

# FEWS Web Feature Service - OpenGIS WFS 2.0

- [Introduction](#)
- [Developers documentation](#)
- [WFS Request methods](#)
  - [GetCapabilities](#)
    - [Request parameters](#)
  - [GetFeature](#)
    - [Request parameters](#)
    - [Vendor-specific request parameters](#)
  - [DescribeFeatureType](#)
    - [Request parameters](#)
  - [ListStoredQueries](#)
    - [Request parameters](#)
  - [DescribeStoredQueries](#)
    - [Request parameters](#)
- [WFS Request Filtering](#)
  - [TIME](#)
  - [BBOX](#)
  - [FILTER](#)
  - [CQL\\_FILTER](#)
- [Properties](#)

## Introduction

The FEWS Web Feature is the Delft-FEWS implementation of the OpenGIS WFS 2.0 OGC standard and is available since 2023.01. It allows requesting GML or GeoJSON feature data from plots that have been configured in the Delft-FEWS Grid Display. The WFS service can be seen as complementary to the WMS service, as it currently supports only gridplot types that contain polygon timeseries. The WMS service allows you to render these plots (feature or gridded data) to images, the WFS service provides access to the underlying vector data if the time series valueType is "polygon".

A valueType of polygon typically indicates that a polygon time series was calculated in FEWS based on a raster model of inundations where the polygon is different for each time step. Scalar time series which are linked to a locationSet containing polygons that are static (districts, catchments) currently can not be rendered using the WFS.

## Developers documentation

For API Developers it is recommended to use the Open API Documentation. Please see: <https://fewsdocs.deltares.nl/webservices/wfs/apidoc/>

The official OpenGIS documentation consists of two documents: the [OpenGIS Web Feature Service 2.0 Interface Standard](#) and the [OpenGIS Filter Encoding 2.0 Encoding Standard](#), both of which are ISO certified standards.

## WFS Request methods

The available request methods with their supported parameters are described here.

### GetCapabilities

Get a list of the available polygon gridPlots. In WFS terminology these are "FeatureTypes" and other information about the service. GetCapabilities will return its content as XML in accordance with the WFS 2.0 standard. Only version 2.0.0 of this standard is supported.

### Request parameters

- the GetCapabilities request has no parameters

### Response

When requesting the GetCapabilities, an XML response will be given with all available polygon gridPlots. The XML schema unfortunately does not allow them to be nested by gridPlotGroups (unlike the WMS schema).

### Example request

```
http://localhost:8080/FewsWebServices/wfs?request=GetCapabilities
```

### Example xml response

```

<?xml version="1.0" encoding="UTF-8"?>
<WFS_Capabilities version="2.0.0" xmlns="http://www.opengis.net/wfs/2.0" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:deltares="http://deltares.nl" xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0/wfs.xsd http://www.opengis.net/ows/1.1 http://schemas.opengis.net/ows/1.1.0/owsAll.xsd">
  <ows:ServiceIdentification>
    <ows:Title>WFS Unit tests</ows:Title>
    <ows:ServiceType>WFS</ows:ServiceType>
    <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>Deltares</ows:ProviderName>
  </ows:ServiceProvider>
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost/FewsWebServices/wfs"></ows:Get>
        </ows:HTTP>
      </ows:DCP>
      <ows:Parameter name="AcceptVersions">
        <ows:AllowedValues>
          <ows:Value>2.0.0</ows:Value>
        </ows:AllowedValues>
      </ows:Parameter>
      <ows:Parameter name="AcceptFormats">
        <ows:AllowedValues>
          <ows:Value>text/xml</ows:Value>
        </ows:AllowedValues>
      </ows:Parameter>
    </ows:Operation>
    <ows:Operation name="DescribeFeatureType">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost/FewsWebServices/wfs"/>
        </ows:HTTP>
      </ows:DCP>
      <ows:Parameter name="outputFormat">
        <ows:AllowedValues>
          <ows:Value>application/gml+xml; version=3.2</ows:Value>
        </ows:AllowedValues>
      </ows:Parameter>
    </ows:Operation>
    <ows:Operation name="ListStoredQueries">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost/FewsWebServices/wfs"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="DescribeStoredQueries">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost/FewsWebServices/wfs"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="GetFeature">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost/FewsWebServices/wfs"/>
        </ows:HTTP>
      </ows:DCP>
      <ows:Parameter name="outputFormat">
        <ows:AllowedValues>
          <ows:Value>application/gml+xml; version=3.2</ows:Value>
          <ows:Value>application/geo+json</ows:Value>
          <ows:Value>application/json</ows:Value>
        </ows:AllowedValues>
      </ows:Parameter>
    </ows:Operation>
  </ows:OperationsMetadata>
</WFS_Capabilities>

```

```

        </ows:Parameter>
    </ows:Operation>
    <ows:Parameter name="AcceptVersions">
        <ows:AllowedValues>
            <ows:Value>2.0.0</ows:Value>
        </ows:AllowedValues>
    </ows:Parameter>
    <ows:Constraint name="ImplementsBasicWFS">
        <ows:NoValues/>
        <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsTransactionalWFS">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsLockingWFS">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="KVPEncoding">
        <ows:NoValues/>
        <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="XMLEncoding">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="SOAPEncoding">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsInheritance">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsRemoteResolve">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsResultPaging">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsStandardJoins">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsSpatialJoins">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsTemporalJoins">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsFeatureVersioning">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ManageStoredQueries">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="wfs:StoredQuery">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
</ows:OperationsMetadata>
<FeatureTypeList>
    <FeatureType>
        <Name>deltares:full_extent_cta_clean</Name>
    </FeatureType>
</FeatureTypeList>

```

```

<Title>SFINCS full extent</Title>
<DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
<OtherCRS>urn:ogc:def:crs:EPSG::4326</OtherCRS>
<OtherCRS>urn:ogc:def:crs:EPSG::3857</OtherCRS>
<ows:WGS84BoundingBox>
  <ows:LowerCorner>125.095000 -37.527500</ows:LowerCorner>
  <ows:UpperCorner>153.637500 -10.020000</ows:UpperCorner>
</ows:WGS84BoundingBox>
</FeatureType>
<FeatureType>
  <Name>deltares:full_extent_cta</Name>
  <Title>SFINCS full extent (not clean)</Title>
  <DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
  <OtherCRS>urn:ogc:def:crs:EPSG::4326</OtherCRS>
  <OtherCRS>urn:ogc:def:crs:EPSG::3857</OtherCRS>
  <ows:WGS84BoundingBox>
    <ows:LowerCorner>125.095000 -37.527500</ows:LowerCorner>
    <ows:UpperCorner>153.637500 -10.020000</ows:UpperCorner>
  </ows:WGS84BoundingBox>
</FeatureType>
</FeatureTypeList>
</WFS_Capabilities>

```

*Note: A Deltares namespace is used in this example to identify items that originate from Delft-FEWS, as identified by `xmlns:deltares="http://deltares.nl"`. This is used as prefix for the `gridplot-id` values that have been taken from the spatial display configuration. The WFS standard more or less requires this. The namespace used by the WFS service can be changed in the `WebServices.xml` configuration.*

## GetFeature

Get the polygon GridPlot vector data in GML or GeoJSON format. The GridPlot to get is specified by the `typenames` parameter. Optional filters can be used to request a specific subset of a feature type.

## Request parameters

- *typenames* (required): the gridPlot id of the plot to display. Only one gridPlot id is supported.
- *version* (optional): supported version is 2.0.0 and is the default if not set. Older versions are not supported.
- *crs* (optional, '**EPSG:4326**' is the default): the output projection of the data. The default projection used by FEWS is always EPSG:4326 (WGS84 geographic coordinates; longitude, latitude). Other coordinate systems are supported only if they are configured in `WebServices.xml`.
- *format* (optional, '**application/gml+xml**; **version=3.2**' is the default): GML is the standard output format that any WFS implementation must support. Optionally '**application/geojson**' can be requested instead. Geojson is more compact and generally easier to process.
- *bbox* (optional): the bounding box (in the projection as defined by the CRS parameter) of the extent of interest. Any features that intersect the BBOX will be returned. Features that lie completely outside the BBOX area will not be returned.
- *filter* (optional): a filter expression in URL-encoded XML format as specified in the [OpenGIS Filter Encoding 2.0 Encoding Standard](#). Only the "Minimum Standard Filter" conformance class is currently implemented and tested. For more information, refer to *WFS Request Filtering* below.

## Vendor-specific request parameters

- *time* (optional): the time associated with the polygon data being requested. Only one time is supported. Time ranges are NOT supported. Time has to be in the XML date format: `yyyy-MM-ddTHH:mm:ssZ`. The time parameter is identical to that of the Delft-FEWS WMS service for practical reasons. If no time parameter is specified, the most recent (current) forecast data will be returned. Please note that any options the OpenGIS Filter specifications offer to define a temporal extent have not yet been implemented in the Delft-FEWS WFS.
- *cql\_filter* (optional): similar to the filter parameter, a filter expression in CQL (Common Query Language) syntax can be specified to request a subset of features. This syntax is easier to use than the XML filter syntax. For more information, refer to *WFS Request Filtering* below.

### Restrictions:

- There currently are no restrictions on the output size of the WFS service. Paging options that may help dealing with very large datasets have not yet been implemented.

### Response

GML3 or GeoJSON vector data of the requested polygon gridPlot for the specified time. If no time is specified, the last available time step will be returned.

### Example request with typeName

```
http://localhost:8080/FewsWebServices/wfs?service=wfs&request=GetFeature&typenames=r-1_shapeFWA
```

### Example request with version

```
<to be added>
```

#### Example request with crs

```
<to be added>
```

#### Example request with format

```
<to be added>
```

#### Example response:

The typical response in GML or GeoJSON for polygon data is too large to be displayed here.

## DescribeFeatureType

Provides a detailed description of one or more feature types. The most important function of this request is to list the available feature attributes (Delft-FEWS timeSeries properties). The output format is an XML schema.

### Request parameters

- *typeNames* (required): the gridPlot id of one or more plots to display. If this is not specified, a list of feature types is provided as a list of uri's.
- *version* (optional): supported version is 2.0.0 and is the default if not set. Older versions are not supported.

#### Example request

```
http://localhost:8080/FewsWebServices/wfs?service=wfs&request=DescribeFeatureType=r-1_shapeFWA&TypeNames=deltares:r-1_shapeFWA
```

#### Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:deltares="http://deltares.nl" targetNamespace="http://deltares.nl">
  <xsd:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <xsd:complexType name="r-1_shapeFWAType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element minOccurs="0" name="geometry" nillable="true" type="gml:MultiGeometryPropertyType"/>
          <xsd:element minOccurs="0" name="model_type" nillable="true" type="xsd:string"/>
          <xsd:element minOccurs="0" name="model_reg" nillable="true" type="xsd:string"/>
          <xsd:element minOccurs="0" name="model_id" nillable="true" type="xsd:string"/>
          <xsd:element minOccurs="0" name="model_name" nillable="true" type="xsd:string"/>
          <xsd:element minOccurs="0" name="remarks" nillable="true" type="xsd:string"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="r-1_shapeFWA" substitutionGroup="gml:AbstractFeature" type="deltares:r-1_shapeFWAType"/>
</xsd:schema>
```

## ListStoredQueries

Get a list of 'stored queries' a WFS supports. The only stored query supported is the so-called "GetFeatureById" stored query. This request has been implemented because it is a requirement for the "simple" implementation of a WFS 2.0 service.

The ids, used mainly for the generation of GML output, are assigned sequentially; they are not linked to a specific property or attribute in Delft-FEWS.

## Request parameters

- *version* (optional): supported version is 2.0.0 and is the default if not set. Older versions are not supported

### Example request

```
http://localhost:8080/FewsWebServices/wfs?service=wfs&request=ListStoredQueries
```

### Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ListStoredQueriesResponse xmlns="http://www.opengis.net/wfs/2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  <StoredQuery id="urn:ogc:def:query:OGC-WFS::GetFeatureById">
    <Title>Get feature by identifier</Title>
  </StoredQuery>
</ListStoredQueriesResponse >
```

## DescribeStoredQueries

Get a more detailed description of the 'stored queries' that the WFS supports. The only stored query supported is "GetFeatureById". This request has been implemented because it is a requirement for the "simple" implementation of a WFS 2.0 service.

The ids, used mainly for the generation of GML output, are assigned sequentially; they are not linked to a specific property or attribute in Delft-FEWS.

## Request parameters

- *version* (optional): supported version is 2.0.0 and is the default if not set. Older versions are not supported.

### Example request

```
http://localhost:8080/FewsWebServices/wfs?service=wfs&request=DescribeStoredQueries
```

### Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeStoredQueriesResponse xmlns="http://www.opengis.net/wfs/2.0" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0.0
/wfs.xsd">
  <StoredQueryDescription id="urn:ogc:def:query:OGC-WFS::GetFeatureById">
    <Title>Get feature by identifier</Title>
    <Parameter name="id" type="xs:string"/>
    <QueryExpressionText isPrivate="false" language="urn:ogc:def:queryLanguage:OGC-WFS::
WFS_QueryExpression" returnFeatureTypes="" />
  </StoredQueryDescription>
</DescribeStoredQueriesResponse>
```

## WFS Request Filtering

An important feature of the GetFeature request is filtering, which allows a subset of a feature set to be requested based on the feature properties /attributes, spatial extent, and/or time. The filtering parameters that can be applied are:

- time
- bbox
- filter
- cql\_filter

## TIME

The filtering using the TIME parameter works the same as it does for the Delft-FEWS WMS service. A time parameter can be given on the request url in the format **yyyy-MM-ddTHH:mm:ssZ**

Polygon-based plots in the Delft-FEWS Spatial Display configuration are typically timeSeries of polygons. By using the TIME parameter you can request the polygons that are in the datastore for a specific timestep. If no TIME parameter is specified, the polygon for the most recent timestep will be returned. It is not currently possible to request multiple time steps (a timeSeries) of polygons in a single request. Instead, a separate request for each timestep in a timeSeries is required.

#### Example request using the TIME parameter

<to be added>

## BBOX

The BBOX (Bounding BOX) filter allows you to specify a rectangular area in the format **BBOX=<minX>,<minY>,<maxX>,<maxY>**. This will return only the features that partially or completely intersect that rectangular area. If a CRS parameter is provided to the GetFeature request to return the data in a specific cartographic projection, the bounding box coordinates should also be provided in that projection.

#### Example request using the BBOX parameter

<to be added>

## FILTER

The standard FILTER parameter, which is part of the official OpenGIS WFS 2.0 specification, uses an XML-based syntax, which is documented in [OpenGIS Filter Encoding 2.0 Encoding Standard](#).

Only the "Minimum Standard Filter" conformance class is currently implemented, which means that the logical operators **PropertyIsEqualTo**, **PropertyIsNotEqualTo**, **PropertyIsLessThan**, **PropertyIsGreaterThan**, **PropertyIsLessThanOrEqualTo**, and **PropertyIsGreaterThanOrEqualTo** can be used and chained together using **And** / **Or** operators.

The problem with the standard XML-based filter syntax is that the Delft-FEWS WFS implementation currently only supports http GET methods, in which all request parameters need to be passed via the request url using NVP (Name-Value Pair) syntax, and the XML for the filter parameter needs to be 'URL encoded'.

#### Example request using the FILTER parameter

For example, a project where the polygons have a property 'region\_id' one could specify a filter for the WFS getFeature request in XML like like this:

```
<Filter>
  <PropertyIsEqualTo>
    <PropertyName>region-id</PropertyName>
    <Literal>123</Literal>
  </PropertyIsEqualTo>
</Filter>
```

Now this would need to be put on a single line and 'URL encoded' to get the FILTER parameter:

```
FILTER=%3CFilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Eregion_id%3C%2FPropertyName%3E%3CLiteral%3123%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E
```

This syntax is more or less unusable for manually putting together or interpreting WFS requests using filters. Fortunately, the CQL\_FILTER parameter syntax (below) is more practical.

<to be added>

## CQL\_FILTER

The CQL\_Filter parameter uses the Common Query Language format, which was developed by the OGC consortium as part of the Catalogue Service for the Web specification (CSW) and is, for example, implemented by Geoserver, which is commonly considered the reference implementation of the WFS protocol. The CQL\_FILTER parameter for the GetFeature request is therefore not part of the official standard specifications, but strongly recommended for use in http GET requests. A nice overview of the CQL syntax is provided in the [Geoserver documentation](#), however a warning is in place that not all possible filters mentioned there are currently supported and only the simple logical operators like in the above example have been tested.

#### Example request using the CQL\_FILTER parameter

The filter that was used as an example in the previous section in CQL syntax looks like this:

```
CQL_FILTER=region_id='123'
```

Multiple logical operators can be grouped together using AND / OR like this:

CQL\_FILTER=region\_id='123'+OR+region\_id='124'

<to be added>

## Properties

The following WFS specific properties can be configured in the webservices.xml configuration:

<Work in progress>