



Dokumentation

Geokodierungsdienste für Adressen und Geonamen Schnittstellenbeschreibung



Version 1.8 (04.01.2021)

Inhaltsverzeichnis

Tabellenverzeichnis	5
Abkürzungsverzeichnis	6
1 Übersicht über den Datenbestand	7
1.1 Änderungshistorie des Dienstes	9
1.1.1 Version 1.8	9
1.1.2 Version 1.7	9
1.1.3 Version 1.6	9
1.1.4 Version 1.5	10
1.1.5 Version 1.4	10
1.1.6 Version 1.3	10
2 Datenmodell	12
2.1 Ortsangabe	12
2.1.1 Attribute	12
2.1.2 Klassifizierung	15
2.2 Vorschlag	17
2.2.1 Attribute	18
3 Rückgabeformate	19
3.1 GeoJSON	19
3.1.1 Content-Type	19
3.1.2 Mapping	19
3.1.3 Beispiel	20
3.2 GML	21
3.2.1 Versionen	21
3.2.2 Content-Type	21
3.2.3 Mapping	21
3.2.4 Beispiel	22
3.3 GeoRSS	23
3.3.1 Content-Type	23
3.3.2 Mapping	23
3.3.3 Beispiel	24
3.4 JSON Vorschlagsliste	25
3.4.1 Content-Type	25
3.4.2 Beispiel	25
3.5 OpenSearch Suggestion Extension	26
3.5.1 Content-Type	26
3.5.2 Format	26
3.5.3 Beispiel	26
4 OpenSearch GeoTemporal Service	27

Geokodierungsdienst für Adressen und Geonamen

4.1	Conformance Class	27
4.2	OpenSearch Description	27
4.3	Suche	28
4.3.1	Query-Syntax	28
4.4	Online Dokumentation	30
4.4.1	Basis URL	30
4.4.2	Vorschlagssuche	30
4.4.3	Ortssuche	31
4.4.4	Strukturierte Ortssuche / Geocode	34
4.4.5	Direktzugriff auf Objekte / GetRecord	37
4.5	OpenSearch Response	38
4.6	Error handling	38
4.7	Anwendungsbeispiele	39
4.7.1	OpenSearch Beschreibung aufrufen	39
4.7.2	Vorschlagssuche	39
4.7.3	Volltextsuche	39
4.7.4	Strukturierte Suche	39
4.7.5	Reverse Geokodierung	40
5	Web Feature Service	41
5.1	GetCapabilities	41
5.1.1	Beschreibung	41
5.1.2	Ergebnis	41
5.1.3	Beispiel	42
5.2	DescribeFeatureType	42
5.2.1	Beschreibung	42
5.2.2	Ergebnis	42
5.2.3	Beispiel	42
5.3	GetFeature	43
5.3.1	Beschreibung	43
5.3.2	Ergebnis	43
5.3.3	Beispiel	44
5.4	Anwendungsbeispiele	46
5.4.1	OpenSearch Beschreibung aufrufen	46
5.4.2	Volltextsuche	46
5.4.3	Strukturierte Suche	47
5.4.4	Strukturierte Suche mit Ausgabe der Einzelgüte	48
5.4.5	Reverse Geokodierung	48
6	Sonstige Hinweise	51
6.1	Suchverhalten	51
6.1.1	Unstrukturierte Volltextsuche	51
6.1.2	Strukturierte Suche	51

Geokodierungsdienst für Adressen und Geonamen

6.1.3	Bewertung	51
6.1.4	Auswertung der Ergebnisse	52
6.1.5	Fehlertoleranz	52
6.2	Räumliche Einschränkung der Anfragen	53
6.3	Angabe des Koordinatenreferenzsystems	53
6.4	Hinweise zu Koordinaten	54
7	Datenschutz	55
8	Anlagen	56
8.1	JSON Schema Ortsangabe	56
8.2	XML Schema Ortsangabe	59
9	Kontaktdaten	64

Tabellenverzeichnis

Tabelle 1 Attribute für Datentyp Ortsangabe.....	12
Tabelle 2 Attribute für Datentyp Vorschlag	18
Tabelle 3 Ausgabeformate für Datentypen	19
Tabelle 4 Abbildung Ortsangabe auf GeoJSON Feature	19
Tabelle 5 Abbildung Ortsangabe auf GML Feature.....	21
Tabelle 6 Abbildung Attribute von Ortsangabe auf Atom	23
Tabelle 7 Übersicht NOT Operatoren	29
Tabelle 8 Parameter für Vorschlagssuche	30
Tabelle 9 Parameter für Ortssuche.....	32
Tabelle 10 Parameter für die strukturierte Suche.....	34
Tabelle 11 Parameter für GetRecord.....	37
Tabelle 12 HTTP-Status Codes im Fehlerfall.....	38

Abkürzungsverzeichnis

EPSG	European Petroleum Survey Group
GA	Georeferenzierte Adressen
GN250	Geographische Namen Deutschlands
GML	OpenGIS Geography Markup Language Encoding Standard
HK-DE	Amtliche Hauskoordinaten Deutschlands
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
KVP	Key Value Pair
OGC	Open Geospatial Consortium
OWS	OGC Web Service, oder Open Web Service
WFS	Web Feature Service
WKT	Well Known Text
XML	Extensible Markup Language

1 Übersicht über den Datenbestand

Der Geokodierungsdienst für Adressen und Geonamen ist ein Gemeinschaftsprojekt von Bund und Ländern. Das Konzept wurde gemeinsam entwickelt. Die Datengrundlage wird von den Bundesländern bereitgestellt. Das Bundesamt für Kartographie und Geodäsie (BKG) bereitet die Daten auf und betreibt den am Dienstleistungszentrum des BKG entwickelten Webdienst. Das vorliegende Dokument beschreibt das Datenmodell und die Schnittstellen des Geokodierungsdienstes und gibt Hinweise zu dessen Anwendung.

Ein *Geokodierungsdienst* ist ein Webdienst, der attributiv beschriebenen Objekten (geographischen Identifikatoren) eine räumliche Lagebeschreibung in Form einer Koordinate zuweist und die Objekte damit georäumlichen Analysen und Bearbeitungen zugänglich macht. Er erweitert die Funktionalität von Anwendungen, in die ein Dienst eingebunden wird und die den jeweiligen fachlichen Kontext abbilden sowie die passende Nutzeroberfläche bereitstellen.

Der vorliegende Dienst vereint in sich die folgenden funktionalen und lizenzrechtlichen Dienstetypen:

- *Ortssuche* (mit Vorschlags- und Ergebnissuche)

Ein Ortssuchdienst ist ein Geodatendienst, der den über geographische Identifikatoren gesuchten Objekten in Geoanwendungen eine Koordinate (Geokodierung) zuweist. Der Ortssuchdienst unterstützt die Suche durch dynamisch generierte Vorschlagslisten sowie durch eine nach der Trefferqualität sortierte Ergebnisliste. Es findet keine persistente Speicherung der Koordinate statt.

- *Geokodierung*

Ein Geokodierungsdienst ist ein Geodatendienst, der einzelnen oder massenhaften Objekten mit geographischen Identifikatoren eine Koordinate (Geokodierung) zu deren persistenten Speicherung zuweist.

- *Reverse Geokodierung*

Ein reverser Geokodierungsdienst ist ein Geodatendienst, der ausgehend von einer Koordinate (Punkt) oder einem umfassenden Gebiet (Rechteck, Polygon) alle geographischen Identifikatoren mit einem bestimmten räumlichen Bezug (vollständig enthalten, angeschnitten) ermittelt.

Datengrundlage des Dienstes sind die Datenbestände *Amtliche Hauskoordinaten Deutschlands (HK-DE)* der deutschen Landesvermessung und *Geographische Namen Deutschlands (GN250)* des Bundesamtes für Kartographie und Geodäsie. Der Dienst unterstützt eine unstrukturierte Volltextsuche ebenso wie eine strukturierte Suche von Adressen und Geonamen über deren Attribute und gibt die gefundenen Geoobjekte einschließlich ihrer Lagekoordinate zurück. Dabei arbeitet der Dienst fehlertolerant und bewertet die gefundenen Objekte mit einem *score*, der die Qualität ihrer Übereinstimmung mit der Suchanfrage ausdrückt.

Die schnelle Vorschlagssuche liefert Suchergebnisse innerhalb von Millisekunden und kann so zur Vervollständigung interaktiver Eingaben eines Suchbegriffs in einer Anwendung eingesetzt werden. Optionale attributive Filter können in allen funktionalen Anwendungsfällen die Suchergebnisse einschränken und so beispielsweise auf bestimmte Verwaltungsgebiete begrenzen. Der zurück gegebene *score* ist im Fall einer strukturierten Adresssuche interpretierbar hinsichtlich der Wahrscheinlichkeit, ob der Treffer das gesuchte Objekt wirklich darstellt oder nicht. Diese Eigenschaft ist für die Geokodierung qualitativ schlechter Ausgangsdaten von besonderer Bedeutung.

Geokodierungsdienst für Adressen und Geonamen

Der Geokodierungsdienst unterstützt die folgenden durch das Open Geospatial Consortium (OGC) international standardisierten Schnittstellen:

1. **OpenSearch GeoTemporal Service (OSGTS)**

Eine zur Spezifikation *OpenSearch Geo and Time Extensions* konforme Schnittstelle, die insbesondere auf die Integration in Webanwendungen optimiert ist.

2. **Web Feature Service (WFS)**

Eine zum *Web Feature Service (WFS)* konforme Schnittstelle für komplexere Anwendungsfälle.

Aus den oben angesprochenen lizenzrechtlichen Gründen werden für beide Schnittstellen Dienste mit und ohne Erlaubnis zur persistenten Speicherung der Ergebnisse angeboten:

Mit Erlaubnis zur persistenten Speicherung:

- https://sg.geodatenzentrum.de/gdz_geokodierung (OSGTS)
- https://sg.geodatenzentrum.de/wfs_geokodierung (WFS)

Ohne Erlaubnis zur persistenten Speicherung:

- https://sg.geodatenzentrum.de/gdz_ortssuche (OSGTS)
- https://sg.geodatenzentrum.de/wfs_ortssuche (WFS)

Für Bundesbehörden werden weitere Dienste zur internen Anwendung bereitgestellt, die auf dem Adressdatenbestand *Georeferenzierte Adressen (GA)* des BKG und den GN250 basieren. Der Datenbestand GA ist ein um Daten aus der Wirtschaft erweiterter Datenbestand HK-DE. Diese Dienste sind unter folgenden URLs aufrufbar:

- https://sg.geodatenzentrum.de/gdz_geokodierung_bund (OSGTS)
- https://sg.geodatenzentrum.de/wfs_geokodierung_bund (WFS)

Die im Dokument nachfolgenden URLs sind exemplarisch nur für eine Variante der Dienste angegeben. Verwendet wird die URL für den Dienst mit persistenter Speicherung auf Grundlage des Datenbestandes HK-DE. Die URLs sind bei abweichender Nutzung oder Datengrundlage entsprechend anzupassen.

1.1 Änderungshistorie des Dienstes

1.1.1 Version 1.8

Mit Version 1.8 wird für die OpenSearch GeoTemporal Service Schnittstelle eine Online Dokumentation zur Verfügung gestellt. Neben der Dokumentation der angebotenen Endpunkte und deren Parameter kann über die Online-Dokumentation die OSGTS Schnittstelle auch praktisch erprobt werden.

1.1.2 Version 1.7

Die Geokodierungsdienste wurden um folgende Funktionen erweitert:

- Das Qualitätskennzeichen (QKZ) wird als neues Attribut aufgenommen. Das Attribut wird bei der strukturierten Suche bestimmt. Es beschreibt die Qualität der Geokodierung auf Grundlage der Attributwerte score und typ.
- Das Attribut treffer wird für alle Typen ermittelt insofern anhand der Suche der angefragte Typ abgeleitet werden kann. Bisher wurde treffer nur für den Typ Haus unterstützt.
- Der Web Feature Service unterstützt beim GetFeature Request den Request-Parameter resulttype. Durch die Angabe des Wertes hits kann der Response auf die Anzahl der Ergebnisse sowie den Zeitpunkt der Ergebnisstellung beschränkt werden.
- Die Ortssuche des OpenSearch Service unterstützt den Request-Parameter properties. Hiermit können explizit die Attribute angegeben werden, die in der Antwort enthalten sein sollen.
- Der OpenSearch Service stellt eine neue Schnittstelle für die strukturierte Suche bereit, die es erlaubt die Adressattribute direkt per Request-Parameter anzugeben. Dies erlaubt eine leichtere Übergabe der Suchparameter und vermeidet fehlerhafte Suchanfragen.

1.1.3 Version 1.6

Die Geokodierungsdienste wurden um folgende Funktionen erweitert:

- Das Datenmodell des Dienstes wurde um das Attribut schluessel erweitert. Dieses Attribut enthält eine Kombination des Regionalschlüssels (12-stellig) mit dem Ortsteilschlüssel (vierstellig) und Straßenschlüssel (fünfstellig).
 - Für Straßen und Hausadressen besitzt schluessel 21 Stellen und setzt sich aus Regional-, Ortsteil und Straßenschlüssel zusammen.
 - Für Objekte vom Typ Ort besitzt schluessel 16 Stellen und setzt sich aus dem Regional- und Ortsteilschlüssel zusammen.
- Das Attribut rs weist ab sofort nur noch den 12-stelligen Regionalschlüssel aus.
- Identische Objekte vom Typ Ort oder Straße, die sich nur im Regionalschlüssel unterscheiden werden ab sofort zusammengefasst.
 - Für diese Objekte kann schluessel mehrere Werte enthalten. Die Angabe der Liste erfolgt als kommaseparierete Zeichenkette.
(Bsp.: "schluessel": "0600000000000000, 0700000000000000")
 - Die Reihenfolge der Liste wird durch die Anzahl der zugehörigen Hausadressen bestimmt.

- Die Werte für ags, rs, bundesland, regbezirk, kreis, verwgem und gemeinde enthalten weiterhin immer nur einen Wert. Dieser bezieht sich immer auf den ersten Wert von schluessel.
- Die Attribute ags und rs enthalten weiterhin nur einen Wert. Der Wert entspricht dem ersten Wert des Attributes schluessel (der mit den meisten Adressen).
- Es ist ab sofort möglich bei strukturierten Suchen zusätzlich zur Gesamtgüte auch die Einzelgütewerte der Attribute ausgeben zu lassen. Das Datenmodell wurde dazu um die Attribute score_<XYZ> erweitert (z.B. score_ort). Die Ausgabe dieser Attribute erfolgt nur, wenn dies explizit in der Anfrage angefordert wurde. In der OpenSearch-Schnittstelle erfolgt dies durch den Parameter allScores. In der WFS-Schnittstelle kann dies über die PropertyName-Auswahl erfolgen (s. 5.4.4).
- Es ist ab sofort möglich Ergebnisse nach Gesamtgüte zu filtern. In der OpenSearch Schnittstelle erfolgt dies mit den Request-Parametern minScore und maxScore. In der WFS Schnittstelle kann dies mit dem PropertyIsLessThan bzw. dem PropertyIsGreaterThan Operator auf das Attribut score erfolgen.

1.1.4 Version 1.5

Es erfolgten Optimierungen am Suchverhalten der Vorschlagssuche. Ziel war die Verbesserung der Relevanz der vorgeschlagenen Suchbegriffe.

Weiterhin wurde das Datenmodell der Geonamen um Information zum Bundesland erweitert. Damit kann künftig die Suche nach Geonamen auch textuell auf Bundeslandebene eingeschränkt werden.

1.1.5 Version 1.4

Mit Version 1.4 sind ab sofort Postleitzahlgebiete im Referenzdatenbestand enthalten. Der Dienst erlaubt damit die ausschließliche Suche nach Postleitzahlen. (s.2.1.2.1)

Die Vorschlagssuche der OpenSearch-Schnittstelle unterstützt nun die Konfiguration des Highlighting. Über die Parameter *hl_pre* und *hl_post* können Prefix und Postfix zum Hervorheben der übereinstimmenden Wörter individuell angepasst werden.

1.1.6 Version 1.3

Mit Version 1.3 erfolgte eine Optimierung der Such- und Bewertungsalgorithmen, damit bessere Suchergebnisse erzielt werden können. Die Schnittstelle ist von diesen Änderungen nicht betroffen und kann in gewohnter Weise weitergenutzt werden.

Folgende Verbesserungen wurden im allgemeinen Suchverhalten vorgenommen:

- Bei reinen Ortssuchen werden Ergebnisse ohne Ortsteil nun erwartungsgerecht bevorzugt. Suchen nach großen Orten (z.B. München) lieferten bisher oft deren Ortsteile vor dem eigentlichen Ort als Ergebnis zurück.
- Die OpenSearch- und WFS-Schnittstelle wurden in ihrer Funktionalität weiter einander angeglichen.

Folgende Verbesserungen wurden für die strukturierte Suche vorgenommen:

- Die Suche nach Straße und Haus im kombinierten Attribut *strasse_haus* wurde optimiert und mit dem Verhalten der getrennten Suche in den Attributen *strasse* und *haus* abgeglichen.
- Das Suchverhalten für Hausnummern wurde überarbeitet. Wird eine Hausnummer nicht gefunden, wird im Suchergebnis die nächstpassende Hausnummer zurückgegeben. Erst wenn die Straße nicht stimmt, wird der übergeordnete Ort zurückgegeben.
- Der Bewertungsalgorithmus für Hausnummern und Postleitzahlen wurde verbessert.

Geokodierungsdienst für Adressen und Geonamen

- Die relative Bewertung (s. 6.1.3.2) hat eine verbesserte Klassifizierung erhalten.

2 Datenmodell

Das nachfolgend beschriebene Datenmodell basiert auf einer flachen Struktur (einfache Objekttypen mit tabellarischen Attributen und ohne relationale Objektverknüpfungen), auf der besonders leistungsfähige Suchalgorithmen implementiert werden können.

Der Dienst unterscheidet zwei Datentypen:

- *Ortsangabe*: zur detaillierten Suche von Objekten aus den verschiedenen Datenbeständen
- *Vorschlag*: zur Generierung von Vorschlagslisten.

Alle Objekte aus den Datenbeständen *HK-DE* und *GN250* können über diese beiden Datentypen abgebildet werden.

2.1 Ortsangabe

2.1.1 Attribute

Der Datentyp *Ortsangabe* besitzt die in Tabelle 1 angegebenen obligatorischen (P), optionalen (O) und virtuellen (V) Attribute. Virtuelle Attribute können nur zur Suche verwendet werden und sind nicht im Ergebnis enthalten.

Tabelle 1 Attribute für Datentyp Ortsangabe

Attribut	P/O/V ¹	Datentyp	Beschreibung
id	P	String	Die eindeutige ID (Zeichenfolge) des Objekts zur Identifizierung im Dienst.
text	P	String	Volltext-Repräsentation des Objekts.
typ	P	String	Klassifizierung des Objekts [Haus Straße Ort PLZ Geoname]
score	O	Double	Qualitative Bewertung des Objektes - Gesamtgüte. Dieses dynamische Attribut gibt Auskunft inwieweit die Suchanfrage mit dem gefundenen Objekt übereinstimmt. Dabei ist zwischen relativer und absoluter Bewertung zu unterscheiden.
score_*	O	Double	Für Strukturierte Suchen mit relativer Bewertung kann bei Bedarf die qualitative Bewertung eines Attributes – Einzelgüte ausgegeben werden. Der Name des Attributes für die Einzelgüte setzt sich aus „score_“ und dem Attributnamen zusammen. Diese dynamischen Attribute geben Auskunft inwieweit ein Attribut der Suchanfrage mit dem Attribut des gefundenen Objektes übereinstimmt.
treffer	O	String	Enthält für Strukturierte Suchen mit relativer Bewertung eine einfache Bewertung des Ergebnisses. Nachfolgende Werte sind möglich:

¹ Pflichtattribut (P) / optionales Attribut (O) / virtuelles Attribut (V)

Geokodierungsdienst für Adressen und Geonamen

			Wird nur bei Objekten mit type=[Haus Strasse Ort] verwendet.
ort	O	String	Ortsname der Adresse. Wird nur bei Objekten mit type=[Haus Strasse Ort] verwendet.
ortsteil	O	String	Ortsteilname der Adresse. Wird nur bei Objekten mit type=[Haus Strasse Ort] verwendet
strasse	O	String	Straßenname der Adresse. Wird nur bei Objekten mit type=[Haus Strasse] verwendet
haus	O	String	Hausnummer mit Hausnummernzusatz der Adresse. Wird nur bei Objekten mit type=Haus verwendet
strasse_haus	V	String	Suchattribut wenn Straßenname und Hausnummer als einzelner String gegeben sind.
qualitaet	O	Character	Qualitätsangabe zur Hauskoordinate: <ul style="list-style-type: none"> • A = Koordinate liegt sicher innerhalb der Gebäudefläche und das Gebäude ist sicher in der Örtlichkeit vorhanden. • B = Koordinate liegt sicher innerhalb der Flurstücksfläche und das Gebäude ist sicher in der Örtlichkeit vorhanden. • R = Koordinate liegt sicher innerhalb der Flurstücksfläche. Das Gebäude ist nicht sicher in der Örtlichkeit vorhanden.
name	O	String	Name des Objekts. Wird nur für Objekte mit type= Geoname verwendet.
kategorie	O	String	Thematische Kategorie (Landschaft, Schule, ...) des Objekts. Wird nur bei Objekten mit type= Geoname verwendet.
qkz	O	String	Enthält für Strukturierte Suchen ein Qualitätskennzeichen, das die geokodierte Koordinate genauer klassifiziert. <ul style="list-style-type: none"> • 1 = Sichere gebäudescharfe Geokodierung (Gesamtgüte > 0.95 und Typ = Haus) • = Gute gebäudescharfe Geokodierung (0.948 <= Gesamtgüte <= 0.95 und Typ = Haus) • = Nachbarschaftsscharfe Geokodierung (0.946 <= Gesamtgüte <= 0.947 und Typ = Haus) • = Straßengenaue Geokodierung (Gesamtgüte > 0.95 und Typ = Strasse) • = Ortsgenaue Geokodierung (Gesamtgüte > 0.95 und Typ = Ort) • = PLZ-genaue Geokodierung (Gesamtgüte > 0.95 und Typ = PLZ)

			<ul style="list-style-type: none">• 9 = Ohne Geokodierung einschließlich aller Fälle, in denen die Bedingungen unter Nummer 1 bis 6 nicht erfüllt werden.
--	--	--	---

2.1.2 Klassifizierung

Für Instanzen vom Datentyp *Ortsangabe* können Klassifizierungen vorgenommen werden. Diese erfolgen durch das Attribut *typ*. Die Klassifizierung kann zur Ermittlung des zugrunde liegenden Datenbestandes herangezogen werden und hat Auswirkungen auf die Verfügbarkeit optionaler Attribute.

Für *typ* gibt es folgende Klassifizierungen:

- *PLZ* (Quelle: *HK-DE*)
- *Ort* (Quelle: *HK-DE*)
- *Strasse* (Quelle: *HK-DE*)
- *Haus* (Quelle: *HK-DE*)
- *Geoname* (Quelle: *GN250*)

Im Folgenden sind die Klassifizierungen und damit verbundenen Eigenschaften näher beschrieben.

2.1.2.1 *PLZ*

Postleitzahlen stammen aus dem Datenbestand *HK-DE* und werden im Attribut *typ* durch den Wert *PLZ* klassifiziert. Neben den Pflichtattributen sind die folgenden Attribute gegeben:

- *plz*
- *bundesland*

Das Attribut *text* enthält den Wert aus *plz*.

2.1.2.2 *Ort*

Orte stammen aus dem Datenbestand *HK-DE* und werden im Attribut *typ* durch den Wert *Ort* klassifiziert. Neben den Pflichtattributen sind die folgenden Attribute gegeben:

- *plz*
- *ort*
- *ortsteil*
- *ags*
- *rs*
- *schluessel*
- *bundesland*
- *regbezirk*
- *kreis*
- *verwgem*
- *gemeinde*

Handelt es sich um einen Ortsteil, ist ebenfalls das Attribut *ortsteil* gefüllt. Andernfalls ist es ein leerer String.

Das Attribut *text* ist wie folgt gebildet: <plz> <ort> [- <ortsteil>]²

2.1.2.3 Strasse

Straßen stammen aus dem Datenbestand *HK-DE* und werden im Attribut *typ* durch den Wert *Strasse* klassifiziert. Neben den Pflichtattributen sind die folgenden Attribute gegeben:

- plz
- ort
- ortsteil
- strasse
- ags
- rs
- schluessel
- bundesland
- regbezirk
- kreis
- verwgem
- gemeinde

Die Geometrie *geometry* beschreibt den Mittelpunkt der Straße. Dieser wird durch die Koordinate der mittleren Hausnummer definiert.

Das Attribut *text* ist wie folgt gebildet: <strasse>, <plz> <ort> [- <ortsteil>]

2.1.2.4 Haus

Ortsangaben vom Typ *Haus* repräsentieren eine komplette Adressangabe (Gebäude oder ähnliches). Diese Objekte stammen aus dem Datenbestand *HK-DE*. Neben den Pflichtattributen sind die folgenden Attribute gegeben:

- plz
- ort
- ortsteil
- strasse
- haus
- qualitaet
- ags
- rs
- schluessel
- bundesland
- regbezirk
- kreis
- verwgem
- gemeinde

Das Attribut *text* ist wie folgt gebildet: <strasse> <haus>, <plz> <ort> [- <ortsteil>]

2.1.2.5 Geoname

Objekte mit der Klassifizierung *typ=Geoname* stammen aus dem Datenbestand *GN250*. In diesem Fall sind neben den Pflichtattributen die folgenden Attribute gegeben:

² Spitze Klammern: Werte aus gleichnamigen Attributen; Eckige Klammern: optionaler Teil

Geokodierungsdienst für Adressen und Geonamen

- name
- kategorie
- bundesland
- ags (Nur Anteil für Bundesland, fehlende Stellen mit 0 aufgefüllt)
- rs (Nur Anteil für Bundesland, fehlende Stellen mit 0 aufgefüllt)
- schluessel (Nur Anteil für Bundesland, fehlende Stellen mit 0 aufgefüllt)

Das Attribut *kategorie* gibt Auskunft über die thematische Kategorie und kann einen der folgenden Werte enthalten:

- Besonderer Hoehepunkt
- Bundesland
- Flugverkehr
- Gemeinde
- Gewaesser
- Gewaessermerkmal
- Heide
- Insel
- Kreis/Region
- Landschaft
- Meer
- Moor
- Nationalstaat
- Naturpark
- Regierungsbezirk
- Sport/Freizeit/Erholung
- Sumpf
- Verwaltungsgemeinschaft
- Weg
- Wald

Sollte das Objekt sich über mehrere Bundesländer erstrecken, wird dies auch in den Attributen *bundesland* und *schluessel* abgebildet. Die Zeichenkette setzt sich dabei aus einer kommaseparierten Liste mit den entsprechenden Werten zusammen.

Das Attribut *text* ist wie folgt gebildet: *<name>* (*<kategorie>*)

2.2 Vorschlag

Der Datentyp *Vorschlag* wird zur Generierung von Vorschlagslisten verwendet. Dieser Datentyp steht nur für die Vorschlagssuche im OpenSearch GeoTemporal Service zur Verfügung.

2.2.1 Attribute

Tabelle 2 Attribute für Datentyp Vorschlag

Attribut	P/O ³	Datentyp	Beschreibung
text	P	String	Text für Suchvorschlag.
score	O	Double	Qualitative Bewertung des Objektes. Dieses dynamische Attribut gibt Auskunft inwieweit die Suchanfrage mit dem gefundenen Objekt übereinstimmt. Bei einem Vorschlag erfolgt die Bewertung immer absolut.
highlighted	O	String	Text für Anzeige in Vorschlagsliste. Enthält Markup mit Hervorhebung der übereinstimmenden Teile.

³ Pflichtattribut (P) / optionales Attribut (O)

3 Rückgabeformate

Der Geokodierungsdienst unterstützt als Rückgabeformat für Ortsangaben GeoJSON, GML und Atom. Vorschläge können in einem einfachen JSON-Format und einem von der OpenSearch Suggestion Extension verwendeten JSON-Format ausgegeben werden. Die folgende Tabelle zeigt in welchen Formaten die einzelnen Datentypen ausgegeben werden können.

Tabelle 3 Ausgabeformate für Datentypen

Format	Ortsangabe	Suggestion
JSON	Ja (GeoJSON)	Ja (JSON-Vorschlagsliste)
GML	Ja	Nein
GeoRSS	Ja	Nein
OpenSearch Suggestion Extension	Nein	Ja

3.1 GeoJSON

Bei GeoJSON⁴ handelt es sich um einen IT-Standard, der auf der JavaScript Object Notation (JSON) basiert, um geographische Objekte zu kodieren. Details zur verwendeten Datenstruktur können dem Schema der Anlage 8.1 entnommen werden.

Alle gefundenen Datensätze werden innerhalb einer FeatureCollection zurückgegeben. Ein einzelner Datensatz wird als Feature abgebildet.

3.1.1 Content-Type

- application/json

3.1.2 Mapping

Der Datentyp Ortsangabe wird in GeoJSON als Feature ausgegeben. Die Attribute sind wie folgt abgebildet:

Tabelle 4 Abbildung Ortsangabe auf GeoJSON Feature

Ortsangabe	GeoJSON Feature
id	feature.id
text	feature.properties.text
typ	feature.properties.typ
score	feature.properties.score
score_*	feature.properties.score_*
treffer	feature.properties.treffer
geometry	feature.geometry
bbox	feature.properties.bbox
ags	feature.properties.ags
rs	feature.properties.rs

⁴ (GeoJSON kein Datum)

Geokodierungsdienst für Adressen und Geonamen

schluessel	feature.properties.schluessel
bundesland	feature.properties.bundesland
regbezirk	feature.properties.regbezirk
kreis	feature.properties.kreis
verwgem	feature.properties.verwgem
gemeinde	feature.properties.gemeinde
plz	feature.properties.plz
ort	feature.properties.ort
ortsteil	feature.properties.ortsteil
strasse	feature.properties.strasse
haus	feature.properties.haus
qualitaet	feature.properties.qualitaet
name	feature.properties.name
kategorie	feature.properties.kategorie
qkz	feature.properties.qkz

3.1.3 Beispiel

```
{
  "type": "FeatureCollection",
  "crs": {
    "type": "name",
    "properties": {"name": "EPSG:4326"}
  },
  "features": [
    {
      "type": "Feature",
      "id": "haus.8022.35523.510079.15032976",
      "geometry": {"type": "Point", "coordinates": [12.13188, 50.96714]},
      "properties": {
        "text": "Karl-Rothe-Straße 10-14, 04105 Leipzig",
        "typ": "Haus",
        "score": 1.0,
        "treffer": "",
        "bbox": {
          "type": "Polygon",
          "coordinates": [[
            [12.13188, 50.96714],
            [12.13188, 51.69659],
            [13.24085, 51.69659],
            [13.24085, 50.96714],
            [12.13188, 50.96714]
          ]]
        },
        "ags": "12345678",
        "rs": "123456789012",
        "schluessel": "1234567890120000",
        "bundesland": "Sachsen",
        "kreis": "...",
        "gemeinde": "...",
        "plz": "04105",
        "ort": "Leipzig",
        "ortsteil": null,
        "strasse": "Karl-Rothe-Straße",
      }
    }
  ]
}
```

```
    "haus": "10-14",
    "qualitaet": "A",
    "qkz": 1
  }
},
{
  "type": "Feature",
  "id": "gnde.DEBKGGND00001CFZ",
  "geometry": {"type": "Point", "coordinates": [12.13188,50.96714]},
  "properties": {
    "text": "Leipzig (Regierungsbezirk)",
    "typ": "GNDE",
    "score": 0.9,
    "treffer": "",
    "bbox": {
      "type": "Polygon",
      "coordinates": [[...]]
    },
    "name": "Leipzig",
    "kategorie": "Regierungsbezirk"
  }
}
]
}
```

3.2 GML

Der OpenGIS Geography Markup Language Encoding Standard (GML)⁵ dient zur Enkodierung von geographischen Objekten per XML. Details zur verwendeten Datenstruktur können dem Schema der Anlage 8.2 entnommen werden.

Alle gefundenen Datensätze werden innerhalb einer gml:FeatureCollection zurückgegeben. Ein einzelner Datensatz wird als Feature vom Typ gdz:Ortsangabe abgebildet.

3.2.1 Versionen

Vom Dienst werden die GML-Versionen 2.1.2 und 3.1.1 unterstützt.

3.2.2 Content-Type

- application/gml+xml;version=3.1
- application/gml+xml;version=2.1

3.2.3 Mapping

Der Datentyp Ortsangabe wird in GML als gdz:Ortsangabe ausgegeben (implementiert gml:AbstractFeatureType). Die Attribute sind wie folgt abgebildet:

Tabelle 5 Abbildung Ortsangabe auf GML Feature

Ortsangabe	GML Feature XPath
id	gdz:Ortsangabe@gml:id (GML 2.1: gdz:Ortsangabe@gml:fid)
text	gdz:Ortsangabe/gdz:text
typ	gdz:Ortsangabe/gdz:typ

⁵ (OGC, OpenGIS Geography Markup Language (GML) Encoding Standard 2007)

Geokodierungsdienst für Adressen und Geonamen

score	gdz:Ortsangabe/gdz:score
score_*	gdz:Ortsangabe/gdz:score_*
treffer	gdz:Ortsangabe/gdz:treffer
geometry	gdz:Ortsangabe/gdz:geometry
bbox	gdz:Ortsangabe/gdz:bbox
ags	gdz:Ortsangabe/gdz:ags
rs	gdz:Ortsangabe/gdz:rs
Schluessel	gdz:Ortsangabe/gdz:schluessel
bundesland	gdz:Ortsangabe/gdz:bundesland
regbezirk	gdz:Ortsangabe/gdz:regbezirk
kreis	gdz:Ortsangabe/gdz:kreis
verwgem	gdz:Ortsangabe/gdz:verwgem
gemeinde	gdz:Ortsangabe/gdz:gemeinde
plz	gdz:Ortsangabe/gdz:plz
ort	gdz:Ortsangabe/gdz:ort
ortsteil	gdz:Ortsangabe/gdz:ortsteil
strasse	gdz:Ortsangabe/gdz:strasse
haus	gdz:Ortsangabe/gdz:haus
qualitaet	gdz:Ortsangabe/gdz:qualitaet
name	gdz:Ortsangabe/gdz:name
kategorie	gdz:Ortsangabe/gdz:kategorie
qkz	gdz:Ortsangabe/gdz:qkz

3.2.4 Beispiel

```
<?xml version="1.0" encoding="utf-8"?>
<gml:FeatureCollection xmlns:gml="https://www.opengis.net/gml"
  xmlns:xlink="https://www.w3.org/1999/xlink"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:xml="https://www.w3.org/XML/1998/namespace"
  xmlns:gdz="https://www.geodatenzentrum.de/location">
  <gml:featureMember>
    <gdz:Ortsangabe gml:id="haus.8022.35523.510079.15032976">
      <gdz:text>Karl-Rothe-Straße 10-14, 04105 Leipzig</gdz:text>
      <gdz:typ>Haus</gdz:typ>
      <gdz:score>1.0</gdz:score>
      <gdz:treffer>T</gdz:treffer>
      <gdz:geometry>
        <gml:Point srsName="EPSG:4326">
          <gml:pos>12.953736304764146 50.67553957644192</gml:pos>
        </gml:Point>
      </gdz:geometry>
      <gdz:bbox>
        <gml:Polygon srsName="EPSG:4326">
          <gml:exterior>
            <gml:LinearRing>
              <gml:posList srsDimension="2">...</gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </gdz:bbox>
      <gdz:ags>12345678</gdz:ags>
    </gdz:Ortsangabe>
  </gml:featureMember>
</gml:FeatureCollection>
```

```

<gdz:rs>123456789012</gdz:rs>
<gdz:schluesssel>1234567890120000</gdz:schluesssel>
<gdz:bundesland>Sachsen</gdz:bundesland>
<gdz:kreis></gdz:kreis>
<gdz:gemeinde></gdz:gemeinde>
<gdz:plz>04105</gdz:plz>
<gdz:ort>Leipzig</gdz:ort>
<gdz:ortsteil/>
<gdz:strasse>Karl-Rothe-Straße</gdz:strasse>
<gdz:haus>10-14</gdz:haus>
<gdz:qualitaet>A</gdz:qualitaet>
<gdz:qkz>1</gdz:qkz>
</gdz:Ortsangabe>
</gml:featureMember>
<gml:featureMember>
  <gdz:Ortsangabe gml:id="gnde.DEBKGGND00001CFZ">
    <gdz:text>Leipzig (Regierungsbezirk)</gdz:text>
    <gdz:typ>GNDE</gdz:typ>
    <gdz:score>0.9</gdz:score>
    <gdz:treffer/>
    <gdz:geometry>...</gdz:geometry>
    <gdz:bbox>...</gdz:bbox>
    <gdz:kategorie>Regierungsbezirk</gdz:kategorie>
    <gdz:name>Leipzig</gdz:name>
  </gdz:Ortsangabe>
</gml:featureMember>
...
</gml:FeatureCollection>

```

3.3 GeoRSS

Suchergebnisse können als XML-Datei im Atom 1.0 Format ausgegeben werden. Dies dient in erster Linie zur Erfüllung der Anforderungen aus OpenSearch. Geometrien werden unter Verwendung von GeoRSS⁶ enkodiert.

3.3.1 Content-Type

- application/atom+xml

3.3.2 Mapping

Objekte vom Typ Ortsangabe werden im XML als *entry*-Element enkodiert. Folgende Attribute können auf spezielle Atom XML-Elemente abgebildet werden.

Tabelle 6 Abbildung Attribute von Ortsangabe auf Atom

Ortsangabe	GML entry XPath
id	entry/id
text	entry/title
geometry	entry/georss:point
bbox	entry/georss:polygon

⁶ (GeoRSS kein Datum)

Innerhalb des entry/content-Elements werden zusätzlich alle Attribute des Datentyps Ortsangabe als Freitext abgebildet (<Attributname>: <Attributwert><Zeilenumbruch>).

3.3.3 Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="https://www.w3.org/2005/Atom"
      xmlns:opensearch=https://a9.com/-/spec/opensearch/1.1/
      xmlns:geo=https://a9.com/-/opensearch/extensions/geo/1.0/
      xmlns:georss="https://www.georss.org/georss">
  <title>Ortssuche</title>
  <subtitle type="html">
    68776 Ergebnisse für &quot;Leipzig&quot; gefunden.
  </subtitle>
  <updated>2014-06-03T18:30:02Z</updated>
  <author>
    <name>
      Dienstleistungszentrum des Bundes für Geoinformation und Geodäsie
    </name>
  </author>
  <id>https://sg.geodatenzentrum.de/gdz_geokodierung/search/?query=?Leipzig&format=ATOM</id>
  <link rel="self" type="application/atom+xml" href="https://sg.geodatenzentrum.de/gdz_geokodierung/search/?query=?Leipzig&format=ATOM"/>
  <link rel="search" type="application/opensearchdescription+xml" href="https://sg.geodatenzentrum.de/gdz_geokodierung/OpenSearchDescription.xml"/>

  <opensearch:totalResults>68776</opensearch:totalResults>
  <opensearch:Query role="request" searchTerms="Leipzig" startIndex="0"/>

  <link rel="alternate" type="application/json" href="https://sg.geodatenzentrum.de/gdz_geokodierung/search/?query=?Leipzig&format=GEOJSON"/>
  <link rel="alternate" type="text/xml; subtype=gml/3.1.1" href="https://sg.geodatenzentrum.de/gdz_geokodierung/search/?query=?Leipzig&format=GML3"/>
  <link rel="alternate" type="text/xml; subtype=gml/2.1.2" href="https://sg.geodatenzentrum.de/gdz_geokodierung/search/?query=?Leipzig&format=GML2"/>

  <georss:box>40.69418 -74.0667 40.7722 -73.9116</georss:box>

  <!-- Start der Ergebnisliste -->
  <entry>
    <title>Leipzig (Regierungsbezirk)</title>
    <link type="application/json" href="https://sg.geodatenzentrum.de/gdz_geokodierung/record/gnde.DEBKGGND00001CFZ.json"/>
    <link type="text/xml; subtype=gml/3.1.1" href="https://sg.geodatenzentrum.de/gdz_geokodierung/record/gnde.DEBKGGND00001CFZ.gml"/>
    <id>gnde.DEBKGGND00001CFZ</id>
    <updated>2014-06-03T18:30:02Z</updated>
    <georss:polygon>12.147653576027 50.9787945499588 12.147653576027 51.6849737160488 13.2166355423497 51.6849737160488 13.2166355423497 50.9787945499588 12.147653576027 50.9787945499588</georss:polygon>
    <content type="text">
      id: gnde.DEBKGGND00001CFZ
      text: Leipzig (Regierungsbezirk)
      name: Leipzig
      typ: GNDE
      kategorie: Regierungsbezirk
    </content>
  </entry>
</feed>
```



```
    geometry: 51.331884133003797,12.6821445591884
  </content>
</entry>
<entry>
  <title>An der Kirchgasse, 04205 Leipzig - Miltitz</title>
  <link type="application/json" href="https://sg.geodaten-
zentrum.de/gdz_geokodierung/record/str.112.35541.510792.json"/>
  <link type="text/xml; subtype=gml/3.1.1" href="https://sg.geodaten-
zentrum.de/gdz_geokodierung/record/str.112.35541.510792.gml"/>
  <id>str.112.35541.510792</id>
  <updated>2014-06-03T18:30:02Z</updated>
  <georss:polygon>12.2573804855347 51.3218231201172 12.2573804855347
51.3225288391113 12.2587175369263 51.3225288391113 12.2587175369263
51.3218231201172 12.2573804855347 51.3218231201172</georss:polygon>
  <content type="text">
    id: str.112.35541.510792
    text: An der Kirchgasse, 04205 Leipzig - Miltitz
    strasse: An der Kirchgasse
    ortsteil: Miltitz
    ort: Leipzig
    plz: 04205
    typ: Strasse
    geometry: 12.258717000000001,51.322028000000003
  </content>
</entry>
<!-- ...-->

</feed>
```

3.4 JSON Vorschlagsliste

Standardmäßig werden Suchvorschläge als Liste einfacher JSON-Objekte ausgegeben. Die enthaltenen Attribute sind dem [Datenmodell](#) zu entnehmen.

3.4.1 Content-Type

- application/json

3.4.2 Beispiel

```
[ {
  "suggestion" : "01665 Leippen",
  "score" : 0.5303301,
  "highlighted" : "01665 <b>Leippen</b>"
}, {
  "suggestion" : "89340 Leipheim",
  "score" : 0.5303301,
  "highlighted" : "89340 <b>Leipheim</b>",
}, {
  "suggestion" : "Leipheim",
  "score" : 0.5303301,
  "highlighted" : "<b>Leipheim</b>"
}, {
  "suggestion" : "Leipzig",
  "score" : 0.5303301,
  "highlighted" : "<b>Leipzig</b>"
} ]
```

3.5 OpenSearch Suggestion Extension

Für Vorschlagslisten kann das in der OpenSearch Suggestion Extension⁷ angegebene Format verwendet werden. Die Ausgabe erfolgt als JSON-Liste.

3.5.1 Content-Type

- application/x-suggestions+json

3.5.2 Format

Die JSON-Liste enthält in angegebener Reihenfolge folgende Elemente:

1. Suchstring – der vom Nutzer angegebene Suchstring
2. Vorschlagsliste – Liste der Vorschläge, die den gegebenen Suchstring vervollständigen
3. Metadaten (optional) – vom Menschen lesbare Informationen zu den Vorschlägen (z.B. Hervorhebung der gefundenen Wörter)
4. Such-URLs (optional) – Liste der URLs, mit denen nach den vorgeschlagenen Suchbegriffen gesucht werden kann

3.5.3 Beispiel

```
[
  "Leip",
  [ "01665 Leippen", "89340 Leipheim", "Leipheim", "Leipzig" ],
  [ "01665 <b>Leippen</b>", "89340 <b>Leipheim</b>", "<b>Leipheim</b>",
  "<b>Leipzig</b>" ],
  [
    "https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch.html?query=01665+
    Leippen",
    "https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch.html?query=89340+
    Leipheim",
    "https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch.html?query=Leiphe
    im",
    "https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch.html?query=Leipzi
    g"
  ]
] 8
```

⁷ (OpenSearch.org, OpenSearch Suggestions Extension kein Datum)

⁸ Einrückung und Formatierung nur zur besseren Lesbarkeit

4 OpenSearch GeoTemporal Service

Diese Schnittstelle stellt einen durch das OGC spezifizierten OpenSearch GeoTemporal Service⁹ bereit. „Bei OpenSearch¹⁰ handelt es sich um eine auf XML basierende Sammlung von Techniken, die es ermöglicht, Suchergebnisse von Suchmaschinen und Websites in einem standardisierten und maschinenlesbaren Format auszugeben.“¹¹ Diese Schnittstelle bietet eine einfache Implementierungsgrundlage und leichte Integrationsmöglichkeiten in Webanwendungen.

Primär unterstützt der Geokodierungsdienst die unstrukturierte Suche und die Generierung von Vorschlagslisten zur interaktiven Unterstützung der Sucheingabe. Darüber hinaus werden auch strukturierte und räumliche Suchfunktionalitäten angeboten.

4.1 Conformance Class

Der Dienst erfüllt folgende Conformance Klassen des OGC Standards OpenSearch Geo and Time Extension:

- **Core (GeoSpatial Service)**
 - Der Server generiert ein gültiges OpenSearch Description Dokument.
 - Der Server definiert URL-Templates für Ergebnisse im Atom-Format
 - Der Server implementiert eine Bounding-Box Suche
- **Arbitrary Geometry Search**
 - Der Server implementiert die Suche mit beliebiger Geometrie
- **Point and Radius Search**
 - Der Server implementiert eine räumliche Suche mittels Punkt und Radius
- **Minimum spatial operator**
 - Der Server implementiert als räumliche Operation “intersects”
- **Spatial relations**
 - Der Server implementiert weitere räumliche Operationen
- **Get record by id.**
 - Der Server kann auf einen Datensatz mittels ID zugreifen.
- **Search by name**
 - Der Server implementiert eine räumliche Suche unter Verwendung des Namen oder der Adresse eines Ortes.

4.2 OpenSearch Description

Wesentlicher Bestandteil des OpenSearch Standards ist das OpenSearch Description Document. Hierbei handelt es sich um eine XML-Datei, die die Schnittstellen des Suchdiensts beschreibt. Diese Metadaten entsprechen den Service Metadaten eines WMS GetCapabilities Request. Konkret enthält die XML-Datei Metainformationen zum Dienst, Dienstanbieter und dessen Kontaktinformationen.

⁹ (OGC, OpenSearch Geo and Time Extensions Version 1.0.0, 2014)

¹⁰ (OpenSearch.org, OpenSearch 1.1 Draft 5, 2014)

¹¹ Quelle: Wikipedia

Weiterer Bestandteil des OpenSearch Description Document sind parametrisierte URL-Templates. Diese instruieren eine Clientanwendung, wie sich die Suchanfragen an den Dienst zusammensetzen. Ein URL-Template besteht aus der HTTPS-GET URL inklusive Template-Parameter, die durch die Clientanwendung mit den entsprechenden Werten ersetzt werden. Im Beispiel `https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch?query={searchTerms}` würde die Clientanwendung `{searchTerms}` durch die konkrete Suchanfrage ersetzen.

Im OpenSearch Description Document wird mindestens ein URL-Template definiert. Weitere URL-Templates können verwendet werden, um weitere Ergebnistypen und Ausgabeformate zu spezifizieren. Bei mehreren Templates entscheidet der Client anhand Ergebnistyp (Attribut *type*) und Ergebnisformat (Attribut *ref*) welches zu verwenden ist.

Die OpenSearch Geo and Time Extension geht standardmäßig davon aus, dass alle Geometrien (sowohl in der Suche als auch im Ergebnis) in der Georeferenzierung EPSG:4326 angegeben werden. Dies wird durch den Geokodierungsdienst mit einem zusätzlichen Parameter *srsName* erweitert, der zusätzliche Koordinatenreferenzsysteme unterstützt.

4.3 Suche

Die Suche unterscheidet die Erzeugung von Vorschlagslisten, die Ortssuche mit Rückgabe der Geometrien und den Direktzugriff auf Objekte.

4.3.1 Query-Syntax

Eine spezielle Suchsyntax steht bei der Orts- und Vorschlagssuche für die Parameter *query* und *filter* zur Verfügung. Dies ermöglicht unter anderem die strukturierte Suche.

4.3.1.1 Terme

Ein Suchstring wird in Terme und Operatoren zerlegt. Es gibt zwei Typen von Termen: Einzeltermine und Phrasen. Ein Einzelterm besteht aus einem einzelnen Wort, wie z.B. `Leipzig` oder `Sachsen`. Eine Phrase besteht aus einer Gruppe von Wörtern, welche durch Hochkommas umschlossen werden. Bsp.: `"04105 Leipzig"`.

Mehrere Terme können über Boolesche-Operatoren zu einer komplexen Suche kombiniert werden ([S.U.](#)).

4.3.1.2 Attribute

Für die Suche nach Termen können die im Datenmodell angegebenen Attribute verwendet werden. Standardmäßig wird das Attribut *text* verwendet. Die attributive Suche nach einem Term erfolgt durch Angabe des Attributnamen, gefolgt von einem Doppelpunkt (":"), gefolgt vom gesuchten Term.

Bsp.:

- `ort:Leipzig AND strasse:Karl-Rothe-Straße`
- `Leipzig AND bundesland:Sachsen`

4.3.1.3 Wildcard Suche

Die Query-Syntax unterstützt in Einzeltermen die Suche nach einzelnen und multiplen Wildcards. Eine einzelne Wildcard wird per "?" angegeben. Multiple Wildcards werden per "*" angegeben. Da es sich in beiden Fällen um Sonderzeichen handelt, ist den Wildcards ein Escape-Zeichen voranzustellen.

Bsp.:

- `Freib\?rg`
- `Frank*`

Achtung: "?" oder "*" können nicht als erstes Zeichen in einem Term verwendet werden.

4.3.1.4 Fuzzy Suche

Eine Fuzzy Suche erlaubt eine tolerante Suche basierend auf dem Levenshtein (Edit) Distanz Algorithmus. Dies wird in der Sucheingabe über den Tilde-Operator ("~") am Ende eines Einzeltermes angegeben. Zusätzlich kann der Grad der Übereinstimmung als Wert zwischen 0 und 1 angegeben werden. Je näher dieser Wert zu 1 ist, desto ähnlicher sind die gefundenen Terme. Der Standardwert ist 0.5.

Bsp.:

- Leipzig~
- Leipzig~0.6

4.3.1.5 Umgebungssuche

Der Tilde-Operator kann bei Phrasen zur Umgebungssuche verwendet werden. Dies bedeutet, dass die einzelnen Wörter innerhalb der Phrase eine bestimmte Distanz auseinander stehen dürfen. In folgendem Beispiel kann zwischen den Wörtern Karl und Straße ein zusätzliches Wort stehen: "karl straße"~1

4.3.1.6 Gewichtung

Suchterme können über einen Faktor gewichtet werden. Dazu kann das Zeichen "^" gefolgt von einem Gewichtungsfaktor verwendet werden.

Bsp.: ort:Leipzig^5.3 typ:Strasse^0.5

Achtung: Bei einer relativen Bewertung kann diese Gewichtung unter Umständen ignoriert werden.

4.3.1.7 Boolesche Operatoren

Die Booleschen Operatoren erlauben es Terme mittels logischer Operatoren zu kombinieren. Dabei werden die Operatoren AND, "+", OR, NOT und "-" als Boolesche Operatoren unterstützt (**Großschreibung beachten**).

Standardmäßig wird der OR Operator zur Kombination der Terme verwendet. Folgende Beispiele sind also identisch:

- 04105 Leipzig
- 04105 OR Leipzig

Aus Gründen der Fehlertoleranz arbeiten die Operatoren AND, "+" und OR identisch und liefern auch Ergebnisse, wenn eine der Teilbedingungen nicht erfüllt wird.

Um Suchbegriffe im Ergebnis auszuschließen steht der NOT bzw. "-" Operator zur Verfügung.

Tabelle 7 Übersicht NOT Operatoren

Operator	Bedeutung	Beispiel
NOT "!"	Es werden alle Objekte ausgeschlossen, die den Term nach dem NOT Operator enthalten. NOT kann nicht mit einem einzelnen Term verwendet werden.	04105 NOT karl-rothe-str
"-"	Es werden alle Objekte ausgeschlossen, die den Term nach "-" enthalten	04105 -"karl rothe str"

4.3.1.8 Gruppierung

Die Query Syntax unterstützt die Parenthese zur Steuerung der booleschen Logik. Dies erfolgt über eine Klammerung mit "(" und ")".

Bsp.: (Leipzig OR Halle) AND typ:Ort

4.3.1.9 Attributive Gruppierung

Die Klammer kann ebenfalls zur Zuweisung mehrerer Klauseln zu einem Attribut verwendet werden.

Bsp.: `strasse:(karl +rothe straÙe)`

4.3.1.10 Sonderzeichen

Folgende Sonderzeichen werden von der Query Syntax verwendet:

`+ - && | ! () { } [] ^ " ~ * ? : \`

Mit dem Escape-Zeichen "\" können diese unschädlich gemacht werden.

Bsp.: `Halle \ (Saale\)`

4.4 Online Dokumentation

Die OpenSearch GeoTemporal Service Schnittstelle stellt eine Online Dokumentation der angebotenen Endpunkte zur Verfügung. Die Dokumentation wurde auf der Grundlage von [Swagger UI](#) bereitgestellt und bietet neben der Beschreibung der Schnittstelle auch die Möglichkeit die Endpunkte zu testen.

Hinweis: Die Dokumentation kann nur in Browsern aufgerufen werden, die ECMAScript 6 und höher unterstützen. Dies schließt den Internet Explorer aus.

4.4.1 Basis URL

https://sg.geodatenzentrum.de/gdz_geokodierung/swagger-ui.html

4.4.2 Vorschlagssuche

Die Vorschlagssuche dient zur Generierung von Vorschlagslisten, die zur automatischen Vervollständigung eines in einer Suchmaske eingegebenen Suchbegriffs eingesetzt werden können. Es werden demnach nur Objekte vom Typ [Vorschlag](#) zurückgegeben.

4.4.2.1 Basis URL

https://sg.geodatenzentrum.de/gdz_geokodierung/suggest

4.4.2.2 Unterstützte Mime-Types

- `application/json`
- `application/x-suggestions+json`

4.4.2.3 Parameter

Tabelle 8 Parameter für Vorschlagssuche

Parameter	P/O	Datentyp	Beschreibung	Default
query	P	Query	Suchanfrage	-
count	O	Integer	Anzahl der Suchergebnisse	0
bbox	O	String	Bounding Box Angabe der Koordinaten in "West,Süd,Ost,Nord"	-
filter	O	Query	Filter zur Einschränkung der Suchanfrage	-

Geokodierungsdienst für Adressen und Geonamen

srsName	O	String (RegEx: „EPSG:\d+“)	Angabe der zu verwendeten Georeferenzierung für Sucheingaben (geo:box, geo:lat, geo:lon, geo:geometry) und Ergebnisse	EPSG:4326
outputformat	O	String: (json, sxjson)	Ergebnisformat	json
minScore	O	Float	Minimale Score-Wert, den die Ergebnisse aufweisen müssen. Dies überschreibt die internen Score-Filterungsmechanismen.	-
maxScore	O	Float	Maximale Score-Wert, den die Ergebnisse aufweisen müssen. Dies überschreibt die internen Score-Filterungsmechanismen.	-
hl_pre	O	String	Prefix für Hervorhebung von übereinstimmende Wörter	
hl_post	O	String	Postfix für Hervorhebung von übereinstimmende Wörter	

4.4.2.4 Beispiele

https://sg.geodatenzentrum.de/gdz_geokodierung/suggest?query=04105+Leipzig

```
[ {
  "suggestion": "04105 Leipzig - Zentrum-Nordwest",
  "score": 0.40000004,
  "highlighted": null
}, {
  "suggestion": "04105 Leipzig - Zentrum-Nord",
  "score": 0.40000004,
  "highlighted": null
}, ...
]
```

4.4.3 Ortssuche

Die Ortssuche erlaubt die Suche nach Ortsangaben. Es können sowohl unstrukturierte Suchanfragen als auch strukturierte Suchanfragen mittels Query-Syntax verarbeitet werden. In den Ergebnissen werden alle attributiven Informationen inklusive Geometrien zurückgegeben.

4.4.3.1 Basis URL

https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch

Anfragen können sowohl als GET als auch per POST gestellt werden. POST-Anfragen empfehlen sich insbesondere bei der Angabe von Geometrien.

4.4.3.2 Unterstützte Mime-Types

- application/json
- application/gml+xml;version=3.1
- application/gml+xml;version=2.1
- application/atom+xml

4.4.3.3 Parameter

Tabelle 9 Parameter für Ortssuche

Parameter	P/O	Datentyp	Beschreibung	Default
query	P ¹²	Query	Suchanfrage	-
count	O	Integer	Anzahl der Suchergebnisse	20
bbox	O	String Angabe der Koordinaten in "West,Süd,Ost,Nord"	Umschließendes Rechteck für eine räumliche Suche, welche sich immer auf die Punktgeometrie des Objekts bezieht.	-
filter	O	Query	Filter zur Einschränkung der Suchanfrage	-
lat	O	Float	Geogr. Breite bzw. Hochwert eines Punktes für eine Umkreissuche. Die Umkreissuche bezieht sich immer auf die Punktgeometrie des Objekts. Sofern <i>query</i> leer ist, werden die Ergebnisse entsprechend Ihrer Entfernung zum Suchpunkt sortiert.	-
lon	O	Float	Geogr. Länge bzw. Rechtswert eines Punktes für eine Umkreissuche. S. Anmerkungen <i>lat</i> .	-
distance	O	Float	Suchradius der Umkreissuche in Meter.	1000
geometry	O	WKT	Geometrie für eine räumliche Suche, welche sich immer auf die Punktgeometrie des Objekts bezieht. Angabe einer Geometrie mittels Well Known Text (WKT). Folgende 2D Geometrien werden unterstützt: POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON	-
relation	O	String: (intersects, contains, disjoint)	Relation für räumliche Suche.	intersects
srsName	O	String (RegEx: „EPSG:d+“)	Angabe der zu verwendeten Georeferenzierung für Sucheingaben (<i>bbox</i> , <i>lat</i> , <i>lon</i> , <i>geometry</i>) und Ergebnisse	EPSG:4326

¹² Werden die räumlichen Suchparameter *lat* + *lon* oder *geometry* angegeben, ist *query* optional. Dies erlaubt die reverse Geokodierungen.

Geokodierungsdienst für Adressen und Geonamen

outputformat	O	String: (atom, gml, gml3, gml2, json)	Ergebnisformat	json
allScores	O	Boolean (0, 1, false, true, on, off)	Mit diesem Parameter kann gesteuert werden, ob die Einzelgütewerte in der Antwort ausgegeben werden sollen.	false
minScore	O	Float	Minimale Score-Wert, den die Ergebnisse aufweisen müssen. Dies überschreibt die internen Score-Filterungsmechanismen.	-
maxScore	O	Float	Maximale Score-Wert, den die Ergebnisse aufweisen müssen. Dies überschreibt die internen Score-Filterungsmechanismen.	-
properties	O	List	Kommaseparierte Liste anzuzeigender properties. Die Properties <i>text</i> , <i>typ</i> , <i>score</i> und <i>bbox</i> werden immer angezeigt.	

4.4.3.4 Beispiele

https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch?query=Leipzig&outputformat=json

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "bbox": [
        12.2880411148071, 51.2924118041992,
        12.5057373046875, 51.4160003662109
      ],
      "geometry": {
        "type": "Point",
        "coordinates": [12.3751, 51.3408]
      },
      "properties": {
        "text": "Leipzig",
        "typ": "Ort",
        "score": 0.5303301,
        "treffer": "",
        "bbox": {
          "type": "Polygon",
          "coordinates": [
            [
              [12.288, 51.2924],
              [12.288, 51.416],
              [12.5057, 51.416],
              [12.5057, 51.2924],
              [12.288, 51.2924]
            ]
          ]
        }
      }
    }
  ]
}
```

```

"ags": "14713000",
"rs": "147130000000",
"schluessel": "1471300000000000",
"bundesland": "Sachsen",
"kreis": "Leipzig",
"verwgem": "Leipzig",
"gemeinde": "Leipzig",
"ort": "Leipzig",
"ortsteil": ""
},
{id": "DEGAC00000081785"
}, {
...
}, ...
]
}

```

4.4.4 Strukturierte Ortssuche / Geocode

Die Schnittstelle erlaubt im Gegensatz zur Ortssuche nur die strukturierte Suche von Ortsangaben. Anstelle der Query sind die erforderlichen Adressattribute direkt als Request-Parameter angegeben werden. In den Ergebnissen werden alle attributiven Informationen inklusive Geometrien zurückgegeben.

4.4.4.1 Basis URL

https://sg.geodatenzentrum.de/gdz_geokodierung/geocode

Anfragen können sowohl als GET als auch per POST gestellt werden. POST-Anfragen empfehlen sich insbesondere bei der Angabe von Geometrien.

4.4.4.2 Unterstützte Mime-Types

- application/json
- application/gml+xml;version=3.1
- application/gml+xml;version=2.1

4.4.4.3 Parameter

Tabelle 10 Parameter für die strukturierte Suche

Parameter	P/O	Datentyp	Beschreibung	Default
ort	O	String	Angabe des zu suchenden Orts.	-
plz	O	String	Angabe der zu suchenden Postleitzahl.	-
ortsteil	O	String	Angabe des zu suchenden Ortsteils.	-
strasse	O	String	Angabe der zu suchenden Straße.	-
haus	O	String	Angabe der zu suchenden Hausnummer.	-
strasse_haus	O	String	Kombinierte Angabe der zu suchenden Straße sowie einer Hausnummer in dieser Straße.	-
count	O	Integer	Anzahl der Suchergebnisse	20

Geokodierungsdienst für Adressen und Geonamen

bbox	<input type="radio"/>	String Angabe der Koordinaten in "West,Süd,Ost,Nord"	Umschließendes Rechteck für eine räumliche Suche, welche sich immer auf die Punktgeometrie des Objekts bezieht.	-
filter	<input type="radio"/>	Query	Filter zur Einschränkung der Suchanfrage	-
lat	<input type="radio"/>	Float	Geogr. Breite bzw. Hochwert eines Punktes für eine Umkreissuche. Die Umkreissuche bezieht sich immer auf die Punktgeometrie des Objekts. Sofern <i>query</i> leer ist, werden die Ergebnisse entsprechend Ihrer Entfernung zum Suchpunkt sortiert.	-
lon	<input type="radio"/>	Float	Geogr. Länge bzw. Rechtswert eines Punktes für eine Umkreissuche. S. Anmerkungen <i>lat</i> .	-
distance	<input type="radio"/>	Float	Suchradius der Umkreissuche in Meter.	1000
geometry	<input type="radio"/>	WKT	Geometrie für eine räumliche Suche, welche sich immer auf die Punktgeometrie des Objekts bezieht. Angabe einer Geometrie mittels Well Known Text (WKT). Folgende 2D Geometrien werden unterstützt: POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON	-
relation	<input type="radio"/>	String: (intersects, contains, disjoint)	Relation für räumliche Suche.	intersects
srsName	<input type="radio"/>	String (RegEx: „EPSG:\d+“)	Angabe der zu verwendeten Georeferenzierung für Sucheingaben (<i>bbox</i> , <i>lat</i> , <i>lon</i> , <i>geometry</i>) und Ergebnisse	EPSG:4326
outputformat	<input type="radio"/>	String: (atom, gml, gml3, gml2, json)	Ergebnisformat	json
allScores	<input type="radio"/>	Boolean (0, 1, false, true, on, off)	Mit diesem Parameter kann gesteuert werden, ob die Einzelgütewerte in der Antwort ausgegeben werden sollen.	false
minScore	<input type="radio"/>	Float	Minimale Score-Wert, den die Ergebnisse aufweisen müssen. Dies überschreibt die internen Score-Filterungsmechanismen.	-

Geokodierungsdienst für Adressen und Geonamen

maxScore	O	Float	Maximale Score-Wert, den die Ergebnisse aufweisen müssen. Dies überschreibt die internen Score-Filterungsmechanismen.	-
properties	O	List	Kommaseparierte Liste anzuzeigender properties. Die Properties <i>text</i> , <i>typ</i> , <i>score</i> und <i>bbox</i> werden immer angezeigt.	

4.4.4.4 Beispiele

https://sg.geodatenzentrum.de/gdz_geokodierung/geocode?ort=Leipzig&outputformat=json

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "bbox": [
        12.2880411148071, 51.2924118041992,
        12.5057373046875, 51.4160003662109
      ],
      "geometry": {
        "type": "Point",
        "coordinates": [12.3751, 51.3408]
      },
      "properties": {
        "text": "Leipzig",
        "typ": "Ort",
        "score": 0.5303301,
        "treffer": "",
        "bbox": {
          "type": "Polygon",
          "coordinates": [
            [
              [12.288, 51.2924],
              [12.288, 51.416],
              [12.5057, 51.416],
              [12.5057, 51.2924],
              [12.288, 51.2924]
            ]
          ]
        }
      },
      "ags": "14713000",
      "rs": "147130000000",
      "schluessel": "1471300000000000",
      "bundesland": "Sachsen",
      "kreis": "Leipzig",
      "verwgem": "Leipzig",
      "gemeinde": "Leipzig",
      "ort": "Leipzig",
      "ortsteil": ""
    },
    {
      "id": "DEGAC00000081785"
    },
    ...
  ],
  ...
}
```

```
]
}
```

4.4.5 Direktzugriff auf Objekte / GetRecord

Der Direktzugriff erlaubt die Abfrage eines bestimmten Datensatzes per ID. Das Ergebnis ist eine Ortsangabe mit allen attributiven Informationen.

4.4.5.1 Basis URL

https://sg.geodatenzentrum.de/gdz_geokodierung/record/{uid}

4.4.5.2 Unterstützte Mime-Types

- application/json
- text/xml; subtype=gml/3.1.1
- text/xml; subtype=gml/2.1.2
- application/atom+xml

4.4.5.3 Parameter

Tabelle 11 Parameter für GetRecord

Parameter	P/O ¹³	Datentyp	Beschreibung	Default
uid	P	String	Identifikator des Datensatzes. Dieser Parameter wird innerhalb des URL-Pfades angegeben. (s. Basis-URL {uid})	-
srs	O	String (Regex: „EPSG:\d+“)	Angabe der zu verwendeten Georeferenzierung für Sucheingaben (geo:box, geo:lat, geo:lon, geo:geometry) und Ergebnisse	EPSG:4326
format	O	String: (atom, gml, gml3, gml2, json)	Ergebnisformat	json

4.4.5.4 Beispiele

https://sg.geodatenzentrum.de/gdz_geokodierung/record/DEGAC00000081785?outputformat=json

```
{
  "type": "Feature",
  "bbox": [
    12.2358822418717, 51.2362041325005,
    12.5397442213646, 51.4340460610466
  ],
  "geometry": {
    "type": "Point",
    "coordinates": [12.37793, 51.33829]
  },
}
```

¹³ Pflichtparameter (P) / optionaler Parameter (O)

```

"properties":{
  "text":"Leipzig",
  "typ":"Ort",
  "score":0.0,
  "bbox":{
    "type":"Polygon",
    "coordinates":[[
      [12.23588,51.2362],
      [12.23588,51.43405],
      [12.53974,51.43405],
      [12.53974,51.2362],
      [12.23588,51.2362]
    ]]
  },
  "ags":"14713000",
  "rs":"147130000000",
  "schluessel":"1471300000000000",
  "bundesland":"Sachsen",
  "kreis":"Leipzig",
  "verwgem":"Leipzig",
  "gemeinde":"Leipzig",
  "ort":"Leipzig",
  "ortsteil":""
},
"id":"DEGAC00000081785"
}

```

4.5 OpenSearch Response

Die OpenSearch-Schnittstelle kann alle oben erwähnten Rückgabeformate bedienen. Dies umfasst auch die zur Erfüllung von OpenSearch geforderte Enkodierung der Ergebnisse mittels Atom 1.0. Über den *outputformat*-Parameter kann das Ausgabeformat explizit gewählt werden.

4.6 Error handling

Fehler werden primär durch den HTTP-Status-Code gekennzeichnet. Dabei unterscheidet man in clientseitige Fehler (4xx) und serverseitige Fehler (5xx)

Tabelle 12 HTTP-Status Codes im Fehlerfall

Code	Name	Details	Beschreibung
500	Internal Server Error		Standardfehler für serverseitige Fehler
501	Not Implemented	OperationNotSupported	Die Anfrage kann vom Server nicht verarbeitet werden, da das entsprechende Feature nicht implementiert ist
503	Service Unavailable		Server ist temporär nicht verfügbar (z.B. wegen Überlastung)
504	Gateway Timeout		Die internen Suchdienste im Backend antworten nicht innerhalb des geforderten Timeout-Intervall.
400	Bad Request	InvalidFormat	Fehlender QUERY-Parameter
400	Bad Request	InvalidParameterValue	Falscher Wert oder Formatierung eines Parameter

413	Request Entity Too Large		Die Anfrage generiert zu viele Treffer
-----	--------------------------	--	--

4.7 Anwendungsbeispiele

4.7.1 OpenSearch Beschreibung aufrufen

https://sg.geodatenzentrum.de/gdz_ortssuche/index.xml

4.7.2 Vorschlagssuche

Vorschlagssuche nach Leipzig

query: leipzig

URL: https://sg.geodatenzentrum.de/gdz_ortssuche/suggest?query=leipzig

Vorschlagssuche nach Leipzig eingeschränkt auf Sachsen-Anhalt

query: leipzig

filter: bundesland:"sachsen-anhalt"

URL: https://sg.geodatenzentrum.de/gdz_ortssuche/suggest?query=leipzig&filter=bundesland:%22sachsen-anhalt%22

4.7.3 Volltextsuche

Suche nach Leipzig

query: leipzig

URL: https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?query=leipzig

Suche nach Karl-Rothe-Straße 10-14 Leipzig

query: Karl-Rothe-Straße 10-14 Leipzig

URL: https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?query=Karl-Rothe-Stra%C3%9Fe%2010-14%20Leipzig

Suche nach Karl-Rothe-Straße in Sachsen

query: Karl-Rothe-Straße

filter: bundesland:Sachsen

URL: https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?query=Karl-Rothe-Stra%C3%9Fe&filter=bundesland:Sachsen

4.7.4 Strukturierte Suche

Suche nach Ort Leipzig

query: ort:Leipzig

URL: https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?query=ort:Leipzig

Suche nach Adresse Karl-Rothe-Straße 10-14 Leipzig

query: strasse:(Karl-Rothe-Straße) haus:(10-14) ort:Leipzig

URL: [https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?query=strasse:\(Karl-Rothe-Stra%C3%9Fe\)%20haus:\(10-14\)%20ort:Leipzig](https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?query=strasse:(Karl-Rothe-Stra%C3%9Fe)%20haus:(10-14)%20ort:Leipzig)

Suche nach Adresse Karl-Rothe-Straße 10-14 Leipzig

Parameter: strasse:(Karl-Rothe-Straße) haus:(10-14) ort:Leipzig

URL: [https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?strasse:\(Karl-Rothe-Stra%C3%9Fe\)&haus:\(10-14\)&ort:\(Leipzig\)](https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?strasse:(Karl-Rothe-Stra%C3%9Fe)&haus:(10-14)&ort:(Leipzig))

4.7.5 Reverse Geokodierung

Alle Straßen im Umkreis von 1000m um 12.389443, 51.342277

lat: 12.389443

lon: 51.342277

distance: 1000

filter: typ:strasse

URL: https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?lon=12.389443&lat=51.342277&distance=1000&filter=typ:strasse

Alle Orte in der BBOX [12.3,51.3,12.4,51.35]

bbox: 12.3,51.3,12.4,51.35

filter: typ:ort

URL: https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?bbox=12.3,51.3,12.4,51.35&filter=typ:ort

Alle Hausadressen im Polygon POLYGON((12.3 51.3,12.35 51.35,12.4 51.3,12.3 51.3))

geometry: POLYGON((12.3 51.3,12.35 51.35,12.4 51.3,12.3 51.3))

filter: typ:haus

URL: [https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?geometry=POLYGON\(\(12.3%2051.3,12.35%2051.35,12.4%2051.3,12.3%2051.3\)\)&filter=typ:haus](https://sg.geodatenzentrum.de/gdz_ortssuche/geosearch?geometry=POLYGON((12.3%2051.3,12.35%2051.35,12.4%2051.3,12.3%2051.3))&filter=typ:haus)

5 Web Feature Service

Als weitere Schnittstelle wird ein Basic Web Feature Service in Version 1.1.0¹⁴ bereitgestellt, der die folgenden Operationen unterstützt:

- GetCapabilities,
- DescribeFeatureType und
- GetFeature

Alle Operationen können über HTTPS aufgerufen werden und unterstützen Anfragen per Key Value Pair (KVP) oder per XML-Post.

Mit dem Web Feature Service können nur Objekte vom Typ Ortsangabe angefragt werden.

5.1 GetCapabilities

5.1.1 Beschreibung

Die GetCapabilities Operation liefert die Metadaten des WFS. Weitere Details können dem Web Feature Service 1.1.0 Standard entnommen werden.

5.1.2 Ergebnis

Ergebnis ist ein XML-Dokument, welches den Dienst und die angebotenen Operationen beschreibt. Die Hauptabschnitte des Dokuments sind:

- ServiceIdentification – Beschreibung des Dienstes (Name, Abstract, Nutzungsbedingungen)
- ServiceProvider – Kontaktinformationen des Anbieters des Dienstes
- OperationsMetadata – Auflistung der angebotenen Operationen und deren Parameter
- FeatureTypeList – Liste der angebotenen Datentypen (nur Ortsangabe)
- Filter_Capabilities – Liste der anwendbaren Suchfilter

Anhand der Capabilities sind folgende Einschränkungen im Dienst erkennbar:

- Die maximale Anzahl der ermittelbaren Features (DefaultMaxFeatures) liegt bei 1000
- Für die GetFeature-Operation werden nur folgende Filterausdrücke zugelassen:
 - And
 - Or
 - Not
 - PropertyIsEqualTo
 - PropertyIsLike
 - BBOX
 - Intersects
 - Disjoint
 - Contains
 - Within
 - DWithin

¹⁴ (OGC, Web Feature Service Implementation Specification 2005)

5.1.3 Beispiel

HTTP-GET:

https://sg.geodatenzentrum.de/wfs_geokodierung?SERVICE=WFS&REQUEST=GetCapabilities

Antwort:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:WFS_Capabilities
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ows="https://www.opengis.net/ows"
  xmlns:xlink="https://www.w3.org/1999/xlink"
  xmlns:gml="https://www.opengis.net/gml"
  updateSequence="0" version="1.1.0">
  <ows:ServiceIdentification>
    <ows:Title>Geokodierungsdienst der AdV</ows:Title>
    ...
</wfs:WFS_Capabilities>
```

5.2 DescribeFeatureType

5.2.1 Beschreibung

Die DescribeFeatureType Operation beschreibt das Datenmodell und die vom Dienst verwendeten Datentypen über eine Schemadefinition. Weitere Details können dem Web Feature Service 1.1.0 Standard entnommen werden.

5.2.2 Ergebnis

Das Ergebnis der Operation ist eine XML-Schemadefinition, welches die XML-Struktur einer GetFeature-Antwort beschreibt.

5.2.3 Beispiel

HTTP-GET:

https://sg.geodatenzentrum.de/wfs_geokodierung?SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType

Antwort:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsd="https://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  targetNamespace="https://www.geodatenzentrum.de/ortsangabe">
  <xsd:import namespace="https://www.opengis.net/gml"
    schemaLocation="https://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
  <xsd:complexType name="Ortsangabe">
    ...
</xsd:schema>
```

5.3 GetFeature

5.3.1 Beschreibung

Die GetFeature Operation führt eine Suche durch und findet Datensätze, die zur Sucheingabe passen. Dazu muss ein Filterausdruck nach der OpenGIS Filter Encoding Implementation Specification¹⁵ angegeben werden. Eine detaillierte Beschreibung der Schnittstelle kann dem Web Feature Service 1.1.0 Standard¹⁶ entnommen werden.

Implementationsbedingt werden nicht alle Filteroperatoren bedient. Zugelassen sind folgende Operatoren:

- Or
- And
- Not
- PropertyIsEqualTo
- PropertyIsLike
- BBOX
- Intersects
- Disjoint
- Contains
- Within
- DWithin

Aufgrund der Implementierungsgrundlage kann es zu Abweichung bei der Interpretation der Filteroperatoren mit dem üblichen WFS-Verhalten kommen. Dies dient in erster Linie zur Bereitstellung einer möglichst fehlertoleranten Suche.

Mit den logischen Operatoren AND oder OR können mehrere Attribute in einer Abfrage kombiniert werden. Aus Gründen der Fehlertoleranz arbeiten beide Operatoren identisch und liefern auch Ergebnisse, wenn eine der Teilbedingungen nicht erfüllt wird.

Die textuelle Suche nach Attributwerten erfolgt mit den Operatoren PropertyIsEqualTo und PropertyIsLike. Der übergebene Wert wird normalisiert und in Token (Teilwörter) zerlegt, nach denen dann gesucht wird. Bei PropertyIsEqualTo muss die Reihenfolge der Token übereinstimmen. Dagegen arbeitet PropertyIsLike etwas toleranter. Hier kann die Reihenfolge der Token abweichen. Generell müssen aber nicht alle Token für einen Treffer vorkommen.

Der BBOX Operator darf im Filterausdruck nur einmal vorkommen. D.h. eine Kombination mehrerer BBOX-Operatoren ist nicht zulässig. Für die Suche per BBOX stehen die Attribute *geometry* und *bbox* zur Verfügung.

5.3.2 Ergebnis

Ergebnisse der GetFeature-Operation können in den oben angegebenen GML- und GeoJSON-Formaten ausgegeben werden. Weitere Formate werden nicht vom WFS unterstützt. Die Wahl des Formats erfolgt über den Parameter *OUTPUTFORMAT*. Standardmäßig wird GML 3.1.1 verwendet.

Die Datensätze sind in der FeatureCollection absteigend nach *score* sortiert.

¹⁵ (OGC, OpenGIS Filter Encoding Implementation Specification 2012)

¹⁶ (OGC, Web Feature Service Implementation Specification 2005)

5.3.3 Beispiel

XML-POST

URL: https://sg.geodatenzentrum.de/wfs_geokodierung

Body:

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.1.0" service="WFS" maxFeatures="20"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:31467">
    <ogc:Filter>
      <ogc:Or>
        <ogc:PropertyIsLike escapeChar="\\" wildCard="*" singleChar="?">
          <ogc:PropertyName>ort</ogc:PropertyName>
          <ogc:Literal>Frankfurt Main</ogc:Literal>
        </ogc:PropertyIsLike>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>typ</ogc:PropertyName>
          <ogc:Literal>Ort</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Or>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

Antwort:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection
  xmlns:xs="https://www.w3.org/2001/XMLSchema"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:ows="https://www.opengis.net/ows"
  xmlns:xlink="https://www.w3.org/1999/xlink" numberOfFeatures="20">
  <gml:boundedBy>
    <gml:Envelope srsDimension="2" srsName="https://www.open-
gis.net/gml/srs/epsg.xml#31467">
      <gml:lowerCorner>3476122.202013193 5552763.767109428</gml:lo-
werCorner>
      <gml:upperCorner>3482078.9145937776 5560820.7907572985</gml:up-
perCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:featureMember>
    <gdz:Ortsangabe gml:id="DEGAC00000034810">
      <gml:boundedBy>
        <gml:Envelope srsDimension="2" srsName="https://www.open-
gis.net/gml/srs/epsg.xml#31467">
          <gml:lowerCorner>3476181.4060013588
```

Geokodierungsdienst für Adressen und Geonamen

```
5558608.530328523</gml:lowerCorner>
  <gml:upperCorner>3477723.8402231266
5559587.7290658</gml:upperCorner>
  </gml:Envelope>
  </gml:boundedBy>
  <gdz:text>60433 Frankfurt am Main - Frankfurter Berg</gdz:text>
  <gdz:typ>Ort</gdz:typ>
  <gdz:score>0.98425376</gdz:score>
  <gdz:geometry>
    <gml:Point srsDimension="2" srsName="https://www.open-
gis.net/gml/srs/epsg.xml#31467">
      <gml:pos>3476840.155170563 5558972.791725892</gml:pos>
    </gml:Point>
  </gdz:geometry>
  <gdz:bbox>
    <gml:Polygon srsDimension="2" srsName="https://www.open-
gis.net/gml/srs/epsg.xml#31467">
      <gml:exterior>
        <gml:LinearRing srsDimension="2">
          <gml:posList>3476181.4060013588
5558615.186362992 3476185.754484719 5559587.7290658 3477723.8402231266
5559581.073398344 3477719.772387507 5558608.530328523 3476181.4060013588
5558615.186362992</gml:posList>
        </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
  </gdz:bbox>
  <gdz:ags>06412000</gdz:ags>
  <gdz:rs>064120000000</gdz:rs>
  <gdz:schluesssel>064120000000048902762, 064120000000048904519,
064120000000048905359, 064120000000048901916, 064120000000048905318,
064120000000048901297, 064120000000048906170, 064120000000049101198,
064120000000048905826, 064120000000048903715, 064120000000048903800,
064120000000048902135, 064120000000048905829, 064120000000048902773,
064120000000048904647, 064120000000048905828, 064120000000048905860,
064120000000048902587, 064120000000048902472, 064120000000048905862,
064120000000048900589, 064120000000048902329, 064120000000048905827,
064120000000048906030, 064120000000048904376, 064120000000048905850,
064120000000048905859, 064120000000048906176, 064120000000048905825,
064120000000048903804, 064120000000048905849, 064120000000048903281,
064120000000048905861, 064120000000049101297, 064120000000048905236,
064120000000049002329</gdz:schluesssel>
  <gdz:bundesland>Hessen</gdz:bundesland>
  <gdz:regbezirk>Darmstadt</gdz:regbezirk>
  <gdz:kreis>Frankfurt am Main</gdz:kreis>
  <gdz:verwgem>Frankfurt am Main</gdz:verwgem>
  <gdz:gemeinde>Frankfurt am Main</gdz:gemeinde>
  <gdz:plz>60433</gdz:plz>
  <gdz:ort>Frankfurt am Main</gdz:ort>
  <gdz:ortsteil>Frankfurter Berg</gdz:ortsteil>
  </gdz:Ortsangabe>
</gml:featureMember>
...
</wfs:FeatureCollection>
```

Über den Parameter RESULTTYPE kann spezifiziert werden ob, eine leere FeatureCollection angezeigt werden soll. Dazu ist der Parameter mit dem Wert *hits* aufzurufen. In der Folge wird eine leere FeatureCollection mit den Attributen *numberOfFeatures* und *timestamp* angezeigt.

Im Fall des GeoJSON-Formats wird das Attribut *numberReturned* anstelle von *numberOfFeatures* angezeigt.

Antwort:

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:FeatureCollection
  xmlns:xs="https://www.w3.org/2001/XMLSchema"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:ows="https://www.opengis.net/ows"
  xmlns:xlink="https://www.w3.org/1999/xlink"
  numberOfFeatures="20" timeStamp="2020-12-02T16:16:12.247Z" />
```

5.4 Anwendungsbeispiele

5.4.1 OpenSearch Beschreibung aufrufen

https://sg.geodatenzentrum.de/gdz_ortssuche/index.xml

5.4.2 Volltextsuche

Suche nach Leipzig

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.1.0" service="WFS" maxFeatures="20"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:31467">
    <ogc:Filter>
      <ogc:PropertyIsLike escapeChar="\ " wildCard="*" singleChar="?">
        <ogc:PropertyName>text</ogc:PropertyName>
        <ogc:Literal>Leipzig</ogc:Literal>
      </ogc:PropertyIsLike>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

Suche nach Karl-Rothe-Straße in Sachsen

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.1.0" service="WFS" maxFeatures="20"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:31467">
    <ogc:Filter>
      <ogc:And>
```

```
<ogc:PropertyIsLike escapeChar="\ " wildCard="*" singleChar="?">
  <ogc:PropertyName>text</ogc:PropertyName>
  <ogc:Literal>Karl-Rothe-Straße</ogc:Literal>
</ogc:PropertyIsLike>
<ogc:PropertyIsEqualTo>
  <ogc:PropertyName>bundesland</ogc:PropertyName>
  <ogc:Literal>Sachsen</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

5.4.3 Strukturierte Suche

Suche nach Ort Leipzig

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.1.0" service="WFS" maxFeatures="20"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:31467">
    <ogc:Filter>
      <ogc:PropertyIsLike escapeChar="\ " wildCard="*" singleChar="?">
        <ogc:PropertyName>ort</ogc:PropertyName>
        <ogc:Literal>Leipzig</ogc:Literal>
      </ogc:PropertyIsLike>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

Suche nach Adresse Karl-Rothe-Straße 10-14 Leipzig

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.1.0" service="WFS" maxFeatures="20"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:31467">
    <ogc:Filter>
      <ogc:And>
        <ogc:PropertyIsLike escapeChar="\ " wildCard="*" singleChar="?">
          <ogc:PropertyName>strasse</ogc:PropertyName>
          <ogc:Literal>Karl-Rothe-Straße</ogc:Literal>
        </ogc:PropertyIsLike>
        <ogc:PropertyIsLike escapeChar="\ " wildCard="*" singleChar="?">
          <ogc:PropertyName>haus</ogc:PropertyName>
          <ogc:Literal>10-14</ogc:Literal>
        </ogc:PropertyIsLike>
        <ogc:PropertyIsLike escapeChar="\ " wildCard="*" singleChar="?">
          <ogc:PropertyName>ort</ogc:PropertyName>
          <ogc:Literal>Leipzig</ogc:Literal>
        </ogc:PropertyIsLike>
      </ogc:And>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

```
</ogc:PropertyIsLike>
</ogc:And>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

5.4.4 Strukturierte Suche mit Ausgabe der Einzelgüte

Bei der strukturierten Suche ist es möglich zusätzlich zur Gesamtgüte auch die berechneten Einzelgütewerte in der Antwort ausgeben zu lassen. Dies erfolgt über die PropertyName-Auswahl der WFS-Schnittstelle. In der Key-Value-Pair Schnittstelle ist dies der Parameter PROPERTYNAME. Über XML-Post ist dies der XPath wfs:GetFeature/wfs:Query/wfs:PropertyName.

Sollte dies genutzt werden, müssen alle optionalen Attribute des Datenmodells (z.B. plz, ort, strasse, ...) angegeben werden, damit diese in der Ausgabe enthalten sind.

Beispiel:

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.1.0" service="WFS" maxFeatures="20"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:31467">
    <ogc:PropertyName>plz</ogc:PropertyName>
    <ogc:PropertyName>ort</ogc:PropertyName>
    <ogc:PropertyName>ortsteil</ogc:PropertyName>
    <ogc:PropertyName>strasse</ogc:PropertyName>
    <ogc:PropertyName>haus</ogc:PropertyName>
    <ogc:PropertyName>score_plz</ogc:PropertyName>
    <ogc:PropertyName>score_ort</ogc:PropertyName>
    <ogc:PropertyName>score_ortsteil</ogc:PropertyName>
    <ogc:PropertyName>score_strasse</ogc:PropertyName>
    <ogc:PropertyName>score_haus</ogc:PropertyName>
    <ogc:Filter>
      <ogc:PropertyIsLike escapeChar="\ " wildcard="*" singleChar="?">
        <ogc:PropertyName>ort</ogc:PropertyName>
        <ogc:Literal>Leipzig</ogc:Literal>
      </ogc:PropertyIsLike>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

5.4.5 Reverse Geokodierung

Alle Straßen im Umkreis von 1000m um 12.389443, 51.342277. Die räumliche Suche bezieht sich auf die Punktgeometrie (*geometry*) des Objekts. Die Sortierung der Ergebnisse basiert auf der Entfernung zum Suchmittelpunkt.

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.1.0" service="WFS"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc">
```


Geokodierungsdienst für Adressen und Geonamen

```
xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
xmlns:gml="https://www.opengis.net/gml"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
<wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:4326">
  <ogc:Filter>
    <ogc:And>
      <ogc:DWithin>
        <ogc:PropertyName>geometry</ogc:PropertyName>
        <gml:Point srsName="EPSG:4326">
          <gml:coordinates decimal="." cs="," ts=" ">
            12.389443,51.342277
          </gml:coordinates>
        </gml:Point>
        <ogc:Distance units="meter">1000</ogc:Distance>
      </ogc:DWithin>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>typ</ogc:PropertyName>
        <ogc:Literal>Strasse</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:And>
  </ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

Alle Orte in der BBOX [12.3,51.3,12.4,51.35]. Die räumliche Suche bezieht sich auf die Punktgeometrie (*geometry*) des Objekts.

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.0.0" service="WFS"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:4326">
    <ogc:Filter>
      <ogc:And>
        <ogc:BBOX>
          <ogc:PropertyName>geometry</ogc:PropertyName>
          <gml:Box srsName="EPSG:4326">
            <gml:coordinates>12.3,51.3 12.4,51.35</gml:coordinates>
          </gml:Box>
        </ogc:BBOX>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>typ</ogc:PropertyName>
          <ogc:Literal>Ort</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:And>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

Alle Hausadressen im Polygon POLYGON((12.3 51.3,12.35 51.35,12.4 51.3,12.3 51.3)). Die räumliche Suche bezieht sich auf die Punktgeometrie (*geometry*) des Objekts.

Geokodierungsdienst für Adressen und Geonamen

```
<?xml version="1.0" encoding="UTF-8" ?>
<wfs:GetFeature version="1.0.0" service="WFS"
  xmlns:wfs="https://www.opengis.net/wfs"
  xmlns:ogc="https://www.opengis.net/ogc"
  xmlns:gdz="https://www.geodatenzentrum.de/ortsangabe"
  xmlns:gml="https://www.opengis.net/gml"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="gdz:Ortsangabe" srsName="EPSG:4326">
    <ogc:Filter>
      <ogc:And>
        <ogc:Intersects>
          <ogc:PropertyName>gdz:geometry</ogc:PropertyName>
          <gml:Polygon srsName="EPSG:4326">
            <gml:exterior>
              <gml:LinearRing>
                <gml:coordinates decimal="." cs="," ts=" ">
                  12.3,51.3 12.35,51.35 12.4,51.3 12.3,51.3
                </gml:coordinates>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </ogc:Intersects>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>typ</ogc:PropertyName>
          <ogc:Literal>Haus</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:And>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

6 Sonstige Hinweise

6.1 Suchverhalten

Der Geokodierungsdienst basiert auf einer Volltextsuche. Dazu werden die Ortsangaben über das in Punkt 2.1 beschriebene Datenmodell indiziert. Sowohl für die Indizierung der Attribute als auch für die Sucheingabe erfolgt eine Normalisierung und Zerlegung in Terme (Wörter). Bei der Suche werden alle Ortsangaben gesucht, die die gesuchten Terme enthalten. Es müssen nicht alle gesuchten Terme in der Ortsangabe enthalten sein. Generell gilt aber: je mehr Terme übereinstimmen desto höher ist die Bewertung (*score*).

6.1.1 Unstrukturierte Volltextsuche

Die Volltextsuche erfolgt über das Attribut *text* oder wenn beim OpenSearch GeoTemporal Service keine Attributnamen in der Suche angegeben werden. Das bedeutet, dass gleichzeitig die Attribute *plz*, *ort*, *ortsteil*, *strasse*, *haus*, *name* und *kategorie* durchsucht werden. Die Bewertung der Ergebnisse erfolgt **absolut**.

6.1.2 Strukturierte Suche

Per Query-Syntax in der OpenSearch-Schnittstelle oder per WFS-Schnittstelle können strukturierte Suchen nach bestimmten Attributwerten durchgeführt werden.

Wird nach mindestens einem der Attribute *plz*, *ort*, *ortsteil*, *strasse* und/oder *haus* gesucht, dann erfolgt eine strukturierte Adresssuche, die abweichend von der unstrukturierten Volltextsuche eine **relative** Bewertung verwendet.

6.1.3 Bewertung

Inwieweit das gefundene Objekt mit der Suchanfrage übereinstimmt, wird im dynamischen Attribut *score* angegeben. Die Bewertung kann dabei absolut oder relativ erfolgen. Allgemein gilt: Je größer der Wert *score* ist desto besser ist das Ergebnis und desto weiter oben steht dies in der Ergebnisliste.

6.1.3.1 Absolute Bewertung

Das Attribut *score* hat bei einer **absoluten** Bewertung einen Wert größer als 1. Nach oben gibt es keinen Maximalwert.

6.1.3.2 Relative Bewertung

Bei der **relativen** Bewertung werden zunächst die Übereinstimmungen zwischen den gesuchten und den gefundenen Attributwerten berechnet. Dazu werden Ähnlichkeitsalgorithmen wie z.B. die Damerau-Levenshtein-Distanz eingesetzt. Die Einzelscores der Attribute werden zu einem Gesamtscore über einen Gewichtungsalgorithmus zusammengeführt. Das Attribut *score* enthält dabei nur Werte zwischen 0 und 1, wobei 0 absolut keine Übereinstimmung und 1 totale Übereinstimmung bedeutet.

Im Gewichtungsalgorithmus wird folgende Klassifizierung des Scores vorgenommen:

- $score > 0.99$: Exakter Treffer
Jedes angegebene Attribut hat eine exakte Übereinstimmung, d.h. die Werte sind identisch. Das Ergebnis kann als identisch zur Suchanfrage betrachtet werden.
- $0.95 < score \leq 0.99$: Sicherer Treffer
Jedes angegebene Attribut hat mindestens eine sichere Übereinstimmung. Bei einer sicheren Übereinstimmung darf der Attributwert minimal abweichen (z.B. Zusätze,

Abkürzungen, Umlaute, einzelne Buchstabendreher, ...). Es handelt sich bei dem Ergebnis um das gesuchte Objekt. Allerdings gibt es minimale Abweichungen die korrigiert werden müssen.

- $0.9 < \text{score} \leq 0.95$: Gutes Ergebnis
Jedes Attribut hat mindestens eine gute Übereinstimmung. Bei einer guten Übereinstimmung gibt es eine Abweichung der Attributwerte, es könnte sich aber um eine Übereinstimmung handeln (z.B. Rechtschreibfehler). Das Ergebnis könnte zur Suchanfrage passen. Die Abweichungen sollten aber kontrolliert werden.
- $\text{score} \leq 0.9$: Schlechtes Ergebnis
Mindestens ein Attribut hat eine schlechte Übereinstimmung. Bei einer schlechten Übereinstimmung gibt es eine deutliche Abweichung. Das Ergebnis passt vermutlich nicht zur Suchanfrage.

6.1.4 Auswertung der Ergebnisse

Die Bewertung im Attribut *score* liefert die Aussage, inwieweit das gefundene Objekt mit der Suchanfrage übereinstimmt. Für strukturierte Suchanfragen nach Hausadressen findet zusätzlich eine Einstufung des Ergebnisses im Attribut *treffer* statt. Dieses Attribut sagt aus, ob es sich bei dem Ergebnis um einen eindeutigen Treffer (T), mehrdeutigen Treffer (M), ein Ergebnis mit geringer Übereinstimmung (F) oder einen anderen Objekttyp (leer) handelt.

Für andere Suchanfragen sollten zur Einschätzung, wie *gut* das Ergebnis wirklich ist, noch folgende Aspekte ausgewertet werden:

- **Mehrdeutigkeit**

Eine Suche kann mehrere Ergebnisse mit gleichwertigem *score* liefern. Ein Beispiel bilden die Anfragen nach *Frankfurt am Main* und *Frankfurt (Oder)*. Je vollständiger und individueller eine Adressangabe ist, umso geringer ist die Wahrscheinlichkeit für solche Mehrdeutigkeiten. Um diese zu erkennen, ist die sortierte Ergebnisliste auszuwerten.

- **Typabgleich**

Wird nicht für alle übergebenen Attributwerte eine Übereinstimmung gefunden, kann als Ergebnis gegebenenfalls ein übergeordnetes Objekt zurückgegeben werden. Das Attribut *typ* ist also in jedem Fall zu kontrollieren, um das Ergebnis korrekt zu interpretieren. Beispielsweise könnte bei einer vollständigen Adresssuche (Ergebnis *typ*=Haus wird erwartet) mit stark fehlerhafter Straßenangabe allein der Ort (*typ*=Ort) als sicherer Treffer ($\text{score} > 0.95$) gefunden werden.

6.1.5 Fehlertoleranz

Die Suche ist mehrstufig fehlertolerant. Wird in einer Stufe nichts gefunden oder treten bei relativer Bewertung nur schlechte Bewertungen auf ($\text{score} < 0,94$), wird sukzessive die Fehlertoleranz erhöht.

Zunächst müssen nicht alle gesuchten Terme in einem Treffer vorkommen. Weiterhin darf auch die Reihenfolge der Wörter abweichen. Beispielsweise liefern die Sucheingaben „*Karl-Rothe-Straße, 04105 Leipzig*“ und „*04105 leipzig karl rothe straße*“ identische Ergebnisse

In weiteren Suchschritten wird eine Ähnlichkeitssuche (Fuzzy-Suche) für die Terme angewandt. Dadurch können z.B. Tippfehler korrigiert werden.

6.2 Räumliche Einschränkung der Anfragen

Um Anfragen gegen den Dienst regional einzuschränken stehen zwei Varianten zur Verfügung: Eine geometrische Einschränkung und eine attributive Einschränkung.

Für die geometrische Einschränkung können abhängig von der verwendeten Schnittstelle entsprechende Parameter der Suchanfrage übergeben werden. Bei der OpenSearch-Schnittstelle sind dies die Request-Parameter *bbox*, *lat + lon + distance* oder *geometry + relation*. In der WFS-Schnittstelle können die räumlichen Filteroperatoren *BBOX*, *Intersects*, *Disjoint*, *Contains*, *Within* oder *DWithin* verwendet werden. Allerdings sei angemerkt, dass die Verwendung von komplexen polygonalen Geometrien zur Einschränkung der Suchanfrage negative Auswirkungen auf die Performance hat.

Eine Alternative ist die attributive Einschränkung. Hierfür können die Attribute *ags*, *rs*, *bundesland*, *regbezirk*, *kreis*, *verwgem* oder *gemeinde* verwendet werden. In der OpenSearch-Schnittstelle kann dazu der Request-Parameter *filter* eingesetzt werden, um die Suchanfrage attributiv einzuschränken. Beispiele hierfür wären:

- Suche auf Regierungsbezirk Mittelfranken einschränken: `regbezirk:Mittelfranken`
- Suche auf Bundesland Sachsen-Anhalt einschränken: `bundesland:"Sachsen-Anhalt"`
- Suche per AGS auf Landkreis Passau (09275) einschränken: `ags:09275*`

Die oben genannten Beispiele wurden in der Query-Syntax angegeben und müssen ggf. noch für die Verwendung im Request-Parameter *filter* URL enkodiert werden. Ein vollständiges Beispiel für eine attributiv eingeschränkte Suche inkl. Fehlerkorrektur sieht wie folgt aus:

- URL ohne URL-Enkodierung:
`https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch?query=strasse:Munsterstraße+haus:11+ort:Münster&filter=bundesland:Bayern`
- URL mit URL-Enkodierung:
`https://sg.geodatenzentrum.de/gdz_geokodierung/geosearch?query=strasse%3AMunsterstra%C3%9Fe+haus%3A11+ort%3AM%C3%BCnster&filter=bundesland%3ABayern`

Die attributive Filterung lässt sich sowohl für die Ortssuche als auch für die Vorschlagssuche einsetzen und liefert bei der OpenSearch-Schnittstelle eine strenge Filterung.

Innerhalb der WFS-Schnittstelle lässt sich die attributive Einschränkung über eine Erweiterung des Filter-Ausdrucks verwenden. Die eigentliche Suchanfrage wird innerhalb eines logischen And-Operators mit der attributiven Einschränkung verwendet. Allerdings ist hierbei anzumerken, dass fehlerhafte oder unscharfe Anfragen aufgrund der gezielt implementierten Fehlertoleranz nicht immer streng auf das gewünschte räumliche Gebiet eingeschränkt sind. Für eine konsequente Einschränkung empfehlen wir deshalb die Verwendung der OpenSearch-Schnittstelle.

6.3 Angabe des Koordinatenreferenzsystems

Die Angabe des (Ziel-)Koordinatensystems erfolgt über EPSG-Codes und ist optional. Dies kann in einer der folgenden Formen erfolgen:

- (1) EPSG:xxxx
- (2) <https://www.opengis.net/gml/srs/epsg.xml#xxxx>
- (3) urn:x-ogc:def:crs:EPSG:xxxx

Fehlen diese Angaben werden Koordinatenangaben in EPSG:4326 (WGS 84) erwartet bzw. auch zurückgegeben.

Die Koordinatenreihenfolge wird von vielen Geoinformationssystemen wie folgt interpretiert:

Länge – Breite bzw. Rechtswert – Hochwert (A)

Im EPSG-System ist für alle geographischen und einige metrische Koordinatensysteme allerdings die umgekehrte traditionelle geodätische Interpretation für die Reihenfolge der Achsen des Koordinatensystems festgelegt:

Breite – Länge bzw. Hochwert – Rechtswert (B)

Um diesen widersprüchlichen Ansätzen und allgemeinen Nutzererwartungen möglichst gut gerecht zu werden, haben wir folgenden Kompromiss gewählt:

- SRS-Angaben der Form (1) werden in der Reihenfolge (A) interpretiert.
- SRS-Angaben der Form (2) und (3) werden in der Reihenfolge (B) interpretiert, sofern dies im EPSG-System so definiert ist.

6.4 Hinweise zu Koordinaten

Die Referenzdaten des Dienstes werden aus Daten, welche in der Georeferenzierung UTM-Abbildung in Zone 32 vorliegen, aufbereitet. Für den Dienst müssen die Geometrien der Daten intern in Geographischen Koordinate (GEO84) vorgehalten werden. Eine Rückgabe der Geometrien in anderen Georeferenzierungen erfolgt durch dynamische Transformation innerhalb des Dienstes. Im Vergleich zu den Originaldaten können diese Transformationen Systembedingt durch den Einsatz von Gleitkommazahlen zu weiteren Nachkommastellen führen.

7 Datenschutz

Der Geokodierungsdienst stellt nur reine Adressangaben und Geonamen ohne jeden Personenbezug bereit und kann nur entsprechende Suchanfragen ohne jeden Personenbezug verarbeiten.

Der Geokodierungsdienst und ebenso die vorgelagerte IT-Infrastruktur des BKG speichern grundsätzlich keine an ihn gerichteten Anfragen mit ihren semantischen Parametern, auch nicht temporär. Die Suchparameter und die Ergebnisse der Suche werden nur flüchtig verarbeitet, um die Antwort an den Client zu senden. So sind auch hier alle Anforderungen an den Datenschutz erfüllt.

Um dieses Datenschutzbedürfnis vieler Nutzer erfüllen zu können, wurde ausschließlich eine synchrone Geokodierung einzelner Adressen ohne jede Zwischenspeicherung implementiert. Eine Massengeokodierung muss deshalb auf Basis von (ggf. parallelen) Einzelanfragen verwirklicht werden. Eine asynchrone Massengeokodierung mit dem Erfordernis einer Zwischenspeicherung auf Serverseite wurde verworfen und nicht realisiert.

In der Sicherheitsarchitektur des Dienstleistungszentrums werden die kostenrelevanten Anfragen bei einem nutzungsbezogenen Tarif mit minimalen Angaben geloggt. Die oben genannten Suchparameter gehören nicht dazu. Geloggt werden technische Angaben, die den kostenrelevanten Zugriff des Nutzers belegen: Zeit, IP-Adresse, Nutzeridentifikator beim BKG, Name des aufgerufenen Dienstes, Anzahl der bereitgestellten Objekte oder die bereitgestellte Fläche bei flächenbezogener Abrechnung von Diensten.

8 Anlagen

8.1 JSON Schema Ortsangabe

```
{
  "$schema": "https://json-schema.org/draft-04/schema#",
  // TODO: "id": "https://json-schema.org/geojson/geojson.json#",
  "title": "Geo JSON object",
  "description": "Schema für GeoJSON Antwort vom Geokodierungsdienst",
  "type": "object",
  "required": ["type"],
  "properties": {
    "crs": {"$ref": "https://json-schema.org/geojson/crs.json#"},
    "bbox": {"$ref": "https://json-schema.org/geojson/bbox.json#"}
  },
  "oneOf": [
    {"$ref": "#/definitions/featureCollection"}
  ],
  "definitions": {
    "feature": {
      "title": "Feature",
      "description": "Ein Geo JSON Feature für ein Ortsangabe-Objekt",
      "required": ["geometry", "properties"],
      "properties": {
        "type": {"enum": ["Feature"]},
        "geometry": {
          "description": "Punktgeometrie des geometrische Mittelpunktes",
          "oneOf": [
            {"type": "null"},
            {"$ref": "https://json-schema.org/geojson/geometry.json#"}
          ]
        },
        "properties": {
          "type": "object",
          "properties": {
            "text": {
              "description": "Volltext-Repräsentation des Objekts.",
              "type": "string",
              "required": true
            },
            "typ": {
              "description": "Klassifizierung des Objekts. (Haus, Straße, Ort, Geoname)",
              "type": "string",
              "required": true
            },
            "score": {
              "description": "Qualitative Bewertung des Objektes. Dieses Attribut ist abhängig von der Sucheingabe und wird nur in der Antwort zum Client übergeben.",
              "type": "number",
              "required": false
            },
            "bbox": {
              "description": "BBOX-Geometrie des Features. Wird zum Zoom auf passenden Kartenausschnitt benötigt.",
              "$ref": "https://json-schema.org/geojson/geometry.json#",
              "required": true
            }
          }
        }
      }
    }
  }
}
```


Geokodierungsdienst für Adressen und Geonamen

```
"ags": {
  "description": "Amtlicher Gemeindeschlüssel 8-stellig.",
  "type": "string",
  "required": false
  // TODO: pattern
},
"rs": {
  "description": "Regionalschlüssel 12-stellig.",
  "type": "string",
  "required": false
  // TODO: pattern
},
"schluessel": {
  "description": "Enthält Kombination aus Regional- (12 Stellen), Orts-teil- (4 Stellen) und Straßenschlüssel (5 Stellen).",
  "type": "string",
  "required": false
  // TODO: pattern
},
"bundesland": {
  "description": "Name des Bundeslandes der das Objekt enthält.",
  "type": "string",
  "required": false
},
"regbezirk": {
  "description": "Name des Regierungsbezirks der das Objekt enthält.",
  "type": "string",
  "required": false
},
"kreis": {
  "description": "Name des Kreises der das Objekt enthält.",
  "type": "string",
  "required": false
},
"verwgem": {
  "description": "Name des Verwaltungsgemeinschaft der das Objekt enthält.",
  "type": "string",
  "required": false
},
"gemeinde": {
  "description": "Name der Gemeinde die das Objekt enthält.",
  "type": "string",
  "required": false
},
"plz": {
  "description": "Postleitzahl der Adresse. Wird nur bei Objekten mit type=[ Haus | Strasse | Ort ] verwendet.",
  "type": "string",
  "required": false
},
"ort": {
  "description": "Ortsname der Adresse. Wird nur bei Objekten mit type=[ Haus | Strasse | Ort ] verwendet.",
  "type": "string",
  "required": false
},
"ortsteil": {
```

Geokodierungsdienst für Adressen und Geonamen

```
    "description": "Ortsteilname der Adresse. Wird nur bei Objekten mit type=[ Haus | Strasse | Ort ] verwendet.",
    "type": "string",
    "required": false
  },
  "strasse": {
    "description": "Straßenname der Adresse. Wird nur bei Objekten mit type=[ Haus | Strasse ] verwendet.",
    "type": "string",
    "required": false
  },
  "haus": {
    "description": "Hausnummer mit Hausnummernzusatz der Adresse. Wird nur bei Objekten mit type=Haus verwendet.",
    "type": "string",
    "required": false
  },
  "qualitaet": {
    "description": "Qualitätsangabe zur Hauskoordinate. [A|B|R]",
    "type": "string",
    "required": false
  },
  "treffer": {
    "description": " Enthält für Strukturierte Suchen mit relativer Bewertung für Objekte eine einfache Bewertung des Ergebnisses. [T|M|F|]",
    "type": "string",
    "required": false
  },
  "qkz": {
    "description": " Enthält für Strukturierte Suchen mit relativer Bewertung für Objekte ein Qualitätskennzeichen für Geokoordinaten und Gitterzellen ausgehend vom score und typ. [1|2|3|4|5|6|9]",
    "type": "integer",
    "required": false
  },
  "name": {
    "description": "Name des Objekts. Wird nur für Objekte mit type= Geoname verwendet.",
    "type": "string",
    "required": false
  },
  "kategorie": {
    "description": "Thematische Kategorie (Landschaft, Schule, ...) des Objekts.",
    "type": "string",
    "required": false
  },
  "score_plz": {
    "description": "Einzelgüte für plz.",
    "type": "string",
    "required": false
  },
  "score_ort": {
    "description": " Einzelgüte für ort.",
    "type": "string",
    "required": false
  },
  "score_ortsteil": {
    "description": "Einzelgüte für ortsteil.",
```

```
        "type": "string",
        "required": false
    },
    "score_strasse": {
        "description": "Einzelgüte für strasse.",
        "type": "string",
        "required": false
    },
    "score_haus": {
        "description": "Einzelgüte für haus.",
        "type": "string",
        "required": false
    }
}
},
"id": {
    "description": "Die eindeutige ID (Zeichenfolge) des Objekts zur
    Identi-fizierung im Dienst.",
    "type": "string"
}
},
"featureCollection": {
    "title": "FeatureCollection",
    "description": "A Geo JSON feature collection",
    "required": ["features"],
    "properties": {
        "type": {"enum": ["FeatureCollection"]},
        "features": {
            "type": "array",
            "items": {"$ref": "#/definitions/feature"}
        }
    }
}
}
}
```

8.2 XML Schema Ortsangabe

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:gdz="https://www.geodatenzentrum.de/location"
    xmlns:gml="https://www.opengis.net/gml"
    xmlns:xsd="https://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    targetNamespace="https://www.geodatenzentrum.de/location">

    <xsd:import namespace="https://www.opengis.net/gml" schemaLoca-
    tion="https://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>

    <!-- ===== -->
    <xsd:simpleType name="ortsTyp">
        <xsd:annotation>
            <xsd:documentation>
                Dient zur Klassifizierung von Ortsangaben. Die Klassifizierung kann
                zur Ermittlung des zugrunde liegenden Datenbestandes herangezogen werden
                und hat Auswirkungen auf die Verfügbarkeit optionaler Attribute.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:simpleType>

```

Geokodierungsdienst für Adressen und Geonamen

```
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="PLZ"/>
  <xsd:enumeration value="Ort"/>
  <xsd:enumeration value="Strasse"/>
  <xsd:enumeration value="Haus"/>
  <xsd:enumeration value="Geoname"/>
  <xsd:enumeration value="POI"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="agsTyp">
  <xsd:annotation>
    <xsd:documentation>
      Datentyp für amtliche Gemeindegchlüssel. Der Amtliche Gemeindeg-
      schlüssel (AGS) besteht aus insgesamt 8 Ziffern, die sich wie folgt zusam-
      mensetzen:
        1. - 2. Stelle      = Kennzahl des Bundeslandes
        3. Stelle          = Kennzahl des Regierungsbezirks
        4. - 5. Stelle      = Kennzahl des Kreises
        6. - 8. Stelle      = Gemeindekennzahl
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9]{8}"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="rsTyp">
  <xsd:annotation>
    <xsd:documentation>
      Datentyp für den Regionalschlüssel. Der Regionalschlüssel (RS) be-
      steht aus insgesamt 12 Ziffern. Im Vergleich zum AGS sind vor den letzten
      drei Ziffern vier Stellen zur Verschlüsselung der Gemeindeverbände
      eingefügt.
        1. - 2. Stelle      = Kennzahl des Bundeslandes
        3. Stelle          = Kennzahl des Regierungsbezirks
        4. - 5. Stelle      = Kennzahl des Kreises
        6. - 9. Stelle      = Verbandsschlüssel
        10. - 12. Stelle    = Gemeindekennzahl
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9]{12}"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="schluesselTyp">
  <xsd:annotation>
    <xsd:documentation>
      Enthält Kombination aus Regional- (12 Stellen), Orts-teil- (4 Stel-
      len) und Straßenschlüssel (5 Stellen).
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9]{12,17}"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="plzTyp">
  <xsd:annotation>
```

Geokodierungsdienst für Adressen und Geonamen

```
<xsd:documentation>Datentyp für Postleitzahlen.</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
  <xsd:pattern value="([0-9]{5}|[0-9]{1,4}\*)" />
</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="OrtsangabeTyp">
  <xsd:annotation>
    <xsd:documentation>
      Ortsangaben dienen zur Kodierung von Objekten aus den Datenbeständen Georeferenzierte Adressen (GA), Geographische Namen Deutschland (GN-DE) und
      Points-of-Interest (POI). Über die Attribute können detaillierte Informationen zu den Objekten ermittelt werden. Das Vorkommen einiger Attribute
      ist abhängig von der Klassifizierung im Attribut typ.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="1" name="text"
nillable="false" type="xsd:string" />
        <xsd:element maxOccurs="1" minOccurs="1" name="typ"
nillable="false" type="gdz:ortsTyp" />
        <xsd:element maxOccurs="1" minOccurs="1" name="score"
nillable="true" type="xsd:float" />
        <xsd:element maxOccurs="1" minOccurs="1" name="geometry"
nillable="true" type="gml:PointPropertyType" />
        <xsd:element maxOccurs="1" minOccurs="1" name="bbox"
nillable="true" type="gml:GeometryPropertyType" />

        <!-- Adressattribute -->
        <xsd:element maxOccurs="1" minOccurs="0" name="ags"
nillable="true" type="gdz:agsTyp" />
        <xsd:element maxOccurs="1" minOccurs="0" name="rs"
nillable="true" type="gdz:rsTyp" />
        <xsd:element maxOccurs="1" minOccurs="0" name="schluessel"
nillable="true" type="gdz:schluesselTyp" />

        <xsd:element maxOccurs="1" minOccurs="0" name="bundesland"
nillable="true" type="xsd:string" />
        <xsd:element maxOccurs="1" minOccurs="0" name="regbezirk"
nillable="true" type="xsd:string" />
        <xsd:element maxOccurs="1" minOccurs="0" name="kreis"
nillable="true" type="xsd:string" />
        <xsd:element maxOccurs="1" minOccurs="0" name="verwgem"
nillable="true" type="xsd:string" />
        <xsd:element maxOccurs="1" minOccurs="0" name="gemeinde"
nillable="true" type="xsd:string" />

        <xsd:element maxOccurs="1" minOccurs="0" name="plz"
nillable="true" type="gdz:plzTyp" />
        <xsd:element maxOccurs="1" minOccurs="0" name="ort"
nillable="true" type="xsd:string" />
        <xsd:element maxOccurs="1" minOccurs="0" name="ortsteil"
nillable="true" type="xsd:string" />
        <xsd:element maxOccurs="1" minOccurs="0" name="strasse"
nillable="true" type="xsd:string" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Geokodierungsdienst für Adressen und Geonamen

```
<xsd:element maxOccurs="1" minOccurs="0" name="haus"
nillable="true" type="xsd:string"/>

    <xsd:element maxOccurs="1" minOccurs="0" name="qualitaet"
nillable="true" type="xsd:string"/>

    <!-- GN250 Attribute -->
    <xsd:element maxOccurs="1" minOccurs="0" name="name"
nillable="true" type="xsd:string"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="kategorie"
nillable="true" type="xsd:string"/>

    <!-- Bewertungsattribute -->
    <xsd:element maxOccurs="1" minOccurs="0" name="treffer"
nillable="true" type="xsd:string"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="qkz"
nillable="true" type="xsd:integer"/>

    <xsd:element maxOccurs="1" minOccurs="0" name="score_text"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_typ"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_geometry"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_bbox"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_agrs"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_rs"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_schluessel"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_bundesland"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_regbezirk"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_kreis"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_verwgem"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_gemeinde"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_plz"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_ort"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_ortsteil"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_strasse"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_haus"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_qualitaet"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_name"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0" name="score_kategorie"
nillable="true" type="xsd:float"/>
    <xsd:element maxOccurs="1" minOccurs="0"
name="score_strasse_haus" nillable="true" type="xsd:float"/>
```

Geokodierungsdienst für Adressen und Geonamen

```
        <xsd:element maxOccurs="1" minOccurs="0" name="score_treffer"
nillable="true" type="xsd:float"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

  <xsd:element name="Ortsangabe" substitutionGroup="gml:_Feature"
type="gdz:OrtsangabeTyp"/>

<!-- ===== -->

<xsd:complexType name="VorschlagTyp">
  <xsd:annotation>
    <xsd:documentation>
      Einfache Repräsentation von Objekten aus den Datenbeständen Geore-
ferenzierte Adressen (GA) und Geographische Namen Deutschland (GN250) zur
Generierung von Vorschlagslisten.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="1" name="text"
nillable="false" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="1" name="score"
nillable="false" type="xsd:float"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>
```

9 Kontaktdaten

Bundesamt für Kartographie und Geodäsie
Referat GDL2 | Dienstleistungszentrum des Bundes für Geoinformation und Geodäsie (DLZ)
| Zentrale Stelle Geotopographie (ZSGT)
Karl-Rothe-Straße 10-14
D-04105 Leipzig

Tel.: +49(0)341 5634-333

Fax: +49(0)341 5634-415

E-Mail: dlz@bkg.bund.de

Weitere Informationen finden Sie auf unserer Homepage www.bkg.bund.de unter der Rubrik „Produkte und Services“.