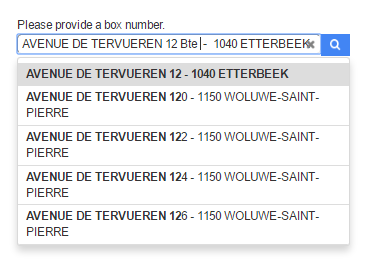
bpAddressAutoComplete for external bpost users



bpAddressAutoComplete is an AngularJS implementation of the Address Auto Complete Service, which makes use of the Web Services @bpost to validate addresses and offer suggestions of addresses based on user input which may be incorrect or incomplete.

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

This document aims to provide information for any external public implementation.

|  |  |
| --- | --- |
| Project | bpAddressAutoComplete for external bpost users |
| Author | BPOST |
| Save Date | 06.10.2016 |

Table of Contents

[Components 2](#_Toc464025131)

[Overview of components 2](#_Toc464025132)

[Options 3](#_Toc464025133)

[bp-address-auto-complete and bp-address-auto-complete-wrapper 3](#_Toc464025134)

[Ng-model 3](#_Toc464025135)

[Min-level 4](#_Toc464025136)

[url 4](#_Toc464025137)

[bp-validator 4](#_Toc464025138)

[url 4](#_Toc464025139)

[Example Implementations 5](#_Toc464025140)

[Events 6](#_Toc464025141)

[Emitting Events 6](#_Toc464025142)

[bpAddressAutoComplete.addressSelected 6](#_Toc464025143)

[Listening to Events 6](#_Toc464025144)

[language-changed 6](#_Toc464025145)

[bpAddressAutoComplete.clearInput 6](#_Toc464025146)

[Endpoints 6](#_Toc464025147)

[Autocomplete Service External 6](#_Toc464025148)

[Main REST Endpoint 6](#_Toc464025149)

[Parameters 8](#_Toc464025150)

[Return Values 8](#_Toc464025151)

[AddressValidation Service External 10](#_Toc464025152)

[Main REST Endpoint - validateAddress 10](#_Toc464025153)

# Components

## Overview of components

The bpAddressAutoComplete project includes the following directories:

* Bp-address-auto-complete : the main core Address Auto Complete directive
* Bp-Address-Auto-Complete-Wrapper : The main wrapper directive. what the output from the core directive is and how it should be displayed on every user input (‘Please fill in a house number …)
* Bp-address-auto-complete-demo-page : The Demo page includes the AAC Wrapper with some parameterized options available to the user to:
  + change the environment endpoints
  + Set the minimum level of validation
  + allows the user to prefill the search bar with unstructured or structured fields

**What is the difference between bp-address-auto-complete and bp-address-auto-complete-wrapper?**

Bp-address-auto-complete (core) is a collection of angularJS directives which includes the address auto complete input window and suggestions list.

Bp-Address-auto-complete-wrapper (wrapper) depends on the core, and displays the messages returned by core (‘please provide a house number … ’). This is the standard look of the demo page.

If you wish to customize your look (for example: do not display the ‘please provide’ message, or hide the input bar when the address has been validated … ) you can create your own wrapper, which depends on core. If the functionality provided by the wrapper is sufficient, you can include the wrapper as-is.

The reason we have separated these two components is to provide customizability in case you wish to make your own wrapper, but also provide a default wrapper in case the default view is sufficient.

**What is bp-address-validator?**

Bp-address-validator includes the search button to the right of the search input bar. When you click this button or press enter in the input bar, a request will be sent to the Address Validation web service (which is different from the Address Auto Complete web service). The Address Validation web service is not geared towards fast and incomplete input, however the service supports street and municipality synonyms and may recognize addresses which are not recognized by address auto complete.

# Options

## bp-address-auto-complete and bp-address-auto-complete-wrapper

### Ng-model

To retrieve the selected address from the component, bind a model variable to the directive with ng-model.

Please note: always use a dot when binding with ng-model. [You may find an explanation here.](https://github.com/angular/angular.js/wiki/Understanding-Scopes#-directives)

|  |  |
| --- | --- |
| **Example**  Example:   |  | | --- | | $scope.selectedAddress = {}  <bp-address-auto-complete-wrapper ng-model=”selectedAddress.address” … | |

The selected address will then become available on selectedAddress.address in the current $scope. Please note: the address only changes when the user selects a valid address (which depends on the min-level setting). If the user selects an address which requires the user to provide more details, it is not yet set to ng-model.

### Min-level

An address will only be set if the clicked suggestion matches the provided minimum level. The following levels are supported:

|  |  |
| --- | --- |
| 1 | **Box Number**. An address is considered complete when the suggestion contains a Box Number or is a Street Number which does not have any box. |
| **2** (default) | **Street Number.** An address is considered complete when either the address is marked as complete by the services (eg. PO Boxes and Special Postal codes), or the address contains a house number, street name, postal code and municipality. |
| **3** | **Street**. An address is considered sufficiently complete when either the address is marked as complete by the services or the address contains at least a street name, postal code and municipality. |
| **4** | **Municipality.** An address is considered sufficiently complete when either the address is marked as complete by the services or the address contains at least a postal code and municipality. Note: this means all suggestions in the suggestion window will be accepted as sufficiently complete addresses, as each suggestion contains at least the postal code and municipality. |

### URL

The URL parameter specifies which endpoint of AddressAutoComplete will be used.

|  |
| --- |
| <bp-address-auto-complete-wrapper ng-model="address.selectedAddress" url="{{aacUrl}}" min-level="3"> </bp-address-auto-complete-wrapper>  or  <bp-address-auto-complete ng-model="address.selectedAddress" url="{{aacUrl}}" min-level="3"> </bp-address-auto-complete> |

## bp-validator

If you wish to include the AddressValidation WebService, include the bp-validator directive in-between your invocation of bp-address-auto-complete or bp-address-auto-complete-wrapper.

Table 1 Example

|  |
| --- |
| <bp-address-auto-complete-wrapper ng-model="selectedAddress" url="{{aacUrl}}" min-level="{{selectedLevel}}">  <bp-address-validator url="{{valUrl}}"></bp-address-validator> </bp-address-auto-complete-wrapper> |

### URL

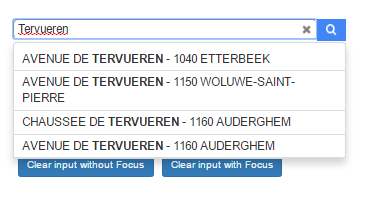
The URL of AddressValidation WebService (AC1).

Table 2 Example

|  |
| --- |
| <bp-address-validator url=" https://webservices-pub.acbpost.be/ws/ExternalMailingAddressProofingCSREST\_v1\_AC1/address/validateAddresses " bp-internal-validator></bp-address-validator> |

# Example of Implementation

While the consumer types the address inside the Address Search Bar Autocomplete Web service brings the 5 top suggestions on the list. (Picture 1)

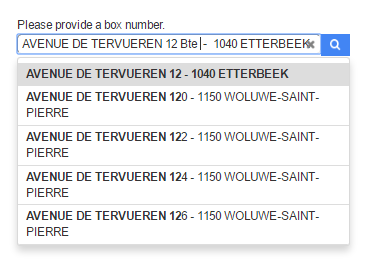


Picture 1. Top Suggestions while the user types the address

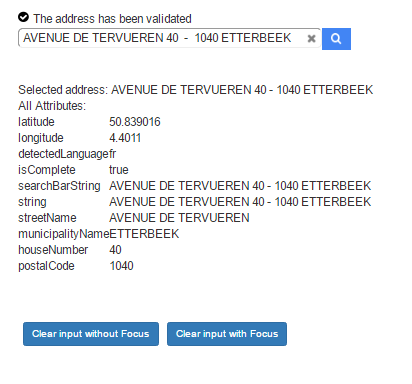
By selecting one of the suggestions or by clicking enter the validateAddresses Web service validates the address taking into account the minimum level of validation (the parameterized option min-level).

In case of success the details of the address point are displayed (Picture 3)

else a validation message on top of the Address Search Bar is prompting the user to enter the missing elements, the cursor being placed at the best place for this purpose (Picture 2).



Picture 2. Validation messages prompt the user to enter more information



Picture 3. Validation success with address details

# Events

## Emitting Events

bp-address-auto-complete will emit the following events:

### bpAddressAutoComplete.addressSelected

When the user selects a validated address, this event will be emitted and selectedAddress will be set. By listening to this event, your code will not need to watch the selectedAddress object for changes and in instances where the user can select multiple times on a validated address, the event will still fire even though the selected address remains the same.

## Listening to Events

bp-address-auto-complete will listen to the following events:

### language-changed

On 'language-changed', bp-address-auto-complete will retranslate its text ('is a submunicipality of', 'please provide input' ...)

### bpAddressAutoComplete.clearInput

Clears the input and selected address. By providing an additional focus argument (set to true), the cursor will be positioned in the address search bar.

|  |
| --- |
| $scope.$broadcast('bpAddressAutoComplete.clearInput', focus); |

Another way to clear the selected address and input text in the controller is by using the shortcut **Ctrl+Del**.

Changing the selected address in ng-model will not work, as the component will not watch for outside changes to this variable.

# Endpoints

## Autocomplete Service External

|  |  |
| --- | --- |
| Environment | Url |
| ac1 | <https://webservices-pub.acbpost.be/ws/ExternalMailingAddressProofingCSREST_v1_AC1/address/autocomplete> |
| Pr | <https://webservices-pub.bpost.be/ws/ExternalMailingAddressProofingCSREST_v1/address/autocomplete> |

# Main REST Endpoint

The AddressAutoComplete REST endpoint allows web applications to request suggestions for a string of words.

Example of a json returned by the AddressAutoComplete Web Service is the following;

Example 1

{

"response": {

"sequenceNumber": 68,

"topSuggestions": [{

"address": {

"latitude": 50.839016,

"longitude": 4.4011,

"detectedLanguage": "fr",

"isComplete": true,

"searchBarString": "AVENUE DE TERVUEREN 40 - 1040 ETTERBEEK",

"string": "AVENUE DE TERVUEREN 40 - 1040 ETTERBEEK",

"streetName": "AVENUE DE TERVUEREN",

"municipalityName": "ETTERBEEK",

"houseNumber": "40",

"postalCode": "1040"

}

},

{

"address": {

"latitude": 50.8317,

"longitude": 4.391034,

"detectedLanguage": "fr",

"searchBarString": "AVENUE DE LA CHASSE 40 Bte - 1040 ETTERBEEK",

"string": "AVENUE DE LA CHASSE 40 - 1040 ETTERBEEK",

"streetName": "AVENUE DE LA CHASSE",

"municipalityName": "ETTERBEEK",

"houseNumber": "40",

"postalCode": "1040"

}

},

{

"address": {

"latitude": 50.828533,

"longitude": 4.390382,

"detectedLanguage": "fr",

"isComplete": true,

"searchBarString": "AVENUE DES CASERNES 40 - 1040 ETTERBEEK",

"string": "AVENUE DES CASERNES 40 - 1040 ETTERBEEK",

"streetName": "AVENUE DES CASERNES",

"municipalityName": "ETTERBEEK",

"houseNumber": "40",

"postalCode": "1040"

}

},

{

"address": {

"latitude": 50.838205,

"longitude": 4.397482,

"detectedLanguage": "fr",

"isComplete": true,

"searchBarString": "AVENUE DES CELTES 40 - 1040 ETTERBEEK",

"string": "AVENUE DES CELTES 40 - 1040 ETTERBEEK",

"streetName": "AVENUE DES CELTES",

"municipalityName": "ETTERBEEK",

"houseNumber": "40",

"postalCode": "1040"

}

},

{

"address": {

"latitude": 50.826434,

"longitude": 4.402987,

"detectedLanguage": "fr",

"searchBarString": "AVENUE DES VOLONTAIRES 40 Bte - 1040 ETTERBEEK",

"string": "AVENUE DES VOLONTAIRES 40 - 1040 ETTERBEEK",

"streetName": "AVENUE DES VOLONTAIRES",

"municipalityName": "ETTERBEEK",

"houseNumber": "40",

"postalCode": "1040"

}

}]

}

}

Example 2

{

"response": {

"sequenceNumber": 18,

"topSuggestions": [{

"address": {

"latitude": 50.579267,

"longitude": 4.066706,

"detectedLanguage": "fr",

"searchBarString": "RUE DE NEUFVILLES - 7060 SOIGNIES",

"string": "RUE DE NEUFVILLES - 7060 SOIGNIES",

"streetName": "RUE DE NEUFVILLES",

"municipalityName": "SOIGNIES",

"postalCode": "7060"

}

},

{

"address": {

"latitude": 50.573554,

"longitude": 4.000031,

"detectedLanguage": "fr",

"searchBarString": "7063 SOIGNIES",

"string": "7063 SOIGNIES",

"municipalityName": "SOIGNIES",

"postalCode": "7063"

},

"messages": [{

"key": "subl",

"args": ["NEUFVILLES",

"SOIGNIES"]

}]

},

{

"address": {

"latitude": 50.538999,

"longitude": 3.377577,

"detectedLanguage": "fr",

"searchBarString": "RUE DE NEUFVILLES - 7620 BRUNEHAUT",

"string": "RUE DE NEUFVILLES - 7620 BRUNEHAUT",

"streetName": "RUE DE NEUFVILLES",

"municipalityName": "BRUNEHAUT",

"postalCode": "7620"

}

},

{

"address": {

"latitude": 50.562085,

"longitude": 3.949722,

"detectedLanguage": "fr",

"searchBarString": "RUE DE NEUFVILLES - 7870 LENS",

"string": "RUE DE NEUFVILLES - 7870 LENS",

"streetName": "RUE DE NEUFVILLES",

"municipalityName": "LENS",

"postalCode": "7870"

}

},

{

"address": {

"latitude": 50.543813,

"longitude": 4.04866,

"detectedLanguage": "fr",

"searchBarString": "7060 SOIGNIES",

"string": "7060 SOIGNIES",

"municipalityName": "SOIGNIES",

"postalCode": "7060"

}

}]

}

}

## Parameters

The API accepts the following input parameters:

|  |  |
| --- | --- |
| parameter | Description |
| **q** | This parameter is mandatory and contains the search string. |
| **id** | This parameter is optional. The id may only consist of numeric characters and is returned as the sequenceNumber in the response. This value is useful to distinguish newer requests from older requests, regardless of how they are routed through the networks. |

## Return Values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | | |  | **Description** |
| response |  |  |  | Root attribute of the response. |
|  | sequenceNumber |  |  | If an id was given with the request, the same value will be returned as the sequenceNumber. When no id was provided, the default value is 0. |
|  | topSuggestions |  |  | Contains an array of suggestions. |
|  |  | address |  | Represents the default address object and its address elements. |
|  |  | dutch |  | (OPTIONAL) If multiple address translations for an address are available, an address object for the dutch address translation may be returned. The client can choose whether or not to choose the dutch address translation object over the default chosen one. |
|  |  | french |  | (OPTIONAL) If multiple address translations for an address are available, an address object for the french address translation may be returned. The client can choose whether or not to choose the french address translation object over the default chosen one. |
|  |  | german |  | (OPTIONAL) If multiple address translations for an address are available, an address object for the german address translation may be returned. The client can choose whether or not to choose the german address translation object over the default chosen one. |
|  |  |  | string | The representation of the address in a one-line string of text. |
|  |  |  | streetName | (OPTIONAL) The street name of the address. |
|  |  |  | houseNumber | (OPTIONAL) The house number of the address |
|  |  |  | postalCode | The postal code of the address |
|  |  |  | municipalityName | The municipality name of the address |
|  |  |  | isComplete | (OPTIONAL) Boolean. Provided when the address is 'complete', even if the house number is missing. |
|  |  |  | Messages | (OPTIONAL) Provides messages about the suggestion. Example: the user searched on the name of a submunicipality.  An address suggestion may contain messages for the user. The messages field is an array of arrays.  Each message is an array of Strings, with the first field indicating the message code and the subsequent fields the parameters of the message code. The message codes are explained in the Messages section.  Example: (See picture 4)  messages: [ ["subl",   "BLANDEN",   "OUD-HEVERLEE"]] |
|  |  |  | detectedLanguage | Language of the address detected by the autocomplete service. |
|  |  |  | searchBarString | Similar to string, the representation of the address in a one-line string of text. The difference is in the fact that this string may contain additional words in order to avoid the user to put it manually.  Example: Bte is added to the string (See picture 2) |



Picture 4. Example of subl message.

## AddressValidation Service External

|  |  |
| --- | --- |
| Environment | Url |
| ac1 | <https://webservices-pub.acbpost.be/ws/ExternalMailingAddressProofingCSREST_v1_AC1/address/validateAddresses> |
| pr | <https://webservices-pub.bpost.be/ws/ExternalMailingAddressProofingCSREST_v1/address/validateAddresses> |

# Main REST Endpoint

The ValidateAddresses Rest endpoint allows web applications to validate an address and return the result of an existing address.

Example of a json returned by ValidateAddresses Web Service is the following:

{

"ValidateAddressesResponse": {

"ValidatedAddressResultList": {

"ValidatedAddressResult": [{

"@id": "1",

"ValidatedAddressList": {

"ValidatedAddress": [{

"PostalAddress": {

"StructuredDeliveryPointLocation": {

"StreetName": "ANSPACHLAAN"

},

"StructuredPostalCodeMunicipality": {

"PostalCode": "1000",

"MunicipalityName": "BRUSSEL"

},

"CountryName": "BELGIE"

},

"AddressLanguage": "nl",

"Label": {

"Line": ["ANSPACHLAAN",

"1000 BRUSSEL"]

}

}]

},

"Error": [{

"ComponentRef": "",

"ErrorCode": "address\_not\_recognized",

"ErrorSeverity": "error"

},

{

"ComponentRef": "UnstructuredDeliveryPointLocation",

"ErrorCode": "field\_not\_recognized",

"ErrorSeverity": "warning"

}],

"DetectedInputAddressLanguage": "nl",

"TransactionID": "66ccb54a-5382-4b2f-a3b9-24ce4989aeba"

}]

}

}

}