

Package: valve (via r-universe)

September 4, 2024

Title Redirects Your Plumbing For You

Version 0.1.3

Description Scales Plumber APIs in parallel on demand. Valve is a Rust library, cli tool, and R package, that spawns plumber APIs in parallel. Valve creates a connection pool of Plumber APIs which are fetched based on availability. Unused connections are terminated. Valve is built using the powerful extendr, tokio, and deadpool libraries.

License BSD_3_clause + file LICENSE

URL <https://github.com/JosiahParry/valve>,
<https://josiahparry.r-universe.dev/valve>,
<https://rsgeo.josiahparry.com/>

BugReports <https://github.com/JosiahParry/valve/issues/>

Encoding UTF-8

Language en

SystemRequirements Cargo (rustc package manager)

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Config/rextendr/version 0.3.1.9000

Imports plumber

Suggests rmarkdown, vetiver

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

RemoteUrl <https://github.com/josiahparry/valve>

RemoteRef HEAD

RemoteSha ab8b177486f11a7bbf40e214b86570e7d5a482b2

Contents

| | |
|-------------------------------|---|
| valve_run | 2 |
| valve_write_vetiver | 3 |

| | |
|-----------|--------------------------|
| valve_run | <i>Start a valve App</i> |
|-----------|--------------------------|

Description

Run a plumber API in parallel using valve. Plumber APIs are spawned on `n_threads` asynchronous threads on random ports. Incoming requests are handled on a single port and redirected to the plumber APIs in a simple round-robin fashion.

Usage

```
valve_run(
  filepath = "plumber.R",
  host = "127.0.0.1",
  port = 3000,
  n_min = 1,
  n_max = 3,
  workers = n_max,
  check_unused = 10,
  max_age = 300
)
```

Arguments

| | |
|---------------------------|--|
| <code>filepath</code> | default "plumber.R". The path to the Plumber API. Provided to the file argument of <code>plumber::plumb()</code> . |
| <code>host</code> | default "127.0.0.1". Where to host the valve app and Plumber APIs. |
| <code>port</code> | default 3000. The port to host the valve app on. |
| <code>n_min</code> | default 1. The minimum number of Plumber APIs made available. Must be smaller than <code>n_max</code> . |
| <code>n_max</code> | default 3. The maximum number of Plumber APIs to run in parallel. |
| <code>workers</code> | default <code>n_max</code> . The number of worker threads in the valve app to execute requests. This number should typically mimic <code>n_max</code> . |
| <code>check_unused</code> | default 10. The time interval, in seconds, to check for unused connections. |
| <code>max_age</code> | default 300 (five minutes). Specifies how long a connection can go unused without being terminated. If a connection reaches this age it will be terminated in the next pool check (interval determined by <code>check_unused</code>), |

Examples

```
if (interactive()) {
  plumber_api_path <- system.file("plumber.R", package = "valve")
  valve_run(plumber_api_path)
}
```

| | |
|---------------------|---|
| valve_write_vetiver | <i>Write a Dockerfile for vetiver using valve</i> |
|---------------------|---|

Description

A Valve powered vetiver Dockerfile will run prediction endpoints concurrently.

Usage

```
valve_write_vetiver(  
  vetiver_model,  
  plumber_file = "plumber.R",  
  path = ".",  
  port = 8000,  
  vetiver_args = list(),  
  workers = NULL,  
  n_max = NULL,  
  check_unused = NULL,  
  max_age = NULL  
)
```

Arguments

| | |
|---------------|--|
| vetiver_model | A deployable <code>vetiver_model()</code> object |
| plumber_file | A path for your Plumber file, created via <code>vetiver_write_plumber()</code> . Defaults to <code>plumber.R</code> in the working directory. |
| path | A path to write the Dockerfile and lockfile, capturing the model's package dependencies. Defaults to the working directory. |
| port | The server port for listening: a number such as 8080 or an expression like <code>'as.numeric(Sys.getenv("PORT"))'</code> when the port is injected as an environment variable. |
| vetiver_args | additional arguments passed to <code>vetiver::vetiver_write_docker()</code> as key-value pairs in a list object. |

Details

This function is a modification of `vetiver::vetiver_write_docker()`. It modifies the created Dockerfile to install Valve, changes the ENTRYPOINT to use the Valve executable instead of a single plumber API via an R command.

Value

The content of the Dockerfile, invisibly.

Index

valve_run, [2](#)
valve_write_vetiver, [3](#)
vetiver_model(), [3](#)
vetiver_write_plumber(), [3](#)