

Address By Component AutoComplete

Address By Component AutoComplete APIs

Address By Component AutoComplete is a software component for searching address information with dynamic autocomplete support.

Based on text input of the end-user, the components present a list of suggested matches which are limited and ordered by a search model.

The service features 4 web services which retrieves suggestions by Component, in cascade. An API key is needed to be able to use these APIs. Please contact addressvalidation@bpost.be to retrieve your personal key, which should be added in the header 'x-api-key'.

The AutoComplete By Component uses the same classification module and SOLR (Apache Foundation) back-end which was built for AddressBlockParser & AutoComplete.

Address By Component Validation Widget

Address By Component Widget is an Angular 13 implementation of the Address By Component AutoComplete which is a combination of 4 web services which offer suggestions of address components based on user input which may be incorrect or incomplete.

When typing in the component autocomplete field, the widget starts searching for entries that match and displays a list of values to choose from.

By entering more characters, the user can filter down the list to better matches. Widget ensures the cascade & coherence between address components

The component can be used internally by Bpost services and externally by the public.

Overview of features

API features

The Address By Component AutoComplete service offers:

- **Up to N top suggestions for a query string of words (By default, N=5 - custom param)**
- 'Typo search' (also known as fuzzy searching), input words do not need to be exactly correct (example: *neuvilles* instead of *neufvilles*).
- Search for incomplete words (example: *bru* instead of *bruxelles*)
- Provide a message when the user has searched for a submunicipality and a suggestion is returned with the fused municipality. (Example: *Blanden* is a submunicipality of *OudHeverlee*).
- For addresses which have multiple address translations (such as Bruxelles or any municipality with facilities), all address translations are returned.
- **Only returns components related to the Address Component endpoint. (postalCode/locality, street, streetNumber, boxNumber)**
- Search for the detail number instead of the concatenated civic number (Example: Search for *19* and return a house with civic number *17-21*). In case of 'Gantoise Numbering', the detail number is returned as the civic number.
- Automatic daily index updates from the Bpost Address database (RoMa).

- Support for abbreviations (*Ch* -> *Chemin* or *Chaussee*) and special characters (-, /, ...)
- Treat *le, la, les, du, des* as synonyms, as a mistake is often made against these words.
- Support for searching and confirming compound words as street names (*Rue de la loi*)
- Support for special postal codes having addresses (example: 7010 *SHAPE*)
- Support for searching for box numbers
- **General Options**
 - Environment: To configure *st/dv/ac/pr* endpoints.
- **Specific Options (by Address component):**
 - Max Number of suggestions in response
 - Response Sort Criteria

API Limitations

- Only Belgian addresses can be validated
- Only Legal/Official addresses can be validated - Synonyms / Aliases cannot be validated (eg. 2000 *Anvers* vs 2000 *Antwerpen*)
- Does not support landmarks (Example: *Muntcentrum*)
- Does not Support special postal codes without any address (example: 1043 *VRT*)
- Does not Support PO Boxes

API KPIs

- Availability: 99,9% - Free tool (Win-win Customer / Bpost - since 6 years, it was down only few times during < 4 hours)
- Response Time: Average between 200ms and 400ms (depending on load, autoscaling is ON). When user is typing, he's not feeling the delay.
- Usage recommendation: No limit foreseen until now. However, if we notice intensive usage which cannot be related to a Web application (eg. detecting DoS), the flow is monitored and the IP address(es) may be blocked + client contacted to understand the situation

Widget features (A front-end Angular13 Module which uses the AddressAutoComplete by Component APIs)

- Properly manage each char typing and order of responses
- Properly manage cascade back-and-forth between address component. By example, a locality limits the possible streets (with some tolerance within same municipality) but a street or a streetNumber may update a locality.
- Word completion (Type-ahead)
- Bold word matching suggestion with user input
- Language logic (3 languages: French - Dutch - German, coming from WEB UI or default settings)
- IsComplete logic (generate event once address is complete, at streetNumber or boxNumber level)
- CSS that can be changed by consuming web application (translation property file)
- Some shortcut keys (like CTRL+DELETE to clean bar)
- ...
- **General options:**
 - Detection of service unavailability: possibility to show the unavailability message or emit a background event.
 - o Clear input with and without focus option

- o Allow Not-In-List: This option is used for street number and box number widget. If enabled, It allows user to submit input which is not present in list.
- **Specific options (by Address component):**
 - Message Display Format: (eg. submunicipality message): possibility to deactivate it, show it as a second line or between parenthesis on same line.
 - o Max number of suggestions to display in suggestion list window.
 - o Pre-filling address component(s)

Others

- A senior DEV should take ½ to 1 day to integrate this widget in a web application
 - Widget is aimed for Web applications, not for native apps. However, as a consumer you may create your own widget and at least enjoy all back-end features. If you try to use the autocomplete rest API from your own home-made widget, pay attention to some specific behaviors. Otherwise you will meet unexpected cases. You may also start from the widget and enrich it, by example with an extra “not-in-list” suggestion, like some consumers did.
- Some tips / use cases:
- Multi-lan: <https://api.mailops.bpost.cloud/roa-info/externalMailingAddressProofingRest/autocomplete/locality?id=3&q=n&maxNumberOfSuggestions=5>
Address block is always present, corresponding to the default language if you don't match any language with the user language.
 - Messages (submunicipalities): <https://api.mailops.bpost.cloud/roa-info/externalMailingAddressProofingRest/autocomplete/locality?id=7&q=neufv&maxNumberOfSuggestions=5>
 - Id = sequence_number (mais d'autres moyens d'ordonnement des réponses existent)
 - isComplete (address is complete and valid): <https://api.mailops.bpost.cloud/roa-info/externalMailingAddressProofingRest/autocomplete/streetNumber?id=3&q=64&postalCode=1140&locality=EVERE&street=RUE%20EDOUARD%20DEKOSTER&maxNumberOfSuggestions=5>
 - cases when locality needs to be updated in cascade:
 - <https://api.mailops.bpost.cloud/roa-info/externalMailingAddressProofingRest/autocomplete/locality?id=12&q=7060&maxNumberOfSuggestions=5>
→ Select 7060 *Soignies* (fused municipality)
 - <https://api.mailops.bpost.cloud/roa-info/externalMailingAddressProofingRest/autocomplete/street?id=7&q=ho&postalCode=7060&locality=Soignies&maxNumberOfSuggestions=5>
→ Select *rue reine de hongrie* (postal code/locality must be updated)
 - <https://api.mailops.bpost.cloud/roa-info/externalMailingAddressProofingRest/autocomplete/streetNumber?id=4&q=2&postalCode=7061&locality=THIEUSIES&street=RUE%20REINE%20DE%20HONGRIE&maxNumberOfSuggestions=5>
→ Select 102 (postal code/locality must be updated again)