

Package ‘feltr’

July 22, 2025

Title Access the Felt API

Version 0.0.4

Description Upload, download, and edit internet maps with the Felt API

([https://feltmaps.notion.site/](https://feltmaps.notion.site/Felt-Public-API-reference-c01e0e6b0d954a678c608131b894e8e1)

[Felt-Public-API-reference-c01e0e6b0d954a678c608131b894e8e1](https://feltmaps.notion.site/Felt-Public-API-reference-c01e0e6b0d954a678c608131b894e8e1)).

Allows users to create new maps, edit existing maps, and extract data.

Provides tools for working with layers, which represent geographic data, and elements, which are interactive annotations. Spatial data accessed from the API is transformed to work with 'sf'.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports cli, curl, fs, geojsonsf, httr2, jsonlite, purrr, rlang, sf, stringr, tibble, tidyr

URL <https://github.com/christopherkenny/feltr>,

<https://christophertkenny.com/feltr/>

Suggests httptest2, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Christopher T. Kenny [aut, cre] (ORCID:

<https://orcid.org/0000-0002-9386-6860>)

Maintainer Christopher T. Kenny <christopherkenny@fas.harvard.edu>

Repository CRAN

Date/Publication 2023-11-05 20:30:02 UTC

Contents

felt_add_map_layers	2
felt_add_map_layers_url	3
felt_create_map	4

felt_delete_map	5
felt_delete_map_layer	5
felt_finish_upload	6
felt_get_comments	6
felt_get_map	7
felt_get_map_elements	8
felt_get_style	8
felt_get_user	9
felt_patch_style	9
key	10
set_felt_key	10
Index	12

felt_add_map_layers	<i>Add Layers to Existing Map</i>
---------------------	-----------------------------------

Description

Add Layers to Existing Map

Usage

```
felt_add_map_layers(  
    map_id,  
    name = NULL,  
    file_names = NULL,  
    fill_color = NULL,  
    stroke_color = NULL  
)
```

Arguments

map_id	map identifier from url, from https://felt.com/map/Readable-Name-map_id
name	Name of the layer
file_names	Files to include
fill_color	Color to fill shape with, typically a hexcode. Defaults to NULL.v
stroke_color	Color to outline shape with, typically a hexcode. Defaults to NULL.

Value

a string with the id for the created layer

Examples

```
layer <- felt_add_map_layers(map_id = 'Rockland-2024-Districts-TBI8sDkmQjuK2GX9CSiHiUA',
                             file_names = fs::path_package('feltr', 'towns.geojson'),
                             name = 'Towns')

layer
# and delete layer
felt_delete_map_layer(map_id = 'TBI8sDkmQjuK2GX9CSiHiUA', layer_id = layer)
```

felt_add_map_layers_url

Add Layers to Existing Map from URL

Description

See [Felt "Upload Anything" documentation](#) for detailed examples of potential URLs.

Usage

```
felt_add_map_layers_url(map_id, url, name = NULL)
```

Arguments

map_id	map identifier from url, from https://felt.com/map/Readable-Name-map_id
url	Link to layer to include
name	Name of the layer

Value

a [tibble::tibble](#) for the created layer

Examples

```
# split the URL for length reasons
url <- paste0(
  'https://www.rocklandgis.com/portal/sharing/rest/',
  'content/items/73fc78cb0fb04580b4788937fe5ee697/data'
)
layer <- felt_add_map_layers_url(
  map_id = 'Rockland-2024-Districts-TBI8sDkmQjuK2GX9CSiHiUA',
  url = url,
  name = 'Parks')
layer
# and delete the new layer
felt_delete_map_layer(map_id = 'TBI8sDkmQjuK2GX9CSiHiUA', layer_id = layer$id)
```

felt_create_map	Create a new map
-----------------	------------------

Description

Create a new map

Usage

```
felt_create_map(  
  title = NULL,  
  basemap = NULL,  
  layer_urls = NULL,  
  lat = NULL,  
  lon = NULL,  
  zoom = NULL  
)
```

Arguments

title	Title to use for the map. Defaults to NULL.
basemap	Basemap for the new map. Defaults to NULL. Can be a URL or color hex code
layer_urls	vector of URLs to generate layers in map. Defaults to NULL.
lat	latitude to center the map. Defaults to NULL.
lon	longitude to center the map. Defaults to NULL
zoom	zoom level to initialize the map with.

Value

a `tibble::tibble` for the new map

Examples

```
map <- felt_create_map(title = 'feltr example')  
map  
# and delete it again  
felt_delete_map(map_id = map$id)
```

felt_delete_map	<i>Delete an existing map</i>
-----------------	-------------------------------

Description

Delete an existing map

Usage

```
felt_delete_map(map_id)
```

Arguments

map_id	map identifier from url, from https://felt.com/map/Readable-Name-map_id
--------	---

Value

response code

Examples

```
map <- felt_create_map(title = 'feltr example')  
felt_delete_map(map_id = map$id)
```

felt_delete_map_layer	<i>Delete Layer from an Existing Map</i>
-----------------------	--

Description

Delete Layer from an Existing Map

Usage

```
felt_delete_map_layer(map_id, layer_id)
```

Arguments

map_id	map identifier from url, from https://felt.com/map/Readable-Name-map_id
layer_id	layer identifier, as returned by <code>felt_get_map_layers()</code> or <code>felt_add_map_layers()</code>

Value

response code

Examples

```
layer <- felt_add_map_layers('TBI8sDkmQjuK2GX9CSiHiUA',
                             'test', fs::path_package('feltr', 'towns.geojson'))
felt_delete_map_layer(map_id = 'Rockland-2024-Districts-TBI8sDkmQjuK2GX9CSiHiUA',
                      layer)
```

felt_finish_upload	<i>Finish Upload</i>
--------------------	----------------------

Description

Uploads are redirected to Amazon S3 buckets and must be confirmed once done. This function is *primarily* internal, but may have uses for custom uploads.

Usage

```
felt_finish_upload(map_id)
```

Arguments

map_id map identifier from url, from `https://felt.com/map/Readable-Name-map_id`

Value

httr2 status code

Examples

```
felt_finish_upload('TBI8sDkmQjuK2GX9CSiHiUA')
```

felt_get_comments	<i>Export Comments on an Existing Map</i>
-------------------	---

Description

Export Comments on an Existing Map

Usage

```
felt_get_comments(map_id)
```

Arguments

map_id map identifier from url, from `https://felt.com/map/Readable-Name-map_id`

Value

a `tibble::tibble` for the map

Examples

```
felt_get_comments(map_id = 'Rockland-2024-Districts-TBI8sDkmQjuK2GX9CSiHiUA')
```

felt_get_map	<i>Get Map Information from Map ID</i>
--------------	--

Description

- `felt_get_map()` returns identifying information for the map
- `felt_get_map_layers()` returns information about each layer in the map
- `felt_get_map_elements()` returns the shapes for each layer in the map

Usage

```
felt_get_map(map_id)
```

```
felt_get_map_layers(map_id)
```

Arguments

`map_id` map identifier from url, from https://felt.com/map/Readable-Name-map_id

Value

a `tibble::tibble` for the map

Examples

```
felt_get_map('Rockland-2024-Districts-TBI8sDkmQjuK2GX9CSiHiUA')
felt_get_map_layers('Rockland-2024-Districts-TBI8sDkmQjuK2GX9CSiHiUA')

# slower, as it has to build the shapes from the API result
felt_get_map_elements('Rockland-2024-Districts-TBI8sDkmQjuK2GX9CSiHiUA')
```

felt_get_map_elements *Get Map Elements from Map ID*

Description

Get Map Elements from Map ID

Usage

```
felt_get_map_elements(map_id)
```

Arguments

map_id map identifier from url, from <https://felt.com/map/Map-ID>

Value

a `tibble::tibble` for the map

Examples

```
felt_get_map_elements(map_id = 'Rockland-2024-Districts-TBI8sDkmQjuK2GX9CSiHiUA')
```

felt_get_style *Get Felt Style Information for a Layer*

Description

For details on the Felt Style Language, see <https://felt.com/blog/felt-style-language>.

Usage

```
felt_get_style(map_id, layer_id)
```

Arguments

map_id map identifier from url, from https://felt.com/map/Readable-Name-map_id
layer_id layer identifier, as returned by `felt_get_map_layers()` or `felt_add_map_layers()`

Value

a list version of style information

Examples

```
felt_get_style(map_id = 'TBI8sDkmQjuK2GX9CSiHiUA', layer_id = 'Qno0gfmkTS9BggbesfQooND')
```

felt_get_user	<i>Obtain information about the user</i>
---------------	--

Description

Obtain information about the user

Usage

```
felt_get_user()
```

Value

a `tibble::tibble` of information about the user

Examples

```
felt_get_user()
```

felt_patch_style	<i>Update Felt Style Information for a Layer</i>
------------------	--

Description

For details on the Felt Style Language, see <https://felt.com/blog/felt-style-language>.

Usage

```
felt_patch_style(map_id, layer_id, fsl)
```

Arguments

map_id	map identifier from url, from https://felt.com/map/Readable-Name-map_id
layer_id	layer identifier, as returned by <code>felt_get_map_layers()</code> or <code>felt_add_map_layers()</code>
fsl	A list indicating the Felt style language to update the layer to

Value

response data

Examples

```
felt_get_style(map_id = 'TBI8sDkmQjuK2GX9CSiHiUA', layer_id = 'Qno0gfmkTS9BggbesfQooND')
```

key	<i>Check or Get Felt API Key</i>
-----	----------------------------------

Description

Check or Get Felt API Key

Usage

```
has_felt_key()
```

```
get_felt_key()
```

Value

logical if has, key if get

Examples

```
has_felt_key()
```

set_felt_key	<i>Add Entry to Renviron</i>
--------------	------------------------------

Description

Adds Felt API key to .Renviron.

Usage

```
set_felt_key(key, overwrite = FALSE, install = FALSE, r_env = NULL)
```

Arguments

key	Character. API key to add to add.
overwrite	Defaults to FALSE. Boolean. Should existing FELT_KEY in Renviron be overwritten?
install	Defaults to FALSE. Boolean. Should this be added to an environment file, r_env?
r_env	Path to install to if install is TRUE.

Value

key, invisibly

Examples

```
example_env <- tempfile(fileext = '.Renviron')
set_felt_key('1234', r_env = example_env)
# r_env should likely be: file.path(Sys.getenv('HOME'), '.Renviron')
```

Index

- * **comments**
 - felt_get_comments, [6](#)
- * **edits**
 - felt_add_map_layers, [2](#)
 - felt_add_map_layers_url, [3](#)
 - felt_delete_map_layer, [5](#)
 - felt_finish_upload, [6](#)
- * **get**
 - felt_get_map, [7](#)
 - felt_get_map_elements, [8](#)
- * **key**
 - key, [10](#)
 - set_felt_key, [10](#)
- * **map**
 - felt_create_map, [4](#)
 - felt_delete_map, [5](#)
- * **style**
 - felt_get_style, [8](#)
 - felt_patch_style, [9](#)
- * **user**
 - felt_get_user, [9](#)

felt_add_map_layers, [2](#)
felt_add_map_layers_url, [3](#)
felt_create_map, [4](#)
felt_delete_map, [5](#)
felt_delete_map_layer, [5](#)
felt_finish_upload, [6](#)
felt_get_comments, [6](#)
felt_get_map, [7](#)
felt_get_map_elements, [8](#)
felt_get_map_layers(felt_get_map), [7](#)
felt_get_style, [8](#)
felt_get_user, [9](#)
felt_patch_style, [9](#)

get_felt_key(key), [10](#)

has_felt_key(key), [10](#)

key, [10](#)

set_felt_key, [10](#)

tibble::tibble, [3](#), [4](#), [7–9](#)