

Package ‘pudu’

January 14, 2025

Type Package

Title C++ Tools for Cleaning Strings

Description Provides function declarations and inline function definitions that facilitate cleaning strings in C++ code before passing them to R.

Version 0.1.0

Suggests cpp11, desc, knitr, mockery, rmarkdown, testthat (>= 3.0.0), withr

Depends R(>= 3.5.0)

License Apache License (>= 2)

BugReports <https://github.com/pachadotdev/pudu/issues>

URL <https://pacha.dev/pudu/>, <https://github.com/pachadotdev/pudu>

RoxygenNote 7.3.2

Encoding UTF-8

Config/testthat/edition 3

NeedsCompilation no

Author Mauricio Vargas Sepulveda [aut, cre]
(<https://orcid.org/0000-0003-1017-7574>),
Munk School of Global Affairs and Public Policy [fnd]

Maintainer Mauricio Vargas Sepulveda <m.sepulveda@mail.utoronto.ca>

Repository CRAN

Date/Publication 2025-01-14 15:00:02 UTC

Contents

cpp_vendor	2
Index	3

`cpp_vendor`*Vendor the cpp11 and pudu dependency*

Description

Vendoring is the act of making your own copy of the 3rd party packages your project is using. It is often used in the go language community.

Usage

```
cpp_vendor(dir = NULL, subdir = "/inst/include")
```

Arguments

<code>dir</code>	The directory to vendor the code into.
<code>subdir</code>	The subdirectory to vendor the code into.

Details

This function vendors `cpp11` and `pudu` into your package by copying the `cpp11` and `pudu` headers into the `'inst/include'` folder and adding `'cpp11 version: XYZ'` and `'pudu version: XYZ'` to the top of the files, where `XYZ` is the version of `cpp11` and `pudu` currently installed on your machine.

Vendoring places the responsibility of updating the code on you. Bugfixes and new features in `cpp11` and `pudu` will not be available for your code until you run `'cpp_vendor()'` again.

Value

The file path to the vendored code (invisibly).

Examples

```
# create a new directory
dir <- tempdir()
dir.create(dir)

# vendor the cpp11 headers into the directory
cpp_vendor(dir)
```

Index

`cpp_vendor`, 2