



## Tutorial: OGC-Webservices



### Normung / Interoperabilität

In dieser Lerneinheit sollen gängige OGC-Webservices beschrieben und deren Nutzung bei der Verarbeitung offener Geodaten erläutert werden.

Zudem werden Tipps und Tricks im Umgang mit OGC-WxS gesammelt.



### Einführung

Jeder, der schon einmal mit Geodaten gearbeitet hat, kennt das Problem: Will man Analysen zu aktuellen Fragestellungen durchführen, ist es schwierig Datensätze zu finden die sich auf demselben zeitlichen Stand befinden. Es liegt in der Natur von Geodaten, dass sie einen Bezug zur Wirklichkeit haben. Und da die Wirklichkeit stetigen Veränderungen unterliegt, müssen alle Datensätze, die nicht explizit einen historischen Stand beschreiben, häufig aktualisiert werden. Arbeitet man hierfür „klassisch“, also mit lokalen Dateien und Ordnern, ist dieser Prozess aufwendig. In der Praxis wird oft an verschiedenen Orten mit unterschiedliche Versionen des gleichen Datensatzes gearbeitet, was schnell zu Diskrepanzen in Analyseergebnissen führen kann.

Wie also löst man dieses Problem? Betrachtet man einmal die Welt außerhalb von Geodaten, ist das beste Werkzeug, um hochaktuelle Informationen zu erlangen, das World Wide Web. Dies trifft nicht nur auf Nachrichten und Trends zu, sondern auch auf Datensammlungen. Und natürlich wollen wir diese Möglichkeiten auch für die Analyse von Geodaten verwenden. Genauer gesagt geht es uns um die Nutzung von Webdiensten.

Die gesamte Struktur des Internets basiert auf Diensten, welche meistens über eine Client-Server-Beziehung realisiert sind. Die grundlegende Struktur ist also vorhanden, nun müssen nur für Geodaten passende Schnittstellen entworfen werden.

Natürlich könnte nun jeder seiner eigenen Implementation folgen. Aber genau wie wir mit einer Sammlung von standardisierten Dateiformaten arbeiten, wollen wir auch unsere Dienste standardisieren. Allseits bekannt ist die International Standard Organization (ISO), welche sich weltweit um Industriestandards kümmert. Speziell für raumbezogene Informationsverarbeitung hat sich 1994 außerdem das Open Geospatial Consortium (OGC) gegründet. Dieses arbeitet eng mit der ISO zusammen, um Datenformate und Prozesse im Geodatenbereich weltweit zu standardisieren. Aus dieser Zusammenarbeit sind unter anderem Normen und Spezifikationen für die von uns benötigten Webdienste hervorgegangen. Man redet hier typischerweise von den „OGC Web Services“.

In diesem Dokument wollen wir uns mit einigen dieser Dienste genauer beschäftigen. Zuerst jedoch eine Auflistung der derzeit definierten Schnittstellen:

Dienstname	Abkürzung	Nutzen
Web Map (Tile) Service	WM(T)S	Liefert Webkarten als Bilddateien, entweder in einem gewählten Bildausschnitt, oder in vordefinierten Kacheln
Web Feature Service	WFS	Liefert räumliche Vektordaten
Web Processing Service	WPS	Führt räumliche Operationen auf Anfragen aus und liefert das Ergebnis zurück
Web Coverage Service	WCS	Liefert räumliche Rasterdaten in verschiedenen Formaten
Web Catalogue Service	WCAS	Zugriff auf und Suche nach Metadaten
Web Gazetteer Service	WGS	Suche nach Geodaten über Namen und Wörter
Web Coordinate Transformation Service	WCTS	Bietet Koordinatentransformationen
Web Pricing and Ordering Service	WPOS	Stell Preis- und Lizenzinformationen für Geodaten bereit, und erlaubt das direkte Bestellen
Web View Service	WVS	Liefert 3D Szenen
Web Terrain Service	WTS	Liefert 3D Terrain

Für die meisten dieser Dienste sind nicht nur Normen, sondern auch technische Spezifikationen und Referenzimplementierungen vorhanden. Besonders die ersten Einträge in dieser Tabelle werden schon seit längerem sehr intensiv in der Geodatenverarbeitung genutzt.

## Grundlagen von Web-Anfragen

Alle hier betrachteten Webdienste basieren auf dem Hypertext Transport Protocol (HTTP). Dieses ist eins der grundlegenden Bausteine des Webs, und erlaubt es verschiedene Arten von Anfragen an Server zu stellen. Die meisten dieser Methoden rufen Daten von einem Server ab, übertragen neue Daten oder modifizieren, was bereits auf dem Server vorhanden ist. In unserem Fall wollen wir gewisse Daten an den Server senden, damit dieser uns die entsprechende Antwort geben kann. Meistens handelt es sich bei diesen Daten um Sammlungen von Parametern, z.B. die Koordinaten eines Rechtecks, welches einen Kartenausschnitt definiert.

Es gibt zwei Anfragemethoden um solche Daten zu übertragen: GET und POST. Bei GET werden die Parameter direkt in die URL geschrieben. Bei POST werden sie im „Körper“ der http-Anfrage mitversandt. Dabei wird die URL nicht modifiziert, stattdessen verhält sich diese Art von Anfrage mehr wie ein klassischer Dateiupload/Download und eignet sich vor allem für größere Anfragen. In beiden Fällen werden die Parameter durch Key-Value-Paare beschrieben. Keys geben dabei den Namen der Parameter an (ohne Beachtung von Großschreibung), Values die Werte dieser Parameter (unter Beachtung von Großschreibung).

Als Beispiel wollen wir eine Anfrage an die Webseite <http://server.com/page.html> stellen.

- GET: Wir schreiben „<http://server.com/page.html?param1=val1&param2=val2>“ in die Adresszeile unseres Browsers und öffnen die Seite.

- POST: Auf einer Webseite werden zwei Formularfelder namens „Param1“ und „Param2“ angeboten, außerdem ein Knopf, um deren Inhalte abzuschicken. Über den Quellcode der Seite wird nach Klicken des Knopfes eine Nachricht an „http://server.com/page.html“ zusammengestellt.

## Ausgewählte OGC-Webservices

Wir wollen uns nun die in der Praxis am häufigsten genutzten OGC Web Services genauer ansehen.

### WMS

Der Web Map Service liefert auf Anfragen Kartenausschnitte als fertige Bilddateien zurück. Angesichts unseres Wissens über HTTP-Anfragen stellt sich nun die Frage: Was für Parameter benötigt der Server?

Zuerst gibt es hier den sogenannten „Service“-Parameter. Er legt fest, welcher Dienst genau angefragt wird, falls ein Server z.B. neben dem WMS noch andere Diensteschnittstellen anbietet. Weiterhin benötigt wird „Request“. Er legt fest welche Funktion des WMS genau benötigt wird. Neben der direkten Datenanfrage können Webdienste z.B. auch auflisten, über was für Funktionen sie genau verfügen. Außerdem kann über „Version“ angegeben werden für welche WMS-Version eine Anfrage gedacht ist. Dies verhindert, dass ältere automatisierte Abläufe nach Systemupdates auf einmal nicht mehr funktionieren.

Um die Funktionsweise eines WMS zu beschreiben, sind vor allem die verschiedenen möglichen „Requests“ interessant, daher fahren wir mit diesen fort:

#### *GetCapabilities*

Dieser Request liefert ein ISO-standardisiertes XML-Metadaten-Dokument zurück. Es enthält Informationen darüber, was für Bildformate der Server zurückgeben kann, in welchen Koordinatensystemen Daten vorliegen, welche Datenlayer angefragt werden können usw.

#### *GetMap*

Der eigentliche Kern eines WMS. Dieser Request gibt ein Rasterbild eines spezifizierten Gebietes zurück. Dieses Gebiet und das Aussehen des Rasterbildes werden über verschiedene Parameter definiert. Eine Auswahl von Keys:

- BBOX – der gewünschte Kartenausschnitt als Rechteck.
- CRS – Koordinatenreferenzsystem, meistens als EPSG-Code
- WIDTH/HEIGHT – die gewünschte Größe des Bildes auf dem Bildschirm
- LAYERS – welche Kartenlayer auf dem Bild ausgegeben werden sollen
- STYLES/SLD – der gewünschte Stil der Karte
- ...

Eine GET-basierte *GetMap*-Anfrage, um z.B. die Rostocker Ortsteile aus dem Geodatengebot der Hansestadt Rostock zu erhalten, könnte also so aussehen:

<https://geo.sv.rostock.de/geodienste/ortsteile/wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap&LAYERS=hro.ortsteile.ortsteile&WIDTH=512&HEIGHT=512&CRS=EPSG:3857&BBOX=1334743.3879,7180341.5631,1364362.7364,7209540.5079&FORMAT=image/png>

Auf den ersten Blick kann einen eine solche Anfrage ganz schön erschlagen. Gehen wir aber die einzelnen Parameter Schritt für Schritt durch: Wir stellen eine Anfrage an die Seite <https://geo.sv.rostock.de/geodienste/ortsteile/wms>. Hier handelt es sich also um einem WMS, welcher Ortsteile der Hansestadt Rostock anbietet. Trotzdem vermutlich auf dieser URL der WMS der

einzigste Dienst ist, müssen wir über „SERVICE“ spezifizieren, dass wir WMS-Funktionalitäten nutzen wollen. Wir benötigen diese dabei in der „VERSION“ 1.3.0 der OGC-WMS-Spezifikation. Da wir ein Kartenbild erhalten wollen ist unser „REQUEST“ *GetMap*. Im Falle von „LAYERS“ ist der benötigte Wert etwas schwieriger zu ermitteln, hier müssen wir vorher über *GetCapabilities* in die genaue Layerdefinition schauen. Der Ortsteillayer nennt sich auf diesem WMS `hro.ortsteile.ortsteile`. „WIDTH“ und „HEIGHT“ werden auf 512 gesetzt, wir werden also ein quadratisches Bild mit entsprechender Pixelauflösung erhalten. Das genutzte „CRS“ ist EPSG:3857, ein projiziertes Koordinatensystem, welches in vielen Internetanwendungen genutzt und daher auch als „Web-Mercator“ bezeichnet wird. Die „BBOX“ definiert zwei Koordinaten in diesem System, welche für links-unten und rechts-oben im gewünschten Rechteck des Geländeausschnitts stehen. Schlussendlich wird noch angegeben, dass wir das Ergebnis im „FORMAT“ PNG (Portable Network Graphics) erhalten wollen.

Kopiert man die obige *GetMap*-Anfrage in einen Browser und ruft die Seite auf erhält man also folgende PNG-Datei als Ergebnis:



Über „STYLE“ und andere Parameter kann man das Ergebnis weiter manipulieren. Für Styles werden bei WMS sogenannte SLD-Dateien (Styled Layer Descriptor) genutzt. Deren Syntax zu beschreiben würde jedoch an dieser Stelle zu weit führen. Ein umfassendes Tutorial findet sich unter <http://docs.geoserver.org/stable/en/user/styling/sld/cookbook/>.

#### *DescribeLayer*

(Optional) Gibt, wenn vorhanden, Detailinformationen zu einzelnen Layers zurück.

#### *GetFeatureInfo*

(Optional) Gibt die Datengrundlage für ein gewähltes Gebiet zurück. Simplifizierte Variante einer WFS-Anfrage.

#### *GetLegendGraphics*

(Optional) Ähnliche Parameter wie *GetMap*, liefert jedoch anstatt des Kartenausschnittes eine Legende für den gewählten Bildbereich. Hierbei handelt es sich auch um eine Bilddatei.

Hinweis: Mit WMS lässt sich beim Erstellen von Webseiten sehr einfach arbeiten. Es muss hier kein komplexes WebGIS aufgesetzt werden, um die Karten korrekt darzustellen. Stattdessen können die Ausgabebilddateien einfach mittels HTML und CSS übereinandergelegt werden, um komplexe Karten mit mehreren Layern und Legende zu erzeugen.

## WFS

Benötigen wir nicht nur die fertige Kartendarstellung, sondern auch die zugrundeliegenden Daten in einem passenden Geodatenformat, dann muss statt eines WMS ein WFS genutzt werden. Hier wird keine Bilddatei zurückgegeben, sondern Vektordaten. Die Daten werden dabei normalerweise in der Geography Markup Language (GML) zurückgegeben, ein ebenso von der OGC entworfenes und durch ISO standardisiertes Format für das Speichern von Geodaten in XML-Datenstrukturen. Andere Formate wie z.B. GeoJSON können jedoch auch definiert werden.

Eine Anfrage ähnelt der eines WMS, es gibt jedoch einige neue Parameter und Requests:

### *GetCapabilities*

Selbe Funktionalität wie im WMS.

### *DescribeFeatureType*

Gibt standardisierte Beschreibungen zu den verschiedenen vom Dienst angebotenen Objektarten und ihrer Struktur zurück.

### *GetFeature*

Der eigentliche Datenaufruf, analog zu *GetMap*. WFS und WMS teilen sich grundlegende Parameter wie „BBOX“. Anstatt „LAYERS“ nutzt man hier jedoch „TYPENAMES“, um zu definieren welche Typen von Vektordaten man benötigt. Analog zum WMS kann man diese über „BBOX“ räumlich filtern. Weitere angebotene Filterparameter sind z.B. „COUNT“ und „SORTBY“, es gibt jedoch auch einen „FILTER“ Parameter mit dem komplexere Abfolgen in GML definiert werden können. Diese ähneln in ihrer Funktionalität SQL-Anfragen. Wollen wir also nur Straßen des Typs „1“ von einem WFS erhalten, könnte der Filter folgendermaßen aussehen:

<Filter>

<PropertyIsEqualTo>

<PropertyName>roadtype</PropertyName>

<Literal>1</Literal>

</PropertyIsEqualTo>

</Filter>

Um einen Filter textuell in dieser Form aufschreiben zu können muss ein POST-Request genutzt werden. Will man diesen Filter über GET anwenden, müssen in der URL die entsprechenden Sonderzeichen-Codes genutzt werden:

**%3CFilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Eroadtype%3C/PropertyName%3E%3CLiteral%3E1%3C/Literal%3E%3C/PropertyIsEqualTo%3E%3C/Filter%3E**

Häufige Datenverarbeitungs-Operationen können so schon angewendet werden, bevor der Datensatz den Nutzer überhaupt erreicht.

## WPS

Die im WFS angebotenen Filteroperationen sind jedoch nur der Anfang. Im Zeitalter von Cloud Computing sollte schnell klar werden, dass auch weitaus fortgeschrittenere Operationen über Anfragen an Server möglich gemacht werden können.

Hierfür wurde der Web Processing Service spezifiziert. Mit ihm ist es möglich Dienste aufzusetzen, welche auf Anfrage komplexe Verarbeitungsprozesse auf dem Server starten und das Ergebnis danach an den Anfragenden zurücksenden.

Die Operationen selbst sind dabei nicht standardisiert und können je nach Bedarf auf dem Server implementiert und angeboten werden. Stattdessen sind genau wie bei den anderen Diensten nur die Schnittstellen zur Eingabe und Ausgabe von Parametern und Daten standardisiert.

### *GetCapabilities*

Selbe Funktionalität wie in den anderen Diensten. Listet unter anderem die möglichen Operationen (Prozesse) auf.

### *DescribeProcess*

Hat man in den Capabilities den gewünschten Prozess entdeckt, schreibt man dessen ID in diese Anfrage. In der Antwort finden sich dann spezifische Informationen darüber wie der Prozess arbeitet, und was für Möglichkeiten für Eingabe und Ausgabe angeboten werden. Eingabe und Ausgabe werden dabei genau wie beim WFS in GML-Strukturen gefasst, es sei denn der Output ist eine einzelne Datei, wie z.B. eine Bilddatei. Diese kann direkt als Rohdatei verschickt werden.

### *Execute*

Der Start des eigentlichen Prozesses. Diese können asynchron und über lange Zeiträume laufen. Zuerst muss der Nutzer festlegen was seine Inputs sind, dann wie die Antwort aussehen soll (in GML oder Rohdaten) und ob die Antwort auch nachträglich über eine Webadresse erreichbar sein soll. Außerdem gibt es weitere Anschlussstellen, um Webadressen zu erhalten, die über den Status einer länger laufenden Operation Auskunft geben. Dieser Teil der WPS-Syntax ist einer der komplexeren Teile der Spezifikation, und würde den Rahmen dieses Dokuments sprengen. Die meisten Tools, die die Arbeit mit WPS anbieten, bringen vereinfachte „Request Builder“ mit sich, welche die Arbeit mit Web-Operationen auch weniger technisch versierten Nutzern möglich machen.

Dem geeigneten Leser fällt vielleicht an dieser Stelle auf, dass bis auf wenige Ausnahmen alle W\*S-Anfragen und Antworten über die standardisierte GML ablaufen. Diese auf XML-Dokumente konzentrierte Standardisierung ermöglicht eine sehr interessante Vorgehensweise namens „Service Chaining“. Aufgrund der gemeinsamen Schnittstellen können Ketten von WPS-Operationen erzeugt werden, welche eventuell sogar die Ausgabe von anderen WFS oder WMS als Eingabe verwenden. Nutzt man die OGC-Webservices also zu ihrem vollen Potential aus, können über das Internet ganze Verarbeitungsketten definiert werden, welche dezentral dafür sorgen, dass jeder Schritt des Prozesses stets mit den aktuellsten Daten arbeitet. Hinter dem Anzeigen einer simplen Ausgabekarte auf einer Webseite könnten also Dutzende komplexe WebGIS-Analyseschritte stehen, ohne dass der Nutzer je etwas davon mitbekommt.

## WCS

Wie vorher erwähnt, bietet ein WFS nur Vektordaten, ein WMS nur einfache Bilddateien an. Was jedoch, wenn wir mit georeferenzierten Rasterdatensätzen arbeiten wollen? Hierfür gibt es den „Web Coverage Service“ (WCS). Die möglichen Requests sind analog zum WFS: *GetCapabilities*,



*DescribeCoverage*, *GetCoverage*. Genau wie beim WFS gibt es auch die Möglichkeit Filteroperationen vom Server durchführen zu lassen. So können noch bevor das Ergebnis ankommt z.B. Zeiträume, gewünschte Spektralbänder und Höhenbeschränkungen spezifiziert werden.

Die möglichen Ausgabeformate eines WCS umfassen einfache Bilddateien wie beim WMS, aber auch georeferenzierte Formate wie GeoTIFF oder ArcGrid.

## WCAS

Der "Web Catalogue Service" (WCAS), teilweise auch als "Catalogue Service for the Web" (CSW) bezeichnet, erlaubt das Bereitstellen und Durchsuchen von Metadaten über Geodatenätze und räumliche Prozesse. In großen Geodateninfrastrukturen, welche mit den OGC-Webservices kompatibel sind, kann der WCAS einen zentralen Anlaufpunkt darstellen. Die genannten Metadaten können eine Vielzahl von Informationen enthalten. Eine Auswahl: Identifikationsnummern, Einschränkungen, Qualität, Aktualität, Räumliche Referenzen, Validität, Ausdehnung, usw.

WCAS profitieren von der dokumentbasierten Architektur, die GML mit sich bringt. Als Ableger von XML können Such- und Filteroperationen mit einer eigens definierten Abfragesprache namens XPath definiert werden, welche mächtige Suchoperationen auch auf großen XML-Datensätzen erlaubt.

Der WCAS-Standard ist sowohl für Betreiber, Datenlieferanten als auch Anwender gedacht und enthält neben Such- und Filteroperationen auch Möglichkeiten zum Aktualisieren von Metadaten, oder sogar zum Übernehmen von Metadaten aus anderen standardkonformen WCAS. So können Netze an Informationen aufgebaut werden, in denen Datensätze aus einer Vielzahl von Quellen zusammengeführt werden, ohne dass Unklarheiten darüber entstehen, wem ein Datensatz ursprünglich gehört. Gerade in Zeiten in denen für viele GIS-Projekte eine Mischung aus offenen und kommerziellen Daten genutzt wird sind solche Quelleninformationen wichtig.


## Nutzung der OGC-Services in GIS

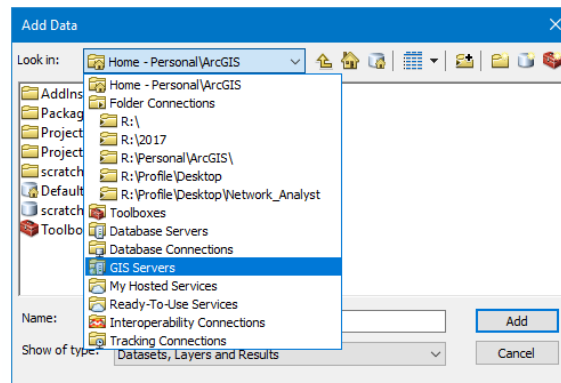
Die im letzten Abschnitt vorgestellte Mischung aus HTTP-Anfragen und GML-Dokumenten ist offensichtlich keine sehr nutzerfreundliche Schnittstelle. Zwar sind sie lesbar genug, dass sie von Menschen direkt verwendet werden können, doch es wird viel Hintergrundwissen vorausgesetzt.

Die meisten Geo-Informationssysteme, ob im Web oder mit Desktop-Client, unterstützen einfaches Arbeiten mit den OGC Services über grafische Nutzungsoberflächen. In den meisten Fällen werden die Ausgabedaten der Services wie jede anderen Datenebene im System behandelt.

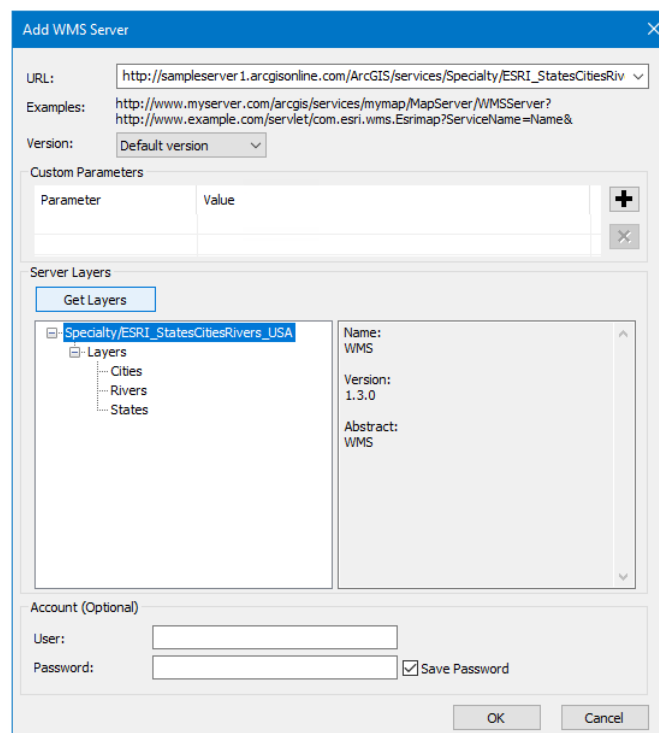
Wir wollen an dieser Stelle beispielhaft betrachten, wie OGC-Webservices in ArcGIS und QGIS eingebunden werden können.

## ArcGIS

Das Hinzufügen von W\*S Daten beginnt wie bei allen anderen Datensätzen: Mit der „Add Data“ Schaltfläche . Anstatt jedoch einen lokalen Ordner auszuwählen, klicken wir auf den Menüpunkt „GIS Servers“. Die ersten Einträge hier sind ESRI-eigene Serverlösungen wie ArcGIS Server, gleich darauf folgen jedoch die zur Verfügung stehenden OGC Services. Nativ unterstützt werden hier WM(T)S und WCS.

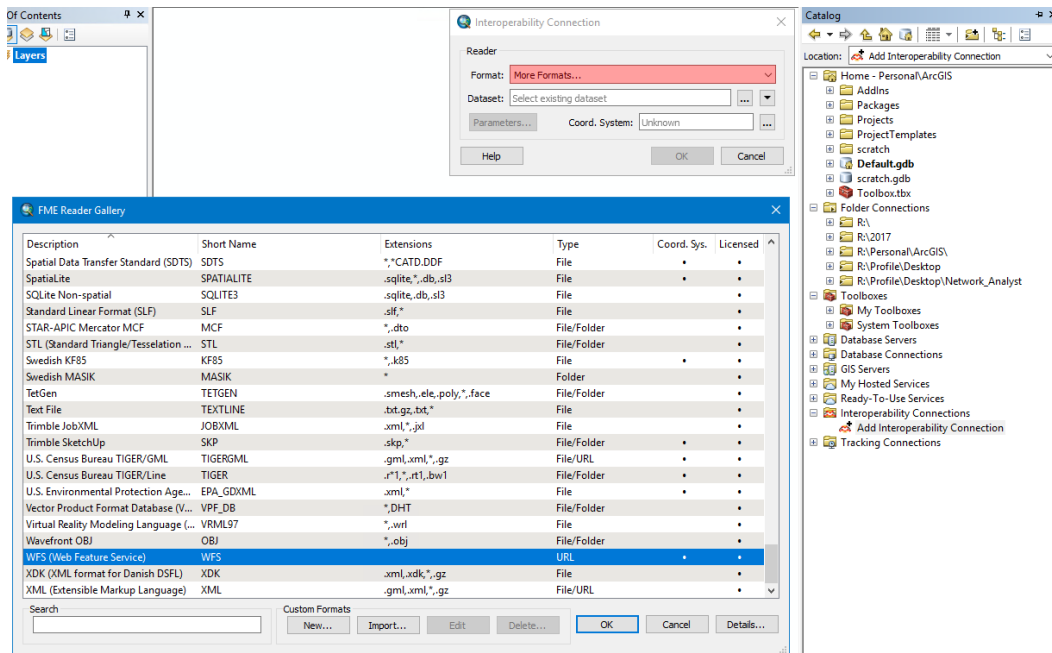


Um einen neuen Server hinzuzufügen muss nur die entsprechende URL des Serviceanbieters angegeben werden. Über „Get Layers“ kann dann direkt eingesehen werden was für Kartenlayer/Raster auf dem Server vorhanden sind.



Nicht angeboten wird an dieser Stelle der WFS. Um diesen nutzen zu können muss ArcGIS so erweitert werden, dass es mit GML-Daten arbeiten kann. Dafür gibt es die „Data Interoperability“-Extension. Ist diese aktiviert, kann im „Catalog“ unter „Interoperability Connections“ der Menüpunkt „Add Interoperability Connection“ genutzt werden. Über diese Schnittstelle kann eine Vielzahl von Geodatenformaten in ArcGIS importiert werden. Unter anderem findet sich hier auch der WFS.



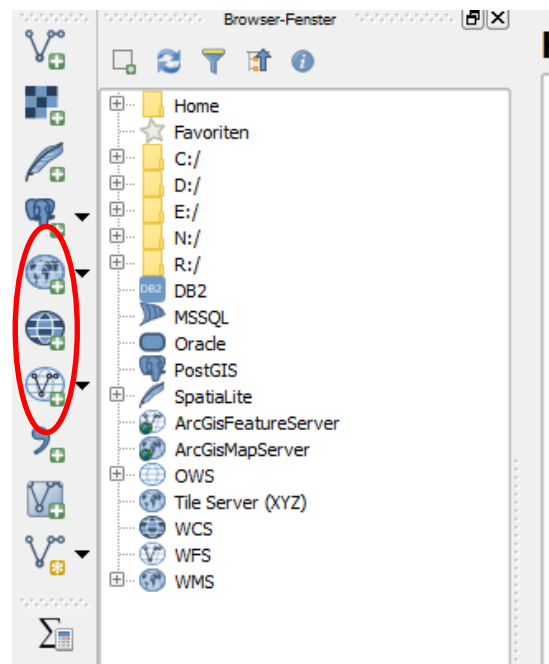


Dieser wird, wie gehabt, über die URL angegeben und kann dann als Layer verwendet werden.

## QGIS

QGIS enthält Unterstützung für die gleichen Dienste wie ArcGIS, macht deren Benutzung dem Nutzer jedoch noch einfacher. Alle drei Optionen befinden sich in der linken Menüleiste, über welche auch alle anderen Daten in QGIS hinzugefügt werden.

Im nächsten Menü können dann Services wie gehabt über ihre URL hinzugefügt werden. Der Nutzer kann sich vorher die verfügbaren Layers ansehen und auswählen, welche Daten genau benötigt werden.



Hinweis: Sowohl in ArcGIS als auch in QGIS funktionieren die Schnittstellen, indem sie Parameter an die vom Nutzer gegebene URL anhängen. Dabei wird von der URL nichts abgeschnitten, gibt man also als Nutzer schon gewisse Parameter mit in die URL, werden diese typischerweise im Ergebnis beachtet. (Es sei denn, es kommt zu Konflikten zwischen den vom Nutzer und den vom Programm spezifizierten Parametern)

## Erstellung eigener OGC-Services

Für Bereitstellen eines eigenen OGC-Services werden drei Dinge benötigt: Ein Webserver, eine Anwendung, die den OGC Standard implementiert, und die Daten, die bereitgestellt werden sollen.

Das Aufsetzen eines Webserver ist heutzutage einfach, und oft in Installationsanleitungen für entsprechende WebGIS enthalten. Der Server ist dazu da, die Anfragen von Nutzern an den Ports entgegenzunehmen und zu verarbeiten. Die im zweiten Abschnitt beschriebenen GET und POST Anfragen landen also beim Webserver, und dieser gibt deren Parameter und Inhalte an lokal laufende Anwendungen weiter.

Die Anwendung ist das Kernstück des Prozesses. Sie muss sich um das Interpretieren von Anfragen, das Laden von Daten, Verarbeitungsschritte und das Zusammenstellen der Antwort kümmern. Typischerweise wird dieser Teil des Dienstes von einem WebGIS übernommen, welches mit Funktionalitäten ähnlich zu denen eines Desktop-GIS auf den webbasierten Datenformaten arbeiten kann. Das populärste Beispiel hierfür ist GeoServer, welcher Unterstützung für fast alle wichtigen offenen Standards implementiert.

Die Daten selbst müssen nicht unbedingt in GML oder anderen in den OGC-Standards vorgegebenen Formaten vorliegen. Das jeweilige WebGIS muss jedoch im Stande sein, sie entweder direkt bereitzustellen oder sie intern in passende Formate zu konvertieren. Ist diese Funktionalität vorhanden, können Daten aus einer Vielzahl von Quellen geliefert werden.

### Geoserver

Soll ein W\*S nicht in eine bestehende Infrastruktur eingebunden werden, sondern einzeln aufgesetzt werden, macht Geoserver einem das Leben sehr leicht. Zuerst muss das Java Runtime Environment in der Version 8 (JRE) installiert werden. Mit den von Geoserver vorgegebenen Installationsdateien kann dann eine normale Installation durchgeführt werden. Der benötigte Webserver wird im Laufe dieser bereits mit aufgesetzt. Die Ports und Schnittstellen sind am Ende schon fertig eingestellt. Die genaue Installationsanleitung ist unter <http://docs.geoserver.org/latest/en/user/> zu finden.

### Literatur

Bill, R. (2016): Grundlagen der Geo-Informationssysteme. 6. Auflage. Wichmann Verlag. Offenbach-Berlin. 867 Seiten. Kapitel 4.3.

Seip, C., Korduan, P., Zehner, M.L. (2017): Web-GIS. Grundlagen, Anwendungen und Implementierungsbeispiele, Wichmann Verlag, Heidelberg, 552 Seiten.