Package: spdl (via r-universe)

July 5, 2024

Type Package

Title Easier Use of 'RcppSpdlog' Functions via Wrapper

Description Logging functions in 'RcppSpdlog' provide access to the logging functionality from the 'spdlog' 'C++' library. This package offers shorter convenience wrappers for the 'R' functions which match the 'C++' functions, namely via, say, 'spdl::debug()' at the debug level. The actual formatting is done by the 'fmt::format()' function from the 'fmtlib' library (that is also 'std::format()' in 'C++20' or later).

Version 0.0.5

Date 2023-06-18

License GPL (>= 2)

Imports RcppSpdlog (>= 0.0.13)

URL https://github.com/eddelbuettel/spdl

BugReports https://github.com/eddelbuettel/spdl/issues

RoxygenNote 6.0.1

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

RemoteUrl https://github.com/eddelbuettel/spdl

RemoteRef HEAD

RemoteSha 4c1eb4bafa730ba0f159d3e311b4e5ea6cb240b2

Contents

setup *Convenience Wrappers for 'RcppSpdlog' Logging From 'spdlog'*

Description

Several short wrappers for functions from 'RcppSpdlog' package are provided as a convenience. Given the potential for clashing names of common and popular functions names we do *not* recommend the import the whole package but rather do `importFrom(RcppSpdlog, set_pattern)` (or maybe `importFrom(RcppSpdlog,set_pattern)`). After that, functionality can be accessed via a convenient shorter form such as for example `spdl::info()` to log at the 'info' level. Format strings suitable for the C++ library 'fmtlib::fmt' and its `fmt::format()` (which as of C++20 becomes 'std::fmt') are supported so the `{}` is the placeholder for simple (scalar) arguments (for which the default R formatter is called before passing on a character representation).

Usage

```
setup(name = "default", level = "warn")

init(level = "warn")

log(level = "warn")

filesetup(s, name = "default", level = "warn")

drop(s)

set_pattern(s)

set_level(s)

trace(s, ...)

debug(s, ...)

info(s, ...)

warn(s, ...)

error(s, ...)

critical(s, ...)

fmt(s, ...)

cat(...)

stopwatch()
```

```
elapsed(w)
```

Arguments

name	Character value for the name of the logger instance
level	Character value for the logging level
s	Character value for filename, pattern, level, or logging message
...	Supplementary arguments for the logging string
w	Stopwatch object

Value

Nothing is returned from these functions as they are invoked for their side-effects.

Examples

```
spdl::setup("exampleDemo", "warn")
# and  spdl::init("warn")  and  spdl::log("warn")  are shortcuts
spdl::info("Not seen as level 'info' below 'warn'")
spdl::warn("This warning message is seen")
spdl::set_level("info")
spdl::info("Now this informational message is seen too")
spdl::info("Calls use fmtlib::fmt {} as we can see {}", "under the hood", 42L)
```

Index