Package: feltr (via r-universe)

November 3, 2024

Type Package

Title Read Spatial Data from Felt

Version 0.1.2.9000

Maintainer Eli Pousson <eli.pousson@gmail.com>

Description Read Felt maps to simple feature objects in R.

License MIT + file LICENSE

URL https://github.com/elipousson/feltr/,
https://elipousson.github.io/feltr/

BugReports https://github.com/elipousson/feltr/issues

Imports cli (>= 2.5.0), grDevices, httr2, lifecycle, rlang (>= 1.1.0), sf, vctrs

Suggests covr, httptest2, rasterpic, RcppSimdJson, testthat (>= 3.0.0), xml2

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 2.10)

Repository https://elipousson.r-universe.dev

RemoteUrl https://github.com/elipousson/feltr

RemoteRef HEAD

RemoteSha 048c27f4b8501e5f9214af0ea6988e229237862c

Contents

`create_felt_layer` *Read layers from a Felt map, delete a layer, or create a new layer*

Description

Read layers from a Felt map with `read_felt_layers()`, delete a single layer with `delete_felt_layer()`, update a layer with `update_felt_layer()`, or create a new layer from a URL, file, or sf or sfc object with `create_felt_layer()`. Note that reading layers does not return layer data—only a list of layers.

Usage

```
create_felt_layer(
  map_id,
  layer,
  name = NULL,
  fileext = "gpkg",
  ...,
  fill_color = NULL,
  stroke_color = NULL,
  webhook_url = NULL,
  token = NULL
)

delete_felt_layer(map_id, layer_id = NULL, safely = TRUE, token = NULL)

read_felt_layers(
  map_id,
  simplifyVector = TRUE,
  token = NULL,
  call = caller_env()
)
```

```
update_felt_layer(
  map_id,
  layer_id,
  name = NULL,
  description = NULL,
  simplifyVector = TRUE,
  token = NULL,
  call = caller_env()
)
```

Arguments

map_id	A Felt map URL, map ID string, or a named list with a id and type element. If map_id is a list, it must be equivalent to the output from get_felt_map() where the list includes a "id" string and a "type" string with the value "map".
layer	Required. A object, file path, a layer source URL, or a sf or sfc object. If layer is a file path or a source URL, the file type or URL type must be supported by Felt. See https://feltmaps.notion.site/Upload-Anything-b26d739e80184127872faa923b55d233e37f06bc38c4971b435fbff2f4da6cb for details. If layer is a sf or sfc object, the object is saved to a temporary file using the supplied fileext.
name	Name for new map layer.
fileext	File extension to use for temporary file if layer is a sf or sfc object.
...	Additional parameters passed to sf::st_write() if layer is a sf or sfc object.
fill_color, stroke_color	Hex string to use as the layer fill or stroke color. Optional.
webhook_url	When the layer finishes processing, Felt will notify to this URL.
token	Felt personal access token
layer_id	Layer ID. Layer IDs for a map can be listed using read_felt_layers()
safely	If TRUE (default), check for user confirmation before deleting a Felt map. If FALSE, delete map without checking.
simplifyVector	Should JSON arrays containing only primitives (i.e. booleans, numbers, and strings) be caused to atomic vectors?
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.
description	Map description

emojis_reference *Emojis reference data*

Description

A simplified data frame with a list of emojis available in Felt through the Emoji Mart picker. More information: https://github.com/missive/emoji-mart

Usage

```
emojis_reference
```

Format

A data frame with 1566 rows and 5 variables:

`id` Emoji ID

`name` Name

`version` Version number

`keywords` Keywords

`alias` Alias

Source

https://raw.githubusercontent.com/missive/emoji-mart/main/packages/emoji-mart-data/sets/5/native.json

felt_user *Get the user information associated with the default (or supplied) token*

Description

List the name, email address, and user ID for the Felt user associated with the default (or supplied) token.

Usage

```
felt_user(token = NULL)
```

Arguments

`token`	Felt personal access token

`get_felt_comments` *Get comments from a Felt map*

Description

Get comments from a Felt map as a data frame or simple feature object. The results include a comment_url column based on the comment ID value.

Usage

```
get_felt_comments(
  map_id,
  flatten = TRUE,
  geometry = TRUE,
  crs = NULL,
  simplifyVector = TRUE,
  token = NULL
)
```

Arguments

`map_id`	A Felt map URL, map ID string, or a named list with a id and type element. If map_id is a list, it must be equivalent to the output from `get_felt_map()` where the list includes a "id" string and a "type" string with the value "map".
`flatten`	If `TRUE` (default) and comments do not include replies, flatten the structure of the results so each row contains a comment and a location. If `FALSE`, comments are included in a list column of data frames.
`geometry`	If `TRUE` (default), return a `sf` object. If `FALSE`, return a data frame.
`crs`	Coordinate reference system to return (if geometry is `TRUE`), Default: `NULL`
`simplifyVector`	Passed to `httr2::resp_body_json()`, Default: `TRUE`
`token`	Felt personal access token

Details

See Felt API documentation on the endpoint for exporting comments.

Value

A data frame or simple feature object (with a list column of comments if flatten is `FALSE`).

get_felt_data *Get Felt map data from the body of a map website*

Description

[Experimental]

`get_felt_data()` returns the parsed JSON included in the body of the HTML for a Felt map website (which includes both features and other user and layer metadata). This data can be used to supplement the Public API and may be deprecated as the API develops.

Usage

```
get_felt_data(map_id, token = NULL, call = caller_env())
```

Arguments

`map_id`	A Felt map URL, map ID string, or a named list with a id and type element. If map_id is a list, it must be equivalent to the output from `get_felt_map()` where the list includes a "id" string and a "type" string with the value "map".
`token`	Felt personal access token
`call`	The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

Value

A list of the parsed JSON found in the "felt-data" div of a Felt map webpage.

get_felt_style *Get Felt layer styles or update a layer style*

Description

[Experimental]

Get one or more Felt layer styles or update a specified layer style. Warning, updating a layer style without a list that can be converted to a valid Felt Style Language (FSL) may get a layer into an *irreversible broken state*.

Usage

```
get_felt_style(map_id, layer_id = NULL, call = caller_env())

update_felt_style(map_id, style, layer_id = NULL, call = caller_env())
```

Arguments

map_id	A Felt map URL, map ID string, or a named list with a id and type element. If map_id is a list, it must be equivalent to the output from get_felt_map() where the list includes a "id" string and a "type" string with the value "map".
layer_id	If NULL (default), all layers for the map are used. Multi-layer maps are not currently supported. Otherwise use a layer ID string. Use read_felt_layers() to list layers for an existing map.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.
style	A named list that can be converted to a valid Felt Style Language string. If style is supplied with a datasets id value matching the layer datasets ids, this function updates an existing layer style. If style is NULL (default), read styles for supplied map and layer. See the documentation on the Felt Style Language and the API endpoint for updating layer styles for more information.

Value

If layer_id is NULL and the map contains multiple styles or if layer_id is a character vector, the function returns a list with style elements named with the layer ID values. If layer_id is a string, the function returns a named list with a single Felt Style Language specification.

---

is_felt_url *Is a object a Felt URL?*

Description

Is a object a Felt URL?

Usage

```
is_felt_url(x)

check_felt_url(x, allow_null = FALSE, arg = caller_arg(x), call = caller_env())
```

Arguments

x	Object to check.
allow_null	If TRUE, check_felt_url() allows a NULL input without an error. Defaults to FALSE.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

read_felt *Read data from a Felt map*

Description

[Superseded]

Read simple features from a Felt map or get data embedded in the website of a Felt map. Superseded by `read_felt_map()`.

Usage

```
read_felt(
  url,
  map_id = NULL,
  ...,
  crs = 3857,
  token = NULL,
  rename = TRUE,
  name_repair = "check_unique"
)
```

Arguments

map_id	A Felt map URL, map ID string, or a named list with a id and type element. If map_id is a list, it must be equivalent to the output from `get_felt_map()` where the list includes a "id" string and a "type" string with the value "map".
...	Additional parameters passed to `sf::read_sf()`.
crs	Coordinate reference system to return. Defaults to 3857.
token	Felt personal access token
rename	If TRUE (default), strip the prefix text "felt-" from all column names.
name_repair	Passed to repair parameter of `vctrs::vec_as_names()`. Defaults to "check_unique".

Value

A simple feature data frame.

See Also

`sf::read_sf()`

read_felt_map *Read Felt map elements, create a Felt map from a URL, or delete a Felt map*

Description

Read elements, create, or delete a Felt map from a URL or map ID. `get_felt_map()` returns a list of map details and optionally (if `read = TRUE`) adds the map elements and layer list as elements in the list.

Usage

```
read_felt_map(map_id, ..., crs = NULL, token = NULL)

get_felt_map(
  map_id,
  ...,
  read = FALSE,
  simplifyVector = TRUE,
  token = NULL,
  call = caller_env()
)

create_felt_map(
  title = NULL,
  description = NULL,
  location = NULL,
  zoom = NULL,
  layer_urls = NULL,
  basemap = c("default", "satellite"),
  token = NULL
)

delete_felt_map(map_id, safely = TRUE, token = NULL)
```

Arguments

`map_id`	A Felt map URL, map ID string, or a named list with a id and type element. If map_id is a list, it must be equivalent to the output from `get_felt_map()` where the list includes a "id" string and a "type" string with the value "map".
`...`	Additional parameters passed to `sf::read_sf()`.
`crs`	Coordinate reference system. Passed to `sf::st_transform()` if supplied.
`token`	Felt personal access token
`read`	If `TRUE`, add a sf object with the map data as an element and a list of Felt layers to the returned list of map attributes. Defaults to `FALSE`.

simplifyVector	Should JSON arrays containing only primitives (i.e. booleans, numbers, and strings) be caused to atomic vectors?
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.
title	Map title
description	Map description
location	Location to center map, either a sf, sfc, or bbox object or a length 2 numeric vector in the form of c("lon", "lat"). To pass coordinates in lat/lon order, set the feltr.latlon option to TRUE (option defaults to FALSE). If location is NULL (default), map is centered on Oakland, California.
zoom	Zoom level number
layer_urls	A character vector or list of raster layer URLs.
basemap	Basemap, string ("default" or "satellite"), a valid layer URL, a color name, or a color hex code.
safely	If TRUE (default), check for user confirmation before deleting a Felt map. If FALSE, delete map without checking.

Value

read_felt_map() returns a sf object, create_felt_map() invisibly returns a list of attributes for the created map, and delete_felt_map() does not return anything.

Examples

```
## Not run:
if (interactive()) {
  map_data <- create_felt_map(title = "Example map")

  url <- map_data$attributes$url

  get_felt_map(url = url)

  delete_felt_map(url = url)

  read_felt_map("https://felt.com/map/Site-Plan-Example-PGTipS2mT8CYBIVlyAm9BkD")
}

## End(Not run)
```

---

read_felt_raster *Use rasterpic to create a* SpatRaster *object from a Felt map*

Description

Read an image feature from Felt and use the rasterpic::rasterpic_img() function and a corresponding image URL or file path to create a SpatRaster object based on the feature geometry.

Usage

```
read_felt_raster(x, images = NULL, ..., col = NULL, crs = 3857)
```

Arguments

x	If x is a Felt map URL, it is passed to `read_felt()` to create a data.frame of features with a "type" and "text" columns. If x is a data.frame, it is expected to be a data.frame created by reading a Felt map with `read_felt()` but could be a sf object with a type column that includes the value "Image" and (if images is named) a text column with matching text. Required.
images	A vector of image file paths or URLs with a "png", "jpeg/jpg", or "tiff/tif" file extension. images must be ordered to match the order of "Image" type features in the input data.frame x or have names that match the text column for x. If images is named, any "Image" features in x with text that does not match the names for images are excluded from the returned list. Defaults to `NULL`. Optional if col is provided.
...	Additional parameters passed to `sf::read_sf()`.
col	If features in x contain an attribute with a file path or URL, set col as the name of the attribute column. col is ignored if images is provided. Defaults to `NULL`.
crs	Coordinate reference system to return. Defaults to 3857.

Value

If images is length 1, a `SpatRaster` object is returned. Otherwise, the function returns a list of `SpatRaster` objects of the same length as images.

`set_felt_token` *Set or get a Felt API personal access token*

Description

An API personal access token is required to use `read_felt()`. See https://feltmaps.notion.site/Felt-Public-API-reference-PUBLIC-c01e0e6b0d954a678c608131b894e8e1 for instructions on how to get a token.

Usage

```
set_felt_token(
  token = NULL,
  install = FALSE,
  overwrite = FALSE,
  default = "FELT_ACCESS_TOKEN"
)

get_felt_token(
  token = NULL,
```

```
  default = "FELT_ACCESS_TOKEN",
  call = caller_env()
)
```

Arguments

token	Felt personal access token
install	If TRUE, this function adds your token to your .Renviron for use in future sessions. Defaults to FALSE.
overwrite	If TRUE, overwrite any existing token in .Renviron using the same environment variable name. Defaults to FALSE.
default	Default name used for environment variable where the token is saved.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Index