

Package ‘overtureR’

August 1, 2024

Title Load 'Overture' Datasets as 'dbplyr' and 'sf'-Ready Data Frames

Version 0.1.0

Description An integrated R interface to the 'Overture' API (<https://docs.overturemaps.org/>). Allows R users to return 'Overture' data as 'dbplyr' data frames or materialized 'sf' spatial data frames.

License MIT + file LICENSE

Suggests bench, duckdbfs, httr, jsonlite, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

URL <https://github.com/arthurgailes/overtureR>,
<https://arthurgailes.github.io/overtureR/>

BugReports <https://github.com/arthurgailes/overtureR/issues>

Imports DBI, dbplyr, dplyr (>= 1.0.0), duckdb (>= 1.0.0), glue, sf

NeedsCompilation no

Author Arthur Gailes [aut, cre, cph] (<https://orcid.org/0009-0006-8176-8653>)

Maintainer Arthur Gailes <agailes1@gmail.com>

Repository CRAN

Date/Publication 2024-08-01 09:30:02 UTC

Contents

collect_sf	2
config_extensions	2
open_curtain	3
stage_conn	4

Index	6
--------------	----------

collect_sf	<i>Convert dbplyr table to sf Object</i>
------------	--

Description

Collects a lazy dbplyr view and materializes it as an in-memory sf table.

Usage

```
collect_sf(tbl, geom_col = NULL, crs = 4326)
```

Arguments

tbl	A dbplyr view object containing a 'geometry' column.
geom_col	The name of the geometry column. Will auto-detect names matching 'geom'.
crs	The coordinate reference system to use for the geometries, specified by its EPSG code. The default is 4326 (WGS 84).

Value

An 'sf' object with the dataset converted to spatial features.

Examples

```
bbox <- c(xmin = -120.5, ymin = 35.5, xmax = -120.0, ymax = 36.0)
lazy_tbl <- open_curtain("building", bbox)
collect_sf(lazy_tbl)
```

config_extensions	<i>Check duckdb extension and config settings</i>
-------------------	---

Description

Check duckdb extension and config settings

Usage

```
config_extensions(conn)
```

Arguments

conn	A connection to a duckdb database.
------	------------------------------------

open_curtain

Retrieve (Spatially Filtered) Overture Datasets

Description

Fetches overture data from AWS. If a bounding box is provided, it applies spatial filtering to only include records within that area. The core code is copied from duckdbfs, which deserves all credit for the implementation

Usage

```
open_curtain(
  type,
  bbox = NULL,
  theme = get_theme_from_type(type),
  conn = NULL,
  as_sf = FALSE,
  mode = "view",
  tablename = ifelse(is.null(type) | type == "*", theme, type),
  union_by_name = FALSE,
  base_url = "s3://overturemaps-us-west-2/release/2024-07-22.0"
)
```

Arguments

type	A string specifying the type of overture dataset to read. Setting to "*" or NULL will read all types for a given theme.
bbox	Optional bounding box to filter the records, expected as a vector of four numbers (xmin, ymin, xmax, ymax). Set to NULL to omit from query
theme	Inferred from type by default. Must be set if type is "*" or NULL
conn	A connection to a duckdb database.
as_sf	If TRUE, return an sf dataframe
mode	Either "view" (default) or "table". If "table", will download the dataset into memory.
tablename	The name of the table to create in the database.
union_by_name	If TRUE, will execute a UNION by column name across all files (NOTE: this can add considerably to the initial execution time)
base_url	Allows user to download data from a different mirror, such as a beta or alpha release.

Value

An dbplyr lazy dataframe, or an sf dataframe if as_sf is TRUE

Examples

```
bbox <- c(xmin = -120.5, ymin = 35.5, xmax = -120.0, ymax = 36.0)
open_curtain("building", bbox)
```

stage_conn

create a cachable duckdb connection. In dev

Description

stage_conn is primarily intended for internal use by other overtureR functions. However, it can be called directly by the user whenever it is desirable to have direct access to the connection object. The core code is copied from duckdbfs, which deserves all credit for the implementation

Usage

```
stage_conn(
  dbdir = ":memory:",
  read_only = FALSE,
  bigint = "numeric",
  config = list(),
  ...
)

strike_stage(conn = stage_conn())
```

Arguments

dbdir	Location for database files. Should be a path to an existing directory in the file system. With the default (or ""), all data is kept in RAM.
read_only	Set to TRUE for read-only operation. For file-based databases, this is only applied when the database file is opened for the first time. Subsequent connections (via the same drv object or a drv object pointing to the same path) will silently ignore this flag.
bigint	How 64-bit integers should be returned. There are two options: "numeric" and "integer64". If "numeric" is selected, bigint integers will be treated as double/numeric. If "integer64" is selected, bigint integers will be set to bit64 encoding.
config	Named list with DuckDB configuration flags, see https://duckdb.org/docs/configuration/overview#configuration-reference for the possible options. These flags are only applied when the database object is instantiated. Subsequent connections will silently ignore these flags.
...	Further arguments passed to DBI::dbConnect
conn	A duckdb_connection object

Details

When first called (by a user or internal function), this function both creates a duckdb connection and places that connection into a cache (`overturer_conn` option). On subsequent calls, this function returns the cached connection, rather than recreating a fresh connection.

This frees the user from the responsibility of managing a connection object, because functions needing access to the connection can use this to create or access the existing connection. At the close of the global environment, this function's finalizer should gracefully shutdown the connection before removing the cache.

`strike_stage` closes the connection.

Value

a `duckdb::duckdb()` connection object

Examples

```
con <- stage_conn()
strike_stage(con)
```

Index

`collect_sf`, 2

`config_extensions`, 2

`DBI::dbConnect`, 4

`duckdb::duckdb()`, 5

`open_curtain`, 3

`stage_conn`, 4

`strike_stage(stage_conn)`, 4