Package 'tgver'

July 22, 2025

Type Package

Title Turing Geovisualization Engine R package

Version 0.3.0

Maintainer Layik Hama <l.hama@leeds.ac.uk>

Description Turing Geovisualization Engine R package for geospatial visualization and analysis.

Language EN-GB

License MIT + file LICENSE

URL https://github.com/tgve/tgver

BugReports https://github.com/tgve/tgver/issues

Encoding UTF-8

LazyData true

Imports plumber, callr, geojsonsf

RoxygenNote 7.2.1

Suggests igraph, covr, curl, devtools, knitr, rmarkdown, testthat (>= 3.0.0), edgebundle, ggplot2, maps, sf, ggraph

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 2.10)

NeedsCompilation no

Author Layik Hama [aut, cre] (ORCID: <https://orcid.org/0000-0003-1912-4890>),
Lydia France [aut],
Nick Barlow [aut],
Roly Perera [aut],
Roger Beecham [aut] (ORCID: <https://orcid.org/0000-0001-8563-7251>),
Nik Lomax [aut] (ORCID: <https://orcid.org/0000-0001-9504-7570>)

Repository CRAN

Date/Publication 2022-09-30 15:20:03 UTC

Contents

`background_run` *Internal helper function to run a 'plumber' instance on specific host and port.*

Description

Internal helper function to run a 'plumber' instance on specific host and port.

Usage

```
background_run(server, port = 8000, host = "127.0.0.1")
```

Arguments

`server`	an instance of 'plumber' class
`port`	numeric port to pass to 'server' instance
`host`	character host value for 'server' instance

`copy_tgve` *Internal helper function to: 1. copy the bundled zip 2. unzip 3. cleanup as required. TODO: return a value?*

Description

Internal helper function to: 1. copy the bundled zip 2. unzip 3. cleanup as required. TODO: return a value?

Usage

```
copy_tgve(path, over.write = TRUE)
```

Arguments

`path`	character path of TGVE instance
`over.write`	boolean whether to cleanup the instance in ‘path‘

`explore_dir` *A function to read and serve at least one file from a directory.*

Description

Current version only tries to: Find two files, one .csv and the other .geojson and pass them to TGVE as ‘defaultURL‘ and ‘geographyURL‘ respectively. It will also look at their column names and try to find matching columns. If there is one file, it passes it to ‘explore_file‘ function.

Usage

```
explore_dir(path, background = FALSE)
```

Arguments

`path`	character of a data directory.
`background`	logical value whether to run instance in ‘callr‘.

Value

no value returned, depending on ‘background‘ either a or not blocking ‘plumber::pr‘ instance is started. A message is displayed with instance details.

Examples

```
{
p = file.path(tempdir(), "data")
dir.create(p)
gURL = paste0("https://raw.githubusercontent.com/saferactive/",
"tgve/main/pf-only-name.geojson")
dURL = "https://raw.githubusercontent.com/saferactive/tgve/main/ksi-pf.csv"
download.file(gURL, destfile = file.path(p, "pf.geojson"))
download.file(dURL, destfile = file.path(p, "data.csv"))

ps = explore_dir(p, background = TRUE)
ps$kill()
unlink(p, recursive = TRUE)
}
```

---

explore_file *A function to explore a spatial csv or geojson file*

Description

Current version can only read geojson/csv files using the given ‘file.uri‘.

Usage

```
explore_file(file.uri, background = FALSE)
```

Arguments

file.uri	character path of file to explore
background	Boolean to run the process in the background, defaults to ‘FALSE‘

Value

no value returned, depending on ‘background‘ either a or not blocking ‘plumber‘ instance is started. A message is displayed with instance details.

Examples

```
{
fp = file.path(tempdir(), "test.geojson")
gj = c(
'[
      {"type":"Point","coordinates":[0,0]},
      {"type":"LineString","coordinates":[[-1,-1],[1,1]]},
        {
      "type": "FeatureCollection",
      "features": [
      {
        "type": "Feature",
```

```
        "properties": {"id":1},
        "geometry": {"type": "Point", "coordinates": [100.0, 0.0]}
      }
    ]
  }
    ]'
)
write(gj, fp)
ps = tgver::explore_file(fp, background = TRUE)
ps$kill()
unlink(fp, recursive = TRUE)
}
```

`explore_sf` *Explore an sf R object using TGVE npm package.*

Description

Explore an sf R object using TGVE npm package.

Usage

```
explore_sf(
  sf = NULL,
  background = FALSE,
  static = FALSE,
  path = tempInstance()
)
```

Arguments

`sf`	a valid sf object that can be converted to geojson
`background`	Boolean to decide whether plumber
`static`	boolean to decide whether data is written to disk and self contained application is built
`path`	path of a TGVE instance, defaults to one in ‘tempdir()‘ should run in the background

Value

depending on ‘background‘ either a or not blocking ‘plumber::pr‘ object is started or returned. In the case of a ‘backgruond‘ FALSE value a message is displayed with object details.

Examples

```
{
gj = c(
'[
      {"type":"Point","coordinates":[0,0]},
      {"type":"LineString","coordinates":[[-1,-1],[1,1]]},
        {
      "type": "FeatureCollection",
      "features": [
      {
        "type": "Feature",
        "properties": {"id":1},
        "geometry": {"type": "Point", "coordinates": [100.0, 0.0]}
      }
    ]
  }
    ]'
)
sf = geojsonsf::geojson_sf(gj)
ps = tgver::explore_sf(sf, background = TRUE)
ps$kill()
}
```

`file_replace` *Function to replace patterns in given files.*

Description

Function to replace patterns in given files.

Usage

```
file_replace(files = NULL, pattern, replacement)
```

Arguments

`files`	character vector of full paths where pattern to be replaced.
`pattern`	character pattern to replace using ‘gsub‘.
`replacement`	character to replace pattern with using ‘gsub‘.

get_url *Helper function to generate URLs*

Description

The function can generate a URL based on a base URL and as many as TGVE API variables provided to the function.

Usage

```
get_url(base = "http://127.0.0.1:8000", ...)
```

Arguments

base	character URL defaults to 'http://127.0.0.1:8000'
...	any or all of the TGVE API variables to replace/add values to.

Value

character URL generated from 'base' and 'tgver::'

Examples

```
{
url = get_url(dark="false")
url == "http://127.0.0.1:8000?dark=false"
url
url = get_url()
url
}
```

help *Function to explore available API variables, their types and examples of using them in the TGVE*

Description

Function to explore available API variables, their types and examples of using them in the TGVE

Usage

```
help()
```

Value

no object is returned

Examples

```
{
help()
}
```

`is_valid_url` *Good enough regex to sanitize URLs*

Description

The task of checking a URL is "hard", see this by J. Hester: https://cran.r-project.org/web/packages/rex/vignettes/url_parsing.
To avoid having a dependency for now, let us not use "rex" R package. To try and understand the
regex please see this gist which includes a breakdown of the regex: https://gist.github.com/dperini/729294

Usage

```
is_valid_url(string)
```

Arguments

`string`	must be valid vector of URLs

`list_api_files` *Function to find what files may contain TGVE API variables for functions like 'file_replace' to consume.*

Description

Function to find what files may contain TGVE API variables for functions like 'file_replace' to consume.

Usage

```
list_api_files(path = NULL)
```

Arguments

`path`	where TGVE instance is located.

openURL *Internal helper function to "browse" a URL.*

Description

Internal helper function to "browse" a URL.

Usage

```
openURL(
  url = NULL,
  host = "127.0.0.1",
  port = 8000,
  browser = FALSE,
  path = "",
  protocol = "http://"
)
```

Arguments

url	character url, if given and valid other parameters will be ignored.
host	character host to pass to plumber
port	integer port to pass to plumber
browser	Boolean whether to specifically launch a browser
path	character path to TGVE instance
protocol	character protocol, this may change

setup *Setup an instance.*

Description

This function initializes an instance of TGVE for permanent use compared with what tgve_server which relies on a ‘tempdir‘ based instance. It requires a path.

Usage

```
setup(path = NULL, create = TRUE)
```

Arguments

path	Character URI to copy tgve instance in.
create	Boolean to create new directory at path, defaults to ‘TRUE‘.

Value

no value returned

Examples

```
{
p = file.path(tempdir(), "tgve")
setup(p)
}
```

`tempInstance` *copy the inst/tgve to a temp in an R session*

Description

copy the inst/tgve to a temp in an R session

Usage

```
tempInstance()
```

`tgve` *Open static TGVE instance*

Description

This is the main and most basic function to run an instance of TGVE without back-end. Compared to tgve_server, this function only uses internal functions to setup an instance then opens the entry HTML file. If a path of an instance is provided it opens it, otherwise creates an instance from a 'tempdir'.

Usage

```
tgve(
  path = Sys.getenv("TEMP_path_ENV"),
  browse = TRUE,
  remote = FALSE,
  url = "https://tgve.github.io/app/"
)
```

Arguments

path	character directory of a current instance to browse. Defaults to 'TEMP_PATH_ENV' environment variable.
browse	boolean to decide whether to browse the instance or not.
remote	boolean whether to run a remote instance of TGVE. If TRUE the above 'path' and 'browse' parameters will be ignored. Defaults to 'FALSE'
url	if 'remote' is true, then this will be used as the parameter to pass to internal function 'openURL'. It defaults, for convenience, to 'https://tgve.github.io/app/' instance.

Value

directory of the new instance if 'path' is not provided.

Examples

```
tgve()
# just get the path of the HTML
p = tgve(browse = FALSE)
file.exists(p)
```

---

tgve_server *Start a TGVE instance server*

Description

The function accepts a 'path' to get a directory containing an instance of TGVE, by default this is done via 'TEMP_DIR_ENV' env variable. If neither is given then the function copies a clean copy of the bundled TGVE version into a temporary directory.

Usage

```
tgve_server(
  path = Sys.getenv("TEMP_path_ENV"),
  port = 8000,
  host = "127.0.0.1",
  background = FALSE,
  run = TRUE
)
```

Arguments

path	location of TGVE path to be served by plumber.
port	to serve from.
host	host to pass to plumber default 'http://127.0.0.1' to 'FALSE'.
background	run the R process in the background using 'callr', defaults to 'TRUE'.
run	whether to start the server, defaults to 'TRUE'. If not, then the created server will be returned.

Value

the valude returned depends on: (1) 'run', if it is FALSE then an instance of 'plumber::pr', (2) if 'run' is true and 'background' is TRUE the 'plumber::pr' instance is started and its process is returned, and (3) if 'run' is TRUE and 'background' is FALSE then a message is displayed showing the blocking 'plumber::pr' instance's 'path', 'port' and 'host'.

Examples

```
{
# This will run in the background using `callr`
ps = tgve_server(background = TRUE)
Sys.sleep(2)
ps$kill()
}
```

---

version *Version of the tgvejs npm package bundled in 'tgver'*

Description

Version of the tgvejs npm package bundled in 'tgver'

Package version included as data

Format

A character vector

Note

This was generated using the script in the 'data' directory ('tgver.R' file).

Author(s)

L Hama <l.hama@leeds.ac.uk>

Examples

```
{
tgver::version
}
```

Index