Package: mvtview (via r-universe)

January 22, 2025

Title View and Serve Mapbox Vector Tile Databases

Version 0.0.3

Description View and Serve Mapbox Vector Tile Databases for mapping development tasks.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE, roclets = c(``collate", ``namespace", ``rd", ``roxyglobals::global_roclet"))

RoxygenNote 7.1.2

Imports callr, DBI, dplyr, fs, glue, httpuv, jsonlite, magrittr, plumber, purrr, rdeck (>= 0.3.0.91000), RSQLite, stats, stringr

Remotes anthonynorth/rdeck, anthonynorth/roxyglobals

Suggests roxyglobals (>= 0.2.1)

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev make libicu-dev libsodium-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

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

RemoteUrl https://github.com/milesmcbain/mvtview

RemoteRef main

RemoteSha 90174f8e4fd77303c4a8eae1ffe6c06b00954b16

Contents

clean_mvt 2
serve_mvt 2
start_mvt_server 3
view_mvt 4

Index **6**

clean_mvt *Stop all running vector tile servers*

Description

As you use serve_mvt or view_mvt servers will accumulate in child processes. This function kills all child processes serving tiles.

Usage

```
clean_mvt()
```

serve_mvt *Serve a .mbtiles database of vectortiles*

Description

Starts a web server in a background R session serving vector tiles from a supplied .mbtiles file.

Usage

```
serve_mvt(tiles_path, port = NULL, .serve_mode = "in-memory")
```

Arguments

tiles_path	The path to an .mbtiles file.
port	The port to for the server to serve mbtiles on. Default is a random available port.
.serve_mode	The way in which the server handles the vector tiles database. "in-memory" is the default and it will read the entire tile database into R as a tibble. "disk" will read tiles from the .mbtiles file as an SQLite database from disk. The default is more performant. Use "disk" only if you have a large vector tileset that would consume too much memory to hold in RAM at once.

See Also

start_mvt_server for more control of server behaviour.

`start_mvt_server` *Start an mvt_server in the current session*

Description

Starts a web server serving vector tiles from a supplied .mbtiles file.

Usage

```
start_mvt_server(
  tiles_path,
  host = "0.0.0.0",
  port = NULL,
  .serve_mode = "in-memory"
)
```

Arguments

`tiles_path`	The path to an .mbtiles file.
`host`	the host to serve tiles on
`port`	the port to serve tiles on
`.serve_mode`	The way in which the server handles the vector tiles database. "in-memory" is the default and it will read the entire tile database into R as a tibble. "disk" will read tiles from the .mbtiles file as an SQLite database from disk. The default is more performant. Use "disk" only if you have a large vector tileset that would consume too much memory to hold in RAM at once.

Details

`serve_mvt()` is likely more convenient. Only use this if if you want more control of the host and port on which your tiles are served on.

Where `serve_mvt()` verifies the server is actually up and responding, this funciton does not. So that's up to you to take on.

Note: This server has been built minimising code written, not ' maximising performance. It is intended for local development work, and will likely not be performant enough for any production use-case.

`view_mvt` *View a local vector tileset on a map*

Description

Given a local .mtiles file containing a vector tiles database, this function will start a local development server to serve the tiles and then return a htmlwidget map that displays the tileset.

Usage

```
view_mvt(
  tiles_path,
  get_fill_color = "#FFFFFF70",
  get_line_color = "#ffffffff",
  get_line_width = 2,
  line_width_units = "pixels",
  get_point_radius = 2,
  point_radius_units = "pixels",
  stroked = TRUE,
  tooltip = TRUE,
  pickable = TRUE,
  ...,
  .serve_mode = "in-memory"
)
```

Arguments

`tiles_path`	The path to an .mbtiles file.
`get_fill_color`	the fill colour of plotted features.
`get_line_color`	the line colour of plotted features.
`get_line_width`	the line width of plotted features (in pixels by default).
`line_width_units`	the units of the value supplied in `get_line_width`. "meters" may be preferred in some cases.
`get_point_radius`	the radius of plotted point features (in pixels by default).
`point_radius_units`	the units of the value supplied in `get_point_radius`. "meters" may be preferred in some cases.
`stroked`	use a line on the borders of polygons or points? TRUE by default.
`tooltip`	generate a tooltip for feature attributes? TRUE by default.
`pickable`	allow map to react to features that get mouse hover? Needs to be enabled to view tooltips. TRUE by default.
`...`	further arguments forwarded to `rdeck::add_mvt_layer()`.

`.serve_mode` The way in which the server handles the vector tiles database. "in-memory" is the default and it will read the entire tile database into R as a tibble. "disk" will read tiles from the .mbtiles file as an SQLite database from disk. The default is more performant. Use "disk" only if you have a large vector tileset that would consume too much memory to hold in RAM at once.

Details

The map is powered by the awesome rdeck package, which is highly recommended for making interactive WebGL maps in R.

The graphics options of this function are passed directed to ' `rdeck::add_mvt_layer()`, and so suppourt rdeck color scales based on attributes. See the rdeck helpfile for more detailed ' descriptions.

The graphics parameters apply only to relvant geometries. For ' example: 'fill color' is not used for line string features.

Index

clean_mvt, 2

rdeck::add_mvt_layer(), *4*, *5*

serve_mvt, 2
serve_mvt(), *3*
start_mvt_server, 3

view_mvt, 4