Package: wdnr.gis (via r-universe)

August 20, 2024

Type Package

Title Pull Spatial Layers from 'WDNR ArcGIS REST API'

Version 0.1.5

Maintainer Paul Frater <paul.frater@wisconsin.gov>

License GPL-3

Encoding UTF-8

LazyData true

Description Functions for finding and pulling data from the 'Wisconsin Department of Natural Resources ArcGIS REST APIs' <https://dnrmaps.wi.gov/arcgis/rest/services> and <https://dnrmaps.wi.gov/arcgis2/rest/services>.

Depends R (>= 3.6.0), arcpullr, sf

Imports dplyr, ggplot2, rlang

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Paul Frater [aut, cre] (<https://orcid.org/0000-0002-7237-6563>), Zac Driscoll [aut] (<https://orcid.org/0000-0002-8233-0980>)

Repository CRAN

Date/Publication 2023-10-25 18:50:02 UTC

Contents

`wdnr.gis-package` *wdnr.gis*

Description

A package to pull spatial layers from the Wisconsin DNR ArcGIS REST API

Details

The wdnr.gis package provides shortcut functions for working with various spatial layers on the WDNR ArcGIS REST API. Currently, these include: get_hydro_layer, get_watershed_layer, get_roads_layer, get_fmdb_site_layer

get_*_layer functions

These functions retrieve spatial layers that are noted by the middle term in the function name. For example, the get_hydro_layer function retrieve's spatial data from Wisconsin's 24K Rivers and Streams Hydrography layer (or lakes if specified). These functions generally have the same arguments and can be queried by county, sf_object, watershed, or a SQL where statement.

check_args *Helper functions to aid in checking arguments to get_*_layer functions*

Description

check_layer_args simply looks at the arguments that is passed to it and checks to make sure that at least one is not NULL. avoid_duplicate_sf_args ensures the presence of only one argument that would result in a downstream spatial query (i.e. only a single sf object can be used in a spatial query – this function ensures that only one will be). deparse_arg_names is just a helper for the above two functions to format argument names in a useful way

Usage

```
check_layer_args(...)

avoid_duplicate_sf_args(...)

deparse_arg_names(...)
```

Arguments

...	Any number of objects to be checked

Value

If any of ... are not NULL, returns nothing. Otherwise stops function execution.

Examples

```
## Not run:
a <- NULL
b <- NULL
check_layer_args(a, b)

## End(Not run)
```

filter_county_poly *Retrieve county polygon layer*

Description

Return specific county polygon layer from wi_counties sf object

Usage

```
filter_county_poly(...)
```

Arguments

...	Any Wisconsin counties provided as character strings, separated by commas

Value

An sf data.frame with the appropriate counties

Examples

```
## Not run:
plot(filter_county_poly("door"))
plot_layer(filter_county_poly("portage"))

## End(Not run)
```

---

get_fmdb_site_layer *Retrieve WDNR's FMDB Site spatial layer*

Description

A function that can be used to retrieve the WDNR's Fish Management Database's (FMDB) monitoring site spatial layer. A spatial query can be performed to limit the output of the function by supplying a county name, watershed code, watershed name, or custom sf polygon object. Use the 'watershed_lookup' to find valid watershed codes and names. FMDB site sequance numbers (site_seq) or SWIMS (swims_site_seq) site sequance numbers can be provided to return specific sites. The 'where' arguement can be used to run custom SQL queries.

Usage

```
get_fmdb_site_layer(
  county = NULL,
  watershed_code = NULL,
  watershed_name = NULL,
  sf_object = NULL,
  site_seq = NULL,
  swims_site_seq = NULL,
  where = NULL,
  layer_type = "points",
  ...
)
```

Arguments

county	A character object specifying a county name
watershed_code	A character object specifying the HUC code for a watershed
watershed_name	A character object specifying the HUC name for a watershed
sf_object	Any sf polygon object

site_seq	A character object or string
swims_site_seq	A character object or string
where	SQL statement
layer_type	Character. Retrieve point stations, polygon stations, or both.
...	Additional parameters to pass to get_spatial_layer

Value

A sf object of class multipoints

Examples

```
## Not run:
get_fmdb_site_layer(county = "milwaukee")
get_fmdb_site_layer(watershed_code = "07070006")
get_fmdb_site_layer(site_seq = c(7511,10175131,128290))
get_fmdb_site_layer(county = "waukesha",
                    where = "STATION_TYPE_CODE = 'LAKE'")

## End(Not run)
```

---

get_hydro_layer *Retrieve WDNR's HYDRO spatial layer*

Description

A function that can be used to retrieve WDNR's 24k Hydrography (HYDRO) layer. Either the"24K Hydrography Streams and Rivers" or the "24K Hydrography Lakes and Open Water" can be queried by setting 'layer_type' to 'lines' or 'polygons' respectively. A spatial query can be performed to limit the output of the function by supplying a county name, watershed code, watershed name, or custom sf polygon object. Use the 'watershed_lookup' to find valid watershed codes and names. WBIC's can also be provided in order to return features for specific waterbodies. The 'where' arguement can be used to run custom SQL queries.

Usage

```
get_hydro_layer(
  county = NULL,
  watershed_code = NULL,
  watershed_name = NULL,
  sf_object = NULL,
  wbic = NULL,
  where = NULL,
  layer_type = "polygons",
  ...
)
```

Arguments

county	A character object specifying a county name
watershed_code	A character object specifying the HUC code for a watershed
watershed_name	A character object specifying the HUC name for a watershed
sf_object	Any sf polygon object
wbic	A character object or string of WBIC's
where	SQL statement
layer_type	"lines", "polygons", or "flowlines"
...	Additional parameters to pass to get_spatial_layer

Details

This function will retrieve WDNR's hydro layer. A county, watershed code, watershed_name, or custom sf polygon can be specifie to filter the layer. The layer type can be specified to query either the polylines or polygons hydro spatial layers.

Value

An sf object of class polylines of polygons

Examples

```
## Not run:
get_hydro_layer(county = "milwaukee", layer_type = "lines")
get_hydro_layer(watershed_code = "07070006", layer_type = "polygons")
get_hydro_layer(wbic = c("549400", "15000"), layer_type = "polygons")
get_hydro_layer(county = "milwaukee", where = "HYDROTYPE = '508'")

## End(Not run)
```

---

get_roads_layer *Retrieve WDNR's roads spatial layer*

Description

A function to retrieve WDNR's roads spatial layers. "layer_type" can be set to "major_roads" or "minor_roads" to query the Major Roads or County and Local Roads respectively. A spatial query can be performed to limit the output of the function by supplying a county name, watershed code, watershed name, or custom sf polygon object. Use the 'watershed_lookup' to find valid watershed codes and names. The "where" argument can be used to run custom SQL queries.

Usage

```
get_roads_layer(
  county = NULL,
  watershed_code = NULL,
  watershed_name = NULL,
  sf_object = NULL,
  where = NULL,
  layer_type = "all",
  ...
)
```

Arguments

county	A character object specifying a county name
watershed_code	A character object specifying the HUC code for a watershed
watershed_name	A character object specifying the HUC name for a watershed
sf_object	Any sf polygon object
where	SQL statement
layer_type	"major_roads" or "minor_roads"
...	Additional parameters to pass to get_spatial_layer

Value

A sf object of class polylines

Examples

```
## Not run:
get_roads_layer(county = "washington", layer_type = "major_roads")
get_roads_layer(watershed_code = "07070006", layer_type = "minor_roads")
get_roads_layer(where ="HWY_NUM = '43'",layer_type = "major_roads")

## End(Not run)
```

---

get_watershed_layer *Retrieve a watershed polygon*

Description

This function will retrieve a watershed boundary from WDNR's ArcGIS Rest Services. A subbasin (HUC8), watershed (HUC 10), or subwatershed (HUC 12) can be retrieved by passing the HUC code or name as a character string. See watershed_lookup for a full list of HUC codes and names. Use filter_huc() to see watersheds by county or classification level.

Usage

```
get_watershed_layer(
  watershed_code = NULL,
  watershed_name = NULL,
  county = NULL,
  sf_object = NULL,
  huc_level = NULL,
  where = NULL,
  ...
)
```

Arguments

watershed_code	A character object specifying the HUC code for a watershed
watershed_name	A character object specifying the HUC name for a watershed
county	A character object specifying a county name
sf_object	Any sf polygon object
huc_level	"HUC_8","HUC_10", or "HUC_12"
where	SQL statement
...	Additional parameters that are passed to get_spatial_layer

Details

A function to retrieve a watershed boundary from WDNR's subbasin (HUC8), watershed (HUC 10), or subwatershed (HUC 12) spatial layers. Use 'watershed_lookup' to see a full list of available HUC codes and names.

Value

A sf polygon object

Examples

```
## Not run:
get_watershed_layer(watershed_code = "07070006")
get_watershed_layer(watershed_name = "Kickapoo")
get_watershed_layer(county = "forest",huc_level = "HUC_12")

## End(Not run)
```

get_wis_rasters *General function to pull Raster layers from a MapServer or ImageServer*

Description

This is a non-exported function that is used as the engine for `get_wis_landcover` and `get_wis_imagery`. It converts watersheds, counties, etc. to the appropriate sf_object and queries the desired service using the function specified in `get_raster_function`

Usage

```
get_wis_rasters(
  service,
  get_raster_function,
  county = NULL,
  watershed_code = NULL,
  watershed_name = NULL,
  sf_object = NULL,
  ...
)
```

Arguments

`service`	Text string describing which service to pull. Will get matched by `match_services(service)`.
`get_raster_function`	The `arcpullr` function to use: either `get_map_layer` or `get_image_layer`
`county`	A character object specifying a county name
`watershed_code`	A character object specifying the HUC code for a watershed
`watershed_name`	A character object specifying the HUC name for a watershed
`sf_object`	Any sf polygon object
`...`	Additional arguments to pass to the `get_raster_function`

Value

A Raster* object dependent on `get_raster_function`

`get_wis_raster_layer` *Get WDNR Image and Map Layers*

Description

Functions to pull layers from the ImageServer and MapServer sections of the Wisconsin Department of Natural Resources ArcGIS REST API . These are raster layers representing various maps and images throughout the state of Wisconsin. Arguments to these function can be used to specify the spatial extent of the output. If no argument is provided, the full raster will be queried.

Usage

```
get_wis_landcover(
  service = "EN_Land_Cover2_Lev2",
  county = NULL,
  watershed_code = NULL,
  watershed_name = NULL,
  sf_object = NULL,
  ...
)

get_wis_imagery(
  service = "EN_Image_Basemap_Leaf_Off",
  county = NULL,
  watershed_code = NULL,
  watershed_name = NULL,
  sf_object = NULL,
  ...
)
```

Arguments

`service`	A string describing the service to be pulled.
`county`	A character object specifying a county name
`watershed_code`	A character object specifying the HUC code for a watershed
`watershed_name`	A character object specifying the HUC name for a watershed
`sf_object`	Any sf polygon object
`...`	Additional arguments to be passed to `get_map_layer`

Details

For a full list of available services use the following search options.

get_wis_landcover – `list_services(section = "DW_Land_Cover")`

get_wis_imagery – `list_services(section = "DW_Image")`

Value

A "RasterLayer" object

Examples

```
## Not run:
mke_forest <- get_wis_landcover(county = c("Milwaukee","Forest"))
plot_layer(mke_forest, outline_poly = wi_poly, legend = FALSE)



## End(Not run)
```

`list_funs` *List available sections, services, layers, and URLs in the WDNR GIS REST API*

Description

These functions can take sections, services, and layers specified as character strings and return either the section, service, layer or url as available in the WDNR GIS REST API

Usage

```
list_sections()

list_services(sections = NULL, pull = TRUE)

list_layers(sections = NULL, services = NULL, pull = TRUE)

list_urls(layers = NULL, sections = NULL, services = NULL, pull = TRUE)
```

Arguments

`sections`	A character vector of available sections to subset by
`pull`	Logical. Pull unique values (TRUE, default) or show the matching rows in the service_urls data.frame
`services`	A character vector of available services to subset by
`layers`	A character vector of available layers to subset by

Value

A vector of matching sections, services, layers, or URLs depending on the function called

Examples

```
list_sections()
list_services(sections = "WT_TMDL")
list_layers(services = match_services("Invasive"))
list_urls(sections = match_sections("WT"),
          services = match_services("inland"))
```

`list_layer_url` *Helper function to re-create* `list_layers` *and* `list_urls`

Description

Helper function to re-create `list_layers` and `list_urls`

Usage

```
list_layer_url(type = "layer", sections = NULL, services = NULL, pull = TRUE)
```

Arguments

`type`	Character. The column of data to retrieve from service_urls
`sections`	See `list_funs`
`services`	See `list_funs`
`pull`	See `list_funs`

Value

A vector of available layers or URLs; depending on `type`

`match_funs` *Find available sections, services, or layers using a regular expression*

Description

These functions allow you to search for sections, services, or layers that are available in the WDNR ArcGIS REST API using a regular expression. This is useful when you don't know the full name of a section, service, or layer but want to search based on keywords

Usage

```
match_sections(..., exact = FALSE)

match_services(..., sections = NULL, pull = TRUE, exact = FALSE)

match_layers(..., sections = NULL, services = NULL, pull = TRUE, exact = FALSE)
```

Arguments

...	Character vector or regular expression to match on
exact	Logical stating whether to match objects in ... exactly or loosely
sections	A character vector of available sections to subset by
pull	Logical. Pull unique values (TRUE, default) or show the matching rows in the service_urls data.frame
services	A character vector of available services to subset by

Value

A character vector of all matching sections, services, or layers appropriate to the called function

Examples

```
match_sections("WT")
match_services("Fish", sections = match_sections("WT"))
match_layers("Fish", sections = match_sections("WT"))
```

---

match_watershed_name *Match a watershed's name based on one or more regex*

Description

This function will match the names of a HUC_8 or a HUC_12 watershed found in the watershed_lookup data set.

Usage

```
match_watershed_name(..., pull = TRUE)
```

Arguments

...	One or more regex passed as character string
pull	Logical. Pull the unique values or

Value

A character string with full watershed names if pull = TRUE, or a data.frame with the number of rows equal to the number of matches otherwise

Examples

```
match_watershed_name("rainbow")
```

standardize_county_names *Standardize county names*

Description

This function alters string text of county names to a standardized format of lower-cased, no punctuation (i.e. st instead of st.), and underscore instead of spaces

Usage

```
standardize_county_names(...)
```

Arguments

...	One or more county names in quotations, or a character vector of county names

Value

A character vector the same length as `name`, but tidied up for easier and standard viewing

watershed_lookup *Various example sf polygons*

Description

These are sf polygons that are used for functions and examples throughout the package

Usage

```
watershed_lookup

wi_counties

wi_poly
```

Format

An object of class `data.frame` with 2232 rows and 3 columns.

An object of class sf and data.frame:

An object of class `sf` (inherits from `data.frame`) with 1 rows and 2 columns.

Source

`map_data`

`wi_example_data` *Various example data and lookup tables*

Description

These datasets are used for functions and examples throughout the package

Usage

```
service_urls
```

Format

A data.frame

Index