Package ‘typetracer’

October 14, 2022

Title Trace Function Parameter Types
Version 0.1.1
Description The ‘R’ language includes a set of defined types, but the language itself is “absurdly dynamic” (Turcotte & Vitek (2019) <doi:10.1145/3340670.3342426>), and lacks any way to specify which types are expected by any expression. The ‘typetracer’ package enables code to be traced to extract detailed information on the properties of parameters passed to ‘R’ functions. ‘typetracer’ can trace individual functions or entire packages.
License MIT + file LICENSE
URL https://github.com/mpadge/typetracer,
https://mpadge.github.io/typetracer/
BugReports https://github.com/mpadge/typetracer/issues
Imports brio, checkmate, methods, tibble, withr
Suggests knitr, rematch, rmarkdown, testthat (>= 3.0.0)
VignetteBuilder knitr
Config/testthat/edition 3
Encoding UTF-8
Language en-GB
RoxygenNote 7.2.1
NeedsCompilation yes
Author Mark Padgham [aut, cre] (<https://orcid.org/0000-0003-2172-5265>), Filip Krikava [ctb] (Author of original ‘injectr’ code on which this package builds; https://github.com/PRL-PRG/injectr), covr authors [cph] (Original authors of sections of code from ‘covr’ package included here in modified form.)
Maintainer Mark Padgham <mark.padgham@email.com>
Repository CRAN
Date/Publication 2022-08-27 08:20:02 UTC
R topics documented:

- clear_traces
- inject_tracer
- load_traces
- trace_package
- uninject_tracer

Index

---

**clear_traces**  
Clear previous traces

**Description**

Traces are by default appended to previous traces. This function can be used to clean those previous ones, to enable subsequent calls to generate new traces that are not appended to previous ones.

**Usage**

```r
clear_traces()
```

**Value**

(Invisibly) A single logical value indicating whether or not traces were successfully cleared.

---

**inject_tracer**  
Inject parameter tracer into one function

**Description**

Inject parameter tracer into one function

**Usage**

```r
inject_tracer(f)
```

**Arguments**

- `f`  
  A function (that is, an object of class "function", and not a character string).

**Value**

Nothing (will error on fail).
load_traces

Note

The tracer is defined in the internal get_types function. This uses an options variable defined on package load for the current tempdir, defining a single location where all traced values are dumped. This is done via options to allow both multi-threaded function calls and calls via callr to be traced.

load_traces | Load traces of parameter types

Description

Load traces of parameter types

Usage

load_traces(files = FALSE, quiet = FALSE)

Arguments

files If TRUE, return paths to all temporary files holding trace data.
quiet If FALSE, issue message when no traces found.

Value

A 'data.frame' of traces, including names of functions and parameters, and values of each parameter traced in both unevaluated and evaluated forms.

trace_package | Trace all parameters for all functions in a specified package

Description

Trace all parameters for all functions in a specified package

Usage

trace_package(
  package = NULL,
  functions = NULL,
  types = c("examples", "tests"),
  pkg_dir = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>package</td>
<td>Name of package to be traced (as character value)</td>
</tr>
<tr>
<td>functions</td>
<td>Optional character vector of names of functions to trace. Defaults to tracing all functions.</td>
</tr>
<tr>
<td>types</td>
<td>The types of code to be run to generate traces: one or both values of &quot;examples&quot; or &quot;tests&quot; (as for tools::testInstalledPackage).</td>
</tr>
<tr>
<td>pkg_dir</td>
<td>For &quot;types&quot; including &quot;tests&quot;, a local directory to the source code of the package. (This is needed because installed versions do not generally include tests.)</td>
</tr>
</tbody>
</table>

Value

A data.frame of data on every parameter of every function as specified in code provided in package examples.

uninject_tracer Remove parameter tracer from one function

Description

This function removes traces previous injected into functions with the inject_tracer function.

Usage

uninject_tracer(f)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>A function (that is, an object of class &quot;function&quot;, and not a character string).</td>
</tr>
</tbody>
</table>

Value

Logical value indicating whether or not tracer was able to be removed ("uninjected").
Index

clear_traces, 2
inject_tracer, 2, 4
load_traces, 3
trace_package, 3
uninjected_tracer, 4