



Geoproxy Freistaat Thüringen

Web Feature Service - Gazetteer

Räumliche Suche

- Beispiele zur Nutzung -

Stand: 24.04.2015

Dokumentenhistorie

Version	Datum	Bemerkungen
1.0	24.04.2015	basierend auf WFS 1.1.0

Inhaltsverzeichnis

1.	Einleitung.....	3
2.	Einsatzmöglichkeiten und Grenzen	3
3.	Anfrage nach WFS-Fähigkeit = GetCapabilities	5
4.	Anfrage nach Beschreibung für ein FeatureType = DescribeFeatureType	5
5.	Hierarchische Anfrage nach postalischen Adressen	7
	5.1. Einfache Anfrage nach Gemeinden (ohne Operator).....	7
	5.2. Hierarchische Anfrage nach einer konkreten Gemeinde (EqualTo).....	9
	5.3. Hierarchische Anfrage nach Gemeinden (Like)	10
	5.4. Hierarchische Anfrage nach allen Straßen einer Gemeinde (EqualTo).....	11
	5.5. Hierarchische Anfrage nach einer konkreten Straße (And / EqualTo).....	12
	5.6. Hierarchische Anfrage nach allen Hausnummern der Straße (EqualTo).....	13
	5.7. Hierarchische Anfrage nach einer konkreten Hausnr (And / EqualTo)	14
	5.8. Weitere Tipps zur sinnvollen Nutzung der Ergebnisse	15
6.	Spezialfall – Abfrage über BBOX oder Polygon	16
7.	Anhang – Konvertierung von Sonderzeichen	17

1. Einleitung

Mittels des **Web Feature Service** (WFS) kann man webbasiert auf die vektoriellen Geodaten des Geoproxy zugreifen. Ein Gazetteer-Service (**WFS-G**) ist eine spezielle Ausprägung eines WFS, der Dienst kann eigenständig oder innerhalb eines „normalen“ WFS nutzbar sein und er dient zur Ortssuche.

Im Geoproxy wurden die Datenstrukturen für den WFS-G auf Grundlage der Spezifikation „Gazetteer Service Profile of the Web Feature Service Implementation Specification“ V0.9.1 aufgebaut, diese Spezifikation wird von der Standard Working Group (SWG) WFS Gazetteer Profile 1.0 des Open Geospatial Consortiums (OGC) weiterentwickelt.

Die Geodaten, welche über einen WFS nutzbar sind, werden allgemein als **Features** bezeichnet. Gleichartig modellierte Geodaten (z.B. Gemeinden) sind in einem **Feature Type** gespeichert und können aus Geometrie- und Attributinformationen bestehen. Ein konkretes Element eines Feature Types (z.B. die Gemeinde Musterdorf) wird als **Feature Instance** bezeichnet.

Im Rahmen der **OGC**- Spezifikationen (Open Geospatial Consortium) werden die Antworten des Dienstes als unabhängiges Dateiformat **GML** (Geography Markup Language) zurückgeliefert, dabei wird auf das Hypertext Transfer Protocol (**HTTP**) zurückgegriffen.

Der Ablauf einer Kommunikation zwischen Nutzer (**Client**) und Anbieter (**Server**) über den Dienst (WFS) verläuft wie folgt und wird in den folgenden Kapiteln näher erläutert.

- **GetCapabilities** Request - Der Nutzer fragt nach den Fähigkeiten des WFS und den im Dienst angebotenen Feature Types. Als Antwort liefert der Dienst ein XML-Dokument mit allgemeinen Angaben zum Anbieter, den Feature Types sowie den möglichen Operationen.
- **DescribeFeatureType** Request - Im zweiten Schritt fragt der Nutzer nach der Beschreibung eines konkreten Feature Types, um Informationen über die Struktur zu erhalten.
- **GetFeature** Request - In den folgenden Schritten werden gezielte Anfragen nach einzelnen oder mehreren Elementen (Feature Instances) gestellt, hierbei werden vom Dienst die eigentlichen Geodaten zurückgeliefert. Dabei kann die Suche und Auswahl über räumliche und/oder vergleichende Operatoren entsprechend detailliert werden.

2. Einsatzmöglichkeiten und Grenzen

Während es für einen „normalen“ WFS bereits diverse Lösungen für CAD/GIS-Clients gibt, wird man bei der angedachten Nutzung eines WFS-G meist auf eine programmtechnische Anpassung setzen. Es kann sehr sinnvoll sein, dass man die nachfolgend beschriebenen Suchmöglichkeiten in eine grafische Oberfläche einbettet, mit deren Hilfe der Nutzer schneller zur gesuchten Position findet.

Nachfolgend werden exemplarische Hinweise zur generischen Nutzung eines WFS-G gegeben, die bspw. im Rahmen von Testung, Vergleich zur eingesetzten Softwarelösung oder eigenständiger Softwareentwicklung genutzt werden können. Die verwendeten real vorkommenden Daten zur Suche nach Adressen und Flurstücken sind willkürlich gewählt und stammen aus dem Bereich der Geobasisdaten des TLVermGeo.

Problematisch für die Verwendung von HTTP Get kann die Länge der Anfrage sein, daher sollten die Anfragen in der Regel über HTTP Post laufen, dies betrifft speziell die konkreten Abfragen nach Features.

- Anfragen über **HTTP Get** können direkt im *Browser* abgesetzt werden
- Anfragen über **HTTP Post** sollten im Rahmen der Testung z.B. über den *generic OGC WebService client* des Geoproxy abgesetzt werden,
URL: <http://www.geoproxy.geoport-th.de/geoproxy/client/client.html>

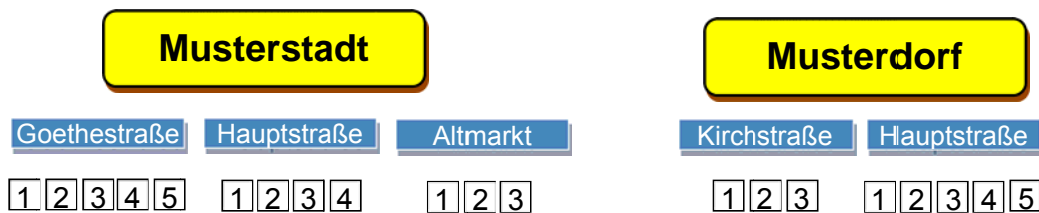
Ein WFS-G kann also zur punktuellen Ortssuche eingesetzt werden, die Möglichkeiten zum sofortigen Einsatz sind jedoch begrenzt. Es ist bspw. nicht möglich, ad hoc eine Liste von Adressen an den Dienst zu liefern und als Ergebnis eine Liste von Koordinaten zu erwarten oder eine „freihändige“ Suche durchzuführen (Beispiel: es werden alle bekannten Begriffe zum Ort eingegeben und der Dienst liefert die

am besten passenden Ergebnisse). Bei Tippfehlern oder einer „unscharfen Suche“ liefert der WFS-G entweder falsche oder keine Ergebnisse.



Am besten ist der Dienst für eine **hierarchische Suche** geeignet. Eine direkte Suche und eine „freihändige“ Suche (wegen möglicher Tippfehler nur im begrenzten Umfang) sind ebenfalls möglich, jedoch müssen als Zwischenschritt diverse Informationen vor der Ergebnisanzeige vom Dienst bezogen und weiterverarbeitet werden.

Folgendes Beispiel für eine Suche nach postalischen Adressen soll die Möglichkeiten zum Einsatz des WFS-G darlegen. Üblicherweise gehören die Hausnummern zu einer Straße und die Straßen zu einer Gemeinde – somit bestehen zwischen diesen hierarchischen Ebenen direkte Verbindungen. Im Beispiel wird weiterhin deutlich, dass es u.U. zu Mehrdeutigkeiten kommen kann: Die Hausnummern 1 bis 4 in der Hauptstraße kommen mehrfach vor – ohne eine Zuordnung zur Gemeinde ist eine eindeutige Ergebnisanzeige nicht möglich!



I) Bei einer **hierarchischen Suche** geht man sinnvollerweise vom Großen ins Kleine:

- (1) vom Nutzer wird die Eingabe der Gemeinde verlangt
- (2) der Dienst liefert die Treffer und der Nutzer wählt bei Mehrfachtreffer das richtige Ergebnis aus
- (3) mit dem Ergebnis der richtigen Gemeinde sucht der Dienst nun alle Straßen der Gemeinde
- (4) der Dienst liefert die Treffer und der Nutzer wählt bei Mehrfachtreffer das richtige Ergebnis aus
- (5) mit dem Ergebnis der richtigen Straße sucht der Dienst nun alle Hausnummern der Straße
- (6) der Dienst liefert die Treffer und der Nutzer wählt bei Mehrfachtreffer das richtige Ergebnis aus
- (7) mit dem Ergebnis der richtigen Hausnummer liefert die Koordinaten für eine Positionierung



Programmtechnisch lässt sich dies zum Beispiel mit einem Pull-Down-Menü umsetzen: zuerst werden alle Gemeinden bezogen (entweder erfolgt die Abfrage der ersten Hierarchieebene schon vor der Nutzereingabe und die Ergebnisse werden mit der Nutzereingabe abgeglichen oder die Nutzereingabe wird zur erstmaligen Suche über die erste Hierarchiestufe verwendet)

Gemeinde (Landkreis):	Magdala (Weimarer Land)
Straße:	Am Linderbach
Haus-Nr.:	3

Werden die Hierarchieebenen entsprechend nacheinander komplett abgefragt und dem Nutzer angezeigt, so dass dieser direkt aus den Ergebnissen auswählen kann, so können fehlerhafte Ergebnisse aufgrund von Falscheingaben weitgehend minimiert werden. Es ist zu beachten, dass zum Beispiel bei großen Orten oder auch bei Flurstücken die Listen sehr lang werden können.

II) Bei einer **direkten Suche** gibt man dem Nutzer sinnvollerweise eine (hierarchisch strukturierte und am Datenmodell orientierte) Eingabemaske vor, in die der Nutzer alle bekannten Daten eingeben kann.

Nach erfolgter Eingabe sollten die Daten vom GUI so an den WFS-G übergeben werden, dass die unter I) beschriebene Routine durchlaufen werden kann. Im Ergebnis sollten die korrekten Koordinaten für eine Positionierung übergeben werden.

III) Bei einer „**freihändigen**“ **Suche** kann der Nutzer alle ihm bekannten Daten meist in einer Zeile eingeben.

Magdala Am Linderbach 3

Nach erfolgter Eingabe sollten die Daten vom GUI so an den WFS-G übergeben werden, dass die unter I) beschriebene Routine durchlaufen werden kann. Im Ergebnis sollten die korrekten Koordinaten für eine Positionierung übergeben werden.

Fazit: Nach bisherigen Erfahrungen lässt sich die hierarchische Suche I) am besten umsetzen, da hier auf den vom Dienst gelieferten Ergebnissen hierarchisch und direkt aufgebaut werden kann. Die direkte und freihändige Suche sind mit mehr Aufwand umsetzbar, jedoch kann sich hier die Fehleranfälligkeit aufgrund von Fehleingaben erhöhen.

3. Anfrage nach WFS-Fähigkeit = GetCapabilities

Anfrage, welche üblicherweise zu Beginn einer Sitzung gestellt wird.

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
REQUEST=GetCapabilities&VERSION=1.1.0&SERVICE=WFS
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetCapabilities service="WFS"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd"/>
```

HTTP POST

→ Antwort: Es wird das Capabilities-Dokument für den WFS geliefert.

Folgende WFS FeatureTypes des TLVermGeo sind Teil des Gazetteers (WFS-G):

FeatureType	Beschreibung
tlvermgeo:GAZHKO_COMMUNE_FREE	Gemeinden
tlvermgeo:GAZHKO_STREET	Straßen
tlvermgeo:GAZHKO_HNR	Hausnummern

Suche nach postalischen Adressen

FeatureType	Beschreibung
tlvermgeo:gaz_alk_kb	Katasteramtsbezirke
tlvermgeo:gaz_alk_gem	Gemarkungen
tlvermgeo:gaz_alk_flur	Fluren
tlvermgeo:gaz_alk_flst	Flurstücke

Suche nach Flurstücken

Der Namespace der FeatureTypes ist:

```
tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
```

4. Anfrage nach Beschreibung für ein FeatureType = DescribeFeatureType

Anfrage, welche Informationen über die Struktur (Geometrie und Sachdaten) der Service zu bestimmten FeatureTypes liefern kann. Wird in der Regel nach GetCapabilities gestellt und dient als Informationsquelle für die nachfolgenden Anfragen.

```
http://www.geoproxy.geoportal-
th.de/geoproxy/services?SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&T
YPENAME=tlvermgeo:GAZHKO_COMMUNE_FREE&NAMESPACE=xmlns(tlvermgeo=http://www.geop
roxy.geoportal-th.de/tlvermgeo)
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:DescribeFeatureType version="1.1.0" service="WFS"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:TypeName>tlvermgeo:GAZHKO_COMMUNE_FREE</wfs:TypeName>
</wfs:DescribeFeatureType>
```

HTTP POST

→ Antwort: Es wird das Schema für den FeatureType geliefert.

```
<?xml version='1.0' encoding='UTF-8'?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" ... targetNamespace="http://www.geoproxy.geoportal-th.de/tlvermgeo" ...>
<import namespace="http://www.opengis.net/gml" schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<element name="GAZHKO_COMMUNE_FREE" substitutionGroup="gml:_Feature">
<complexType>
<complexContent>
<extension base="gml:AbstractFeatureType">
<sequence>
<element name="gemkennzahl" minOccurs="0" type="decimal"/>
<element name="kskennzahl" minOccurs="0" type="string"/>
<element name="gemeindenname" minOccurs="0" type="string"/>
<element name="geom" minOccurs="0" type="gml:GeometryPropertyType"/>
</sequence>
</extension>
</complexContent>
</complexType>
</element>
</schema>
```

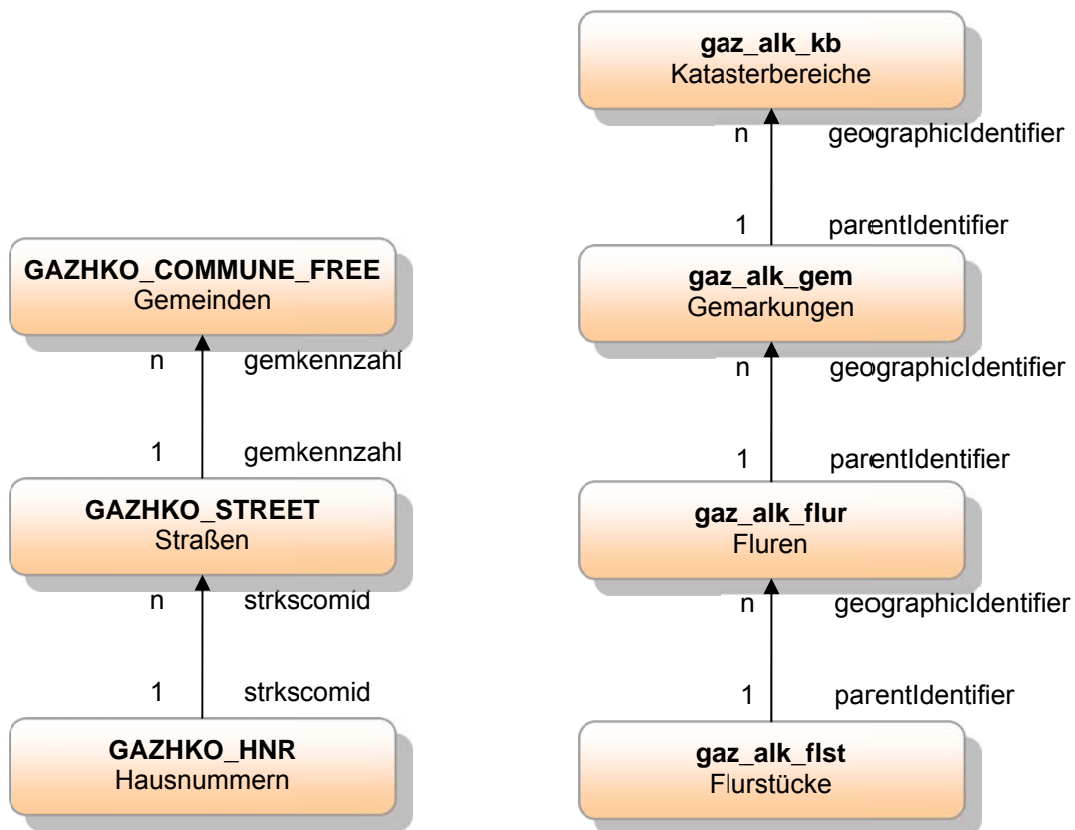


Für die programmtechnische Umsetzung ist es ratsam, dass zunächst für alle benötigten FeatureTypes die DescribeFeatureType-Abfrage erfolgt. Anhand der Angaben erschließt sich das Datenmodell für die jeweilige räumliche Suche.



Änderungen am Datenmodell (Attribute, Bezeichnungen, Datentypen, Namespaces) müssen im Regelfall auf Seiten der Nutzer programmtechnisch nachgepflegt werden. Es empfiehlt sich daher, mit Variablen oder Mapping zu arbeiten, damit der Änderungsaufwand überschaubar bleibt.

Die in Kapitel 2 genannten direkten Verbindungen werden hier dargestellt: die FeatureTypes sind hierarchisch miteinander verknüpft und jedes FeatureType „kennt“ über das Attribut `parentIdentifier` (Flurstückssuche) bzw. die Attribute `gemkennzahl` und `strkscomid` (Adressuche) das jeweils übergeordnete FeatureType per Verweis auf dessen Attribut `geographicIdentifier` (Flurstückssuche) bzw. die Attribute `gemkennzahl` und `strkscomid` (Adressuche)



Suche nach postalischen Adressen

Suche nach Flurstücken

5.-6.) Hinweise für konkrete Anfragen nach Features = GetFeature

- Die folgenden Anfragen stellen häufig verwendete Anwendungsfälle dar, die Möglichkeiten richten sich nach den Angaben aus den GetCapabilities.
- Im Schema deaktivierte Attribute werden nicht übertragen.
- Für detaillierte Anfragen kann das Filter Encoding genutzt werden, die Möglichkeiten richten sich nach den Angaben aus den GetCapabilities. Näheres zu den Filtern entnehmen Sie bitte der OGC Spezifikation „Filter Encoding Implementation Specification“.
- Anfragen sollten möglichst über HTTP Post gestellt werden.

5. Hierarchische Anfrage nach postalischen Adressen

Zur Suche werden hier im Regelfall keine räumliche Operatoren sondern Vergleichsoperatoren eingesetzt, diese sind in den Eigenschaften des Dienstes (Abruf siehe unter 3.) angegeben. Diese können auf die Attribute des jeweiligen Feature Types (Abruf siehe unter 4.) angewendet werden.

In diesem Beispiel soll die Adresse **Am Linderbach 3** in **Magdala** gesucht werden.

5.1. Einfache Anfrage nach Gemeinden (ohne Operator)

Dieses Beispiel kann **ohne Vergleichsoperatoren** genutzt werden, wenn zu Beginn der räumlichen Suche dem Nutzer bereits alle Gemeinden vorgegeben werden sollen.

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=tlvermgeo:GAZHKO_COMMUNE_
FREE&NAMESPACE=xmlns(tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo)
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:Query typeName="tlvermgeo:GAZHKO_COMMUNE_FREE">
</wfs:Query>
</wfs:GetFeature>
```

HTTP POST

Antwort: Informationen über 849 Features werden übertragen

```
<?xml version="1.0" encoding="UTF-8"?><wfs:FeatureCollection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/gml"
xsi:schemaLocation="http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.1.0/wfs.xsd
http://www.geoproxy.geoportal-th.de/tlvermgeo http://www.geoproxy.geoportal-
th.de/geoproxy/services?SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&OUTPUTF
ORMAT=text%2Fxml%3B+subtype%3Dgml%2F3.1.1&TYPENAME=tlvermgeo:GAZHKO_COMMUNE_FREE&N
AMESPACE=xmlns(tlvermgeo=http%3A%2F%2Fwww.geoproxy.geoportal-th.de%2Ftlvermgeo)" timeStamp="2015-04-
23T15:29:55Z" numberOfFeatures="849">
-----
<gml:boundedBy>
  <gml:Envelope srsName="EPSG:25832">
    <gml:lowerCorner>562498.968 5564438.004</gml:lowerCorner>
    <gml:upperCorner>755872.818 5721994.116</gml:upperCorner>
  </gml:Envelope>
</gml:boundedBy>
<gml:featureMember>
  <tlvermgeo:GAZHKO_COMMUNE_FREE xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
gml:id="ID_TLVERMGEO_GAZHKO_COMMUNE_FREE.77052">
  <tlvermgeo:gemkennzahl>77052</tlvermgeo:gemkennzahl>
  <tlvermgeo:kskennzahl>77</tlvermgeo:kskennzahl>
  <tlvermgeo:gemeindenname>Windischleuba</tlvermgeo:gemeindenname>
  <tlvermgeo:geom>
    <gml:Polygon gml:id="ID_TLVERMGEO_GAZHKO_COMMUNE_FREE.77052_TLVERMGEO_GEOM"
srsName="EPSG:25832">
      <gml:exterior>
        <gml:LinearRing srsName="EPSG:25832">
          <gml:posList>742246.117 5655235.795 742246.117 5659307.795 747530.913 5659307.795 747530.913
5655235.795 742246.117 5655235.795</gml:posList>
        </gml:LinearRing>
```

*„Kopfdaten“ der
FeatureCollection,
Zeitstempel
849 Features folgen*

*Envelope = umschließendes
Rechteck für alle Features*

*Das erste FeatureMember,
848 Features folgen*

Zu beachten ist, dass der Dienst die Koordinaten im **nativen Koordinatensystem** zurückliefert, in dem die Features in der Datenbank abgelegt sind (in den WFS Capabilities als <DefaultSRS> beschrieben).

Da die hier verwendeten FeatureTypes für die Suche nach postalischen Adressen und nach Flurstücken aus dem AAA® ALKIS-Modell abgeleitet worden sind, ergibt sich für die Daten das amtliche Lagebezugssystem ETRS89 UTM Zone 32 (EPSG:25832) als Standard-Bezugssystem für die Ausgabe.

Sollen die Daten in einem anderen Bezugssystem geliefert werden, so ist der SRS-Parameter mitzusenden. Im Beispiel sollen die Daten im Gauß-Krüger-System **EPSG:31468** geliefert werden.

```
http://www.geoproxy.geoportal-th.de/geoproxy/services? HTTP GET  
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=tlvermgeo:GAZHKO_COMMUNE_  
FREE&NAMESPACE=xmlns(tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo)  
&SRSNAME=EPSG:31468
```

```
<?xml version="1.0" encoding="UTF-8"?> HTTP POST  
<wfs:GetFeature version="1.1.0"  
xmlns:wfs="http://www.opengis.net/wfs"  
xmlns:gml="http://www.opengis.net/gml"  
xmlns:ogc="http://www.opengis.net/ogc"  
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://www.opengis.net/wfs  
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">  
<wfs:Query typeName="tlvermgeo:GAZHKO_COMMUNE_FREE" srsName="EPSG:31468">  
</wfs:Query>  
</wfs:GetFeature>
```

Antwort: Informationen über 849 Features werden im Gauß-Krüger-System übertragen



Es ist ratsam, den SRS-Parameter immer zu setzen, zum einen werden die Anfragen so robuster gestaltet und zum anderen kann der Nutzer - wenn der Parameter als Variable gesetzt und zum Nutzer durchgereicht wird - bei Bedarf zwischen verschiedenen Systemen wählen.

Neben der Angabe unter <DefaultSRS> können alle im Capabilities-Dokument unter <OtherSRS> angegebenen Bezugssysteme genutzt werden.

5.2. Hierarchische Anfrage nach einer konkreten Gemeinde (EqualTo)

In der Regel liegen dem Nutzer **konkrete** Angaben vor, so dass der `PropertyIsEqualTo` Operator für die **direkte** Suche nach dem Gemeindennamen **Magdala** eingesetzt werden kann.

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature
&TYPENAME=tlvermgeo:GAZHKO_COMMUNE_FREE
&NAMESPACE=xmlns%28tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo%29
&FILTER=<ogc:Filter xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:gemeindenname</ogc:PropertyName>
<ogc:Literal>Magdala</ogc:Literal>
</ogc:PropertyIsEqualTo></ogc:Filter>
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:Query typeName="tlvermgeo:GAZHKO_COMMUNE_FREE">
<ogc:Filter>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:gemeindenname</ogc:PropertyName>
<ogc:Literal>Magdala</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

HTTP POST

Antwort: Informationen von 1 Feature werden übertragen

```
<?xml version='1.0' encoding='UTF-8'?><wfs:FeatureCollection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/gml" xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd http://www.geoproxy.geoportal-th.de/tlvermgeo http://www.geoproxy.geoportal-
th.de/geoproxy/services?SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&OUTPUTFORMAT=text%2Fxml%3B+sub
type%3Dgml%2F3.1.1&TYPENAME=tlvermgeo:GAZHKO_COMMUNE_FREE&NAMESPACE=xmlns(tlvermgeo=http%3A%2F%2Fwww.geopr
oxy.geoportal-th.de%2Ftlvermgeo)" timeStamp="2015-04-23T15:42:24Z" numberOfFeatures="1">
<gml:boundedBy>
<gml:Envelope srsName="EPSG:25832">
<gml:lowerCorner>669395.308 5639180.746</gml:lowerCorner>
<gml:upperCorner>673439.007 5644294.181</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<gml:featureMember>
<tlvermgeo:GAZHKO_COMMUNE_FREE xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
gml:id="ID_TLVERMGEO_GAZHKO_COMMUNE_FREE.71053">
<tlvermgeo:gemkennzahl>71053</tlvermgeo:gemkennzahl>
<tlvermgeo:kskennzahl>71</tlvermgeo:kskennzahl>
<tlvermgeo:gemeindenname>Magdala</tlvermgeo:gemeindenname>
<tlvermgeo:geom>
<gml:Polygon gml:id="ID_TLVERMGEO_GAZHKO_COMMUNE_FREE.71053_TLVERMGEO_GEOM" srsName="EPSG:25832">
<gml:exterior>
<gml:LinearRing srsName="EPSG:25832">
<gml:posList>669395.308 5639180.746 669395.308 5644294.181 673439.007 5644294.181 673439.007 5639180.746 669395.308
5639180.746</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</tlvermgeo:geom>
</tlvermgeo:GAZHKO_COMMUNE_FREE>
</gml:featureMember>
</wfs:FeatureCollection>
```

5.3. Hierarchische Anfrage nach Gemeinden (Like)

Der zuvor beschriebene Fall eignet sich dann, wenn die gesuchten Begriffe eindeutig sind. Soll der Nutzer zum Beispiel nur die ersten Buchstaben (oder Zahlen) eines Begriffes eingeben, dann kann der Like Operator für die Suche nach dem Gemeindegamen **Magdala** eingesetzt werden.

Im Beispiel werden vom Nutzer nur zwei Buchstaben erwartet, dabei wird der Rest des Wortes durch das WildCard-Zeichen „%“ ergänzt.

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature
&TYPENAME=tlvermgeo:GAZHKO_COMMUNE_FREE
&NAMESPACE=xmlns%28tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo%29
&FILTER=<ogc:Filter xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<ogc:PropertyIsLike wildCard="% " singleChar="?" escape="/">
<ogc:PropertyName>tlvermgeo:gemeindenname</ogc:PropertyName>
<ogc:Literal>Ma%</ogc:Literal>
</ogc:PropertyIsLike></ogc:Filter>
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:Query typeName="tlvermgeo:GAZHKO_COMMUNE_FREE">
<ogc:Filter>
<ogc:PropertyIsLike wildCard="% " singleChar="?" escape="/">
<ogc:PropertyName>tlvermgeo:gemeindenname</ogc:PropertyName>
<ogc:Literal>Ma%</ogc:Literal>
</ogc:PropertyIsLike>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

HTTP POST

Antwort: Informationen von 10 Features werden übertragen



Wird der Suchbegriff verfeinert, also je mehr Buchstaben (oder Zahlen) des Suchbegriffes eingegeben werden (z.B. Magda% anstelle Ma%), desto weniger mögliche Treffer werden im Regelfall zurückgeliefert.

Beispiel Magda%: Informationen von 1 Feature werden übertragen (Auszug)

```
...
<gml:featureMember>
<tlvermgeo:GAZHKO_COMMUNE_FREE xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
gml:id="ID_TLVERMGE0_GAZHKO_COMMUNE_FREE.71053">
<tlvermgeo:gemkennzahl>71053</tlvermgeo:gemkennzahl>
<tlvermgeo:kskennzahl>71</tlvermgeo:kskennzahl>
<tlvermgeo:gemeindenname>Magdala</tlvermgeo:gemeindenname>
<tlvermgeo:geom>
<gml:Polygon gml:id="ID_TLVERMGE0_GAZHKO_COMMUNE_FREE.71053_TLVERMGE0_GEOM" srsName="EPSG:25832">
<gml:exterior>
<gml:LinearRing srsName="EPSG:25832">
<gml:posList>669395.308 5639180.746 669395.308 5644294.181 673439.007 5644294.181 673439.007 5639180.746 669395.308
5639180.746</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
...

```

5.4. Hierarchische Anfrage nach allen Straßen einer Gemeinde (EqualTo)

Aufgrund des zuvor erhaltenen Ergebnisses können mit dem als Verbindungselement fungierendem Wert **gemkennzahl 71053** die Straßen der Gemeinde **Magdala** als nächste Hierarchiestufe aufgerufen werden. Diese Abfrage bietet sich an, wenn dem Nutzer in der Auswahl alle Straßen einer Gemeinde präsentiert werden sollen. Dazu wird wieder der `PropertyIsEqualTo` Operator verwendet.

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature
&TYPENAME=tlvermgeo:GAZHKO_STREET
&NAMESPACE=xmlns%28tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo%29
&FILTER=<ogc:Filter xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:gemkennzahl</ogc:PropertyName>
<ogc:Literal>71053</ogc:Literal>
</ogc:PropertyIsEqualTo></ogc:Filter>
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:Query typeName="tlvermgeo:GAZHKO_STREET">
<ogc:Filter>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:gemkennzahl</ogc:PropertyName>
<ogc:Literal>71053</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

HTTP POST

Antwort: Informationen von 46 Features werden übertragen (Auszug)

```
...
<gml:featureMember>
  <tlvermgeo:GAZHKO_STREET xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
gml:id="ID_TLVERMGEO_GAZHKO_STREET.7105301043">
  <tlvermgeo:gemkennzahl>71053</tlvermgeo:gemkennzahl>
  <tlvermgeo:strassesch>01043</tlvermgeo:strassesch>
  <tlvermgeo:strkscomid>7105301043</tlvermgeo:strkscomid>
  <tlvermgeo:strasse>Am Linderbach</tlvermgeo:strasse>
  <tlvermgeo:geom>
  <!--Inlined geometry 'ID_TLVERMGEO_GAZHKO_STREET.7105301043_TLVERMGEO_GEOM'-->
  <gml:Polygon gml:id="ID_TLVERMGEO_GAZHKO_STREET.7105301043_TLVERMGEO_GEOM" srsName="EPSG:25832">
  <gml:exterior>
  <gml:LinearRing srsName="EPSG:25832">
  <gml:posList>671400.594 5642487.814 671400.594 5642506.981 671528.232 5642506.981 671528.232 5642487.814 671400.594
5642487.814</gml:posList>
  </gml:LinearRing>
  </gml:exterior>
  </gml:Polygon>
  </tlvermgeo:geom>
  <tlvermgeo:gemarknr>5362</tlvermgeo:gemarknr>
  <tlvermgeo:gemarkname>Magdala</tlvermgeo:gemarkname>
  </tlvermgeo:GAZHKO_STREET>
</gml:featureMember>
...
```

5.5. Hierarchische Anfrage nach einer konkreten Straße (And / EqualTo)

Da der Nutzer im Beispiel bereits über den konkreten Straßennamen **Am Linderbach** verfügt, kann die Anfrage konkretisiert werden, indem der Filter neben dem Wert **gemkennzahl 71053** für die Gemeinde **Magdala** auf den Straßennamen mit dem **And** Operator erweitert wird.



Auch wenn Leerzeichen, wie sie in Straßenbezeichnungen häufig vorkommen, im Regelfall problemlos verarbeitet werden, so ist es sinnvoll, alle Sonderzeichen durch die entsprechenden Äquivalente zu ersetzen. Im Beispiel wurde das Leerzeichen in http Get-Anfragen durch „%20“ und in http Post-Anfragen durch „ “ ersetzt. → siehe Kapitel zur Konvertierung am Ende!

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature
&TYPENAME=tlvermgeo:GAZHKO_STREET
&NAMESPACE=xmlns%28tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo%29
&FILTER=<ogc:Filter xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<ogc:And><ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:gemkennzahl</ogc:PropertyName>
<ogc:Literal>71053</ogc:Literal></ogc:PropertyIsEqualTo>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:strasse</ogc:PropertyName>
<ogc:Literal>Am%20Linderbach</ogc:Literal>
</ogc:PropertyIsEqualTo></ogc:And>
</ogc:Filter>
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:Query typeName="tlvermgeo:GAZHKO_STREET">
<ogc:Filter>
<ogc:And>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:gemkennzahl</ogc:PropertyName>
<ogc:Literal>71053</ogc:Literal>
</ogc:PropertyIsEqualTo>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:strasse</ogc:PropertyName>
<ogc:Literal>Am&#32;Linderbach</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

HTTP POST

Antwort: Informationen von 1 Feature werden übertragen (Auszug)

```
... <gml:featureMember>
<tlvermgeo:GAZHKO_STREET xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
gml:id="ID_TLVERMGE0_GAZHKO_STREET.7105301043">
<tlvermgeo:gemkennzahl>71053</tlvermgeo:gemkennzahl>
<tlvermgeo:strassesch>01043</tlvermgeo:strassesch>
<tlvermgeo:strkscomid>7105301043</tlvermgeo:strkscomid>
<tlvermgeo:strasse>Am Linderbach</tlvermgeo:strasse>
<tlvermgeo:geom> ...
```

5.6. Hierarchische Anfrage nach allen Hausnummern der Straße (EqualTo)

Aufgrund des zuvor erhaltenen Ergebnisses können mit dem als Verbindungselement fungierendem Wert `strkscomid 7105301043` die Hausnummern der Straße `Am Linderbach` der Gemeinde `Magdala` als letzte Hierarchiestufe aufgerufen werden. Diese Abfrage bietet sich an, wenn dem Nutzer in der Auswahl alle Hausnummern der Straße präsentiert werden sollen. Dazu wird wieder der `PropertyIsEqualTo` Operator verwendet.

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature
&TYPENAME=tlvermgeo:GAZHKO_HNR
&NAMESPACE=xmlns%28tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo%29
&FILTER=<ogc:Filter xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:strkscomid</ogc:PropertyName>
<ogc:Literal>7105301043</ogc:Literal>
</ogc:PropertyIsEqualTo></ogc:Filter>
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:Query typeName="tlvermgeo:GAZHKO_HNR">
<ogc:Filter>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:strkscomid</ogc:PropertyName>
<ogc:Literal>7105301043</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

HTTP POST

Antwort: Informationen von 10 Features werden übertragen (Auszug)

```
...
<gml:featureMember>
  <tlvermgeo:GAZHKO_HNR xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
gml:id="ID_TLVERMGE0_GAZHKO_HNR.1600282484">
  <tlvermgeo:id>1600282484</tlvermgeo:id>
  <tlvermgeo:strkscomid>7105301043</tlvermgeo:strkscomid>
  <tlvermgeo:hausnr>3</tlvermgeo:hausnr>
  <tlvermgeo:qm>B</tlvermgeo:qm>
  <tlvermgeo:geom>
  <!--Inlined geometry 'ID_TLVERMGE0_GAZHKO_HNR.1600282484_TLVERMGE0_GEOM'-->
  <gml:Point gml:id="ID_TLVERMGE0_GAZHKO_HNR.1600282484_TLVERMGE0_GEOM" srsName="EPSG:25832">
    <gml:pos>671517.066 5642494.634</gml:pos>
  </gml:Point>
  </tlvermgeo:geom>
  </tlvermgeo:GAZHKO_HNR>
</gml:featureMember>
...
```

5.7. Hierarchische Anfrage nach einer konkreten Hausnr (And / EqualTo)

Da der Nutzer im Beispiel bereits über die konkrete Hausnummer **3** verfügt, kann die Anfrage konkretisiert werden, indem der Filter neben dem Wert `strkscomid 7105301043` auf die Hausnummer mit dem `And` Operator erweitert wird.



Wenn die Hausnummer noch einen **Zusatz** hat, wie zum Beispiel „**3 a**“, dann muss dieser Zusatz in einem dritten `PropertyIsEqualTo` Operator abgefragt werden, anderenfalls werden sämtliche Features mit Hausnummer „**3**“ angezeigt und es ist eine weitere Abfrage zu stellen.

```
<ogc:PropertyName>tlvermgeo:zusatz</ogc:PropertyName>
<ogc:Literal>a</ogc:Literal>
```

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature
&TYPENAME=tlvermgeo:GAZHKO_HNR
&NAMESPACE=xmlns%28tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo%29
&FILTER=<ogc:Filter xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<ogc:And><ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:strkscomid</ogc:PropertyName>
<ogc:Literal>7105301043</ogc:Literal></ogc:PropertyIsEqualTo>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:hausnr</ogc:PropertyName>
<ogc:Literal>3</ogc:Literal>
</ogc:PropertyIsEqualTo></ogc:And>
</ogc:Filter>
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:Query typeName="tlvermgeo:GAZHKO_HNR">
<ogc:Filter>
<ogc:And>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:strkscomid</ogc:PropertyName>
<ogc:Literal>7105301043</ogc:Literal>
</ogc:PropertyIsEqualTo>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>tlvermgeo:hausnr</ogc:PropertyName>
<ogc:Literal>3</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

HTTP POST

Antwort: Informationen von 1 Feature werden übertragen (Auszug)

```
... <gml:Envelope srsName="EPSG:25832">
  <gml:lowerCorner>671517.066 5642494.634</gml:lowerCorner><gml:upperCorner>671517.066 5642494.634</gml:upperCorner>
... <gml:featureMember>
  <tlvermgeo:GAZHKO_HNR xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
gml:id="ID_TLVERMGEO_GAZHKO_HNR.1600282484">
  <tlvermgeo:id>1600282484</tlvermgeo:id><tlvermgeo:strkscomid>7105301043</tlvermgeo:strkscomid><tlvermgeo:hausnr>3</tlvermgeo:hausnr>
  <gml:Point gml:id="ID_TLVERMGEO_GAZHKO_HNR.1600282484_TLVERMGEO_GEOM" srsName="EPSG:25832">
    <gml:pos>671517.066 5642494.634</gml:pos></gml:Point>...
```


5.8. Weitere Tipps zur sinnvollen Nutzung der Ergebnisse

1. Mit der Antwort aus dem Abschnitt 5.7 steht nun eine korrekte Position der Adresse Am Linderbach 3 in Magdala fest. Die Koordinaten können entweder aus dem oberen Abschnitt `<gml:Envelope>` oder im Feature bei `<gml:Point>` bezogen werden, sie sollten identisch sein, da es sich um eine einfache Punktkoordinate handelt.

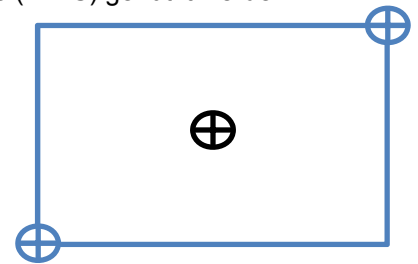
```

1 <?xml version='1.0' encoding='UTF-8'?><wfs:FeatureCollection xmlns:xsi="http://www.w3.org/2001/XMLSchema
2 <gml:boundedBy>
3   <gml:Envelope srsName="EPSG:25832">
4     <gml:lowerCorner>671517.066 5642494.634</gml:lowerCorner>
5     <gml:upperCorner>671517.066 5642494.634</gml:upperCorner>
6   </gml:Envelope>
7 </gml:boundedBy>
8 <gml:featureMember>
9   <tlvermgeo:GAZHKO_HNR xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo" gml:id="ID_
10   <tlvermgeo:id>1600282484</tlvermgeo:id>
11   <tlvermgeo:strkscomid>7105301043</tlvermgeo:strkscomid>
12   <tlvermgeo:hausnr>3</tlvermgeo:hausnr>
13   <tlvermgeo:gm>B</tlvermgeo:gm>
14   <tlvermgeo:geom>
15     <!--Inlined geometry 'ID_TLVERMGE0_GAZHKO_HNR.1600282484_TLVERMGE0_GEOM'-->
16     <gml:Point gml:id="ID_TLVERMGE0_GAZHKO_HNR.1600282484_TLVERMGE0_GEOM" srsName="EPSG:25832">
17       <gml:pos>671517.066 5642494.634</gml:pos>
18     </gml:Point>
19   </tlvermgeo:geom>
20 </tlvermgeo:GAZHKO_HNR>
21 </gml:featureMember>
22 </wfs:FeatureCollection>

```



Dieses Ergebnis kann zur Weitergabe an den Web Map Service (WMS) genutzt werden. Sinnvollerweise soll das Ergebnis in Kartenform präsentiert werden, dabei soll der Treffer im Regelfall im Zentrum des Kartenbildes platziert werden. In Abhängigkeit der Bildschirmgröße (ausgedrückt durch die Parameter `WIDTH` und `HEIGHT`, Angaben in Pixel) und dem gewünschten Anzeigemaßstab können die Werte für die `BBOX` (Angaben in der Maßeinheit des Bezugssystems) bestimmt werden. Alternativ kann man mit vordefinierten Offset-Parametern arbeiten, um die Werte für die linke untere und rechte obere Ecke der `BBOX` festzulegen.



Es empfiehlt sich, das Bezugssystem, welches für die Kartenanzeige genutzt werden soll, bereits in den WFS-G Anfragen zu verwenden, siehe Abschnitt 5.1

2. Bei der Abfrage der Straßen werden in der jetzigen Ausbaustufe des Gazetteers auch die zugehörige Gemarkung unter `gemarkname` und die Gemarkungsnummer unter `gemarknr` übermittelt:

```

- <gml:featureMember>
- <tlvermgeo:GAZHKO_STREET gml:id="ID_TLVERMGE0_GAZHKO_STREET.7105301043">
  <tlvermgeo:gemkennzahl>71053</tlvermgeo:gemkennzahl>
  <tlvermgeo:strassesch>01043</tlvermgeo:strassesch>
  <tlvermgeo:strkscomid>7105301043</tlvermgeo:strkscomid>
  <tlvermgeo:strasse>Am Linderbach</tlvermgeo:strasse>
- <tlvermgeo:geom>
  <!--
  Inlined geometry 'ID_TLVERMGE0_GAZHKO_STREET.7105301043_TLVERMGE0_GEOM'
  -->
- <gml:Polygon gml:id="ID_TLVERMGE0_GAZHKO_STREET.7105301043_TLVERMGE0_GEOM" srsName="EPSG:25832">
  <gml:exterior>
  - <gml:LinearRing srsName="EPSG:25832">
    - <gml:posList>
      671400.594 5642487.814 671400.594 5642506.981 671528.232 5642506.981 671528.232 5642487.814 671400.594 5642487.814
    </gml:posList>
  </gml:LinearRing>
  </gml:exterior>
  <gml:Polygon>
  <tlvermgeo:geom>
  <tlvermgeo:gemarknr>5362</tlvermgeo:gemarknr>
  <tlvermgeo:gemarkname>Magdala</tlvermgeo:gemarkname>
  </tlvermgeo:GAZHKO_STREET>

```



Es kann sinnvoll sein, diese Informationen dem Nutzer ebenfalls zur Verfügung zu stellen. Besonders hilfreich können die Angaben bei großen Gemeinden und nach Eingemeindungen sein oder wenn im Fall einer Eingemeindung noch nicht alle doppelt vorkommenden Straßennamen der vormals getrennt verwalteten Einzelgemeinden umbenannt worden sind.

6. Spezialfall – Abfrage über BBOX oder Polygon

Ein möglicher Anwendungsfall zur Nutzung der WFS-G Features kann darin liegen, dass für eine BoundingBox oder ein Polygon, welche vom Nutzer geliefert werden, die darin enthaltenen Hausnummern ausgegeben werden sollen. Beispielsweise sollen alle Anwohner im Bereich einer geplanten Maßnahme (=Polygon, BBOX) informiert werden.

Es ist bei der Beschreibung der Koordinaten innerhalb des Elementes „LinearRing“ auf die korrekte Formatierung zu achten, die Anfangs- und Endkoordinaten müssen identisch sein.

```
http://www.geoproxy.geoportal-th.de/geoproxy/services?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature
&TYPENAME=tlvermgeo:GAZHKO_HNR
&SRSNAME=EPSG:25832
&NAMESPACE=xmlns%28tlvermgeo=http://www.geoproxy.geoportal-th.de/tlvermgeo%29
&FILTER=<ogc:Filter xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd"><ogc:Intersects>
<ogc:PropertyName>tlvermgeo:geom</ogc:PropertyName>
<gml:Polygon srsName="EPSG:25832"><gml:outerBoundaryIs><gml:LinearRing>
<gml:coordinates decimal="." cs="," ts=" ">671400,5642487 671400,5642506
671528,5642506 671528,5642487 671400,5642487
</gml:coordinates></gml:LinearRing></gml:outerBoundaryIs>
</gml:Polygon></ogc:Intersects></ogc:Filter>
```

HTTP GET

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:tlvermgeo="http://www.geoproxy.geoportal-th.de/tlvermgeo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:Query typeName="tlvermgeo:GAZHKO_HNR" srsName="EPSG:25832">
<ogc:Filter>
<ogc:Intersects>
<ogc:PropertyName>tlvermgeo:geom</ogc:PropertyName>
<gml:Polygon srsName="EPSG:25832">
<gml:outerBoundaryIs>
<gml:LinearRing>
<gml:coordinates>671400,5642487 671400,5642506 671528,5642506 671528,5642487
671400,5642487</gml:coordinates>
</gml:LinearRing>
</gml:outerBoundaryIs>
</gml:Polygon>
</ogc:Intersects>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

HTTP POST

Antwort: Informationen von 9 Features werden übermittelt.



Aus den gewonnenen Informationen muss dann eine Anfrage über die bei den Hausnummern-Features hinterlegten Werte `strkscomid` an die Straßen-Features `GAZHKO_STREET` gestellt werden, um die korrekten Straßennamen `strasse` in Erfahrung zu bringen. An dieser Stelle wird bereits die Gemarkung mit ausgegeben, zur eindeutigen Adressierung ist jedoch eine weitere Anfrage an die Gemeinde-Features `GAZHKO_COMMUNE_FREE` über die bei den Straßen-Features hinterlegten Werte `gemkennzahl` nötig, um den korrekten Gemeindennamen unter `gemeindenname` abzufragen. Das Datenmodell wird hier also rekursiv durchlaufen.

7. Anhang – Konvertierung von Sonderzeichen

Bei HTTP-Get Requests werden einige Sonderzeichen benötigt, im Zweifelsfall sind die folgenden Elemente aus den Beispielen wie folgt in den URL zu ersetzen:

Zeichen	URL-Kodierung
Leerzeichen	%20
"	%22
(%28
)	%29
<	%3C
>	%3E

Bei Anfragen über Attributinhalte kann es vorkommen, dass besondere Zeichen wie beispielsweise Umlaute benötigt werden. Mit UTF-8 sollte die direkte Eingabe von Sonderzeichen möglich sein, dennoch kann es zu Fehlermeldungen des Dienstes kommen. In solchen Fällen sind die Sonderzeichen durch die entsprechende Kodierung zu ersetzen.

Im Beispiel sollen FeatureTypes angefragt werden, deren geographischer Name mit „Mü“ beginnt, dazu wird der Like-Operator (Abruf siehe unter 5.3) verwendet. Funktioniert die Suche über das Kriterium „Mü“ nicht, so ist anstelle des Umlautes „ü“ der numerische Wert `ü` zu verwenden.

```
<ogc:Filter>
  <ogc:PropertyIsLike wildCard='*' singleChar='.' escape='!'>
    <ogc:PropertyName>tlvermgeo:geographicName</ogc:PropertyName>
    <ogc:Literal>M&#252;*</ogc:Literal>
  </ogc:PropertyIsLike>
</ogc:Filter>
```

Nachfolgende Tabelle gibt die Kodierung für im deutschen Sprachraum übliche Sonderzeichen wieder. Bitte beachten: unter Umständen können bei bestimmten Attributinhalten wie bspw. Straßennamen die Akut- bzw. Gravis-Zeichen durch einfache Vokalzeichen ersetzt sein, Beispiel: é → e.

Kategorie	Zeichen	Numerisch	Beschreibung
Alphabet, latein, Akut	á	&#225;	kleines lateinisches a mit Akut
Alphabet, latein, Akut	é	&#233;	kleines lateinisches e mit Akut
Alphabet, latein, Akut	í	&#237;	kleines lateinisches i mit Akut
Alphabet, latein, Akut	ó	&#243;	kleines lateinisches o mit Akut
Alphabet, latein, Gravis	à	&#224;	kleines lateinisches a mit Gravis
Alphabet, latein, Gravis	è	&#232;	kleines lateinisches e mit Gravis
Alphabet, latein, Gravis	ì	&#236;	kleines lateinisches i mit Gravis
Alphabet, latein, Gravis	ò	&#242;	kleines lateinisches o mit Gravis
Alphabet, latein, Gravis	ù	&#249;	kleines lateinisches u mit Gravis
Alphabet, latein	ß	&#223;	kleines lateinisches ß = ess-zed
Grafik	"	&#34;	Anführungszeichen
Typo	" "	&#32;	einfaches Leerzeichen (Space)
Typo	" "	&#160;	nicht brechendes Leerzeichen
Typo	"-"	&#173;	weiches Trennzeichen
Typo	'	&#8242;	einfaches Anführungszeichen = Minuten = Fuss
Typo	"	&#8243;	doppeltes Anführungszeichen = Sekunden = Inch
Typo	&	&#38;	Ampersand
Typo	%	&#37;	Prozent
Typo	©	&#169;	Copyright
Umlaut	Ä	&#196;	Ä
Umlaut	Ö	&#214;	Ö
Umlaut	Ü	&#220;	Ü
Umlaut	ä	&#228;	ä
Umlaut	ö	&#246;	ö
Umlaut	ü	&#252;	ü
Währung	€	&#8364;	Euro