

eGeodata Austria Adresssuch-Service

1	Dokumenteninformation.....	2
1.1	Dokumentenhistorie und Versionskontrolle	2
2	Ziel und Zweck	3
3	Benutzerhandbuch - Dokumentenversionierung	4
4	Adresssuch-Service / Tokenservice	5
4.1	Zugang zum Tokenservice via Portalverbund	5
4.1.1	Portalverbundprotokoll (PVP) - Prinzipien	5
4.1.2	Kunden ohne Zugang zu einem eigenen PVP-Stammportal	6
4.1.3	Zertifikatsbeschreibung des BEV-PVP-Stammportals.....	6
4.1.4	Prozessablauf zum Bezug eines Tokens.....	7
4.1.5	Tokenservice Request / Response	8
4.2	Zugang zum Adresssuch-Service.....	8
4.2.1	Adresssuch-Service Request / Response	8
4.2.2	Adresssuch-Service Request-Parameter	13
4.2.3	Adresssuch-Service Response-Parameter	14
4.2.4	Sortierung und Limits.....	15
4.2.5	Verarbeitungslogik.....	15
4.3	Fehlermeldungen.....	17
4.4	URL für den aktuellen Status des Adresssuch-Service.....	17
4.5	Unterschiede zwischen der TU und der PU	18
5	Anhang	19
5.1	Offene Punkte.....	19
5.2	Begriffsbestimmungen und Abkürzungen	19

1 Dokumenteninformation

1.1 Dokumentenhistorie und Versionskontrolle

Version	Status	Datum	Verantwortlicher, Teilnehmer	Änderungsbeschreibung
1.00	F	16.04.2014	BEV Architekturmanagement (architekturmanagement@bev.gv.at)	Neuanlegung des Dokumentes
2.00	F	24.06.2014	BEV Architekturmanagement (architekturmanagement@bev.gv.at)	Diverse Änderungen / Finalisierung
3.00	F	03.10.2017	BEV Architekturmanagement (architekturmanagement@bev.gv.at)	Externes (eigenes) Stammportal entfernt
4.00	F	29.08.2018	BEV Architekturmanagement (architekturmanagement@bev.gv.at)	URL schreibweise für Adressuche angepasst Zertifikat Format angepasst
5.00	F	13.12.2019	BEV Architekturmanagement (architekturmanagement@bev.gv.at)	Corporate-Design Anpassung
6.00	F	16.12.2019	BEV Zentrales Marketing (zentrales.marketing@bev.gv.at)	Anpassung Wording (Adresssuch-Service)
7.00	F	07.01.2020	BEV Architekturmanagement (architekturmanagement@bev.gv.at)	Codebeispiele für geotype =3 hinzugefügt
Hinweis zur Status- und Versionsverwaltung				
Status	A R F	In Arbeit = beim Erstellen und Bearbeiten eines Dokuments in Review = beim Einholen von Reviewkommentaren Freigegeben = als Abschluss der Bearbeitung und Reviewphase.		
Versionierung	0.01, 0.02 ... 1.00 1.01, 1.02 ... 2.00	nicht freigegebene Versionen, mit dem Status ‚in Arbeit‘ oder ‚in Review‘ erste freigegebene Version mit dem Status ‚freigegeben‘ Versionen, welche das ursprüngliche Produkt in Version 1.00 ergänzen und ‚in Arbeit‘ oder ‚in Review‘ sind zweite freigegebene Version mit dem Status ‚freigegeben‘		

2 Ziel und Zweck

Dieses Dokument ist an Softwarehersteller gerichtet, die eine Software zur direkten Benutzung der Adresssuche mittels Adresssuch-Service entwickeln wollen. Es beschreibt den Ablauf einer Adresssuche und die Abholung eines Tokens, welcher für die Abfrage einer Adresse benötigt wird.

Die Schnittstelle ermöglicht es den Kunden, direkt aus eigenen Anwendungen nach Adressen zu recherchieren.

3 Benutzerhandbuch - Dokumentenversionierung

Das Benutzerhandbuch wird in einer Version geführt, welche für die Test- und für die Produktivumgebung gleich ist.

4 Adresssuch-Service / Tokenservice

Der Zugang zum Adresssuch-Service ist über Token geregelt, welche über den österreichischen Portalverbund (PV) abgeholt werden. Der Portalverbund entspricht dem Portalverbundprotokoll (<http://portal.bmi.gv.at/ref/downloads/PortalVerbundProtokollV1.5.pdf>).

4.1 Zugang zum Tokenservice via Portalverbund

Damit können alle Kunden für das Adresssuch-Service freigeschaltet werden, die Zugänge zu einem Stammportal haben und dieses entsprechend zum BEV Anwendungsportal verbunden ist. Kunden, die nicht Teilnehmer des PV sind, haben die Möglichkeit, mittels BEV Stammportal auf das Adresssuch-Service zuzugreifen.

Die angestrebte Adresssuch-Service Schnittstelle soll es Kunden ermöglichen, direkt aus eigenen Anwendungen heraus nach Adressen zu recherchieren. Diese Lösung soll es den Kunden ermöglichen, Arbeitsabläufe mit Bestellungen von BEV-Produkten besser in die eigenen Prozesse zu integrieren und dadurch den Workflow zu optimieren.

4.1.1 Portalverbundprotokoll (PVP) - Prinzipien

- Der Benutzer wird nur im Stammportal verwaltet, dort wird auch verwaltet, auf welche Applikationen/Services der Benutzer zugreifen kann.
- Die Authentifizierung erfolgt auf Basis von Zertifikaten.

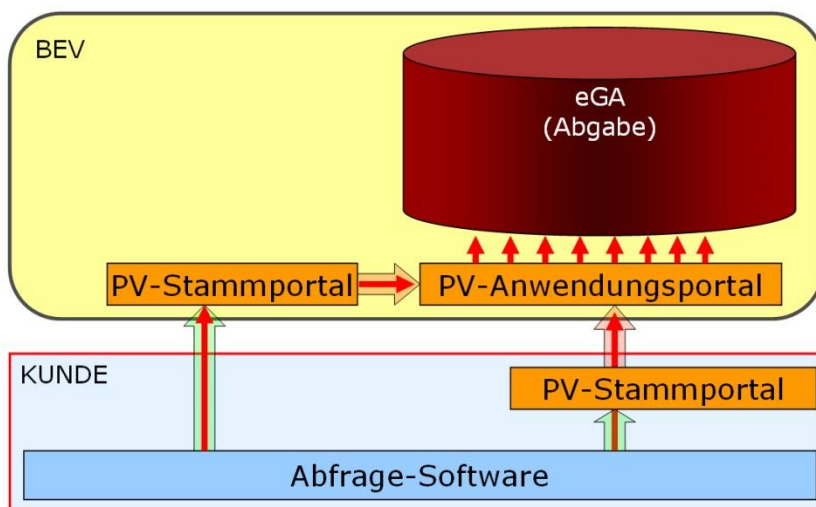


Abbildung 1: Systemanordnung des Produkt-Webservices

4.1.2 Kunden ohne Zugang zu einem eigenen PVP-Stammportal

Diese Kunden bekommen im Rahmen der Freischaltung für das Adresssuch-Service ein Client-Zertifikat vom BEV ausgestellt, dieses ist am Client zu installieren und wird bei der Kommunikation mit dem PV-Stammportal des BEV von der Abfragesoftware zur Authentifizierung mitgesendet.

Technische Voraussetzungen:

- Abfragesoftware
- Client-Zertifikat des BEV

4.1.3 Zertifikatsbeschreibung des BEV-PVP-Stammportals

Technischer Hintergrund:

Wenn sich Clients mittels eines Zertifikates identifizieren, ist der Client der Sender und der Webserver ist der Empfänger, d.h. der Webserver muss dem gesendeten Zertifikat vertrauen. Wenn der Client von einem Webserver eine Antwort erwartet, muss er dem Zertifikat des Webservers vertrauen (für die Antwort ist der Webserver der Absender).

Zertifikatsformat:

Bei der Prüfung eines Zertifikates auf Vertrauenswürdigkeit wird unter anderem der Aussteller des Zertifikates ermittelt und nachgesehen, ob dieser Aussteller (bzw. dessen Zertifikat) in einer jeweils lokal gespeicherten Liste von „vertrauenswürdigen Ausstellern“ enthalten ist. In Fall des Zugriffs auf <https://sti.bev.gv.at> muss der Client dem BEV Server-Zertifikat (ausgestellt von VeriSign) und der BEV Server dem Client-Zertifikat (ausgestellt vom BEV) vertrauen. Die VeriSign-Aussteller-Zertifikate sind in praktisch allen aktuellen Betriebssystemen und Entwicklungsumgebungen in der Liste von vertrauenswürdigen Ausstellern vorhanden und werden auch regelmäßig mit Updates (z.B. WindowsUpdate) auf den neuesten Stand gebracht.

Es wird vorgeschlagen, in Fällen wo ein Ausstellerzertifikat benötigt wird, den Import des Ausstellerzertifikates und die Einstufung als vertrauenswürdig nicht dem Kunden zu überlassen, sondern beim Installationsprozess der Client-Software automatisch durchzuführen.

Dieses Zertifikat wird im PKCS#12-Format übermittelt. Zusätzlich kann das Zertifikat auch im Java-üblichen JKS-Format übermittelt werden.

Hinweis für die Implementierung:

Es ist bekannt, dass es zumindest in WindowsXP nicht möglich ist, den vom BEV in einem automatischen Prozess erstellten PKCS12-Zertifikatsstore in den Windows-eigenen Zertifikatsstore für "persönliche Zertifikate" zu importieren. Im Zertifikats-Import-Assistenten

unter WindowsXP können Kennwörter mit einer maximalen Länge von 32 Zeichen angegeben werden und die BEV-Kennwörter haben eine Länge von 88 Zeichen.

BEV-Zertifikate sind als "elektronischer Ausweis" vorgesehen, mit dem eine bestimmte Client-Applikation die Identität eines Antragstellers zur Benützung eines kostenpflichtigen Dienstes an die Server-Applikation übermittelt. Durch einen Import des Zertifikates in den für den jeweiligen User "öffentlichen Zertifikatstore für persönliche Zertifikate" wird dieses Zertifikat automatisch nicht nur für die bestimmte Applikation zugänglich, sondern auch für jede andere Software und natürlich auch den User selbst. Es wäre somit ohne Kenntnis des Zertifikat-Kennwortes eine (auch missbräuchliche) Verwendung des Zertifikates möglich. Eine falsche Verwendung des Zertifikates kann zwar nicht wirklich verhindert werden, sollte aber zumindest möglichst erschwert werden.

Durch einen Import des Zertifikates in den Windows-Zertifikatstore können zwar Kosten für die Software-Entwicklung gespart werden, aber das auf Kosten der Sicherheit. Im BEV werden die Zertifikate bei jedem Start einer Software aus dem PKCS12-File gelesen und das zugehörige Kennwort ist in der Konfiguration der jeweiligen Applikation verschlüsselt gespeichert.

4.1.4 Prozessablauf zum Bezug eines Tokens

- Der Webserver, der das Adresssuch-Service benutzen will (= Client), sendet einen http Request an das Stammportal
- Das Stammportal prüft den Request auf Zulässigkeit und generiert im positiven Fall einen PVP-Request, der an das Anwendungsportal gesendet wird.
- Das Anwendungsportal prüft die Gültigkeit und Zulässigkeit des PVP-Requests (Zertifikat).
- Das Anwendungsportal sendet den Request an das Tokenservice weiter.
- Das Tokenservice retourniert eine http Reponse (JSON) an das Anwendungsportal.
- Das Anwendungsportal leitet die http Response an das Stammportal weiter.
- Das Stammportal leitet die http Response an den Client weiter.

4.1.5 Tokenservice Request / Response

Das Tokenservice dient zur Anforderung eines Tokens, das für die Verwendung des Adresssuch-Service notwendig ist.

Über die PVP Infrastruktur wird das Tokenservice mit dem folgenden Request aufgerufen:

Codebeispiel 1: Tokenservice-Request

<https://sti.bev.gv.at:8443/adr-service/TokenService>

Das Webservice antwortet Content-Type „application/json;charset=UTF-8“.

Die Antwort auf die Suchanfrage erfolgt in Form eine JSON Files.

Codebeispiel 2: Tokenservice-Response

```
[{"token":"c2b00fc4-8ec9-46ca-854b-6c0de27fd371"}]</body>
```

4.2 Zugang zum Adresssuch-Service

Das angestrebte Adresssuch-Service soll es Kunden ermöglichen, direkt aus eigenen Anwendungen heraus nach Adressen zu recherchieren. Diese Lösung soll es den Kunden ermöglichen, Arbeitsabläufe, die mit Adressen im Zusammenhang stehen, besser in die eigenen Prozesse zu integrieren und dadurch den Workflow zu optimieren.

4.2.1 Adresssuch-Service Request / Response

Der Aufruf des Webservices erfolgt über eine URL, die Übergabe von Suchbegriffen erfolgt über den Pflichtparameter „term“.

Codebeispiel 3: Adresssuch-Service -Request (term+token)

[http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?
term=Schi%ffamt&
token=60cfd451-1dcd-4edc-a952-044dc7db6869](http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?term=Schi%ffamt&token=60cfd451-1dcd-4edc-a952-044dc7db6869)

Der Wert des Parameters „term“ muss UTF-8 enkodiert sein.

Als zweiter erforderlicher Parameter muss ein Token übergeben werden.

Das Token muss zuvor über ein Tokenservice bezogen werden und ist nur für einen begrenzten Zeitraum gültig (eine Stunde).

Das Webservice antwortet bei Verwendung einer Callback Funktion mit Content-Type „application/json;charset=UTF-8“, ansonsten wird text/html retourniert.

Die Antwort auf die Suchanfrage erfolgt in Form eine JSON Files. Falls eine Callbackfunktion

mitübergeben wird (z.B: bei der Verwendung von jQuery UI Autocomplete), ist eine JSON Integration mit einer externen Datenquelle möglich.

Codebeispiel 4: Adresssuch-Service -Response (term+token)

```
[
  {
    "label": "Wien, <b>Schi ffamt</b>sgasse",
    "value": "Wien, Schi ffamtsgasse",
    "x": "",
    "y": "",
    "geotype": "2",
    "id": "S904240"
  },
  {
    "label": "Wien, <b>Schi ffamt</b>sufer",
    "value": "Wien, Schi ffamtsufer",
    "x": "",
    "y": "",
    "geotype": "2",
    "id": "S918279"
  }
]
```

Hierbei wird in drei Geotypes unterschieden:

- "geotype": "1"

Dies bedeutet man befindet sich in der Gemeinde- bzw. Ortsteilebene.

Codebeispiel 5: Adresssuch-Service -Request (geotype=1)

[http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?
term=Wien&token=60cfd451-1dcd-4edc-a952-044dc7db6869](http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?term=Wien&token=60cfd451-1dcd-4edc-a952-044dc7db6869)

Codebeispiel 6: Adresssuch-Service -Response (geotype=1)

```
[
  {
    "label": "<b>Wien</b>",
    "value": "Wien",
    "x": "225973.5226",
    "y": "345567.1651",
    "geotype": "1",
    "id": "M90001"
  },
  {
    "label": "<b>Wien</b>er Neudorf",
    "value": "Wiener Neudorf",
    "x": "222048.7903",
    "y": "331749.1427",
    "geotype": "1",
    "id": "H5168"
  },
  ...
]
```

- "geotype": "2"

Dies bedeutet man befindet sich in der Straßenebene. Diese Ebene gilt ab dem Beistrich nach der Gemeinde bzw. Ortsteil.

Codebeispiel 7: Adresssuch-Service -Request (geotype=2)

<http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?term=Wien,Schiffamt&token=60cfd451-1dcd-4edc-a952-044dc7db6869>

Codebeispiel 8: Adresssuch-Service -Response (geotype=2)

```
[
  {
    "label": "Wien, <b>Schiffamt</b>sgasse",
    "value": "Wien, Schiffamtsgasse",
    "x": "",
    "y": "",
    "geotype": "2",
    "id": "S904240"
  },
  {
    "label": "Wien, <b>Schiffamt</b>sufer",
    "value": "Wien, Schiffamtsufer",
    "x": "",
    "y": "",
    "geotype": "2",
    "id": "S918279"
  }
]
```

- "geotype": "3"

Dies bedeutet man befindet sich in der Hausnummerenebene. Diese Ebene gilt sobald eine Hausnummer mitgesendet wird.

Codebeispiel 9: Adresssuch-Service -Request (geotype=2)

<http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?term=Wien,Schiffamtsgasse 1&token=60cfd451-1dcd-4edc-a952-044dc7db6869>

Codebeispiel 10: Adresssuch-Service -Response (geotype=2)

```
[
  {
    "label": "<b>Wien, Schiffamtsgasse 1</b> - 3",
    "value": "Wien, Schiffamtsgasse 1 - 3",
    "x": "225926.971023",
    "y": "346722.022514",
    "geotype": "3",
    "id": "6796719"
  },
  {
    "label": "<b>Wien, Schiffamtsgasse 1</b>>0",
    "value": "Wien, Schiffamtsgasse 10",
    "x": "226003.30023",
    "y": "346742.439775",
    "geotype": "3",
    "id": "6796726"
  }
  ...
]
```

Die Eingabe von weiteren optionalen Parametern ist ebenfalls möglich.

Bei allen räumlichen Parametern ist es erforderlich, die Information zur jeweiligen Projektion mit dem Parameter „projection“ zu übergeben.

Folgende Werte sind erlaubt:

- epsg_4258 (GEO_ETRS89)
- epsg_4312 (GEO_MGI)
- epsg_31255 (GK_MGI_M31)

Codebeispiel 11: Adresssuch-Service -Request (term+token+projection)

[http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?](http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?term=Schi%20ffamt&token=60cfd451-1dcd-4edc-a952-044dc7db6869&projection=epsg_4258)

```
term=Schi%20ffamt&
token=60cfd451-1dcd-4edc-a952-044dc7db6869&
projection=epsg_4258
```

Codebeispiel 12: Adresssuch-Service -Response (term+token+projection)

```
[
  {
    "label": "Wi en, <b>Schi ffamt</b>sgasse",
    "value": "Wi en, Schi ffamtsgasse",
    "x": "16.3730227906",
    "y": "48.2186556616",
    "geotype": "2",
    "id": "S904240"
  },
  {
    "label": "Wi en, <b>Schi ffamt</b>sufer",
    "value": "Wi en, Schi ffamtsufer",
    "x": "16.3741718741",
    "y": "48.216261026",
    "geotype": "2",
    "id": "S918279"
  }
]
```

Zur Benützung von ortsbezogenen Sortierungen kann ein Koordinatenpunkt übergeben werden, z.B. (48.312342279083595, 16.298046112060547).

Codebeispiel 13: Adresssuch-Service -Request (term+token+projection+point)

[http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?](http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?term=Schi%20ffamt&token=60cfd451-1dcd-4edc-a952-044dc7db6869&point=(48.312342279083595,16.298046112060547)&projection=epsg_4258)

```
term=Schi%20ffamt&
token=60cfd451-1dcd-4edc-a952-044dc7db6869&
point=(48.312342279083595,16.298046112060547)&
projection=epsg_4258
```

Codebeispiel 14: Adresssuch-Service -Response (term+token+projection+point)

```
[
  {
```

```

    "label": "Wien, <b>Schiffamt</b>sufer",
    "value": "Wien, Schiffamtsufer",
    "x": "16.3741718741",
    "y": "48.216261026",
    "geotype": "2",
    "id": "S918279"
  },
  {
    "label": "Wien, <b>Schiffamt</b>sgasse",
    "value": "Wien, Schiffamtsgasse",
    "x": "16.3730227906",
    "y": "48.2186556616",
    "geotype": "2",
    "id": "S904240"
  }
]

```

Zur Einschränkung der Suchergebnisse auf einen definierten Extent kann ein entsprechender Wert übergeben werden.

Codebeispiel 15: Adresssuch-Service -Request (term+token+projection+extent)

<http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?>

```

  term=Wien&
  token=60cfd451-1dcd-4edc-a952-044dc7db6869&
  extent=[[16.3,48],[16.4,48.05]]&
  projection=epsg_4258

```

Codebeispiel 16: Adresssuch-Service -Response (term+token+projection+extent)

```

[
  {
    "label": "Münchendorf, <b>Wien</b>erstraße",
    "value": "Münchendorf, Wienerstraße",
    "x": "16.3801157945",
    "y": "48.0332998783",
    "geotype": "2",
    "id": "S17648"
  },
  {
    "label": "Traiskirchen, <b>Wien</b>erbergerstraße",
    "value": "Traiskirchen, Wienerbergerstraße",
    "x": "16.3058800157",
    "y": "48.0401801023",
    "geotype": "2",
    "id": "S11271"
  }
]

```

Es besteht die Möglichkeit, eine Callback Funktion als Parameter zu übergeben, die Antwort wird dann als Argument dieser Funktion zurückgeliefert. Dieser Modus ist beispielsweise beim Einsatz von jQuery UI Autocomplete notwendig, um den Zugriff auf eine fremde Domäne zu ermöglichen (siehe dataType: "jsonp" in der entsprechenden Dokumentation des jQuery Plugins).

Codebeispiel 17: Adresssuch-Service -Request (callback+term+token)

```
http://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON?  
callback=jsonp12345&  
term=Schi ffamt&  
token=fc8b0a40-319a-4a39-b68b-cb001af1f2d8
```

Codebeispiel 18: Adresssuch-Service -Response (callback+term+token)

```
jsonp12345(  
  [  
    {  
      "label": "Wien, <b>Schi ffamt</b>sgasse",  
      "value": "Wien, Schi ffamtsgasse",  
      "x": "",  
      "y": "",  
      "geotype": "2",  
      "id": "S904240"  
    },  
    {  
      "label": "Wien, <b>Schi ffamt</b>sufer",  
      "value": "Wien, Schi ffamt sufer",  
      "x": "",  
      "y": "",  
      "geotype": "2",  
      "id": "S918279"  
    }  
  ]  
)
```

4.2.2 Adresssuch-Service Request-Parameter

Parameter	Beschreibung	Req/Opt
term	Suchbegriff	Erforderlich
token	Token des Tokenservice	Erforderlich
projection	Projektion, derzeit: epsg_4258 (GEO_ETRS89) epsg_4312 (GEO_MGI) epsg_31255 (GK_MGI_M31)	Erforderlich, falls räumliche Parameter übergeben werden
point	Punkt, nach dem sortiert werden soll	Optional
extent	Extent, auf den die Suchergebnisse eingeschränkt werden sollen	Optional
callback	Callback Funktion, Wert muss mit „jsonp“ beginnen und hat danach nur Zahlen Bsp. „jsonp12345“ Dies wird für die Verwendung von jQuery UI Autocomplete benötigt	Optional

4.2.3 Adresssuch-Service Response-Parameter

Rückgabewert	Beschreibung
label	String, der mit dem übergebenen Suchbegriff beginnt
value	Normalisierter String (sollte dem Benutzer zur Auswahl angezeigt werden)
x	x-Wert in der angeforderten Projektion
y	y-Wert in der angeforderten Projektion
geotype	Typ des Suchergebnisses Gemeinde = 1 Straße = 2 Adresse =3
id	Eindeutiger Gemeinde-, Straßen- oder Adresscode je nach geotype

4.2.4 Sortierung und Limits

Rückgabewerte von Gemeinden und Straßen werden grundsätzlich alphabetisch nach „value“ sortiert.

Falls bei einer Suche nach Straße oder Gemeinde ein Koordinatenpunkt übergeben wird, erfolgt die Sortierung an Hand des Abstands von diesem Punkt.

Falls bei einer Suche nach Straße oder Gemeinde ein Extent übergeben wird, ist die Antwort generell unsortiert. In diesem Fall muss der Client eine Sortierung durchführen.

Adressen werden nach aufsteigender Hausnummer sortiert.

Es werden maximal 100 Hits geliefert.

4.2.5 Verarbeitungslogik

Die normierten Angaben einer Gemeinde, einer Straße oder einer Adresse sind folgende:

Typ	Darstellung	Beispiele
Gemeinde bzw. Ort	<Gemeindenamen> <Gemeindenamen> (<Ortsname>)	„Wien“ „Klosterneuburg“ „Klosterneuburg (Kierling)“
Straße	<Gemeindenamen>, <Straßennamen> <Gemeindenamen> (<Ortsname>), <Straßennamen>	„Wien, Burggasse“ „Klosterneuburg (Kierling), Feldgasse“
Adresse	<Gemeindenamen>, <Straßennamen> <Hausnummer> <Gemeindenamen> (<Ortsname>), <Straßennamen> <Hausnummer>	„Wien, Burggasse 120“ „Klosterneuburg (Kierling), Feldgasse 1“

Ein Ort wird grundsätzlich in Klammer angeführt, außer

- Wenn der Gemeindenamen und der Ortsnamen übereinstimmen
- Bei einigen großen Städten (z.B. werden in Wien nicht die Bezirke=Orte angezeigt)

Als Sonderfall wird auch auf Adressebene – wenn also bei der Suche nur noch die Adressen einer Straße angezeigt werden – noch die Straße selbst zur Auswahl geschickt. Diese soll es Anwendungen ermöglichen, auch die Adressen einer ganzen Straße effizient zu selektieren.

Bsp.:

„Klosterneuburg (Kierling), Feldgasse (Straße)“

Neben den normierten Bezeichnungen werden auf Gemeinde- und Straßenebene auch alternative Schreibweisen gefunden:

Alternative	Beispiele
Ortsangaben	Orte finden die zugehörigen Gemeinde/Ort Kombination Bsp. „Kierling“ findet „Klosterneuburg (Kierling)“
Straßenangaben	Straßenangaben finden Orte mit entsprechenden Straßen „naglergasse“ findet die Naglergassen in Wien, Graz und Zistersdorf Das geht bis auf die Hausnummernliste hinunter, „naglergasse 30“ findet nur mehr die Naglergasse 30 in Graz.
Groß- und Kleinschreibung	Alls Suchbegriffe werden case-insensitive behandelt
„ss“ statt „ß“	„strassb“ findet auch „Straßburg“ „wien, landstrasser“ findet auch „Wien, Landstraßer Hauptstraße“
Weglassen bzw. Ersetzen von Punkten durch Leerzeichen	„dr emm“, „dr. emm“ finden „Stockerau, Dr. Emmerich Czermak-Straße“ ACHTUNG: trennende Leerzeichen werden auf eines reduziert. „dr emm“ (zwei Leerzeichen) liefert kein Ergebnis
Weglassen bzw. Ersetzen von Bindestrichen durch Leerzeichen	„Dr. Emmerich Czermak Straße“ findet „Stockerau, Dr. Emmerich Czermak-Straße“
Unvollständige Bezeichnung (Wortteile)	„Czermak“ oder „Czerm“ finden „Stockerau, Dr. Emmerich Czermak-Straße“
Straßen werden auch ohne Gemeinde-Angabe gefunden	„Dr. Emmerich Czermak-Straße“ findet „Stockerau, Dr. Emmerich Czermak-Straße“

Durchsucht werden die Daten in folgender Weise:

- Suche nach Gemeinden/Orten
- Wortteilen von Gemeinden/Orten
- Suche nach Straßen
- Wortteile von Straßen

Das Ergebnis wird in folgender Reihenfolge sortiert

- Gemeinde/Orte
- Wortteile von Gemeinden/Orten
- Straßen
- Wortteile von Straßen

4.3 Fehlermeldungen

Die folgenden Fehlermeldungen können bei falscher Handhabung und Verwendung des Adresssuch-Service auftreten:

Bei falsch definierter Projektion:

```
[{"err": "projection invalid"}]
```

Wenn ein falscher, abgelaufener oder kein Token mitgeschickt wird:

```
[{"err": "invalid token"}]
```

Falls der zu suchende String (term) fehlt:

```
[{"err": "no term provided"}]
```

Bei einer inkorrekten Definition des Extent:

```
[{"err": "extent not formatted correctly (should be [[x1,y1], [x2,y2]])"}]
```

Wenn der Ausgangspunkt für die Suche falsch definiert wurde:

```
[{"err": "point not formatted correctly (should be (x,y))"}]
```

Falls sowohl ein Extent als auch ein Ausgangspunkt im Request geschickt wurde:

```
[{"err": "cannot base geospatial search both in extent and point"}]
```

Wenn ein Extent oder Ausgangspunkt, aber keine Projektion geschickt wurde:

```
[{"err": "no projection provided for geospatial search"}]
```

Alle anderen Fehler, die keine genaue Definition erhalten, retournieren einen allgemeinen Systemfehler (z.B. wenn der Request nicht korrekt aufgebaut ist):

```
[{"err": "system error"}]
```

4.4 URL für den aktuellen Status des Adresssuch-Service

Um den aktuellen Status des Adresssuch-Service abzurufen, wurde eine Sub-Domain eingerichtet. Diese Seite liefert Informationen über den aktuellen Status oder über zukünftige Ereignisse (z.B.: Wartungsfenster) des Adress-Suchservices. Diese URL kann in einen Client eingebunden werden, somit weiß der Kunde sofort über den Status des Adresssuch-Service Bescheid.

Produktivumgebung:

Die URL für den Status des Adress-Suchservices auf der Produktivumgebung lautet:

<http://www.bev.gv.at/adresssuchservice/status>

4.5 Unterschiede zwischen der TU und der PU

TokenService-URL

Während der Testphase ist die URL für den Aufruf des Tokenservices:

<https://sti.bev.gv.at:8443/adr-servicetest/TokenService>.

Für die produktive Verwendung muss folgende URL verwendet werden:

<https://sti.bev.gv.at:8443/adr-service/TokenService>.

Adresssuch-Service-URL

Während der Testphase ist die URL für den Aufruf des Adresssuch-Service:

<https://sd-servicetest.bev.gv.at/adr-service/servlet/SuggestJSON>.

Für die produktive Verwendung muss folgende URL verwendet werden:

<https://sd-service.bev.gv.at/adr-service/servlet/SuggestJSON>.

Des Weiteren wurde der Port 443 freigeschaltet:

TU: <https://sti.bev.gv.at:443/adr-servicetest/TokenService>

PU: <https://sti.bev.gv.at:443/adr-service/TokenService>

5 Anhang

5.1 Offene Punkte

Thema	Ergebnis/Beschreibung

5.2 Begriffsbestimmungen und Abkürzungen

Begriff / Abkürzung	Definition, Beschreibung
EGA	eGeodata Austria
PVP	Portalverbundprotokoll (reference.e-government.gv.at)
Token	Deutsch: Kennzeichen, Merkmal. Eine ID, die zeitlich begrenzte, sichere Schlüssel für den Zugang zu Datenverarbeitungs-Systemen generiert.
JSON	Javascript Object Notation
http	Hypertext Transfer Protocol, vgl. http://rfc.net/rfc2616.html .
XML	Extensible Markup Language
TU / PU	Testumgebung / Produktivumgebung
URL	Uniform Resource Locator - URLs identifizieren und lokalisieren eine Ressource über das verwendete Netzwerkprotokoll
MGI	Militärgeographisches Institut – geodätisches Datum der österreichischen Landesvermessung
ETRS89	European Terrestrial Reference System 1989
UTM	Universal Transverse Mercator