

# Locatieserver Location server v3\_1 OAS3

/bzk/locatieserver/search/v3\_1/openapi.json

/bzk/locationserver/search/v3\_1/openapi.json

De PDOK Locatieserver is een open en gratis geocodeerservice waarmee gezocht kan worden op gegevens uit verschillende overheidsregistraties (bijvoorbeeld adressen, percelen en rijkswegen) om vervolgens doorgestuurd te worden naar de locatie van de zoekopdracht.

The PDOK Location Server is an open and free geocoding service that allows you to search for data from various government records (e.g. addresses, plots and national roads) and then be forwarded to the location of the search.

De Locatieserver bestaat uit meerdere RESTful API's die eenvoudig geïntegreerd kunnen worden in viewers en/of applicaties. De API's zijn open, gratis en bevatten gegevens uit verschillende overheidsregistraties zoals adressen (postcodes, straten, huisnummer), kadastrale percelen, wegen en hectometerpalen maar ook woonplaatsen, wijken, buurten en waterschappen. De onderliggende gegevens worden zeer frequent bijgewerkt.



The Location Server consists of multiple RESTful APIs that can be easily integrated into viewers and/or applications. The APIs are open, free and contain data from various government registrations such as addresses (postal codes, streets, house number), cadastral plots, roads and hectometre poles, but also residences, districts, neighbourhoods and water boards. The underlying data is updated very frequently.

De Locatieserver bestaat totaal uit vier RESTful API's die gepubliceerd worden volgens de Open API Specificatie [https://api.pdok.nl/bzk/locatieserver/search/v3\\_1/ui/](https://api.pdok.nl/bzk/locatieserver/search/v3_1/ui/). Hiermee zijn de mogelijkheden van de Locatieserver API's (input en output) gedocumenteerd.

The Location Server consists of a total of four RESTful APIs that are published according to the Open API Specification [https://api.pdok.nl/bzk/locatieserver/search/v3\\_1/ui/](https://api.pdok.nl/bzk/locatieserver/search/v3_1/ui/). This documents the capabilities of the Location Server APIs (input and output).

Het zoeken op gegevens wordt vereenvoudigd met behulp van de **Suggest API** die de mogelijkheid biedt om een (gedeelte van een) zoekopdracht op te voeren, waarnaar er suggesties teruggegeven worden. Dankzij slimme technieken als auto-aanvullen en ondersteuning van alternatieve schrijfwijzen en synoniemen vinden gebruikers snel waar ze naar op zoek zijn. Het is ook mogelijk om bij zoekopdrachten een locatie van de gebruiker mee te geven zodat suggesties op basis van die locatie worden gesorteerd.

Searching for data is simplified using the **Suggest API**, which offers the possibility to enter a (part of) a search, to which suggestions are returned. Thanks to smart techniques such as auto-completion and support for alternative spellings and synonyms, users can quickly find what they are looking for. It is also possible to include a location of the user in searches so that suggestions are sorted based on that location.

Zodra er op basis van suggesties van de Suggest API een keuze is gemaakt, wordt de **Lookup API** aangeroepen, welke o.a. een (versimpelde) geometrie van de zoekopdracht teruggeeft. Hiermee kan een gebruiker naar de locatie van de zoekopdracht worden gestuurd op de kaart. Dankzij de highlight functie kan het betreffende gegeven (object) vervolgens worden opgelicht op de kaart.

As soon as a choice has been made based on suggestions from the Suggest API, the **Lookup API**, which returns a (simplified) geometry of the search. This allows a user to be sent to the location of the search on the map. Thanks to the highlight function, the relevant data (object) can then be highlighted on the map.

De suggest- en de lookup API werken dus als een tweetrapsraket. Er is ook een mogelijkheid om vrij te zoeken (klassiek geocoderen) met de **Free API** waar zonder tussenkomst van suggesties de API direct resultaten teruggeeft op basis van de zoekopdracht.

The suggest and lookup APIs therefore work like a two-stage rocket. There is also a possibility to search freely (classic geocoding) with the **Free API** where without the intervention of suggestions the API returns results directly based on the search.

De **Reverse API** biedt de mogelijkheid om een locatie (punt geometrie) op te geven om vervolgens verschillende gegevens in een range rondom deze locatie te ontvangen. Bijvoorbeeld het ophalen van alle adressen of wegen in de buurt (op basis van de opgegeven locatie). Dit wordt ook wel Reverse Geocoderen genoemd. Met de **Reverse API** kan je op basis van een coördinaat en een range (bijvoorbeeld 1 kilometer) de gewenste gegevens binnen die range opvragen en visualiseren.

The **Reverse API** offers the possibility to specify a location (point geometry) and then receive different data in a range around this location. For example, retrieving all addresses or roads in the vicinity (based on the specified location). This is also called Reverse Geocoding. With the **Reverse API**, you can retrieve and visualize the desired data within that range based on a coordinate and a range (e.g. 1 kilometer).

De Locatieserver is primair een geocodeerservice. Voor andere doeleinden zoals het rechtstreeks opvragen van bepaalde gegevens of het downloaden van bulk data biedt PDOK andere mogelijkheden via webservices en/of downloads.

The Location Server is primarily a geocoding service. For other purposes, such as requesting certain data directly or downloading bulk data, PDOK offers other options via web services and/or downloads.

Meer informatie over de Locatieserver is ook te vinden op de Locatieserver Wiki op [Github](#) of volg discussies op het [PDOK forum](#).

More information about the Location Server can also be found on the Location Server Wiki at [Github](#) or follow discussions on the [PDOK forum](#).

## Servers

[https://api.pdok.nl/bzk/locatieserver/search/v3\\_1](https://api.pdok.nl/bzk/locatieserver/search/v3_1)

# Locatieserver Location server



De Free API biedt de mogelijkheid om vrij te zoeken (klassiek geocoderen), waar zonder tussenkomst van suggesties de API direct resultaten teruggeeft op basis van de zoekopdracht.

**GET** /free

The Free API offers the possibility of free search (classic geocoding), where without the intervention of suggestions, the API immediately returns results based on the search query.

^

## Parameters

**Try it out**

Name	Description
------	-------------

<b>q</b> <b>string</b> <i>(query)</i>	Hiermee worden de zoektermen opgegeven. De Solr-syntax voor zoektermen kan hier worden toegepast, bijv. combineren met "and", en het gebruik van dubbele quotes voor opeenvolgende zoektermen. Zoektermen mogen incompleet zijn. Ook wordt er gebruik gemaakt van synoniemen.
---	---

Specifies the search terms. The Solr syntax for search terms can be applied here, e.g. combining with "and", and using double quotes for consecutive search terms. Search terms may be incomplete. Synonyms are also used.



Voorbeelden: **q=Utrecht** : geeft resultaten terug met de zoekterm Utrecht, bijv. adressen in de stad Utrecht, woonplaatsen en gemeenten in de provincie Utrecht.

Examples: **q=Utrecht** : returns results with the search term Utrecht, e.g. addresses in the city of Utrecht, places of residence and municipalities in the province of Utrecht.

**q="De Bilt"** : geeft resultaten terug met de zoekterm De Bilt, bijv. de woonplaats en gemeente De Bilt, of adressen in deze woonplaats. Let op dat bij het daadwerkelijk verzenden van het request onder water de URL-encodingregels toegepast worden, dus een spatie wordt verzonden als een plusteken.

**q="De Bilt"** : returns results with the search term De Bilt, e.g. the place of residence and municipality of De Bilt, or addresses in this place of residence. Note that when actually sending the request underwater, the URL encoding rules are applied, so a space is sent as a plus sign.

**q="Sint Jacob" Utre** : geeft o.a. adressen terug waarvan er delen achtereenvolgens beginnen met "Sint" en "Jacob", of met "St" (synoniem) en "Jacob", en waar ook een deel met "Utre" begint. Een voorbeeld is het adres St.-

Name	Description
	<p>Jacobsstraat 200 (officiële schrijfwijze) in Utrecht.</p> <p><b>q="Sint Jacob" Utre</b> : returns addresses of which parts start successively with "Sint" and "Jacob", or with "St" (synonym) and "Jacob", and where a part also starts with "Utre". An example is the address St.-Jacobsstraat 200 (official spelling) in Utrecht.</p>

lat lon

**number(\$float)** Hiermee kan een coördinaat (in lat/lon, ofwel WGS84) worden opgegeven. Met behulp van deze parameters worden de gevonden zoekresultaten gesorteerd op afstand van het meegegeven punt. Wanneer de locatie van de gebruiker bekend is, kan op deze manier effectiever worden gezocht.

This allows a coordinate (in lat/lon, or WGS84) to be specified. With the help of these parameters, the search results are sorted by distance from the given point. When the user's location is known, this makes it possible to search more effectively.



Het meegeven van een coördinaat is met name nuttig voor de suggest- en vrije geocoder-services. Hier worden meestal meerdere resultaten teruggegeven. Als decimaal scheidingsteken moet een punt worden opgegeven i.p.v. een komma.

Providing a coordinate is particularly useful for the suggest and free geocoder services. This usually returns several results. A period must be specified as a decimal separator instead of a comma.

lon

**number(\$float)** Hiermee kan een coördinaat (in lat/lon, ofwel WGS84) worden opgegeven. Met behulp van deze parameters worden de gevonden zoekresultaten gesorteerd op afstand van het meegegeven punt. Wanneer de locatie van de gebruiker bekend is, kan op deze manier effectiever worden gezocht.

This allows a coordinate (in lat/lon, or WGS84) to be specified. With the help of these parameters, the search results are sorted by distance from the given point. When the user's location is known, this makes it possible to search more effectively.

Het meegeven van een coördinaat is met name nuttig voor de suggest- en vrije geocoder-services. Hier worden meestal meerdere resultaten teruggegeven. Als

Name	Description
	decimaal scheidingsteken moet een punt worden opgegeven i.p.v. een komma.

Providing a coordinate is particularly useful for the suggest and free geocoder services. This usually returns several results. A period must be specified as a decimal separator instead of a comma.

Voorbeeld: `lat=52.09&lon=5.12`

Example: `lat=52.09&lon=5.12`

De resultaten worden gesorteerd op afstand van een bepaald punt in het centrum van Utrecht.

The results are sorted by distance from a certain point in the center of Utrecht.

lon

fl  
string  
(query)

Hiermee worden de velden opgegeven die teruggegeven dienen te worden.



This specifies the fields that need to be returned.

*Default value :* id identificatie weergavenaam bron type openbareruimte\_id  
nwb\_id openbareruimtetype straatnaam straatnaam\_verkort  
adresseerbaarobject\_id nummeraanduiding\_id huisnummer huisletter  
huisnummertoevoeging huis\_nlt postcode buurtnaam buurtcode wijknaam  
wijkcode woonplaatscode woonplaatsnaam gemeentecode gemeentenaam  
provinciecode provincienaam provincieafkorting kadastrale\_gemeentecode  
kadastrale\_gemeentenaam kadastrale\_sectie perceelnummer  
kadastrale\_grootte gekoppeld\_perceel kadastrale\_aanduiding volgnummer  
gekoppeld\_appartement wegnummer hectometernummer zijde hectometerletter

Name	Description
	<p><i>Default value</i> : id identification display name source type type      openbareruimte_id nwb_id public space type street name straatnaam_verkort      adresseerbaarobject_id nummeraanduiding_id house number house letter      house number addition huis_nlt postal code neighborhood name      neighborhood code neighborhood name neighborhood code residence code      city name municipality name province code province code province name      province name province abbreviation kadastrale_gemeentecode      kadastrale_gemeentenaam kadastrale_sectie lot number kadastrale_grootte      gekoppeld_perceel kadastrale_aanduiding serial number      gekoppeld_appartement road number hectometer number side hectometer      letter water board name water board code rdf_seealso centroide_ll      centroide_rd score</p>

id identificatie weergavenaam bron type open

**fq**

**array[string]** Hiermee kan een filter query worden opgegeven, bijv. **fq=bron:BAG** .  
**(query)**

Allows a filter query to be specified, e.g. **fq=source:BAG** .



*Default value* : List [ "type:(gemeente OR woonplaats OR weg OR postcode OR adres)" ]

*Default value* : List [ "type:(municipality OR city OR city OR road OR postal code OR address)" ]

type:(gemeente OR woonplaats OR weg OR p

**df**

**string**  
**(query)**

Hiermee wordt het default zoekveld opgegeven. Dit is het veld waar standaard in gezocht wordt, wanneer de veldnaam niet wordt meegegeven.

Specifies the default search field. This is the field that is searched in by default, when the field name is not provided.

*Default value* : tekst

*Default value* : text

tekst

Name	Description
qf string (query)	<p>Met behulp van deze parameter kan aan bepaalde <i>velden</i> een extra boost worden meegegeven. Hiermee kan de scoreberekening worden aangepast.</p> <p>With the help of this parameter, certain <i>fields</i> can be given an extra boost. This allows the score calculation to be adjusted.</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">qf</div>

bq array[string] (query)	<p>Met behulp van deze parameter kan aan bepaalde <i>veldwaarden</i> een extra boost worden meegegeven. Ook hiermee kan de scoreberekening worden aangepast.</p>
--------------------------------	--

With the help of this parameter, certain *field values* can be given an extra boost. This can also be used to adjust the score calculation.

Voor elke boost query moet een aparte `bq=<boost>` worden gebruikt.  
Bijvoorbeeld: `bq=type:gemeente^1.5&bq=type:woonplaats^1.5`.

For each boost query, a separate `bq=<boost>` should be used. For example:  
`bq=type:gemeente^1.5&bq=type:woonplaats^1.5`.



*Default value* : List [ "type:provincie^1.5", "type:gemeente^1.5", "type:woonplaats^1.5", "type:weg^1.5", "type:postcode^0.5", "type:adres^1" ]

*Default value* : List [ "type:province^1.5", "type:municipality^1.5", "type:city^1.5", "type:road^1.5", "type:postcode^0.5", "type:address^1" ]

type:provincie^1.5

type:gemeente^1.5

type:woonplaats^1.5

type:weg^1.5

type:postcode^0.5

type:adres^1

Name	Description
start integer (query)	<p>Hiermee wordt opgegeven wat de index is van het eerste resultaat dat teruggegeven wordt. Dit is zero-based. In combinatie met de rows-parameter kunnen deze services gepagineerd worden gevraagd. Het maximum is "10.000".</p> <p>Specifies the index of the first result that is returned. This is zero-based. In combination with the rows parameter, these services can be queried paginated. The maximum is "10,000".</p>
	<p><i>Default value : 0</i></p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">0</div>
rows integer (query)	<p>Hiermee wordt opgegeven wat het maximale aantal rijen (ofwel resultaten) is dat teruggegeven moet worden op deze bevraging. Het maximum is "100".</p> <p>This specifies the maximum number of rows (or results) that must be returned to this query. The maximum is "100".</p>
	<p><i>Default value : 10</i></p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">10</div>
sort string (query)	<p>Hiermee kan worden opgegeven hoe de sortering plaatsvindt.</p> <p>This allows you to specify how the sorting takes place.</p>
	<p><i>Default value : score desc,sortering asc,weergavenaam asc</i></p> <p><i>Default value : score desc,sortering asc,display name asc</i></p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">score desc,sortering asc,weergavenaam asc</div>
wt string (query)	<p>Hiermee wordt opgegeven wat het outputformaat is van de bevraging.</p> <p>This specifies the output format of the query.</p>
	<p><i>Available values : json, xml</i></p>

Name	Description
	<i>Default value : json</i>
	<i>Default value: json</i>

json

## Responses

Code	Description	Links
200	200 OK	No links

Media type

application/json

Controls Accept header.



Example Value Schema

```
{  
  "response": {  
    "docs": [  
      {}  
    ],  
    "maxScore": 0,  
    "numFound": 0,  
    "numFoundExact": true,  
    "start": 0  
  }  
}
```

Code	Description	Links						
400	400 Bad Request	No links						
<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json</div>								
<p><b>Example Value</b> Schema</p> <pre>{   "error": {     "code": 400,     "metadata": [       "error-class"     ],     "msg": "undefined field"   } }</pre>								
<p>Zodra er op basis van suggesties van de Suggest API een keuze is gemaakt, wordt de Lookup API aangeroepen, welke o.a. een (versimpelde) geometrie van de zoekopdracht teruggeeft.</p>								
GET	/lookup	<p>As soon as a choice has been made based on suggestions from the Suggest API, the Lookup API is called, which returns a (simplified) geometry of the search.</p> <div style="float: right; background-color: red; color: white; border-radius: 50%; width: 20px; height: 20px; display: flex; align-items: center; justify-content: center;">  </div>						
<p><b>Parameters</b></p>		<b>Try it out</b>						
<table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td><b>id</b> * required string (query)</td><td>Hiermee wordt het object-ID opgegeven.  Specifies the object ID.</td></tr> <tr> <td><b>fl</b> string (query)</td><td>Hiermee worden de velden opgegeven die teruggegeven dienen te worden.  This specifies the fields that need to be returned.</td></tr> </tbody> </table>			Name	Description	<b>id</b> * required string (query)	Hiermee wordt het object-ID opgegeven.  Specifies the object ID.	<b>fl</b> string (query)	Hiermee worden de velden opgegeven die teruggegeven dienen te worden.  This specifies the fields that need to be returned.
Name	Description							
<b>id</b> * required string (query)	Hiermee wordt het object-ID opgegeven.  Specifies the object ID.							
<b>fl</b> string (query)	Hiermee worden de velden opgegeven die teruggegeven dienen te worden.  This specifies the fields that need to be returned.							
<p><i>Default value : id identificatie weergavenaam bron type openbareruimte_id nwb_id openbareruimtetype straatnaam straatnaam_verkort adresseerbaarobject_id nummeraanduiding_id huisnummer huisletter huisnummertoevloeging huis_nlt postcode buurtnaam buurtcode wijknaam</i></p>								

Name	Description
	wijkcode woonplaatscode woonplaatsnaam gemeentecode gemeentenaam provinciecode provincienaam provincieafkorting kadastrale_gemeentecode kadastrale_gemeentenaam kadastrale_sectie perceelnummer kadastrale_grootte gekoppeld_perceel volgnummer gekoppeld_appartement kadastrale_aanduiding wegnummer hectometernummer zijde hectometerletter waterschapsnaam waterschapscode rdf_sealso centroide_ll centroide_rd

*Default value :* id identification display name source type type  
 openbareruimte\_id nwb\_id public space type street name straatnaam\_verkort  
 adresseerbaarobject\_id nummeraanduiding\_id house number house letter  
 house number addition huis\_nlt postal code neighborhood name  
 neighborhood code neighborhood name neighborhood code residence code  
 city name municipality name province code province code province name  
 province name province abbreviation kadastrale\_gemeentecode  
 kadastrale\_gemeentenaam kadastrale\_sectie lot number kadastrale\_grootte  
 gekoppeld\_perceel serial number gekoppeld\_appartement  
 kadastrale\_aanduiding road number hectometer number side hectometer  
 letter water board name water board code rdf\_sealso centroide\_ll  
 centroide\_rd

id identificatie weergavenaam bron type open



**fq**

**array[string]** Hiermee kan een filter query worden opgegeven, bijv. **fq=bron:BAG** .  
**(query)**

Allows a filter query to be specified, e.g. **fq=source:BAG** .

**wt**

**string**  
**(query)**

Hiermee wordt opgegeven wat het outputformaat is van de bevraging.

This specifies the output format of the query.

*Available values :* json, xml

*Default value :* json

*Default value:* json

json

## Responses

Code	Description	Links
200	<p>200 OK</p> <p>Media type</p> <div style="border: 2px solid green; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p><b>Example Value</b> Schema</p> <pre>{   "response": {     "docs": [       {}     ],     "maxScore": 0,     "numFound": 0,     "numFoundExact": true,     "start": 0   } }</pre>	No links
400	<p>400 Bad Request</p> <p>Media type</p> <div style="border: 2px solid green; padding: 2px; display: inline-block;">application/json</div> <p><b>Example Value</b> Schema</p> <pre>{   "error": {     "code": 400,     "metadata": [       "error-class"     ],     "msg": "undefined field"   } }</pre>	No links 
GET	<p>/reverse</p> <p>The Reverse API offers the possibility to specify a location (point geometry) and then receive different data in a range around this location.</p>	^

**Parameters****Try it out**

Name	Description
X <code>number(\$float) (query)</code>	Hiermee wordt het coördinaat in X/Y (RD), of in lat/lion (WGS84) opgegeven. Vanaf dit coördinaat gaat de reverse geocoder zoeken.

Specifies the coordinate in X/Y (RD), or in lat/lion (WGS84). From this coordinate, the reverse geocoder starts searching.

*Example : 194195.304*

194195.304

Y <code>number(\$float) (query)</code>	Hiermee wordt het coördinaat in X/Y (RD), of in lat/lion (WGS84) opgegeven. Vanaf dit coördinaat gaat de reverse geocoder zoeken.
---	--

Specifies the coordinate in X/Y (RD), or in lat/lion (WGS84). From this coordinate, the reverse geocoder starts searching.

*Example : 465885.902*



465885.902

lat <code>number(\$float) (query)</code>	Hiermee wordt het coördinaat in X/Y (RD), of in lat/lion (WGS84) opgegeven. Vanaf dit coördinaat gaat de reverse geocoder zoeken.
---	--

Specifies the coordinate in X/Y (RD), or in lat/lion (WGS84). From this coordinate, the reverse geocoder starts searching.

lat

lon <code>number(\$float) (query)</code>	Hiermee wordt het coördinaat in X/Y (RD), of in lat/lion (WGS84) opgegeven. Vanaf dit coördinaat gaat de reverse geocoder zoeken.
---	--

Specifies the coordinate in X/Y (RD), or in lat/lion (WGS84). From this coordinate, the reverse geocoder starts searching.

lon

Name	Description
type string (query)	<p>Hiermee kan worden opgegeven welke resultaat types je terug krijgt in het resultaat. Voor elke type dat je wilt terugkrijgen moet je een aparte type parameter toevoegen, bijv. <code>type=adres&amp;type=gemeente&amp;type=perceel</code> . Met <code>type=*</code> krijg je alle types terug. Deze wordt voor het bepalen van de resultaten gebruikt en heeft dus geen invloed op aantal resultaten dat wordt teruggegeven.</p> <p>This allows you to specify which result types you get back in the result. For each type you want to get back, you need to add a separate type parameter, e.g. <code>type=adres&amp;type=gemeente&amp;type=perceel</code> . With <code>type=*</code> you get all types back. This is used to determine the results and therefore has no influence on the number of results that are returned.</p> <p><i>Default value : adres</i></p> <p><i>Default value : address</i></p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">adres</div>
distance integer (query)	<p>Hiermee kan een maximale zoekstraal in meters worden opgegeven. Er zullen dan geen resultaten worden teruggegeven die verder liggen dan deze waarden.</p> <p>Specifies a maximum search radius in meters. No results will be returned that are beyond these values.</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">distance</div>
fl string (query)	<p>Hiermee worden de velden opgegeven die teruggegeven dienen te worden.</p> <p>This specifies the fields that need to be returned.</p> <p><i>Default value : id type weergavenaam score afstand</i></p> <p><i>Default value : id type display name score distance</i></p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">id type weergavenaam score afstand</div>
fq string (query)	<p>Hiermee kan een filter query worden opgegeven, bijv. <code>fq=bron:BAG</code> . Deze wordt uitgevoerd nadat de dichtsbijzijnde resultaten zijn gevonden, dus resultaten die hier niet aan voldoen worden weggefilterd waardoor je minder resultaten dan het</p>

Name	Description
	aantal rows kan terugkrijgen.

Allows a filter query to be specified, e.g. `fq=source:BAG`. This is executed after the nearest results have been found, so results that do not meet this are filtered out, so you can get fewer results than the number of rows back.

start  
integer  
(query)

Hiermee wordt opgegeven wat de index is van het eerste resultaat dat teruggegeven wordt. Dit is zero-based. In combinatie met de rows-parameter kunnen deze services gepagineerd worden gevraagd. Het maximum is "10.000".

Specifies the index of the first result that is returned. This is zero-based. In combination with the rows parameter, these services can be queried paginated. The maximum is "10,000".

*Default value : 0*



rows  
integer  
(query)

Hiermee wordt opgegeven wat het maximale aantal rijen (ofwel resultaten) is dat teruggegeven moet worden op deze bevraging. Het maximum is "100".

This specifies the maximum number of rows (or results) that must be returned to this query. The maximum is "100".

*Default value : 10*

wt  
string  
(query)

Hiermee wordt opgegeven wat het outputformaat is van de bevraging.

This specifies the output format of the query.

*Available values : json, xml*

*Default value : json*

*Default value: json*

Name	Description
json	

## Responses

Code	Description	Links
------	-------------	-------

200	200 OK	No links
-----	--------	----------

Media type

**application/json**

Controls Accept header.

**Example Value** Schema

```
{
  "response": {
    "docs": [
      {}
    ],
    "maxScore": 0,
    "numFound": 0,
    "numFoundExact": true,
    "start": 0
  }
}
```



400	400 Bad Request	No links
-----	-----------------	----------

Media type

**application/json****Example Value** Schema

```
{
  "error": {
    "code": 400,
    "metadata": [
      "error-class"
    ],
    "msg": "undefined field"
  }
}
```

De Suggest API biedt de mogelijkheid om een (gedeelte van een) zoekopdracht op te voeren, waarnaar er suggesties teruggegeven worden.

**GET****/suggest**

The Suggest API offers the possibility to enter a (part of a) search query, to which suggestions are returned.

^

**Parameters****Try it out**

Name	Description
------	-------------

**q \* required****string  
(query)**

Hiermee worden de zoektermen opgegeven. De Solr-syntax voor zoektermen kan hier worden toegepast, bijv. combineren met "and", en het gebruik van dubbele quotes voor opeenvolgende zoektermen. Zoektermen mogen incompleet zijn. Ook wordt er gebruik gemaakt van synoniemen.

Specifies the search terms. The Solr syntax for search terms can be applied here, e.g. combining with "and", and using double quotes for consecutive search terms. Search terms may be incomplete. Synonyms are also used.

Voorbeelden: **q=Utrecht** : geeft resultaten terug met de zoekterm Utrecht, bijv. adressen in de stad Utrecht, woonplaatsen en gemeenten in de provincie Utrecht.



Examples: **q=Utrecht** : returns results with the search term Utrecht, e.g. addresses in the city of Utrecht, places of residence and municipalities in the province of Utrecht.

**q="De Bilt"** : geeft resultaten terug met de zoekterm De Bilt, bijv. de woonplaats en gemeente De Bilt, of adressen in deze woonplaats. Let op dat bij het daadwerkelijk verzenden van het request onder water de URL-encodingregels toegepast worden, dus een spatie wordt verzonden als een plusteken.

**q="De Bilt"** : returns results with the search term De Bilt, e.g. the place of residence and municipality of De Bilt, or addresses in this place of residence. Note that when actually sending the request underwater, the URL encoding rules are applied, so a space is sent as a plus sign.

**q="Sint Jacob" Utre** : geeft o.a. adressen terug waarvan er delen achtereenvolgens beginnen met "Sint" en "Jacob", of met "St" (synoniem) en "Jacob", en waar ook een deel met "Utre" begint. Een voorbeeld is het adres St.-

Name	Description
	Jacobsstraat 200 (officiële schrijfwijze) in Utrecht.  <b>q="Sint Jacob" Utre</b> : returns addresses of which parts start successively with "Sint" and "Jacob", or with "St" (synonym) and "Jacob", and where a part also starts with "Utre". An example is the address St.-Jacobsstraat 200 (official spelling) in Utrecht.

lat lat number(\$float) (query)	Hiermee kan een coördinaat (in lat/lon, ofwel WGS84) worden opgegeven. Met behulp van deze parameters worden de gevonden zoekresultaten gesorteerd op afstand van het meegegeven punt. Wanneer de locatie van de gebruiker bekend is, kan op deze manier effectiever worden gezocht.
---------------------------------------	--

This allows a coordinate (in lat/lon, or WGS84) to be specified. With the help of these parameters, the search results are sorted by distance from the given point. When the user's location is known, this makes it possible to search more effectively.



Het meegeven van een coördinaat is met name nuttig voor de suggest- en vrije geocoder-services. Hier worden meestal meerdere resultaten teruggegeven. Als decimaal scheidingsteken moet een punt worden opgegeven i.p.v. een komma.

Providing a coordinate is particularly useful for the suggest and free geocoder services. This usually returns several results. A period must be specified as a decimal separator instead of a comma.

Voorbeeld: **lat=52.09&lon=5.12**

Example: **lat=52.09&lon=5.12**

De resultaten worden gesorteerd op afstand van een bepaald punt in het centrum van Utrecht.

The results are sorted by distance from a certain point in the center of Utrecht.

lon number(\$float) (query)	Hiermee kan een coördinaat (in lat/lon, ofwel WGS84) worden opgegeven. Met behulp van deze parameters worden de gevonden zoekresultaten gesorteerd op afstand van het meegegeven punt. Wanneer de locatie van de gebruiker bekend
-----------------------------------	---

Name	Description
	<p>is, kan op deze manier effectiever worden gezocht.</p> <p>This allows a coordinate (in lat/lon, or WGS84) to be specified. With the help of these parameters, the search results are sorted by distance from the given point. When the user's location is known, this makes it possible to search more effectively.</p>

Het meegeven van een coördinaat is met name nuttig voor de suggest- en vrije geocoder-services. Hier worden meestal meerdere resultaten teruggegeven. Als decimaal scheidingsteken moet een punt worden opgegeven i.p.v. een komma.

Providing a coordinate is particularly useful for the suggest and free geocoder services. This usually returns several results. A period must be specified as a decimal separator instead of a comma.

Voorbeeld: `lat=52.09&lon=5.12`

Example: `lat=52.09&lon=5.12`

De resultaten worden gesorteerd op afstand van een bepaald punt in het centrum van Utrecht.



The results are sorted by distance from a certain point in the center of Utrecht.

lon

fl  
string  
(query)

Hiermee worden de velden opgegeven die teruggegeven dienen te worden.

This specifies the fields that need to be returned.

*Default value* : id weergavenaam type score adrestype

*Default value* : id display name type score address type

id weergavenaam type score adrestype

fq  
array[string]  
(query)

Hiermee kan een filter query worden opgegeven, bijv. `fq=bron:BAG`.

Allows a filter query to be specified, e.g. `fq=source:BAG`.

Name	Description
	<p><i>Default value : List [ "type:(gemeente OR woonplaats OR weg OR postcode OR adres)" ]</i></p> <p><i>Default value : List [ "type:(municipality OR city OR city OR road OR postal code OR address)" ]</i></p>
	<p><code>type:(gemeente OR woonplaats OR weg OR p</code></p>

**qf**  
string  
(query)

Met behulp van deze parameter kan aan bepaalde *velden* een extra boost worden meegegeven. Hiermee kan de scoreberekening worden aangepast.

With the help of this parameter, certain *fields* can be given an extra boost. This allows the score calculation to be adjusted.

*Default value : exacte\_match^1 suggest^0.5 huisnummer^0.5 huisletter^0.5 huisnumbertoevoeging^0.5*

`exacte_match^1 suggest^0.5 huisnummer^0.5`



**bq**  
array[string]  
(query)

Met behulp van deze parameter kan aan bepaalde *veldwaarden* een extra boost worden meegegeven. Ook hiermee kan de scoreberekening worden aangepast.

With the help of this parameter, certain *field values* can be given an extra boost. This can also be used to adjust the score calculation.

Voor elke boost query moet een aparte `bq=<boost>` worden gebruikt.  
Bijvoorbeeld: `bq=type:gemeente^1.5&bq=type:woonplaats^1.5` .

For each boost query, a separate `bq=<boost>` should be used. For example:  
`bq=type:gemeente^1.5&bq=type:woonplaats^1.5` .

*Default value : List [ "type:provincie^1.5", "type:gemeente^1.5", "type:woonplaats^1.5", "type:weg^1.5", "type:postcode^0.6", "type:adres^1" ]*

*Default value : List [ "type:province^1.5", "type:municipality^1.5", "type:city^1.5", "type:road^1.5", "type:postcode^0.6", "type:address^1" ]*

`type:provincie^1.5`

`type:gemeente^1.5`

Name	Description
	type:woonplaats^1.5
	type:weg^1.5
	type:postcode^0.6
	type:adres^1

start integer (query)	Hiermee wordt opgegeven wat de index is van het eerste resultaat dat teruggegeven wordt. Dit is zero-based. In combinatie met de rows-parameter kunnen deze services gepagineerd worden gevraagd. Het maximum is "10.000".  Specifies the index of the first result that is returned. This is zero-based. In combination with the rows parameter, these services can be queried paginated. The maximum is "10,000".
	<i>Default value : 0</i>
	<input type="text" value="0"/>
rows integer (query)	Hiermee wordt opgegeven wat het maximale aantal rijen (ofwel resultaten) is dat teruggegeven moet worden op deze bevraging. Het maximum is "100".  This specifies the maximum number of rows (or results) that must be returned to this query. The maximum is "100".
	<i>Default value : 10</i>
	<input type="text" value="10"/>
sort string (query)	Hiermee kan worden opgegeven hoe de sortering plaatsvindt.  This allows you to specify how the sorting takes place.
	<i>Default value : score desc,sortering asc,weergavenaam asc</i>
	<i>Default value : score desc,sortering asc,display name asc</i>

Name	Description
<code>score desc,sortering asc,weergavenaam asc</code>	
<code>wt</code> <code>string</code> (query)	Hiermee wordt opgegeven wat het output formaat is van de bevraging. This specifies the output format of the query.
<i>Available values : json, xml</i>	
<i>Default value : json</i>	
<i>Default value: json</i>	
<code>json</code>	

## Responses

Code	Description	Links
200	200 OK	<i>No links</i>

Media type

`application/json`

Controls Accept header.

**Example Value** Schema

**Code****Links**

```
        "numFound": 0,  
        "startOffset": 0,  
        "suggestion": [  
            "string"  
        ]  
    }  
}  
}
```

400

400 Bad Request

No links

Media type

**application/json****Example Value** Schema

```
{  
    "error": {  
        "code": 400,  
        "metadata": [  
            "error-class"  
        ],  
        "msg": "undefined field"  
    }  
}
```

**Schemas****error****highlighting****response****spellcheck****VALID**

{...}