

# Package: pgfeatureserv (via r-universe)

September 26, 2024

**Title** Client for pg\_featureserv RESTful web service  
**Version** 0.0.0.9005  
**Description** Client for pg\_featureserv, a RESTful geospatial feature server for PostGIS.  
**License** MIT + file LICENSE  
**Encoding** UTF-8  
**Roxygen** list(markdown = TRUE)  
**RoxygenNote** 7.3.2  
**Imports** chk, geojsonsf, glue, httr, jsonlite, lifecycle, tibble  
**Suggests** covr, httpptest, readr, sf, testthat (>= 3.0.0)  
**Config/testthat/edition** 3  
**Repository** <https://poissonconsulting.r-universe.dev>  
**RemoteUrl** <https://github.com/poissonconsulting/pgfeatureserv>  
**RemoteRef** HEAD  
**RemoteSha** 023b1ceed3577298a457c7362464d39e865e5b95

## Contents

pgf_collections	2
pgf_collection_bbox	3
pgf_collection_crs	4
pgf_collection_description	5
pgf_collection_feature	6
pgf_collection_features	7
pgf_collection_geometry_type	9
pgf_collection_properties	10
pgf_functions	12
pgf_function_description	13
pgf_function_parameters	14
pgf_function_properties	15
pgf_function_result	16

<b>Index</b>	<b>19</b>
--------------	-----------

---

pgf\_collections      *Get collections*

---

### Description

Get information about the collections.

### Usage

```
pgf_collections(  
  base_url,  
  path,  
  nocache = NULL,  
  user = gh_user(),  
  verbose = FALSE,  
  response = FALSE  
)
```

### Arguments

base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

### Value

A tibble.

### See Also

Other collections: [pgf\\_collection\\_bbox\(\)](#), [pgf\\_collection\\_crs\(\)](#), [pgf\\_collection\\_description\(\)](#), [pgf\\_collection\\_feature\(\)](#), [pgf\\_collection\\_features\(\)](#), [pgf\\_collection\\_geometry\\_type\(\)](#), [pgf\\_collection\\_properties\(\)](#)

### Examples

```
## Not run:  
base_url <- "https://features.hillcrestgeo.ca/"  
path <- "fwa"  
pgf_collections(base_url = base_url, path = path)  
  
## End(Not run)
```

---

pgf\_collection\_bbox     *Get collection bbox*

---

### Description

Get bounding box of a collection.

### Usage

```
pgf_collection_bbox(  
    collection_id,  
    base_url,  
    path,  
    nocache = NULL,  
    user = gh_user(),  
    verbose = FALSE,  
    response = FALSE  
)
```

### Arguments

collection_id	A character string of the collection id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

### Value

A list.

### See Also

Other collections: [pgf\\_collection\\_crs\(\)](#), [pgf\\_collection\\_description\(\)](#), [pgf\\_collection\\_feature\(\)](#), [pgf\\_collection\\_features\(\)](#), [pgf\\_collection\\_geometry\\_type\(\)](#), [pgf\\_collection\\_properties\(\)](#), [pgf\\_collections\(\)](#)

## Examples

```
## Not run:
base_url <- "https://features.hillcrestgeo.ca/"
path <- "fwa"
collection_id <- "whse_basemapping.fwa_named_streams"
pgf_collection_bbox(
  collection_id = collection_id, base_url = base_url, path = path
)

## End(Not run)
```

---

pgf\_collection\_crs      *Get collection crs*

---

## Description

Get crs of a collection.

## Usage

```
pgf_collection_crs(
  collection_id,
  base_url,
  path,
  nocache = NULL,
  user = gh_user(),
  verbose = FALSE,
  response = FALSE
)
```

## Arguments

collection_id	A character string of the collection id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

## Value

A string.

## See Also

Other collections: [pgf\\_collection\\_bbox\(\)](#), [pgf\\_collection\\_description\(\)](#), [pgf\\_collection\\_feature\(\)](#), [pgf\\_collection\\_features\(\)](#), [pgf\\_collection\\_geometry\\_type\(\)](#), [pgf\\_collection\\_properties\(\)](#), [pgf\\_collections\(\)](#)

## Examples

```
## Not run:
base_url <- "https://features.hillcrestgeo.ca/"
path <- "fwa"
collection_id <- "whse_basemapping.fwa_named_streams"
pgf_collection_crs(
  collection_id = collection_id, base_url = base_url, path = path
)

## End(Not run)
```

---

pgf\_collection\_description

*Get collection description*

---

## Description

Get description of a collection.

## Usage

```
pgf_collection_description(
  collection_id,
  base_url,
  path,
  nocache = NULL,
  user = gh_user(),
  verbose = FALSE,
  response = FALSE
)
```

## Arguments

collection_id	A character string of the collection id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

**Value**

A string.

**See Also**

Other collections: [pgf\\_collection\\_bbox\(\)](#), [pgf\\_collection\\_crs\(\)](#), [pgf\\_collection\\_feature\(\)](#), [pgf\\_collection\\_features\(\)](#), [pgf\\_collection\\_geometry\\_type\(\)](#), [pgf\\_collection\\_properties\(\)](#), [pgf\\_collections\(\)](#)

**Examples**

```
## Not run:
base_url <- "https://features.hillcrestgeo.ca/"
path <- "fwa"
collection_id <- "whse_basemapping.fwa_named_streams"
pgf_collection_description(
  collection_id = collection_id, base_url = base_url, path = path
)

## End(Not run)
```

---

pgf\_collection\_feature

*Get collection feature*

---

**Description**

Get collection feature given collection\_id, feature\_id and optional query parameters (see [API specs here](#)).

**Usage**

```
pgf_collection_feature(
  collection_id,
  feature_id,
  base_url,
  path,
  properties = NULL,
  transform = NULL,
  nocache = NULL,
  user = gh_user(),
  verbose = FALSE,
  response = FALSE
)
```

### Arguments

collection_id	A character string of the collection id.
feature_id	A positive whole number of the feature id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
properties	A vector of strings of the column names to include. If NULL (default), all columns are retained.
transform	A character vector with the name of the <i>valid transform</i> function followed by the parameter values (e.g. c("ST_Simplify", 100)).
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

### Value

A sf object.

### See Also

Other collections: [pgf\\_collection\\_bbox\(\)](#), [pgf\\_collection\\_crs\(\)](#), [pgf\\_collection\\_description\(\)](#), [pgf\\_collection\\_features\(\)](#), [pgf\\_collection\\_geometry\\_type\(\)](#), [pgf\\_collection\\_properties\(\)](#), [pgf\\_collections\(\)](#)

### Examples

```
## Not run:  
pgf_collection_feature("whse_basemapping.fwa_wetlands_poly", limit = 1)  
  
## End(Not run)
```

---

pgf\_collection\_features

*Get collection features*

---

### Description

Get collection features given collection\_id and optional query parameters (see API specs [here](#)).

**Usage**

```

pgf_collection_features(
  collection_id,
  base_url,
  path,
  filter = NULL,
  limit = 10000,
  offset = 0,
  sortby = NULL,
  bbox = NULL,
  properties = NULL,
  precision = NULL,
  transform = NULL,
  groupby = NULL,
  nocache = NULL,
  user = gh_user(),
  verbose = FALSE,
  response = FALSE
)

```

**Arguments**

collection_id	A character string of the collection id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
filter	A named vector or list of the filter(s) to apply, where the list names correspond to column names and the list values correspond to the desired value, e.g. <code>list(gnis_name = "Sangan River")</code> .
limit	A positive whole number that limits the number of features in the response.
offset	A positive whole number that starts the response at an offset.
sortby	A string that sorts the response items by a property. Default is ascending but property name can be prepended with '-' to indicate descending.
bbox	A vector of four numbers indicating bounding box in lon/lat (e.g. <code>c(minLon, minLat, maxLon, maxLat)</code> ) that features in response must intersect.
properties	A vector of strings of the column names to include. If NULL (default), all columns are retained.
precision	A positive whole number indicating precision (i.e., number of decimal places) of GeoJSON coordinates.
transform	A character vector with the name of the <i>valid transform</i> function followed by the parameter values (e.g. <code>c("ST_Simplify", 100)</code> ).
groupby	A string of the property name (i.e., column name) to group response features by. This would typically be used with an aggregate transform function.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.



user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

### Value

A sf object.

### See Also

Other collections: [pgf\\_collection\\_bbox\(\)](#), [pgf\\_collection\\_crs\(\)](#), [pgf\\_collection\\_description\(\)](#), [pgf\\_collection\\_feature\(\)](#), [pgf\\_collection\\_geometry\\_type\(\)](#), [pgf\\_collection\\_properties\(\)](#), [pgf\\_collections\(\)](#)

### Examples

```
## Not run:  
pgf_collection_features("whse_basemapping.fwa_wetlands_poly", limit = 1)  
  
## End(Not run)
```

---

```
pgf_collection_geometry_type  
    Get collection geometry type
```

---

### Description

Get geometry type of a collection.

### Usage

```
pgf_collection_geometry_type(  
  collection_id,  
  base_url,  
  path,  
  nocache = NULL,  
  user = gh_user(),  
  verbose = FALSE,  
  response = FALSE  
)
```

**Arguments**

collection_id	A character string of the collection id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

**Value**

A string.

**See Also**

Other collections: [pgf\\_collection\\_bbox\(\)](#), [pgf\\_collection\\_crs\(\)](#), [pgf\\_collection\\_description\(\)](#), [pgf\\_collection\\_feature\(\)](#), [pgf\\_collection\\_features\(\)](#), [pgf\\_collection\\_properties\(\)](#), [pgf\\_collections\(\)](#)

**Examples**

```
## Not run:
base_url <- "https://features.hillcrestgeo.ca/"
path <- "fwa"
collection_id <- "whse_basemapping.fwa_named_streams"
pgf_collection_geometry_type(
  collection_id = collection_id, base_url = base_url, path = path
)

## End(Not run)
```

---

pgf\_collection\_properties

*Get collection properties*

---

**Description**

Get information about a collection's column names.

**Usage**

```
pgf_collection_properties(  
  collection_id,  
  base_url,  
  path,  
  nocache = NULL,  
  user = gh_user(),  
  verbose = FALSE,  
  response = FALSE  
)
```

**Arguments**

collection_id	A character string of the collection id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

**Value**

A tibble.

**See Also**

Other collections: [pgf\\_collection\\_bbox\(\)](#), [pgf\\_collection\\_crs\(\)](#), [pgf\\_collection\\_description\(\)](#), [pgf\\_collection\\_feature\(\)](#), [pgf\\_collection\\_features\(\)](#), [pgf\\_collection\\_geometry\\_type\(\)](#), [pgf\\_collections\(\)](#)

**Examples**

```
## Not run:  
base_url <- "https://features.hillcrestgeo.ca/"  
path <- "fwa"  
collection_id <- "whse_basemapping.fwa_named_streams"  
pgf_collection_properties(  
  collection_id = collection_id, base_url = base_url, path = path  
)  
  
## End(Not run)
```

---

pgf\_functions

*Get functions*

---

### Description

Get information on available functions.

### Usage

```
pgf_functions(  
  base_url,  
  path,  
  nocache = NULL,  
  user = gh_user(),  
  verbose = FALSE,  
  response = FALSE  
)
```

### Arguments

base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server you are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

### Value

A tibble.

### See Also

Other functions: [pgf\\_function\\_description\(\)](#), [pgf\\_function\\_parameters\(\)](#), [pgf\\_function\\_properties\(\)](#), [pgf\\_function\\_result\(\)](#)

### Examples

```
## Not run:  
base_url <- "https://features.hillcrestgeo.ca/"  
path <- "fwa"  
pgf_functions(  
  base_url = base_url, path = path  
)  
  
## End(Not run)
```

---

pgf\_function\_description  
*Get function description*

---

## Description

Get information on function description.

## Usage

```
pgf_function_description(  
    function_id,  
    base_url,  
    path,  
    nocache = NULL,  
    user = gh_user(),  
    verbose = FALSE,  
    response = FALSE  
)
```

## Arguments

function_id	A character string of the function id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

## Value

A string.

## See Also

Other functions: [pgf\\_function\\_parameters\(\)](#), [pgf\\_function\\_properties\(\)](#), [pgf\\_function\\_result\(\)](#), [pgf\\_functions\(\)](#)

**Examples**

```
## Not run:
function_id <- "fwa_locatealong"
base_url <- "https://features.hillcrestgeo.ca/"
path <- "fwa"
pgf_function_description(
  function_id = function_id, base_url = base_url, path = path
)

## End(Not run)
```

---

pgf\_function\_parameters

*Get function parameters*


---

**Description**

Get information on function parameters.

**Usage**

```
pgf_function_parameters(
  function_id,
  base_url,
  path,
  nocache = NULL,
  user = gh_user(),
  verbose = FALSE,
  response = FALSE
)
```

**Arguments**

function_id	A character string of the function id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

**Value**

A tibble.

**See Also**

Other functions: [pgf\\_function\\_description\(\)](#), [pgf\\_function\\_properties\(\)](#), [pgf\\_function\\_result\(\)](#), [pgf\\_functions\(\)](#)

**Examples**

```
## Not run:
function_id <- "fwa_locatealong"
base_url <- "https://features.hillcrestgeo.ca/"
path <- "fwa"
pgf_function_parameters(
  function_id = function_id, base_url = base_url, path = path
)

## End(Not run)
```

---

pgf\_function\_properties

*Get function properties*

---

**Description**

Get information on function properties.

**Usage**

```
pgf_function_properties(
  function_id,
  base_url,
  path,
  nocache = NULL,
  user = gh_user(),
  verbose = FALSE,
  response = FALSE
)
```

**Arguments**

function_id	A character string of the function id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server your are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

**Value**

A tibble.

**See Also**

Other functions: [pgf\\_function\\_description\(\)](#), [pgf\\_function\\_parameters\(\)](#), [pgf\\_function\\_result\(\)](#), [pgf\\_functions\(\)](#)

**Examples**

```
## Not run:
function_id <- "fwa_locatealong"
base_url <- "https://features.hillcrestgeo.ca/"
path <- "fwa"
pgf_function_properties(
  function_id = function_id, base_url = base_url, path = path
)

## End(Not run)
```

---

pgf\_function\_result    *Get function result*

---

**Description**

Call a spatial function and get results.

**Usage**

```
pgf_function_result(
  function_id,
  base_url,
  path,
  parameters,
  limit = 10000,
  offset = 0,
  sortby = NULL,
  bbox = NULL,
  properties = NULL,
  precision = NULL,
  transform = NULL,
  nocache = NULL,
  user = gh_user(),
  verbose = FALSE,
  response = FALSE
)
```



**Arguments**

function_id	A character string of the function id.
base_url	A character string of the base URL.
path	A character string of the path or endpoint.
parameters	A named list of the parameters and values for a specific function.
limit	A positive whole number that limits the number of features in the response.
offset	A positive whole number that starts the response at an offset.
sortby	A string that sorts the response items by a property. Default is ascending but property name can be prepended with '-' to indicate descending.
bbox	A vector of four numbers indicating bounding box in lon/lat (e.g. c(minLon, minLat, maxLon, maxLat)) that features in response must intersect.
properties	A vector of strings of the column names to include. If NULL (default), all columns are retained.
precision	A positive whole number indicating precision (i.e., number of decimal places) of GeoJSON coordinates.
transform	A character vector with the name of the <i>valid transform</i> function followed by the parameter values (e.g. c("ST_Simplify", 100)).
nocache	A character string indicating whether to cache results or not. Possible values include 'true' or 'false' or NULL.
user	A character string of the user. This allows whoever is running the server you are querying to contact you in case of misuse.
verbose	A flag indicating whether to provide verbose GET response.
response	A flag indicating whether to return the raw response object.

**Value**

A sf object.

**See Also**

Other functions: [pgf\\_function\\_description\(\)](#), [pgf\\_function\\_parameters\(\)](#), [pgf\\_function\\_properties\(\)](#), [pgf\\_functions\(\)](#)

**Examples**

```
## Not run:
function_id <- "fwa_locatealong"
base_url <- "https://features.hillcrestgeo.ca/"
path <- "fwa"

parameters <- list(
  blue_line_key = 356308001,
  downstream_route_measure = 0
)
pgf_function_result(function_id,
```

```
base_url = base_url,  
path = path,  
parameters = parameters  
)  
  
## End(Not run)
```

# Index

## \* collections

- pgf\_collection\_bbox, 3
- pgf\_collection\_crs, 4
- pgf\_collection\_description, 5
- pgf\_collection\_feature, 6
- pgf\_collection\_features, 7
- pgf\_collection\_geometry\_type, 9
- pgf\_collection\_properties, 10
- pgf\_collections, 2

## \* functions

- pgf\_function\_description, 13
- pgf\_function\_parameters, 14
- pgf\_function\_properties, 15
- pgf\_function\_result, 16
- pgf\_functions, 12

- pgf\_collection\_bbox, 2, 3, 5–7, 9–11
- pgf\_collection\_crs, 2, 3, 4, 6, 7, 9–11
- pgf\_collection\_description, 2, 3, 5, 5, 7, 9–11
- pgf\_collection\_feature, 2, 3, 5, 6, 6, 9–11
- pgf\_collection\_features, 2, 3, 5–7, 7, 10, 11
- pgf\_collection\_geometry\_type, 2, 3, 5–7, 9, 9, 11
- pgf\_collection\_properties, 2, 3, 5–7, 9, 10, 10
- pgf\_collections, 2, 3, 5–7, 9–11
- pgf\_function\_description, 12, 13, 15–17
- pgf\_function\_parameters, 12, 13, 14, 16, 17
- pgf\_function\_properties, 12, 13, 15, 15, 17
- pgf\_function\_result, 12, 13, 15, 16, 16
- pgf\_functions, 12, 13, 15–17