not yet reached relevant results from outside the filter are added to results.
		`prox=Latitude,Longitude`
		`prox=Latitude,Longitude,Radius`
		Can be combined with the country spatial filter.
			`prox=37.86946,-122.26811,10000`

beginHighlight	`string`	Mark the beginning of the match in a token. This can be any character sequence. Common usage is a HTML tag such as <b> for bold. But it can also be a "[".
		Default: no marker
			`query=Barcelo &beginHighlight=<b> &endHighlight=</b>`
			Result:
			`"label": "Espanya, Catalunya, <b>Barcelo</b>na"`

endHighlight	`string`	Mark the end of the match in a token	see previous example

Response Structure

The Geocoder Autocomplete API response is in JSON format.

Table 4: Response Elements

Element	Description	Example
suggestions	Top-most element holding list of suggestions. The element is absent in case of an error.	`suggestions: [ .... ]`
label	Formatted response label including support for match highlights.	`label: "Deutschland, [Berli]n, [Berli]n, 10117, [Berli]n, Mitte, [Pariser] Platz [1]"`
language	Language of the match. 2-letter ISO language code.	`language: "de"`
countryCode	Country of the location. 3-letter ISO country code.	`countryCode: "DEU"`
locationId	Client applications use the `locationId` for location lookup.	`locationId: "NT_5mGkj3z90Fbj4abzMbUE4C_xA"`
address	Structured address block. The address elements support match highlighting.	`address: { country: "Deutschland", state: "Berlin", county: "Berlin", city: "Berlin", district: "Mitte", street: "Pariser Platz", houseNumber: "1", postalCode: "10117" }`
matchLevel	Match precision. One of: `houseNumber`, `street`, `postalCode`, `district`, `city`, `county`, `state`, `country`	`matchLevel: "houseNumber"`

Errors

In case of an error Geocoder Autocomplete API will return an empty response. Client applications can check for the existence of the `suggestions` element to identify a successful request.

Authentication errors are indicated using HTTP status codes:

HTTP Status Code	Status	Description
400	Bad Request	Missing `app_id` or `app_code` parameter
401	Unauthorized	Validation of `app_id`/`app_code` pair failed. Either `app_code` does not match the `app_id` or the `app_id` is not entitled to use the Geocoder Autocomplete API.