Package 'bindrcpp'

May 7, 2026

Title An 'Rcpp' Interface to Active Bindings

Version 0.2.4

Date 2026-02-05

Description Provides an easy way to fill an environment with active bindings that call a C++ function.

License MIT + file LICENSE

URL https://github.com/krlmlr/bindrcpp, https://krlmlr.github.io/bindrcpp/

BugReports https://github.com/krlmlr/bindrcpp/issues

Imports bindr (>= 0.1.1), Rcpp (>= 0.12.16)

Suggests testthat

LinkingTo Rcpp

Config/Needs/check RcppCore/Rcpp

Encoding UTF-8

RoxygenNote 7.3.3.9000

NeedsCompilation yes

Author Kirill Müller [aut, cre] (ORCID: <https://orcid.org/0000-0002-1416-3412>), RStudio [cph]

Maintainer Kirill Müller <kirill@cynkra.com>

Repository CRAN

Date/Publication 2026-02-06 10:50:02 UTC

Contents

bindrcpp-package *bindrcpp: An 'Rcpp' Interface to Active Bindings*

Description

Provides an easy way to fill an environment with active bindings that call a C++ function.

Details

Use `LinkingTo: bindrcpp` in `DESCRIPTION` and `#include <bindrcpp.h>` in your C++ headers and/or modules to access the C++ functions provided by this package:

- `create_env_string()` creates an environment with active bindings, with names given as a character vector. Access of these bindings triggers a call to a C++ function with a fixed signature (`GETTER_FUNC_STRING`); this call contains the name of the binding (as character) and an arbitrary payload (`PAYLOAD`, essentially a wrapped `void*`).
- `create_env_symbol()` is similar, the callback function accepts the name of the binding as symbol instead of character (`GETTER_FUNC_SYMBOL`).
- `populate_env_string()` and `populate_env_symbol()` populate an existing environment instead of creating a new one.

Author(s)

Maintainer: Kirill Müller <kirill@cynkra.com> (ORCID)

Other contributors:

- RStudio [copyright holder]

See Also

Useful links:

- https://github.com/krlmlr/bindrcpp
- https://krlmlr.github.io/bindrcpp/
- Report bugs at https://github.com/krlmlr/bindrcpp/issues

Index