

Package: kwb.heatsine.opencpu (via r-universe)

October 10, 2026

Title Wrapper for R Package kwb.heatsine for OpenCPU Usage

Version 0.1.0

Description Contains higher level functions build on top of R package kwb.heatsine for connecting with OpenCPU.

License MIT + file LICENSE

URL <https://github.com/KWB-R/kwb.heatsine.opencpu>

BugReports <https://github.com/KWB-R/kwb.heatsine.opencpu/issues>

Depends R (>= 2.10)

Imports jsonlite (>= 1.7.1), kwb.heatsine (>= 0.1.5), tibble (>= 3.0.4)

Suggests knitr (>= 1.30), rmarkdown (>= 2.4), testthat (>= 2.3.2)

VignetteBuilder knitr

Remotes github::kwb-r/kwb.heatsine@v0.1.5

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libx11-dev

Repository <https://kwb-r.r-universe.dev>

Date/Publication 2020-10-19 11:18:15 UTC

RemoteUrl <https://github.com/KWB-R/kwb.heatsine.opencpu>

RemoteRef HEAD

RemoteSha 8933a1daf003f8941078dd5b16741456eef501ce

Contents

json_args	2
run_optimisation	2

Index	4
-------	---

json_args	<i>json_args: test dataset for run_optimisation()</i>
-----------	---

Description

test dataset generated with R script (data-raw/DATASET.R)

Usage

data(json_args)

Format

A json list with all parameters used by [run_optimisation](#), which are passed to [run_optimisation](#)

Examples

json_args

run_optimisation	<i>JSON wrapper function for kwb.heatsine::run_optimisation</i>
------------------	---

Description

JSON wrapper function for kwb.heatsine::run_optimisation

Usage

```
run_optimisation(  
  data_sw_selected,  
  data_gw_selected,  
  retardation_factor = 1.8,  
  sw_monitoring_id = attr(data_sw_selected, "monitoring_id"),  
  gw_monitoring_id = attr(data_gw_selected, "monitoring_id"),  
  limits = c(100, 500),  
  tolerance = 0.001,  
  debug = FALSE  
)
```

Arguments

data_sw_selected	data.frame with daily data temperature data of surface water monitoring point with columns "date" (format: "YYYY-MM-DD") and "value" (format: double, temperature in degree Celsius) for selected time period
data_gw_selected	data.frame with daily data temperature data of groundwater monitoring point with columns "date" (format: "YYYY-MM-DD") and "value" (format: double, temperature in degree Celsius) for selected time period
retardation_factor	hydraulic retardation factor (default: 2)
sw_monitoring_id	optional label for surface water monitoring id (default: "surface-water monitoring point" or attr(data_sw_selected, "monitoring_id") if data imported with load_temperature_from_csv), otherwise can be any user-defined character string to be used as label for the monitoring point
gw_monitoring_id	optional label for groundwater monitoring id (default: "surface-water monitoring point" or attr(data_sw_selected, "monitoring_id") if data imported with load_temperature_from_csv), otherwise can be any user-defined character string to be used as label for the monitoring point
limits	minimum/maximum period length for sinus optimisation in days (default: c(100, 500))
tolerance	the desired accuracy (default: 0.001)
debug	show debug messages (default: FALSE)

Value

json list with sim/observation data ("data") fit parameters ("paras"), goodness-of-fit values ("gof") traveltimes ("traveltimes") and special (min, max, turning) points ("points") as returned by [get_predictions](#)

Index

* **datasets**

 json_args, [2](#)

get_predictions, [3](#)

json_args, [2](#)

load_temperature_from_csv, [3](#)

run_optimisation, [2](#), [2](#)