



Klimadatastyrelsen

Brugervejledning – Transitionsguide for entitetsbaserede WFS-tjenester

Drift og modernisering af Datafordeleren

September 2025

Version 0.4 – 19-09-2025



Indholdsfortegnelse

1	Indledning	3
1.1	Eksterne links	4
1.2	Begreber.....	5
2	WFS på Datafordeleren	6
2.1	Om entitetsbaserede WFS-tjenester	6
2.1.1	Forskelle fra traditionelle/sammenstillede WFS-tjenester.....	6
2.2	Udstilling af WFS-tjenester	6
2.3	Standardparametre.....	7
2.4	WFS-Metoder	9
2.4.1	GetCapabilities	9
2.4.2	DescribeFeatureType	9
2.4.3	GetFeature	11
2.5	Outputformater	12
2.5.1	Valg af format.....	12
2.6	Bitemporalitet	13
2.7	Filtrering	13
2.7.1	Filtrering med bbox.....	13
2.7.2	Filtrering med cql_filter.....	15
2.8	Versionering	18
2.9	Sådan anvender du entitetsbaserede WFS-tjenester	18
2.9.1	Separate REST-kald	19
2.9.2	GIS-klienter: QGIS	23
3	Autentifikation for WFS-tjenester.....	28



1 Indledning

Dette dokument er en guide til hvordan man som Anvender kan bruge Datafordelerens entitetsbaserede WFS-tjenester.

Dokumentet er opdelt i tre dele:

- 1) Første del beskriver hvad entitetsbaserede WFS-tjenester er, hvordan de adskiller sig fra traditionelle WFS-tjenester, og hvordan de anvendes gennem standard OGC-metoder.
- 2) Anden del forklarer den tekniske implementering inklusiv URL-struktur, autentifikation, versionering og tilgængelige dataformater.
- 3) Tredje del dækker avanceret anvendelse såsom bitemporal håndtering, performance optimering, fejlhåndtering og praktisk integration med GIS-software.

Dokumentet er tiltænkt følgende læsere:

- 1) Eksisterende anvendere af Datafordeleren der bruger de nuværende Snowflake WFS-tjenester, som ønsker at udnytte de nye automatisk genererede entitetsbaserede WFS-tjenester.
- 2) GIS-brugere og systemintegratorer der skal arbejde med geografiske data fra danske registre gennem standardiserede WFS 2.0.0 grænseflader.
- 3) Nye anvendere der ønsker at komme i gang med at hente geodata fra Datafordeleren via WFS.



1.1 Eksterne links

Den følgende tabel indeholder links, der henvises til i hele dokumentet.

Reference	URL
WFS Standard	https://www.ogc.org/standards/wfs/
GeoServer	https://geoserver.org/
GeoServer WFS Reference	https://docs.geoserver.org/stable/en/user/services/wfs/reference.html
GeoServer GetCapabilities	https://docs.geoserver.org/stable/en/user/services/wfs/reference.html#getcapabilities
GeoServer DescribeFeatureType	https://docs.geoserver.org/stable/en/user/services/wfs/reference.html#describefeaturetype
GeoServer GetFeature	https://docs.geoserver.org/stable/en/user/services/wfs/reference.html#getfeature
GeoServer WFS Output Formats	https://docs.geoserver.org/stable/en/user/services/wfs/outputformats.html
Bitemporalitet på Datafordeleren	https://datafordeler.dk/vejledning/grunddata/datamodel/bitemporalitet/
EPSG:25832 Koordinatsystem	https://epsg.io/25832
GeoServer CQL Tutorial	https://docs.geoserver.org/stable/en/user/tutorials/cql/cql_tutorial.html
GeoServer ECQL Reference	https://docs.geoserver.org/stable/en/user/filter/ecql_reference.html#filter-ecql-reference
QGIS	https://qgis.org/

Tabel 1: Eksterne links



1.2 Begreber

I dette dokument bruges begreberne i Tabel 2: Bemærk at begreberne også er yderligere forklaret i løbet af dokumentet med eksempler.

Begreb	Beskrivelse
WFS	Web Feature Service – OGC-standard protokol for udveksling af geografiske vektordata
OGC	Open Geospatial Consortium - international standardiseringsorganisation for geografiske data
Feature	En geografisk enhed med geometri og attributter (f.eks. en bygning eller et vejstykke)
Lag	Et lag i WFS der repræsenterer en samling features af samme type. Kaldes også "Layer" i engelsksprogede GIS-programmer
FeatureType	Definition af strukturen for en type features (attributnavne, datatyper mv.)
GML	Geography Markup Language - XML-baseret format til geografiske data
GeoJSON	JSON-baseret format til geografiske data, populært i webapplikationer
CQL	Common Query Language - tekstbaseret forespørgselsprog til filtrering (GeoServer-specifik udvidelse)

Tabel 2: Begrebsliste.



2 WFS på Datafordeleren

Datafordeleren tilbyder Web Feature Service (WFS)-tjenester til distribution af geografiske vektordata fra danske registre. WFS er en international OGC-standard, der muliggør struktureret udveksling af geodata mellem systemer og GIS-applikationer.

Datafordeleren udstiller entitetsbaserede WFS-tjenester, som er automatisk genererede baseret direkte på registrenes datamodeller. Tjenesterne følger OGC WFS 2.0.0 standarden. Denne guide fokuserer på hvordan disse tjenester anvendes i praksis.

Yderligere information om WFS og GeoServer kan findes på:

- WFS: [Web Feature Service \(WFS\) Standard | OGC Publications](#)
- GeoServer: [GeoServer](#)

2.1 Om entitetsbaserede WFS-tjenester

Entitetsbaserede WFS-tjenester repræsenterer en automatiseret tilgang til at eksponere geodata, hvor hver tabel i et registers datamodel bliver direkte tilgængelig som et WFS-lag gennem en standardiseret proces. Dette koncept bygger på OGCs WFS 2.0.0 standard, men implementerer en én-til-én mapping mellem databasetabeller og WFS FeatureTypes.

Grundprincippet er at data eksponeres præcis som det ligger i registrenes databaser - uden forudgående filtrering, aggregering eller transformation. Hver entitet i datamodellen bliver automatisk et tilgængeligt lag i WFS-tjenesten. Dette betyder at hvis et register har en tabel kaldet "bygning", vil denne automatisk blive eksponeret som et lag kaldet "bygning_current" for aktuelle data og "bygning_hist" for historiske data.

Tilgangen inkluderer både entiteter med og uden geospatiale data. Dette betyder at selv administrative tabeller som Matriklens Jordstykke-entitet, der ikke indeholder geospatiale data, bliver tilgængelige gennem den samme standardiserede WFS-grænseflade med geometry-feltet sat til null.

2.1.1 Forskelle fra traditionelle/sammenstillede WFS-tjenester

Datafordeleren tilbyder to forskellige tilgange til WFS-tjenester:

1. **Sammenstillede WFS-tjenester:** De eksisterende WFS-tjenester, som er migreret fra Snowflake GoPublisher til GeoServer-plattformen. Disse tjenester er manuelt konfigurerede med specifikke lag, visninger og attributmappings optimeret til bestemte anvendelsesformål. Eksempler inkluderer DAGI Multigeometri, DHM Højdekurver, Fikspunkter og GeoDanmark Vektor tjenesterne. Disse tjenester leverer data gennem foruddefinerede lag med udvalgte felter og strukturer.
2. **Entitetsbaserede WFS-tjenester:** Automatisk genererede tjenester hvor hver tabel i registrenes datamodel bliver direkte tilgængelig som et WFS-lag. Alle felter i databasetabellen bliver til attributter i WFS-laget med identiske navne og datatyper. For entiteter med bitemporal datamodel tilbydes både aktuelle data (`_current`) og komplet historisk sporbarhed (`_hist`). Denne tilgang kræver ingen manuel konfiguration og opdateres automatisk når datamodellerne ændres.

Begge typer kører på GeoServer og følger OGC WFS 2.0.0 standarden. Denne guide fokuserer på entitetsbaserede WFS-tjenester.

2.2 Udstilling af WFS-tjenester

WFS-tjenesterne udstilles gennem URL'er, der følger formen:

`https://wfs.datafordeler.dk/<register>/<register>_wfs/<version>/wfs?<query-parametre>`

Figur 1: WFS URL oversigt



<register> er forkortelsen for det register anvenderen forsøger at hente data fra, <register>_WFS er servicenavnet der følger et fast mønster, <version> er versionen af tjenesten for det pågældende register (aktuelt 1.0.0 for alle initiale services) og <query-parametre> er de forskellige parametre hver tjeneste kan tage. Parametrene er yderligere beskrevet i afsnit 2.3, 2.4, 2.5 og 2.6 og versionering beskrives i afsnit 2.8.

De registre, der udstiller én eller flere entiteter gennem de entitetsbaserede WFS-tjenester, er vist i Tabel 3: nedenfor. Bemærk at et register kan vælge, hvilke entiteter de ønsker udstillet gennem tjenesterne, og at det derfor ikke nødvendigvis er alle entiteter i et register der kan tilgås gennem disse tjenester.

Register	Forkortelse	Service Version	Eksempel URL
Bygnings- og Boligregistret	BBR	1.0.0	https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS
Danmarks Administrative Geografiske Inddeling	DAGI	1.0.0	https://wfs.datafordeler.dk/DAGI/DAGI_WFS/1.0.0/WFS
Danmarks Adresseregister	DAR	1.0.0	https://wfs.datafordeler.dk/DAR/DAR_WFS/1.0.0/WFS
DHM Højdekurver	DHMHøjdekurver	1.0.0	https://wfs.datafordeler.dk/DHMHøjdekurver/DHMHøjdekurver_WFS/1.0.0/WFS
DHM	DHMOprindelse	1.0.0	https://wfs.datafordeler.dk/DHMOprindelse/DHMOprindelse_WFS/1.0.0/WFS
Danmarks Fikspunktsregister	FIKSPUNKT	1.0.0	https://wfs.datafordeler.dk/FIKSPUNKT/FIKSPUNKT_WFS/1.0.0/WFS
GeoDanmark Vektor	GEODKV	1.0.0	https://wfs.datafordeler.dk/GEODKV/GEODKV_WFS/1.0.0/WFS
Matriklen	MAT	1.0.0	https://wfs.datafordeler.dk/MAT/MAT_WFS/1.0.0/WFS

Tabel 3: Oversigt over registre der udstilles via WFS

Nogle registre har data opdelt i flere logiske enheder kaldet replikeringskanaler. For eksempel har DHM separate kanaler for Højdekurver og Oprindelse. Disse eksponeres som separate WFS-services med hver deres URL som vist i tabellen ovenfor.

2.3 Standardparametre

Alle WFS-forespørgsler består af en base-URL og påkrævede query-parametre. Base-URL'en følger strukturen:

```
https://wfs.datafordeler.dk/<register>_WFS/<serviceVersion>/WFS
```

Figur 2: Base url for WFS



Query-parametrene inkluderer tre standard OGC-parametre (service-type, protokolversion og ønsket funktion) samt autentifikation. Disse parametre skal altid inkluderes som query-parametre for at få et gyldigt svar fra tjenesten.

Følgende query-parametre skal altid angives ved WFS forespørgsler:

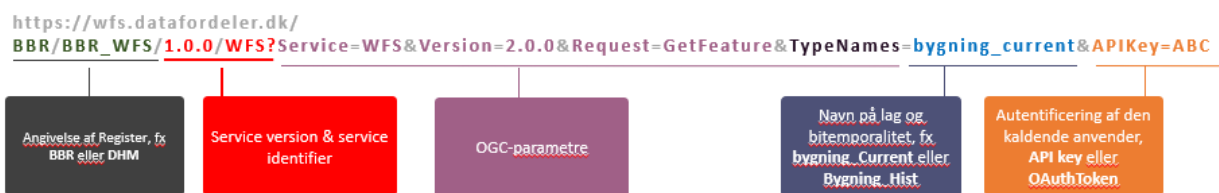
Parameter	Obligatorisk	Mulige inputparametre	Beskrivelse
service	Ja	WFS	Servicetype
OGC-version	Ja	2.0.0	Versionen af WFS-protokollen
request	Ja	GetCapabilities/ DescribeFeatureType/ GetFeature	Navnet på WFS-metode
apikey/Oauth token	Ja	Din api-nøgle/token	Autentifikation. Se afsnit 3.

Tabel 4: Standard WFS-parametre

Det er vigtigt at skelne mellem to forskellige versionsnumre i WFS-forespørgsler. Service-versionen i URL'en (f.eks. /1.0.0/) angiver version af servicen, mens WFS-protokol-versionen i parametrene (version=2.0.0) angiver versionen af WFS-standarden. WFS-protokol-versionen skal altid være 2.0.0, uanset hvilken serviceversion der tilgås.

I eksemplet nedenfor bruges:

- **Register:** BBR
- **Service version:** 1.0.0
- **OGC parametre**
 - Service: WFS
 - WFS-protokol-version: 2.0.0
 - Metode: GetFeature
- **Entitet_bitemporalitet:** Bygning_current



72

Figur 3: URL opbygning - Oversigt

Som vist i eksemplet ovenfor, i Figur 3, indeholder URL'en både service-versionen (1.0.0) og WFS-protokol-versionen (2.0.0) på forskellige steder.



2.4 WFS-Metoder

WFS-tjenesterne understøtter standard OGC-metoder der giver mulighed for at opdage tilgængelige data, forstå datastrukturen og hente de faktiske geografiske features. Hver metode har et specifikt formål i arbejdet med geografiske data.

Metod	Formål	Typisk anvendelse	Returnerer
GetCapabilities	Henter tilgængelige lag og service metadata	Første kald til en WFS-service	XML metadata
DescribeFeatureType	Henter datastrukturen for specifikke lag	Kaldes før opbygning af queries	XSD-skema
GetFeature	Henter faktiske geografiske data	Dataudtræk og analyse	GML/GeoJSON

Tabel 5: Oversigt over WFS-metoder

For yderligere information om WFS-metoder, GeoServer WFS Reference: [GeoServer's officielle dokumentation](#).

2.4.1 GetCapabilities

GetCapabilities-metoder bruges til at hente oplysning om de tilgængelige lag der er i en WFS-tjeneste samt hvilke metoder og formater der understøttes for disse lag. Dette er typisk det første kald man laver til en ny WFS-tjeneste for at forstå hvad der er tilgængeligt. GetCapabilities kan kaldes med ingen yderligere parametre end de standardparametre beskrevet i Tabel 4.

Eksempel:

```
https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?service=WFS&version=2.0.0&request=GetCapabilities&apikey={YOUR_API_KEY}
```

Figur 4: GetCapabilities request til BBR

Kald til GetCapabilities returnerer et længere XML-dokument der beskriver de pågældende tjenester. Kaldet returner blandt andet metadata om:

- Understøttede outputformater (GML, GeoJSON)
- Geografisk udstrækning for hvert lag
- Maksimalt antal features der kan returneres (10.000)
- Understøttede koordinatsystemer (EPSG:25832 som standard)

Læs mere om GeoServer GetCapabilities i [GeoServer dokumentationen](#).

2.4.2 DescribeFeatureType

DescribeFeatureType-metode returnerer den detaljerede struktur i form af et skema for et lag. Dette bruges til at forstå, hvilke attributter der er tilgængelige, deres datatyper og eventuelle begrænsninger. GIS-software bruger ofte denne metode automatisk for at konfigurere attributtabeller korrekt. DescribeFeatureType bruger de samme standardparametre som beskrevet i Tabel 4 (service, version, request, apikey) plus følgende yderligere parameter beskrevet i Tabel 6 for at specificere hvilken entitet du vil se strukturen for.



Parameter	Standardværdi	Beskrivelse
typeName	Alle lag	Den featureType der ønskes beskrevet

Tabel 6: Parametre for DescribeFeatureType

For entitetsbaserede tjenester følger lagnavnene mønsteret {entitet}_{bitemporalitet} hvor entitet er bygning, tekniskanlæg, adresse osv., og bitemporalitet er enten current (aktuelle data) eller hist (historiske data). Eksempler: bygning_current, tekniskanlæg_hist, adresse_current.

Brug af typeName-parameteren:

- Uden typeName: Henter skemaer for alle tilgængelige lag samtidig - nyttigt for at få komplet overblik over servicens datamodel
- Med typeName: Henter kun skema for det specifikke lag - hurtigere og mere fokuseret når du ved præcis hvilket lag du skal arbejde med

2.4.2.1 For at kunne finde en korrekte typeName-værdier:

1. Kald GetCapabilities for at se alle tilgængelige lag, for eksempel:

```
https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?service=WFS&version=2.0.0&request=GetCapabilities&apikey={YOUR_API_KEY}
```

2. Find lagnavne i FeatureTypeList i GetCapabilities-responset:

```
</ows:OperationsMetadata>
<FeatureTypeList>
  <FeatureType xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3">
    <Name>bbr_v001:bygning_current</Name>
    <Title>bygning_current</Title>
    <Abstract>bygning_current</Abstract>
    <DefaultCRS>urn:ogc:def:crs:EPSG::25832</DefaultCRS>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-60.0 54.5</ows:LowerCorner>
      <ows:UpperCorner>15.2 84.0</ows:UpperCorner>
    </ows:WGS84BoundingBox>
  </FeatureType>
  <FeatureType xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3">
    <Name>bbr_v001:bygning_hist</Name>
    <Title>bygning_hist</Title>
    <Abstract>bygning_hist</Abstract>
    <DefaultCRS>urn:ogc:def:crs:EPSG::25832</DefaultCRS>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-60.0 54.5</ows:LowerCorner>
      <ows:UpperCorner>15.2 84.0</ows:UpperCorner>
    </ows:WGS84BoundingBox>
  </FeatureType>
```

Figur 5: GetCapabilities-svar fra BBR

3. Brug lagnavnet direkte som vist i FeatureTypeList **<Title>**, f.eks. bygning_current eller bygning_hist og du vil modtage den detaljerede struktur, du anmodede om

DescribeFeatureType eksempel med bygning_current:

```
https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?service=WFS&version=2.0.0&request=DescribeFeatureType&typeName=bygning_current&apikey={YOUR_API_KEY}
```



```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" targetNamespace="http://datafordeler.dk/schemas/bbr/v001/gml3">
  <xsd:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="https://test07-wfs.datafordeler.dk/geoserver/schemas/gml/3.2.1/gml.xsd"/>
  <xsd:complexType name="bygning_currentType">
    <xsd:complexContent base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningshaendelse" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningsomraade" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningsproces" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="id_namespace" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="id_lokalid" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="kommunekode" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="registreringFra" nillable="true" type="xsd:dateTime"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="registreringsaktør" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="registreringTil" nillable="true" type="xsd:dateTime"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="virkningFra" nillable="true" type="xsd:dateTime"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="virkningsaktør" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="virkningTil" nillable="true" type="xsd:dateTime"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="status" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg007Bygningsnummer" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg021BygningsAnvendelse" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg024AntalletJlighederMedKøkken" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg025AntalletJlighederUdenKøkken" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg026Opførelsesår" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg027OmTilbygningsår" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg029DatoForMidlertidigOpførtBygning" nillable="true" type="xsd:dateTime"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg030Vandforsyning" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg031Afløbsforhold" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg032YdervæggensMateriale" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg033TagdækningsMateriale" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg034SupplerendeYdervæggensMateriale" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg035SupplerendeTagdækningsMateriale" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg036AsbestholdigtMateriale" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg037KildeTilBygningsMaterialer" nillable="true" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg038SamletBygningsareal" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg039BygningsSamledeBoligAreal" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg040BygningsSamledeErhvervsAreal" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg041BebyggetAreal" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg042ArealIndbyggetGarage" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg043ArealIndbyggetCarport" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg044ArealIndbyggetUdhus" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg045ArealIndbyggetUdestueEllerLign" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg046SamletArealAfLukkedeOverdækningerPåBygningen" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg047ArealAfAflukkbareTerasser" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg048AndetAreal" nillable="true" type="xsd:long"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="byg049ArealAfOverdækketAreal" nillable="true" type="xsd:long"/>
      </xsd:sequence>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Figur 6: DescribeFeatureType-svar fra BBR

Læs mere om DescribeFeatureType i [GeoServer dokumentationen](#).

2.4.3 GetFeature

GetFeature-metode bruges til at hente de faktiske geografiske data. Dette er den primære metode når man skal arbejde med data i GIS-software eller lave analyser. GetFeature understøtter forskellige filtrerings- og udvælgelsesmuligheder gennem query-parametre. Alle inputparametre for GetFeature findes i Tabel 7:

Parameter	Obligatorisk?	Standardværdi	Beskrivelse
typeName	Ja	-	Den featureType der ønskes beskrevet
count	Nej	10000	Maksimalt antal features der ønskes returneret (1-10000)
startIndex	Nej	0	Startposition for paging
outputFormat	Nej	application/gml+xml; version=3.2	Outputformatet på data (GML eller GeoJSON). Se afsnit 2.5
bbox	Nej	-	Bounding box for geometrisk filtrering (kan ikke kombineres med cql_filter). Se afsnit 2.7.1.
cql_filter	Nej	-	CQL (Common Query Language) expression for attribut-/geometrisk filtrering (kan ikke kombineres med bbox). Se afsnit 2.7.2.
sortBy	Nej	-	Navnet på feltet der ønskes sorteret på. Tilføj +D (descending) for omvendt sortering.

Tabel 7: Parametre for GetFeature



GetFeature understøtter flere avancerede query-muligheder, såsom:

- Paging: Via startIndex- og count-parameteren
- Sortering: Via sortBy-parameteren
- Attribut-selektion: Via propertyName-parameteren
- Geometrisk filtrering: Via bbox-parameteren eller CQL geometriske operatorer
- Attribut-filtrering: Via cql_filter-parameteren

Eksempel:

```
https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?service=WFS&version=2.0.0&request=GetFeature&typeName=tek  
niskanlaeg_current&count=100&outputFormat=application/json&apikey={YOUR_API_KEY}
```

Figur 7: Eksempel på GetFeature request til bbr, tekniskanlaeg

Læs mere om GetFeature og alle query-muligheder i [GeoServer dokumentationen](#).

For en mere detaljeret beskrivelse af brugen af bbox- og cql_filter-parameteren, se afsnit 2.7.

2.5 Outputformater

WFS-tjenesterne understøtter to standard-outputformater der dækker forskellige anvendelsesbehov. Valg af format afhænger af hvilken type applikation eller GIS-software der skal forbruge data.

Format	MIME Type	Parameterværdi	Typisk anvendelse
GML	application/gml+xml;version=3.2	gml32	Desktop GIS, arkivering, fuld præcision
GeoJSON	application/json	application/json	Webapplikationer, JavaScript, REST APIs

Tabel 8: Understøttede outputformater

Outputformat specificeres via outputFormat-parameteren i GetFeature- og DescribeFeatureType-forespørgsler. Hvis ingen outputFormat-parameter angives, returneres data i henholdsvis GML format som standard for GetFeature, og som XSD-skema som standard for DescribeFeatureType.

For GetFeature returneres geografiske data i det valgte format (GML eller GeoJSON). For DescribeFeatureType returneres strukturbeskrivelsen enten som XSD-skema (standard) eller som JSON-skema når outputFormat=application/json specificeres.

GetCapabilities-metoden understøtter ikke outputFormat-parameteren og returnerer altid XML uanset hvilken værdi der angives.

2.5.1 Valg af format

GML bør vælges når:

- Data skal bruges i desktop GIS-software som eksempelvis QGIS eller ArcGIS



- Fuld OGC-compliance er påkrævet
- Data skal arkiveres eller udveksles mellem offentlige systemer
- Maksimal præcision og metadata er vigtig

GeoJSON bør vælges når:

- Data skal bruges i webapplikationer eller JavaScript
- Integration med moderne REST APIs
- Filstørrelse er en begrænsning (GeoJSON er typisk 20-30% mindre)
- Hurtig parsing og simpel datastruktur er prioriteret

For yderligere information om output formater i GeoServer se: [WFS output formats](#)

2.6 Bitemporalitet

Entitetsbaserede WFS-tjenester håndterer bitemporal data gennem separate lag for aktuelle og historiske data. Alle entiteter med bitemporal datamodel eksponeres i to varianter, identificeret gennem endelsen på lagnavnet:

- **_current**: Viser kun aktuelt gyldige data
- **_hist**: Viser komplet bitemporal historik med alle versioner over tid

Det aktuelle lag, for eksempel bygning_current, returnerer kun features der er gyldige på nuværende tidspunkt. Dvs. de underliggende data filtreres automatisk på både virkningstid og registreringstid for at vise det aktuelle billede af data.

Det historiske lag, for eksempel bygning_hist, returnerer alle versioner af hver feature gennem tiden, inklusive både aktuelle og historiske versioner. Dette giver mulighed for at analysere hvordan data har ændret sig over tid. Det historiske lag giver mulighed for at filtrere på de bitemporale felter for at finde gyldige data i specifikke tidsperioder. De relevante felter er:

- registreringFra / registreringTil: Hvornår data blev registreret i systemet
- virkningFra / virkningTil: Hvornår data er/ var gyldigt i virkeligheden

Man kan filtrere på disse felter ved brug af cql_filter-parameteren, som er beskrevet i afsnit 2.7.2.1.

For yderligere information om [Bitemporalitet på Datafordeleren](#).

2.7 Filtrering

De entitetsbaserede WFS-tjenester understøtter både attribut- og geometrisk filtrering når data hentes ved brug af GetFeature-metoden beskrevet i afsnit 2.4.3. Man kan filtrere på baggrund af udvalgte attributter ved brug af cql_filter-parameteren mens filtrering på baggrund af geospatial data kan gøres på en af to måder: ved brug af enten bbox- eller cql_filter-parameteren. Parametrene kan ikke benyttes samtidig. Brug af bbox-parameteren er beskrevet i afsnit 2.7.1, mens brug af cql_filter-parameteren er beskrevet i afsnit 2.7.2 nedenfor.

2.7.1 Filtrering med bbox

bbox-parameteren gør det muligt at søge efter features, der er indeholdt (eller delvist indeholdt) inden for en såkaldt *bounding box*. En bounding box er en rektangulær, geografisk afgrænsning, som angiver det område indenfor hvilket, der returneres features til anvenderen. Bounding box'en er defineret ved angivelse af to sæt af koordinater, som alle skal være inden for [EPSG-25832](#) grænserne. Nedenfor er vist et eksempel på en bounding box for koordinaterne bbox=721148,617549,722618,6176355



Figur 8: Bounding box-visualisering

Med hensyn til hvilke hjørner af afgrænsningsboksen der skal angives, er det eneste krav, at et af de nederste hjørner (venstre eller højre) angives først. For eksempel nederste venstre og øverste højre, eller nederste højre og øverste venstre. I Figur 8 svarer koordinaterne (minX, minY) til nederst venstre og (maxX, maxY) til øverste højre. Man ville således få samme bounding box ved at angive koordinaterne maxX, minY og minX, maxY, hvilket svarer til nederste højre og øverste venstre hjørne.

Bemærk at bbox-parameteren *ikke* kan benyttes samtidig med cql_filter-parameteren.

Hvis man for eksempel vil finde tekniske anlæg inden for bounding box'en ovenfor, kan man benytte bbox-parameteren til at lave en filtrering på tekniskanlæg_current-laget. Dette kan gøres ved følgende kald:

```
https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?service=WFS&version=2.0.0&request=GetFeature&typeName=tekniskanlæg_current&bbox=721148,617549,722618,6176355&apikey={YOUR_API_KEY}
```

Figur 7: GetFeature request til bbr med tekniskanlæg_current

Kaldet returnerer en lang XML-fil med følgende data:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection ... >
  <wfs:member>
    <bbr_v001:tekniskanlæg_current gml:id= ... > # ID
    <bbr_v001:forretningshændelse>TekniskAnlæg</bbr_v001:forretningshændelse>
    <bbr_v001:forretningsområde>54.15.05.05</bbr_v001:forretningsområde>
    <bbr_v001:forretningsproces>25</bbr_v001:forretningsproces>
    <bbr_v001:id_namespace>http://data.gov.dk/bbr/tekniskanlæg</bbr_v001:id_namespace>
    <bbr_v001:id_lokalId>fa8c34e6-4d61-474f-969e-4565f4760121</bbr_v001:id_lokalId>
    <bbr_v001:kommunekode>0147</bbr_v001:kommunekode> # Frederiksberg kommune
    ... # Flere properties
  </wfs:member>
  <wfs:member>
    ... # Properties
  </wfs:member>
  ... # Flere features
</wfs:FeatureCollection>
```

Figur 9: Eksempel på XML-respons returneret fra bbox-filtreret GetFeature-forespørgsel for tekniske anlæg



2.7.2 Filtrering med cql_filter

cql_filter-parameteren tilbyder en avanceret og fleksibel metode til at filtrere de features, der returneres fra en WFS-tjeneste. I modsætning til den simple bbox-filtrering, giver cql_filter mulighed for at formulere komplekse forespørgsler baseret på både attribut- og geospatiale data. Filtret skrives ved hjælp af OGC's Common Query Language (CQL). Du kan læse mere om CQL og hvordan det kan bruges i GeoServers [CQL Tutorial](#) og [dokumentation for ECQL](#).

2.7.2.1 Attributfiltrering

Attributfiltrering giver mulighed for at udvælge features baseret på deres attributværdier, såsom tekststreng, tal eller datoer. Filtret opbygges typisk som *attribut-operator-værdi*. Følgende operatører understøttes:

Operator	Beskrivelsen
• = (lig med) • <> (ikke lig med) • < (mindre end) • <= (mindre end eller lig med) • > (større end) • >= (større end eller lig med)	Sammenligningsoperatører
• (NOT) BETWEEN value1 AND value2	Tjekker om en værdi er inden eller uden for et interval (value1, value2).
• (NOT) LIKE • (NOT) ILIKE	Tjekker om en tekststreng (ikke) følger et mønster gennem simpel pattern-matching. LIKE bruger "%" som wild-card-karakter. ILIKE laver case-insensitiv matching.
• (NOT) IN	Tjekker om en værdi (ikke) er indeholdt i et sæt af værdier.
• IS (NOT) NULL	Tjekker om en værdi (ikke) er null.

Tabel 9: Understøttede operatører for attributfiltrering

Hvis man for eksempel vil finde tekniske anlæg der var gyldige pr. 1. januar 2023, kan man benytte cql_filter til at lave bitemporal filtrering på tekniskanlaeg_hist-laget. Dette kan gøres ved følgende kald:

```
https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?service=WFS&version=2.0.0&request=GetFeature&typeName=tekniskanlaeg_hist&cql_filter=virkningFra <= '2023-01-01' AND (virkningTil IS NULL OR virkningTil > '2023-01-01') AND registreringTil IS NULL&outputFormat=application/json&apikey=avcaqTElssKjaj2f90GTVqrOcb38tmDuTBtEi9ITe5bkdSDQxXJHTTrfc6iHwz7d8MekUustEb8ftzbJAII04NQwfHOeRvRBj&count=1
```

Figur 10: Eksempel på GetFeature-forespørgsel med bitemporal CQL-filtrering

I dette kald skal cql_filter-filtrene læses således:

- Returnerer kun data der var gyldigt den 1. januar 2023 eller før (virkningFra <= '2023-01-01').
- Returnerer kun data der stadig er gyldigt (virkningTil IS NULL) eller (OR) som var gyldigt indtil efter 1. januar 2023 (virkningTil > '2023-01-01')
- Returnerer kun data der har en registrering der stadig er gyldig (registreringTil IS NULL)



Som vist i kaldet ovenfor, kombineres de forskellige filtre ved hjælp af AND/OR-operatorer.

Bemærk at outputFormat-parameteren er sat til *json*, hvorfor kaldet returnerer en lang json-fil med følgende data:

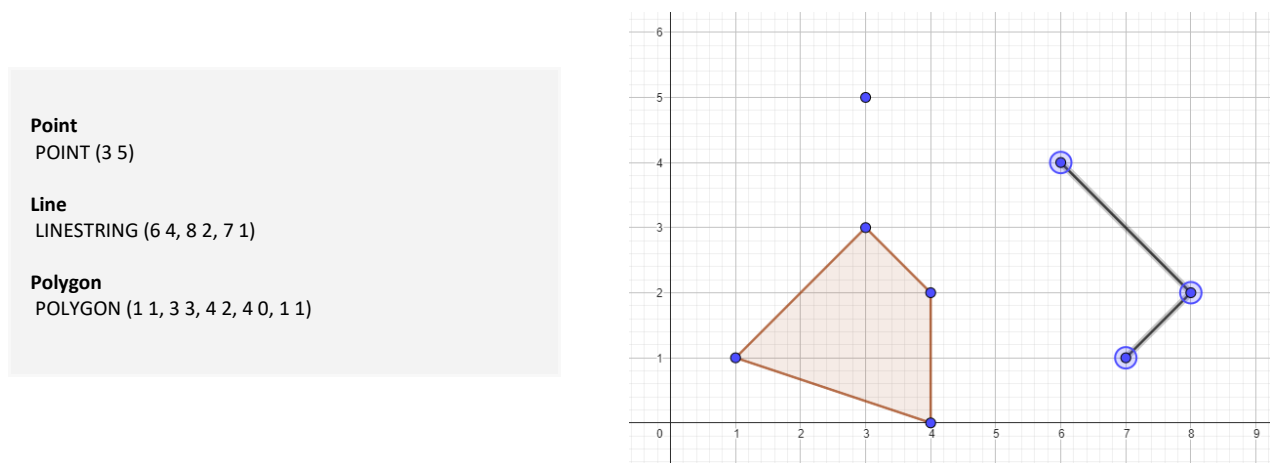
```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "tekniskanlaeg_hist.fid-6429244c_1989dfaf394_-32a4",
      "geometry": null,
      "geometry_name": "tek109Koordinat",
      "properties": {
        "forretningshaendelse": "TekniskAnlæg",
        "forretningsomraade": "54.15.05.05",
        "forretningsproces": "0",
        "id_namespace": "http://data.gov.dk/bbr/tekniskanlaeg",
        "id_lokalId": "1039c906-7847-4300-b9c2-01cd132ea689",
        "kommunekode": "0330",
        "registreringFra": "2017-06-02T12:29:37Z",
        "registreringsaktør": "BBR",
        "registreringTil": null, # IS NULL
        "virkningFra": "2012-03-09T07:15:38.180Z", # <= '2023-01-01'
        "virkningsaktør": "Konvertering2017",
        "virkningTil": null, # IS NULL
        ... # Flere properties
      }
    },
    {
      ... # Properties
    },
    ... # Flere features
  ]
}
```

Figur 11: JSON-respons fra GetFeature-forespørgsel med bitemporal CQL-filtrering

2.7.2.2 Geometrisk filtrering

Udover attributfiltrering kan `cql_filter` også bruges til at filtrere features på baggrund af geospatiale data. Dette gøres i form af WKT-strenger (Well Known Text) som gives som input gennem `cql_filter`-parameteren. Disse WKT'er repræsenterer geospatiale data, enten i form af en koordinattype, en geometri-type eller lignende datatyper.

De mest almindelige typer af geometriske former er punkter, linjer og polygoner. Et punkt repræsenteres af et koordinatsæt med to eller tre værdier, afhængigt af om det er et punkt i to eller tre dimensioner. En linje repræsenteres af en liste af de punkter, der udgør linjen. En polygon repræsenteres af en liste af de punkter, der udgør hjørnerne af polygonen, hvor første og sidste punkt i listen er identisk. Et eksempel på WKT-formatet for hver af de tre geometrityper i 2D, er vist i Figur 12:



Figur 12: Eksempler på 2D-geometrier

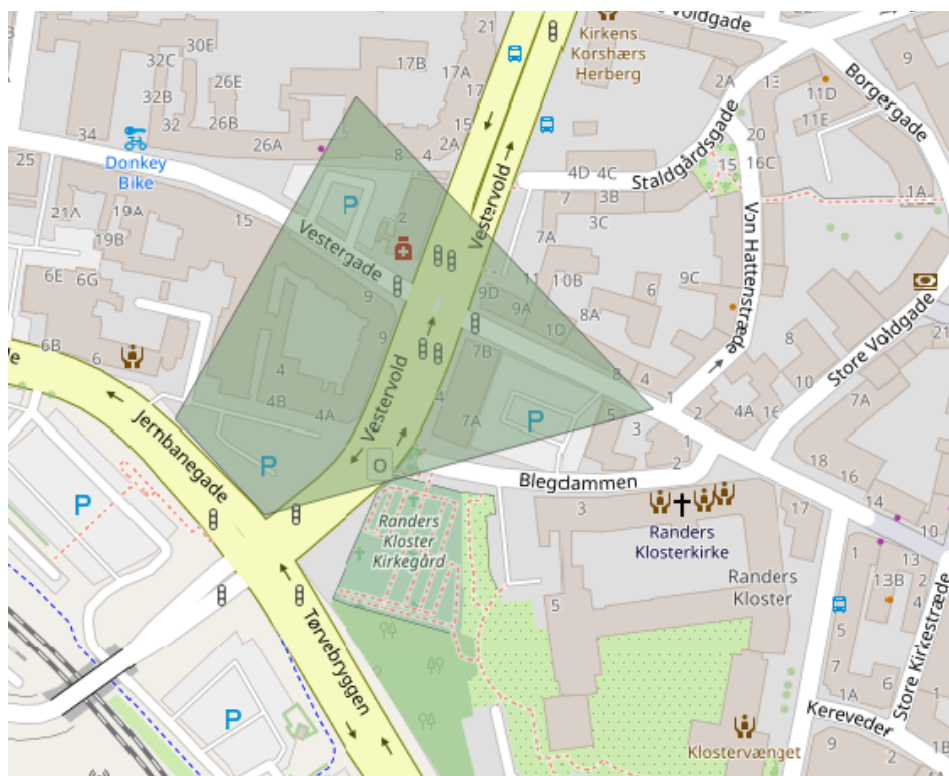


De entitetsbaserede WFS-tjenester understøtter en række forskellige geometriske operatoren. Alle operatoren fremgår af Tabel 10: . Kombinationer af geometriske operatoren understøttes ved hjælp af tilhørende AND/OR-operatoren.

Operator	Beskrivelsen
BBOX(geometri, minX, minY, maxX, maxY)	Tjekker om den angivne geometri <u>overlapper</u> med den angivne bounding box
INTERSECTS(geometri, wkt)	Tjekker om den angivne geometri <u>overlapper</u> med den angivne WKT-geometri
WITHIN(geometri, wkt)	Tjekker om den angivne geometri <u>er indeholdt</u> i den angivne WKT-geometri
CONTAINS(geometri, wkt)	Tjekker om den angivne geometri <u>indeholder</u> i den angivne WKT-geometri
DWITHIN(geometri, wkt, dist, units)	Tjekker om den angivne geometri og den angivne WKT-geometri maksimalt er den angivne distance "dist" fra hinanden. "dist" tager en numerisk værdi og angiver den maksimale distance mellem de to geometrier og "units" er enheden distancen er angivet i. "units" kan tage af følgende: feet, meters, statute miles, nautical miles, kilometers

Tabel 10: Understøttede geometriske filtreringsfunktioner

Nedenfor er vist et eksempel på en WKT-geometri: POLYGON ((563436.97 6257781.63, 563505.88 6257908.45, 563623.98 6257788.20, 563472.62 6257744.47, 563436.97 6257781.63))



Figur 13: Polygon-visualisering



I dette eksempel bruges tek109Koordinat, som er geometrifeltet for tekniske anlæg (som vist i JSON-responset i afsnit 2.7.2.1). Hvis man for eksempel vil finde alle tekniske anlæg, hvor geometrien tek109Koordinat er indeholdt i polygonen oven for, kan man benytte WITHIN-operatorer til at lave en filtrering på tekniskanlaeg_current-laget. Dette kan gøres ved følgende kald:

- [https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?request=GetFeature&typeName=tekniskanlaeg_current&cql_filter=WITHIN\(tek109Koordinat,POLYGON\(\(563436.97 6257781.63, 563505.88 6257908.45, 563623.98 6257788.20, 563472.62 6257744.47, 563436.97 6257781.63\)\)\)&apikey={YOUR_API_KEY}](https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?request=GetFeature&typeName=tekniskanlaeg_current&cql_filter=WITHIN(tek109Koordinat,POLYGON((563436.97%206257781.63,%20563505.88%206257908.45,%20563623.98%206257788.20,%20563472.62%206257744.47,%20563436.97%206257781.63)))&apikey={YOUR_API_KEY})

Kaldet returnerer en lang XML-fil med følgende data:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection ... >
  <wfs:member>
    <bbr_v001:tekniskanlaeg_current gml:id= ... > # ID
      <bbr_v001:forretningshaendelse>TekniskAnlaeg</bbr_v001:forretningshaendelse>
      <bbr_v001:forretningsomraade>54.15.05.05</bbr_v001:forretningsomraade>
      <bbr_v001:forretningsproces>5</bbr_v001:forretningsproces>
      ... # Flere properties
      <bbr_v001:tek109Koordinat>
        <gml:Point srsName="urn:ogc:def:crs:EPSG::25832" srsDimension="3" gml:id= ... > # ID
          <gml:pos>563494.18 6257813.71 0</gml:pos>
        </gml:Point>
      </bbr_v001:tek109Koordinat>
      ... # Flere properties
    </wfs:member>
    ... # Flere features
  </wfs:FeatureCollection>
```

Figur 14: XML-respons fra GetFeature-forespørgsel med WITHIN geometrisk filtrering

2.8 Versionering

I de entitetsbaserede WFS-tjenester versioneres hvert register separat, og alle entiteter, der tilhører det samme register, versioneres under det samme versionsnummer som registret. Det vil sige, at DAR og BBR kan eksistere separat som DAR_WFS/1.0.0 og BBR_WFS/2.0.0. Men hvis BBR har en ændring til Bygning-entiteten i deres datamodel, vil alle entiteter i BBR blive opgraderet til BBR_WFS/3.0.0, og BBR_WFS/2.0.0 vil derefter efter en overgangsperiode blive udfaset.

2.9 Sådan anvender du entitetsbaserede WFS-tjenester

De entitetsbaserede WFS-tjenester kan til anvendes på to forskellige måder: Gennem separate REST-kald og gennem GIS-klienter, som QGIS, ArcGIS og MapInfo. Anvendelsesmønsteret for brug af de entitetsbaserede WFS-tjenester forløber i begge tilfælde således:

1. **GetCapabilities:** Hent oplysninger om de tilgængelige lag for det pågældende register
2. **DescribeFeatureType:** Hent metadataskema for det pågældende lag
3. **GetFeature:** Hent data for det pågældende lag

Selvom det er nøjagtigt samme data der returneres i begge tilfælde, er arbejdsgangen ved brug af separate REST-kald og GIS-klienter lidt forskellig.



2.9.1 Separate REST-kald

Hvis man vil tilgå de entitetsbaserede WFS-tjenester gennem almindelige REST-kald, skal man først kalde den pågældende WFS-tjeneste med GetCapabilities-metoden:

Metoden: GetCapabilities	
URL	<a href="https://wfs.datafordeler.dk/<register>/<register>_WFS/<version>/WFS?service=WFS&version=2.0.0&request=GetCapabilities&apikey={YOUR_API_KEY}">https://wfs.datafordeler.dk/<register>/<register>_WFS/<version>/WFS?service=WFS&version=2.0.0&request=GetCapabilities&apikey={YOUR_API_KEY}
HTTP-metode	GET
Returværdier	• Returnerer HTTP 200 – OK, ved succes. • Returnerer HTTP 400 – Bad Request, ved angivelse af forkerte parametre. • Returnerer HTTP 404 – Not Found, hvis den angivne ressource ikke kan findes. • Returnerer HTTP 500 – Internal Server Error, hvis der er sket en ukendt fejl
Adgang	IT-system med API-nøgle, OAuth Shared Secret eller OAuth Certifikat.

Tabel 11: API-specifikation for GetCapabilities-metoden

GetCapabilities kan kaldes med følgende parametre:

Parameter	Type	Beskrivelse	Obligatorisk?
register	String	Angiver hvilket register der skal hentes oplysninger for. Se Tabel 2 for en oversigt over registre der udstilles via WFS.	Ja
service-version	String	Angiver hvilken serviceversion af tjenesten der kaldes	Ja
service	String	Angiver hvilken service man forsøger at kalde. Kaldes kun værdien "WFS".	Ja
OGC-version	String	Angiver hvilken OGC WFS-version der anvendes. Kan kaldes kun med værdien "2.0.0"	Ja

Tabel 12: Parametre for GetCapabilities-metoden

Denne tjeneste returnerer altid en xml-fil ved HTTP 200 – OK. Filen indeholder metadata om WFS-tjenesten, der beskriver hvilke lag der findes for det pågældende register.

Et eksempel på et kald til tjenesten ved brug af API-nøgle:

- https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0&WFS?service=WFS&version=2.0.0&request=GetCapabilities&apiKey={YOUR_API_KEY}

Svaret returnerer en længere XML-fil, som blandt andet indeholder elementet FeatureTypeList:



```
<FeatureTypeList>
  <FeatureType xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3">
    <Name>bbr_v001:tekniskanlaeg_current</Name>
    <Title>tekniskanlaeg_current</Title>
    <Abstract>bbr_tekniskanlaeg_v001_current</Abstract>
    <DefaultCRS>urn:ogc:def:crs:EPSG::25832</DefaultCRS>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-60.0 54.5</ows:LowerCorner>
      <ows:UpperCorner>15.2 84.0</ows:UpperCorner>
    </ows:WGS84BoundingBox>
  </FeatureType>
  <FeatureType xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3">
    <Name>bbr_v001:tekniskanlaeg_hist</Name>
    <Title>tekniskanlaeg_hist</Title>
    <Abstract>bbr_tekniskanlaeg_v001_hist</Abstract>
    <DefaultCRS>urn:ogc:def:crs:EPSG::25832</DefaultCRS>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-60.0 54.5</ows:LowerCorner>
      <ows:UpperCorner>15.2 84.0</ows:UpperCorner>
    </ows:WGS84BoundingBox>
  </FeatureType>
</FeatureTypeList>
```

Figur 15: FeatureTypeList fra GetCapabilities-respons for BBR

Fra denne liste fremgår det at der for BBR udstilles to lag:

- tekniskanlaeg_current
- tekniskanlaeg_hist

Ønsker man at hente data for et af disse lag kan man således bruge DescribeFeatureType-metoden til først at hente information om det specifikke lag:

Metoden: DescribeFeatureType	
URL	https://wfs.datafordeler.dk/<register>/<register>_WFS/<service-version>/WFS?service=<service>&version=<OGC-version>&request=DescribeFeatureType&typeName=<typeName>
HTTP-metode	GET
Returværdier	• Returnerer HTTP 200 – OK, ved succes. • Returnerer HTTP 400 – Bad Request, ved angivelse af forkerte parametre. • Returnerer HTTP 404 – Not Found, hvis den angivne ressource ikke kan findes. • Returnerer HTTP 500 – Internal Server Error, hvis der er sket en ukendt fejl
Adgang	IT-system med API-nøgle, OAuth Shared Secret eller OAuth Certifikat.

Tabel 13: API-specifikation for DescribeFeatureType-metoden



DescribeFeatureType kan kaldes med følgende parametre:

Parameter	Type	Beskrivelse	Obligatorisk?
register	String	Angiver hvilket register der skal hentes oplysninger for. Se Tabel 2 for en oversigt over registre der udstilles via WFS.	Ja
service-version	String	Angiver hvilken serviceversion af tjenesten der kaldes	Ja
service	String	Angiver hvilken service man forsøger at kalde. Kaldes kun med værdien "WFS".	Ja
Metode	String	DescribeFeatureType + TypeName	
OGC-version	String	Angiver hvilken OGC WFS-version der anvendes. Kaldes kun med værdien "2.0.0"	Ja
typeName	String	Angiver hvilket lag der skal hentes metadata for. Hvis denne ikke angives, hentes metadata for alle tilgængelige lag.	Nej

Tabel 14 : Parametre for DescribeFeatureType-kald

Denne tjeneste returnerer altid en xml-fil ved HTTP 200 – OK. Filen indeholder et metadataschema om WFS-tjenesten, der beskriver hvilke data der findes for det pågældende lag.

Et eksempel på et kald til tjenesten ved brug af API-nøgle:

- https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0&WFS?service=WFS&version=2.0.0&request=DescribeFeatureType&typeName=tekniskanlaeg_current&apiKey=placeholderNoegle

Svaret returnerer en længere XML-fil, som blandt andet indeholder elementet complexType:

```
<xsd:complexType name="tekniskanlaeg_currentType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningshaendelse" nillable="true"
type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningsomraade" nillable="true"
type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningsproces" nillable="true"
type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="id_namespace" nillable="true"
type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="id_lokalId" nillable="true"
type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="kommunekode" nillable="true"
type="xsd:string"/>
        ... # Flere properties
        <xsd:element maxOccurs="1" minOccurs="0" name="tek109Koordinat" nillable="true"
type="gml:PointPropertyType"/>
        ... # Flere properties
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Figur 16: XSD-skema fra DescribeFeatureType-respons for tekniskanlaeg_current

Herfra kan man læse hvilke attributter der eksisterer på et givent lag og hvilken type den pågældende attribut er. Man kan bruge oplysningerne fra dette kald til at sortere eller filtrere data når man efterfølgende kalder WFS-tjenesterne med GetFeature-metoden. Bemærk at tek109Koordinat er af typen "gml:PointPropertyType", hvilket er en geometri. Man kan derfor i



GetFeature-metoden lave geografisk filtrering af data på baggrund af denne attribut ved hjælp af bbox- eller cql_filter-parameteren.

For at hente data for det pågældende lag, skal man bruge GetFeature-metoden:

Metoden: GetFeature	
URL	<code>https://wfs.datafordeler.dk/<register>/<register>_WFS/<service-version>/WFS?service=<service>&version=<OGC-version>&request=GetFeature&typeName=<typeName>&count=<count>&startIndex=<startIndex>&outputFormat=<outputFormat>&bbox=<bbox>&cql_filter=<cql_filter>&sortBy=<sortBy></code>
HTTP-metode	GET
Returværdier	• Returnerer HTTP 200 – OK, ved succes. • Returnerer HTTP 400 – Bad Request, ved angivelse af forkerte parametre. • Returnerer HTTP 404 – Not Found, hvis den angivne ressource ikke kan findes. • Returnerer HTTP 500 – Internal Server Error, hvis der er sket en ukendt fejl
Adgang	IT-system med API-nøgle, OAuth Shared Secret eller OAuth Certifikat.

Tabel 15: API-specifikation for GetFeature-metoden

GetFeature kan kaldes med følgende parametre:

Parameter	Type	Beskrivelse	Obligatorisk?
register	String	Angiver hvilket register der skal hentes oplysninger for. Se Tabel 2 for en oversigt over registre der udstilles via WFS.	Ja
service-version	String	Angiver hvilken version af servicen registeret kan kalde.	Ja
service	String	Angiver hvilken service man forsøger at kalde. aldes kun med værdien "WFS".	Ja
OGC-version	String	Angiver hvilken OGC WFS-version der anvendes. Kaldes kun med værdien "2.0.0"	Ja
typeName	String	Angiver hvilket lag der skal hentes metadata for. Hvis denne ikke angives, hentes metadata for alle tilgængelige lag.	Nej
count	String	Angiver det maksimal antal features der returneres. Mulige værdier mellem 1 og 10.000.	Nej
startIndex	String	Startposition for paging	Nej
outputFormat	String	Outputformatet på data (GML eller GeoJSON). Se afsnit 2.5.	Nej
bbox	String	Bounding box for geometrisk filtrering (kan ikke kombineres med cql_filter). Se afsnit 2.7.1.	Nej
cql_filter	String	CQL (Common Query Language) expression for attribut-/geometrisk filtrering (kan ikke kombineres med bbox). Se afsnit 2.7.2.	Nej
sortBy	String	Navnet på feltet der ønskes sorteret på. Tilføj +D (descending) for omvendt sortering.	Nej

Tabel 16: Parametre for GetFeature-kald



Denne tjeneste returnerer enten en xml- eller en json-fil ved HTTP 200 – OK. Filen indeholder data fra WFS-tjenesten for det pågældende lag.

Et eksempel på et kald til tjenesten ved brug af API-nøgle:

- https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0&WFS?service=WFS&version=2.0.0&request=GetFeature&typeName=tekniskanlaeg_current&outputFormat=json&apiKey=placeholderNoegle

Svaret returnerer en længere json-fil med følgende data:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "tekniskanlaeg_current.fid--7d828cb6_198a8bc895e_-72d8",
      "geometry": {
        "type": "Point",
        "coordinates": [
          563733.21,
          6242798.55,
          0
        ]
      },
      "geometry_name": "tek109Koordinat",
      "properties": {
        "forretningshaendelse": "TekniskAnlaeg",
        "forretningsomraade": "54.15.05.05",
        "forretningsproces": "4",
        "id_namespace": "http://data.gov.dk/bbr/tekniskanlaeg",
        "id_lokalId": "407bdc4d-dd2b-4c69-ab2f-c382d9a6e8a2",
        "kommunekode": "0710",
        ... # Flere properties
      }
    },
    ... # Flere features
  ]
}
```

Figur 17: Eksempel på GeoJSON FeatureCollection returneret fra GetFeature med geometri og attributdata

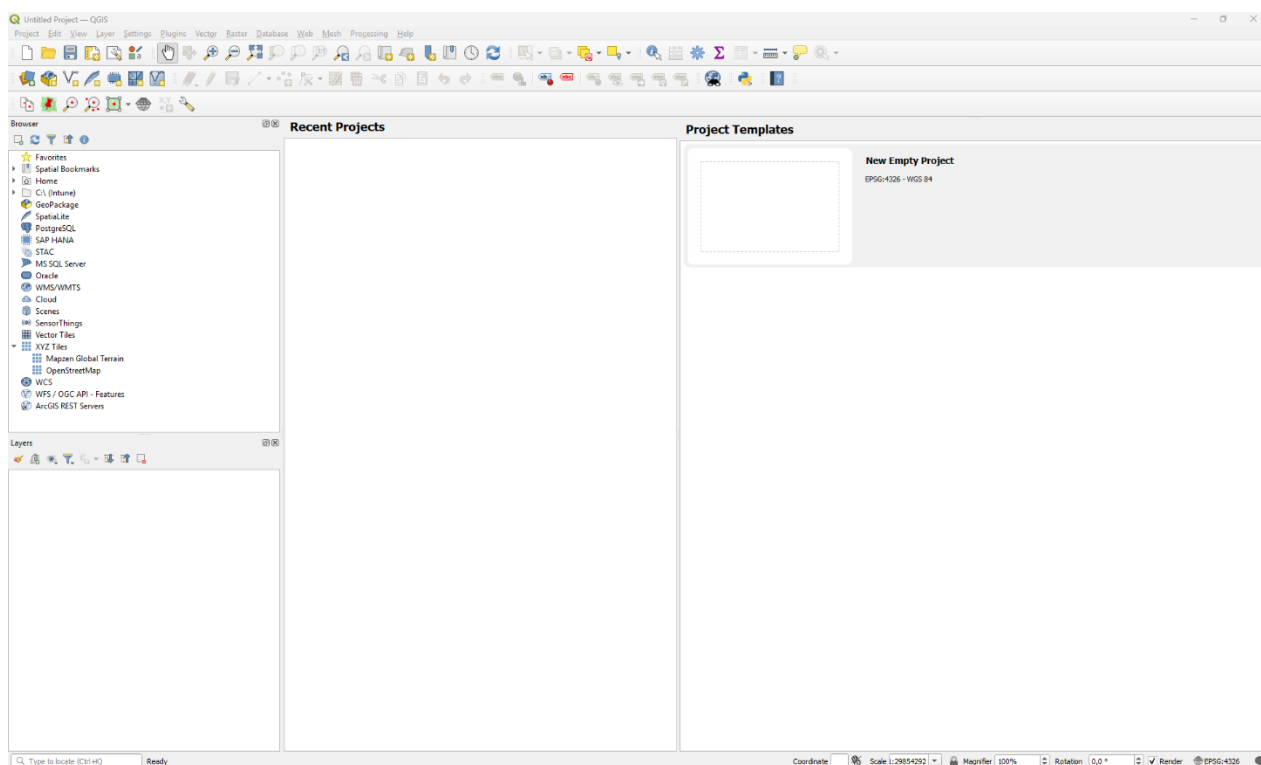
Herfra kan lave gentagne kald med forskellige filtrerings- eller pagingsparametre for at hente alt det ønskede data.

2.9.2 GIS-klienter: QGIS

I stedet for at tilgå de entitetsbaserede WFS-tjenester gennem almindelige REST-kald, kan man vælge at gøre det gennem en GIS-klient. GIS-klienter er software der blandt andet er specialudviklet til behandling af geodata der udstilles gennem WFS-tjenester.

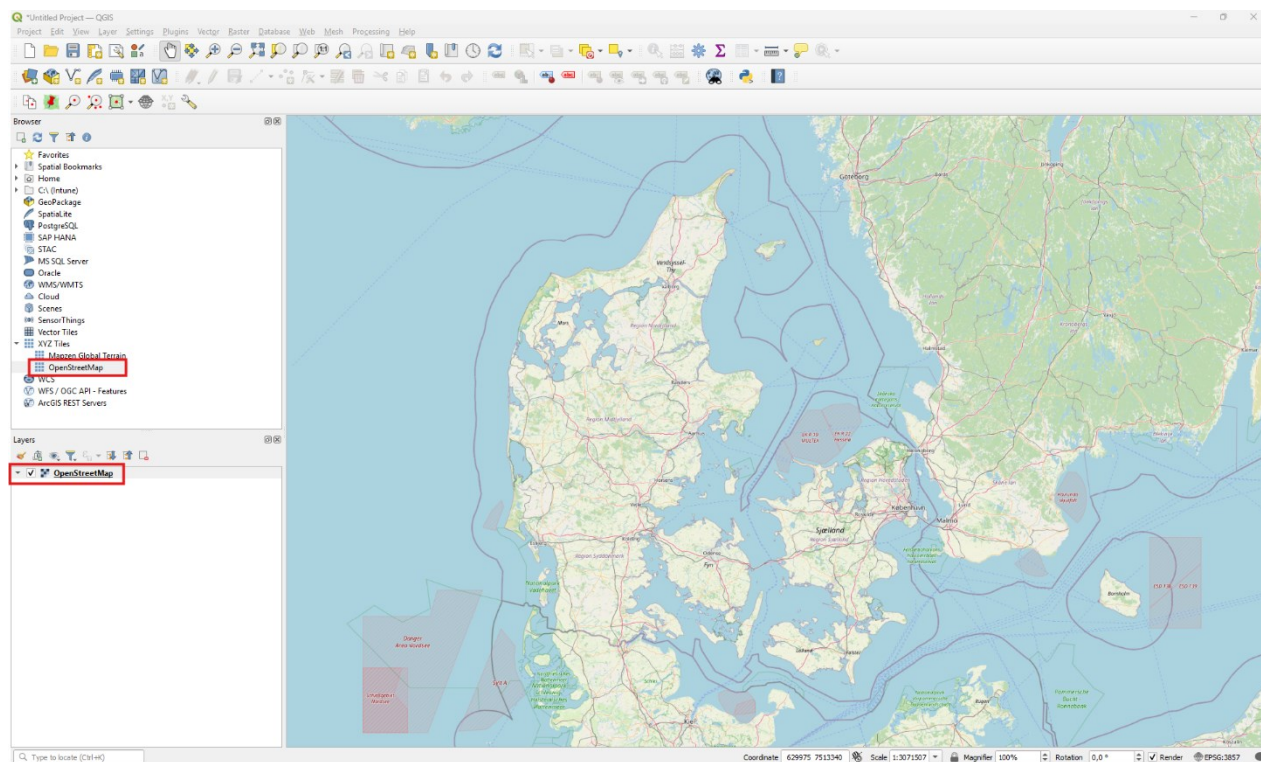
I dette eksempel bruges GIS-klienten [QGIS](#).

Når programmet åbnes, ser brugerfladen ud som vist i Figur 18.



Figur 18: QGIS startskærm

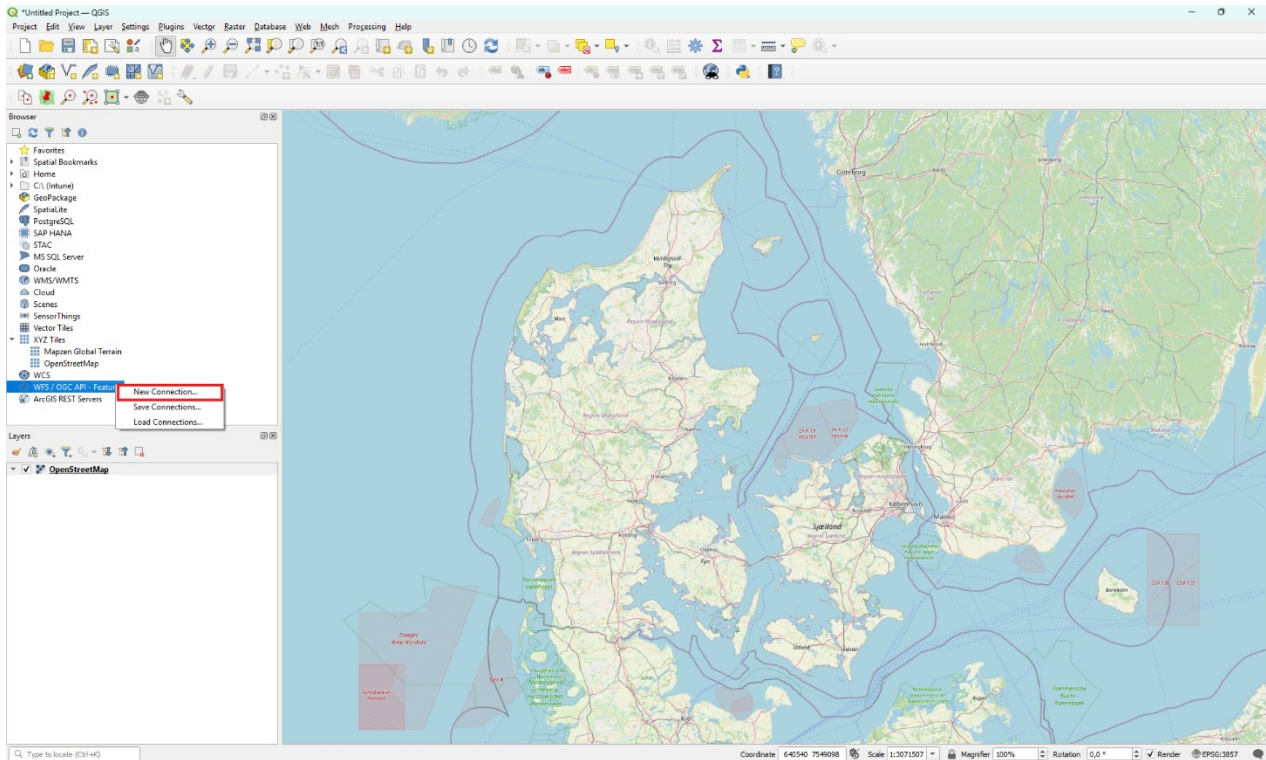
Hvis man dobbeltklikker "OpenStreetMap" til venstre i "Browser"-menuen, vil et verdenskort blive tilføjet til brugerfladen og laget "OpenStreetMap" vil fremgå af "Layers"-fanen. Zoomer man ind på Danmark ser klienten ud som vist i Figur 19: .



Figur 19: QGIS med "OpenStreetMap"

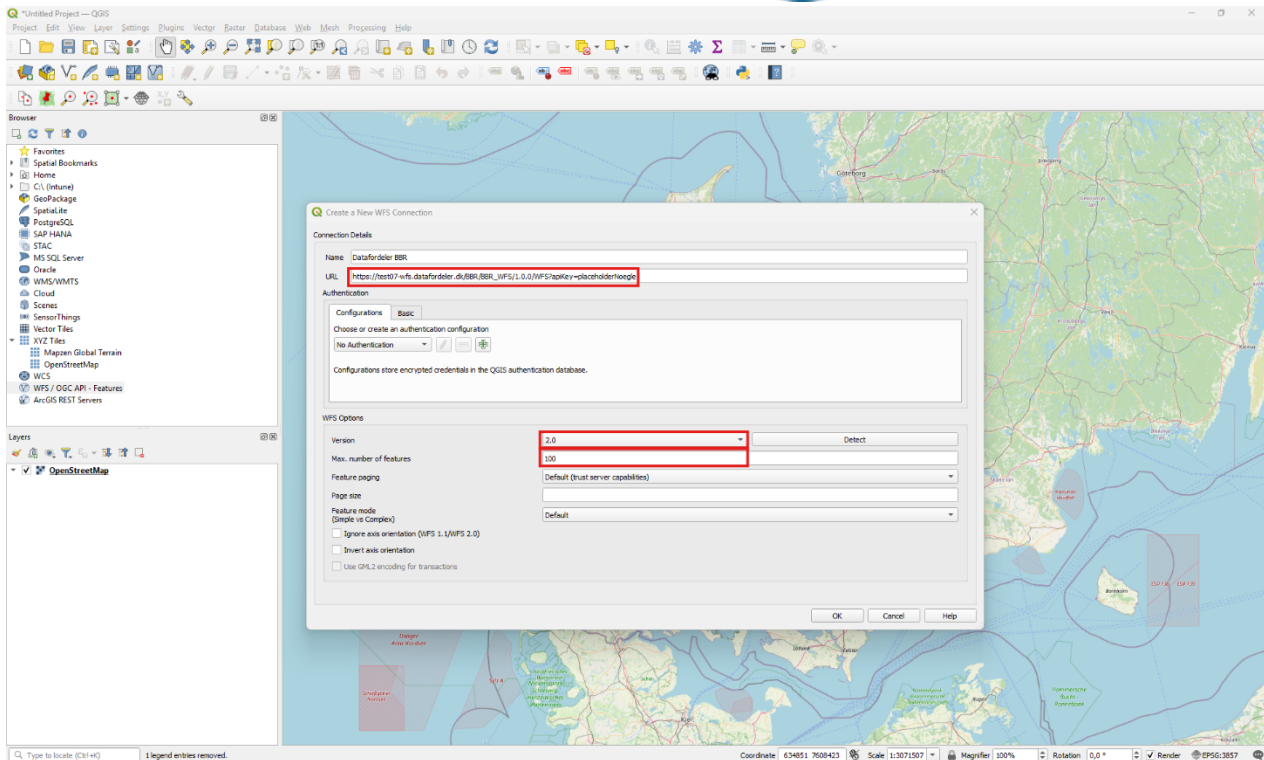


For at oprette forbindelse til en WFS-tjeneste, højreklikker man "WFS / OGC API – Features" i "Browser"-menuen og klikker "New Connection..." som vist i Figur 20 .



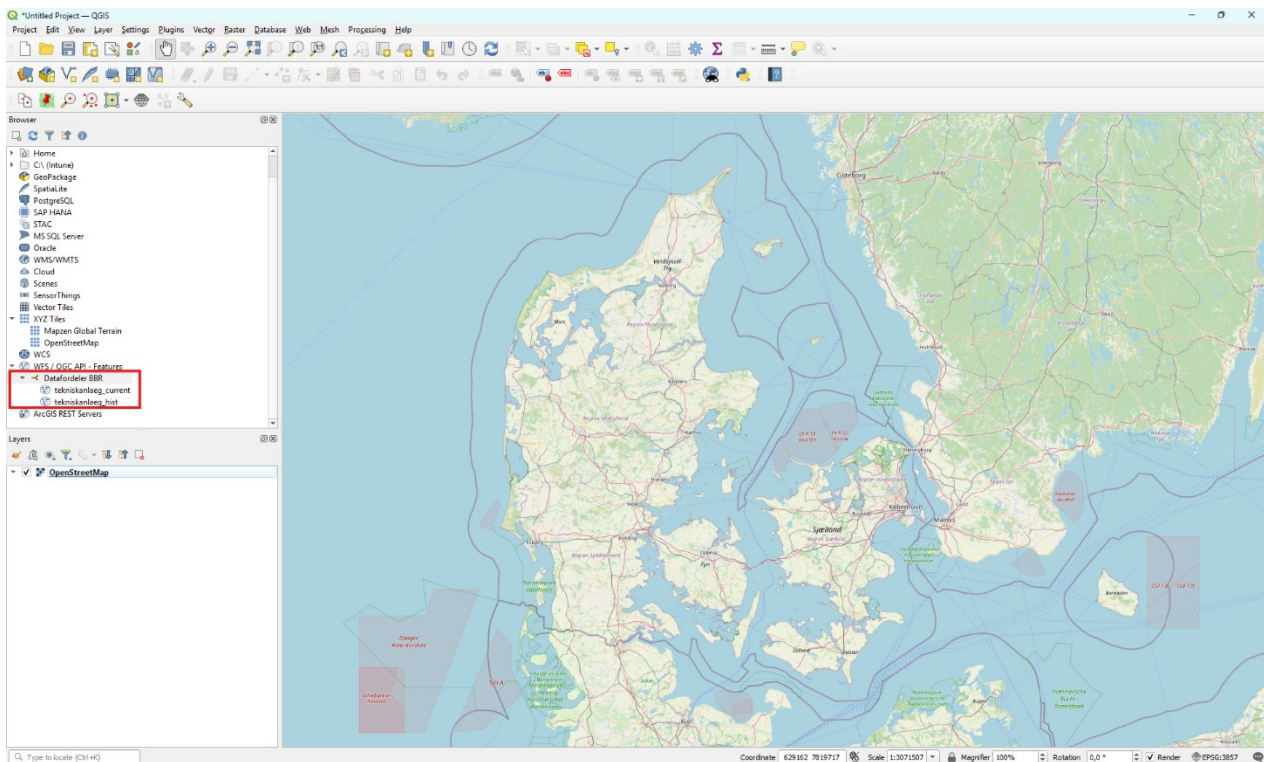
Figur 20 Tilføj "New Connection" i QGIS

Klikker man "New Connection..." kan man nu konfigurere en ny WFS-forbindelse. Her sættes URL'en, autentifikation og WFS-versionsnummer. Man kan også sætte andre parametre såsom "Max. number of features", hvilket svarer til count-parameteren i GetFeature-metoden. For at lave et kald til WFS-tjenesten for BBR indsættes værdierne som vist i Figur 21. Bemærk at "Max. number of features" er sat til 100.



Figur 21: Tilføj ny WFS-forbindelse til BBR i QGIS

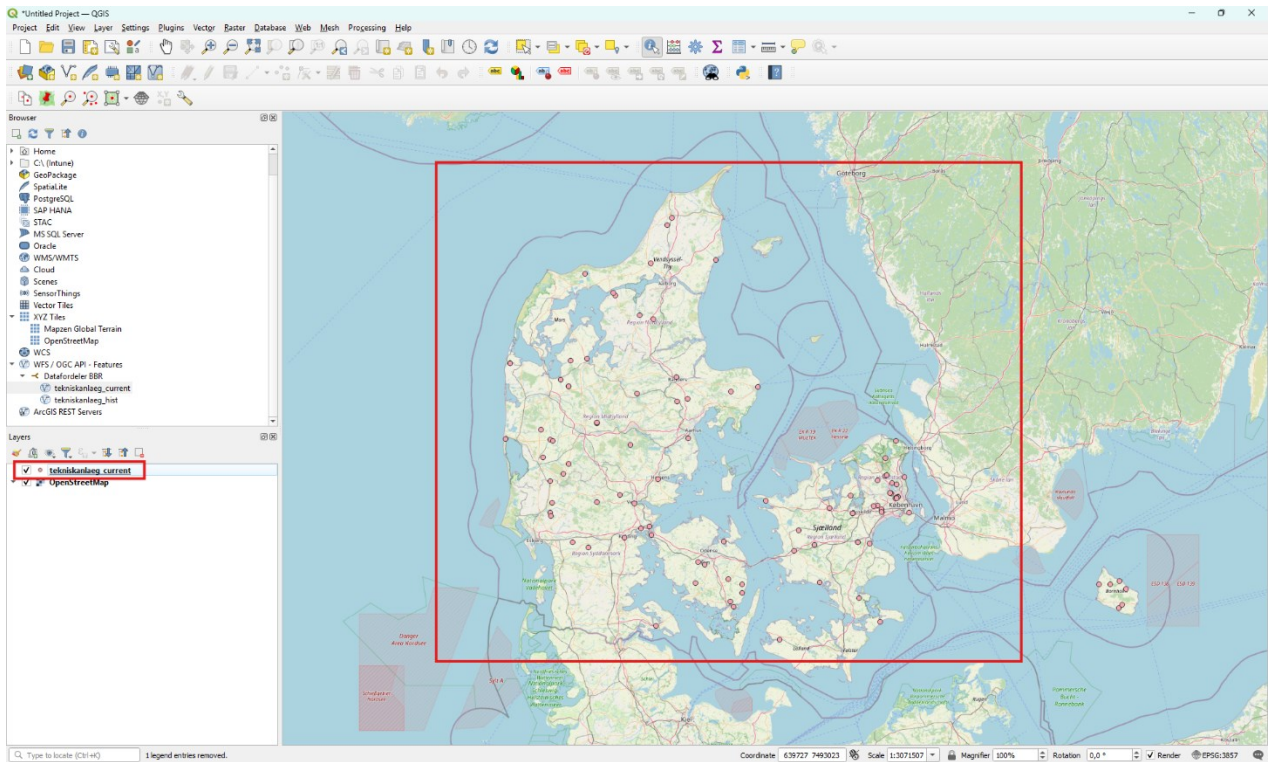
Klikker man "OK", vil "Datafordeler BBR" fremgå som et punkt under "WFS / OGC API – Features" i "Browser"-menuen. Dobbeltklikker man derefter "Datafordeler BBR" laver QGIS et GetCapabilities-kald til Datafordeleren, som derefter vil vise alle tilgængelige lag for BBR, som vist i Figur 22.



Figur 22: Hent tilgængelige lag for BBR i QGIS



QGIS kalder herefter automatisk `DescribeFeatureType` og bruger resultatet til at konfigurere attributtabeller korrekt, så man ikke selv behøver at tage stilling til hvordan data er struktureret. Man kan derfor blot dobbeltklikke et af lagene, for eksempel `tekniskanlaeg_current`, hvorefter klienten laver et `GetFeature`-kald til Datafordeleren, tilføjer det pågældende lag i "Layers"-fanen og viser det resulterende data på kortet, som vist i Figur 23.



Figur 23: Hent data for `tekniskanlaeg_current` i QGIS

Herfra kan blot navigere kortet og klikke på datapunkterne for at finde det ønskede data.



3 Autentifikation for WFS-tjenester

Adgang til de entitetsbaserede WFS-tjenester kræver autentifikation. Datafordeleren understøtter følgende autentifikationsmetoder for WFS:

- IT-system med API-nøgle
- IT-system med OAuth Shared Secret
- IT-system med OAuth Certifikat

For detaljeret vejledning om oprettelse og anvendelse af disse autentifikationsmetoder, se [Autentificeringsmetoder på Datafordeler Administration](#).

Bemærk: Det er *ikke* muligt at benytte tjenestebruger til autentifikation for de entitetsbaserede WFS-tjenester.