Package: gtfsanalyzer (via r-universe)

November 26, 2026

Type Package

Title Validate and Analyse 'GTFS' Schedule Feeds with 'GTFS Analyzer'

Version 0.1.0

Description An interface to the 'GTFS Analyzer' command line validator <https://github.com/ttezer/gtfs-analyzer> for 'GTFS' (General Transit Feed Specification) Schedule feeds. Runs the validator on a feed archive and returns publishability, quality scores and findings as R objects, and compares two versions of a feed. The validator is a standalone executable; it can be installed into the user cache directory with a checksum-verified download, or an existing executable can be used.

License MIT + file LICENSE

URL https://github.com/ttezer/gtfs-analyzer

BugReports https://github.com/ttezer/gtfs-analyzer/issues

Encoding UTF-8

SystemRequirements GTFS Analyzer command line executable (<https://github.com/ttezer/gtfs-analyzer/releases>), installed with ga_install() or supplied through the GTFS_ANALYZER_BIN environment variable.

Depends R (>= 4.1)

Imports jsonlite, stats, tools, utils

Suggests openssl, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

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

Date/Publication 2026-11-26 13:59:32 UTC

RemoteUrl https://github.com/ttezer/gtfs-analyzer

RemoteRef HEAD

RemoteSha af3138b4b713dfc3cad1dc2ca6b14a672e8ae3ef

RemoteSubdir r

Contents

ga_analyze 2
ga_binary 3
ga_compare 3
ga_install 4
ga_rules 5

Index **6**

`ga_analyze` *Validate a GTFS feed*

Description

Runs GTFS Analyzer on a GTFS Schedule ZIP archive and returns the publishability verdict, quality scores and findings.

Usage

```
ga_analyze(feed, lang = "en", today = NULL, max_notices_per_rule = NULL)
```

Arguments

`feed`	Path to a GTFS ZIP archive.
`lang`	Language of titles, messages and remediations: one of ‘"en"‘, ‘"tr"‘, ‘"ja"‘, ‘"fr"‘ or ‘"es"‘.
`today`	Validation date as ‘"YYYYMMDD"‘, or ‘NULL‘ for the current date.
`max_notices_per_rule`	Keep at most this many findings per rule, or ‘NULL‘ for all. Scores and the verdict always cover every finding.

Details

The validator’s exit code 1 only means that findings exist; it is not treated as an error. A feed the validator cannot read at all (exit code 2) signals an error of class ‘gtfsanalyzer_fatal‘.

Value

An object of class ‘ga_result‘: a list with ‘publishable‘, ‘scores‘ (named numeric), ‘notices‘ (data frame), ‘status‘, ‘engine‘ and ‘raw‘ (the parsed JSON).

Examples

```
if (nzchar(ga_binary(required = FALSE))) {
  res <- ga_analyze(system.file("extdata", "sample-feed.zip", package = "gtfsanalyzer"))
  res
}
```

ga_binary *Locate the GTFS Analyzer executable*

Description

Looks, in order, at the ‘GTFS_ANALYZER_BIN‘ environment variable, the ‘gtfsanalyzer.bin‘ option, the copy installed by [ga_install()] for the pinned engine version, and finally ‘gtfs-analyzer‘ on the ‘PATH‘.

Usage

```
ga_binary(required = TRUE)
```

Arguments

required	If ‘TRUE‘, signal an error when no executable is found.

Value

The path to the executable, or ‘""‘ when none is found and ‘required‘ is ‘FALSE‘.

Examples

```
ga_binary(required = FALSE)
```

ga_compare *Compare two versions of a GTFS feed*

Description

Validates both feeds (or takes existing [ga_analyze()] results) and reports per rule how the number of findings changed, plus the change in scores. Useful before and after editing a timetable.

Usage

```
ga_compare(before, after, ...)
```

Arguments

before, after	Paths to GTFS ZIP archives, or ‘ga_result‘ objects.
...	Passed to [ga_analyze()] when a path is given.

Value

An object of class ‘ga_comparison‘: a list with ‘rules‘ (data frame with ‘rule_id‘, ‘severity‘, ‘before‘, ‘after‘, ‘change‘ and ‘state‘, one of ‘"new"‘, ‘"resolved"‘, ‘"increased"‘, ‘"decreased"‘), ‘scores‘ (data frame of before/after/delta) and ‘publishable‘ (before/after).

Examples

```
if (nzchar(ga_binary(required = FALSE))) {
  f <- system.file("extdata", "sample-feed.zip", package = "gtfsanalyzer")
  ga_compare(f, f)
}
```

`ga_install` *Download and install the GTFS Analyzer executable*

Description

Downloads the release archive for the current platform from GitHub, verifies its SHA256 checksum against the value pinned in this package and unpacks it into the user cache directory ([tools::R_user_dir()]). Nothing is downloaded unless this function is called. Verification needs the 'openssl' package; without it the function stops before downloading.

Usage

```
ga_install(version = ga_engine_version, force = FALSE)
```

Arguments

`version`	Engine version to install. Only versions with a pinned checksum can be installed.
`force`	Reinstall even if the executable is already present.

Value

The path to the installed executable, invisibly.

Examples

```
## Not run:
ga_install()

## End(Not run)
```

ga_rules *List the validator's rules*

Description

List the validator's rules

Usage

```
ga_rules(lang = "en")
```

Arguments

lang	Language of titles: one of ‘"en"‘, ‘"tr"‘, ‘"ja"‘, ‘"fr"‘, ‘"es"‘.

Value

A data frame with one row per rule: ‘id‘, ‘severity‘, ‘class‘, ‘authority_source‘, ‘base_effort‘ and ‘title‘.

Examples

```
if (nzchar(ga_binary(required = FALSE))) {
  head(ga_rules())
}
```

Index

ga_analyze, 2
ga_binary, 3
ga_compare, 3
ga_install, 4
ga_rules, 5