

# Package: rwfs (via r-universe)

January 30, 2025

**Type** Package

**Title** R Clients for Open Geospatial Consortium's Web Feature Services

**Version** 0.2.0

**Date** 2017-12-22

**Maintainer** Joona Lehtomaki <joona.lehtomaki@gmail.com>

**Description** R clients for Open Geospatial Consortium's (OGC) Web Feature Services (WFS).

**License** BSD\_2\_clause + file LICENSE

**URL** <https://github.com/ropengov/rwfs>, <https://ropengov.github.io/rwfs/>

**BugReports** <https://github.com/rOpenGov/rwfs/issues>

**Depends** R (>= 3.0.1)

**Imports** digest, R6, raster, sf, sp

**Suggests** knitr, rmarkdown, roxygen2, testthat

**Encoding** UTF-8

**LazyLoad** yes

**RoxygenNote** 7.1.2

**X-schema.org-isPartOf** <http://ropengov.org/>

**X-schema.org-keywords** ropengov

**Config/pak/sysreqs** libgdal-dev gdal-bin libgeos-dev libssl-dev  
libproj-dev libsqlite3-dev libudunits2-dev

**Repository** <https://ropengov.r-universe.dev>

**RemoteUrl** <https://github.com/rOpenGov/rwfs>

**RemoteRef** HEAD

**RemoteSha** b346cc2f076449fa07f96580934869f32fa27574

## Contents

asISO8601 . . . . .	2
GMLFile . . . . .	3
rwfs . . . . .	4
WFSCachingClient . . . . .	4
WFSCachingRequest . . . . .	5
WFSCClient . . . . .	6
WFSRequest . . . . .	8
WFSStreamingClient . . . . .	9
WFSStreamingRequest . . . . .	10
<b>Index</b>	<b>12</b>

---

asISO8601	<i>Returns date time string in ISO8601 format</i>
-----------	---

---

### Description

Converts an object which can be converted to a POSIXlt object to a ISO8601 date time string.

### Usage

```
asISO8601(dt)
```

### Arguments

dt                    Date time object which can be converted to a POSIXlt object.

### Value

Character string in ISO8601 format.

### Author(s)

Jussi Jousimo <jvj@iki.fi>

### Examples

```
asISO8601("2014-01-01")
```

---

GMLFile

*A class for providing a file name reference to a GML document*

---

## Description

A class for providing a file name reference to a GML document.

## Super class

`rwfs::WFSRequest` -> GMLFile

## Methods

### Public methods:

- `GMLFile$new()`
- `GMLFile$getDataSource()`
- `GMLFile$clone()`

### Method `new()`:

*Usage:*

`GMLFile$new(srcFile)`

### Method `getDataSource()`:

*Usage:*

`GMLFile$getDataSource()`

**Method `clone()`:** The objects of this class are cloneable with this method.

*Usage:*

`GMLFile$clone(deep = FALSE)`

*Arguments:*

`deep` Whether to make a deep clone.

## Author(s)

Jussi Jousimo <jvj@iki.fi>

---

rwfs

*WFS client for R*

---

### Description

This R package provides a client to access Web Feature Services (WFS) (<http://www.opengeospatial.org/standards/wfs>).

### Details

The client relies on the GDAL (<http://www.gdal.org/>) library and the **rgdal** (<https://cran.r-project.org/package=rgdal>) package, which should be installed first.

See examples in the **fmi** (<http://www.github.com/rOpenGov/fmi>) and the **gisfin** packages (<http://www.github.com/rOpenGov/gisfin>) how to inherit the abstract R6 classes (<https://cran.r-project.org/package=R6>) provided by the package.

### Author(s)

Jussi Jousimo <jvj@iki.fi>

---

WFSCachingClient

*Downloads response from a WFS and parses the intermediate file*

---

### Description

Dispatches a WFS request, saves the response to a file and parses the file. The data can be converted using ogr2ogr of RGDAL. Provides a caching mechanism for subsequent queries on the same data.

### Super class

`rwfs::WFSCient -> WFSCachingClient`

### Methods

#### Public methods:

- `WFSCachingClient$saveGMLFile()`
- `WFSCachingClient$loadGMLFile()`
- `WFSCachingClient$listLayers()`
- `WFSCachingClient$getLayer()`
- `WFSCachingClient$clone()`

#### Method `saveGMLFile()`:

*Usage:*

`WFSCachingClient$saveGMLFile(destFile)`

**Method** loadGMLFile():*Usage:*

WFS Caching Client \$loadGMLFile(fromFile)

**Method** listLayers():*Usage:*

WFS Caching Client \$listLayers()

**Method** getLayer():*Usage:*

WFS Caching Client \$getLayer(layer, ...)

**Method** clone(): The objects of this class are cloneable with this method.*Usage:*

WFS Caching Client \$clone(deep = FALSE)

*Arguments:*

deep Whether to make a deep clone.

**Author(s)**

Jussi Jousimo &lt;jvj@iki.fi&gt;

**See Also**[WFSRequest](#), [WFSStreamingClient](#)

---

WFS Caching Request	<i>An abstract class for building a URL reference to a WFS with a caching</i>
---------------------	---

---

**Description**

The abstract method `getURL` must be overloaded in a subclass to provide a request URL to a WFS service.

**Super classes**[rwfs::WFSRequest](#) -> [rwfs::WFSStreamingRequest](#) -> WFS Caching Request

**Methods****Public methods:**

- [WFSCachingRequest\\$getDataSource\(\)](#)
- [WFSCachingRequest\\$clone\(\)](#)

**Method** `getDataSource()`:*Usage:*`WFSCachingRequest$getDataSource()`**Method** `clone()`: The objects of this class are cloneable with this method.*Usage:*`WFSCachingRequest$clone(deep = FALSE)`*Arguments:*`deep` Whether to make a deep clone.**Author(s)**

Jussi Jousimo &lt;jvj@iki.fi&gt;

---

`WFSCient`*Class representing a WFS client*

---

**Description**

An abstract class to represent OGC's WFS client in R. Other client classes in this package inherit this this class.

**Format**`R6Class` object.**Methods**

`new(request)` This method is used to create object of this class with `request` as the request object containing WFS connection information and methods. **NOTE:** as this is abstract class, you shouldn't be creating instances of it.

**setRequest(request)** Set client's request object to `request`, which must inherit from [WFSRequest](#).

**listLayers()** Not implemented in this abstract class, but it classes inheriting this class.

**getLayer** Not implemented in this abstract class, but it classes inheriting this class.

**getRaster** Get a raster layer from WFS

**Public fields**`test`

**Active bindings**

test

**Methods****Public methods:**

- [WFSCient\\$new\(\)](#)
- [WFSCient\\$setRequest\(\)](#)
- [WFSCient\\$listLayers\(\)](#)
- [WFSCient\\$getLayer\(\)](#)
- [WFSCient\\$getRaster\(\)](#)
- [WFSCient\\$clone\(\)](#)

**Method new():**

*Usage:*

WFSCient\$new(request)

**Method setRequest():**

*Usage:*

WFSCient\$setRequest(request)

**Method listLayers():**

*Usage:*

WFSCient\$listLayers()

**Method getLayer():**

*Usage:*

WFSCient\$getLayer(layer, ...)

**Method getRaster():**

*Usage:*

WFSCient\$getRaster(parameters)

**Method clone():** The objects of this class are cloneable with this method.

*Usage:*

WFSCient\$clone(deep = FALSE)

*Arguments:*

deep Whether to make a deep clone.

**Author(s)**

Jussi Jousimo <jvj@iki.fi>, Joonas Lehtomaki <joona.lehtomaki@gmail.com>

**See Also**

[WFSSstreamingClient](#), [WFSCachingClient](#), [WFSRequest](#)

---

WFSRequest

*An abstract class for referencing a WFS or a GML document*

---

## Description

This class should be inherited and the abstract method `getDataSource` overloaded in a subclass to provide a reference.

## Methods

### Public methods:

- [WFSRequest\\$getDataSource\(\)](#)
- [WFSRequest\\$print\(\)](#)
- [WFSRequest\\$clone\(\)](#)

### Method `getDataSource()`:

*Usage:*

```
WFSRequest$getDataSource()
```

### Method `print()`:

*Usage:*

```
WFSRequest$print(...)
```

### Method `clone()`: The objects of this class are cloneable with this method.

*Usage:*

```
WFSRequest$clone(deep = FALSE)
```

*Arguments:*

`deep` Whether to make a deep clone.

## Author(s)

Jussi Jousimo <jvj@iki.fi>

## See Also

[WFSClient](#), [GMLFile](#)



---

WFSStreamingClient      *Streams response from a WFS*

---

### Description

Dispatches a WFS request and parses response from the stream directly.

### Super class

[rwfs::WFSCClient](#) -> WFSStreamClient

### Methods

#### Public methods:

- [WFSStreamingClient\\$listLayers\(\)](#)
- [WFSStreamingClient\\$getLayer\(\)](#)
- [WFSStreamingClient\\$clone\(\)](#)

#### Method listLayers():

*Usage:*

`WFSStreamingClient$listLayers()`

#### Method getLayer():

*Usage:*

`WFSStreamingClient$getLayer(layer, ...)`

#### Method clone():

The objects of this class are cloneable with this method.

*Usage:*

`WFSStreamingClient$clone(deep = FALSE)`

*Arguments:*

`deep` Whether to make a deep clone.

### Author(s)

Jussi Jousimo <[jvj@iki.fi](mailto:jvj@iki.fi)>

### See Also

[WFSRequest](#), [WFSCachingClient](#)

---

WFSSstreamingRequest    *An abstract class for building a URL reference to a WFS*

---

### Description

An abstract class for building a URL reference to a WFS.

### Super class

`rwfs: :WFSRequest` -> WFSSstreamingRequest

### Methods

#### Public methods:

- `WFSSstreamingRequest$getParameters()`
- `WFSSstreamingRequest$setPath()`
- `WFSSstreamingRequest$setParameters()`
- `WFSSstreamingRequest$getCapabilities()`
- `WFSSstreamingRequest$getFeature()`
- `WFSSstreamingRequest$clone()`

#### Method `getParameters()`:

*Usage:*

`WFSSstreamingRequest$getParameters()`

#### Method `setPath()`:

*Usage:*

`WFSSstreamingRequest$setPath(path)`

#### Method `setParameters()`:

*Usage:*

`WFSSstreamingRequest$setParameters(...)`

#### Method `getCapabilities()`:

*Usage:*

`WFSSstreamingRequest$getCapabilities(version = "1.0.0", ...)`

#### Method `getFeature()`:

*Usage:*

`WFSSstreamingRequest$getFeature(version = "1.0.0", typeName, ...)`

#### Method `clone()`: The objects of this class are cloneable with this method.

*Usage:*

`WFSSstreamingRequest$clone(deep = FALSE)`

*Arguments:*

`deep` Whether to make a deep clone.

**Author(s)**

Jussi Jousimo <jvj@iki.fi>

# Index

asISO8601, [2](#)

GMLFile, [3](#), [8](#)

R6Class, [6](#)

rwfs, [4](#)

rwfs::WFSCient, [4](#), [9](#)

rwfs::WFSRequest, [3](#), [5](#), [10](#)

rwfs::WFSStreamingRequest, [5](#)

WFSCachingClient, [4](#), [7](#), [9](#)

WFSCachingRequest, [5](#)

WFSCient, [6](#), [8](#)

WFSRequest, [5–7](#), [8](#), [9](#)

WFSStreamingClient, [5](#), [7](#), [9](#)

WFSStreamingRequest, [10](#)