

# Package ‘promr’

October 14, 2022

**Type** Package

**Title** Prometheus 'PromQL' Query Client for 'R'

**Version** 0.1.3

**Description** A native 'R' client library for querying the 'Prometheus' time-series database, using the 'PromQL' query language.

**URL** <https://github.com/domodwyer/promr>

**BugReports** <https://github.com/domodwyer/promr/issues>

**License** Apache License (>= 2)

**Encoding** UTF-8

**Suggests** httpptest, testthat (>= 3.0.0)

**Imports** httr, tibble, urltools

**Config/testthat/edition** 3

**RoxygenNote** 7.2.0

**NeedsCompilation** no

**Author** Dom Dwyer [aut, cre]

**Maintainer** Dom Dwyer <dom@itsallbroken.com>

**Repository** CRAN

**Date/Publication** 2022-08-24 09:10:02 UTC

## R topics documented:

build_url . . . . .	2
cast_timestamp . . . . .	2
query_range . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

build_url	<i>Construct a URL for the specified query.</i>
-----------	---

---

**Description**

Construct a URL for the specified query.

**Usage**

```
build_url(base, query, start, end, step, timeout = NA)
```

**Arguments**

base	A hostname and schema to base the generated path off of.
query	A PromQL query.
start	A RFC3339 timestamp string, numerical unix timestamp, or POSIXct object.
end	A RFC3339 timestamp string, numerical unix timestamp, or POSIXct object.
step	A query resolution step width.
timeout	An optional query timeout value, defaulting to server-side limit. Note this timeout is capped to the server-side value.

**Value**

A URL to execute the query.

---

cast_timestamp	<i>A helper function to map an input of various types to a timestamp string suitable for use with Prometheus.</i>
----------------	---

---

**Description**

A helper function to map an input of various types to a timestamp string suitable for use with Prometheus.

**Usage**

```
cast_timestamp(input)
```

**Arguments**

input	A RFC3339 timestamp string, numerical unix timestamp, or POSIXct object.
-------	--

**Value**

A Prometheus-compatible timestamp that can be coerced to a string.

---

query_range	<i>Evaluate an expression query over a range of time.</i>
-------------	---

---

### Description

Evaluate an expression query over a range of time.

### Usage

```
query_range(
  query,
  start,
  end,
  host = "http://127.0.0.1:9090",
  step = "10s",
  timeout = NA
)
```

### Arguments

query	A PromQL query.
start	A RFC3339 timestamp string, numerical unix timestamp, or POSIXct object.
end	A RFC3339 timestamp string, numerical unix timestamp, or POSIXct object.
host	An optional host - defaulting to <code>http://127.0.0.1:9090</code>
step	An optional query resolution step width, defaulting to <code>10s</code>
timeout	An optional query timeout value, defaulting to server-side limit. Note this timeout is capped to the server-side value.

### Value

A tibble of all series returned by the server, with nested measurements.

### Examples

```
## Not run:
# Run a simple range query against the specified host.
query_range(
  "up",
  "2022-08-20T00:00:00Z",
  "2022-08-21T00:00:00Z",
  host = "http://127.0.0.1:9090"
)

# Run a server-side aggregation query, using the default local host.
query_range(
  "rate(http_requests_total[5m])",
  "2022-08-20T00:00:00Z",
```

```
"2022-08-21T00:00:00Z"  
)  
  
# Specify the time range using POSIXct objects, and set the optional "step"  
query_range(  
  "rate(http_requests_total[5m])",  
  strptime(  
    "2022-08-20T20:10:30",  
    format = "%Y-%m-%dT%H:%M:%S"  
  ),  
  strptime(  
    "2022-08-21T20:10:30",  
    format = "%Y-%m-%dT%H:%M:%S"  
  ),  
  step = "30s"  
)  
  
# Specify the time range using unix timestamps, and set an optional "timeout"  
query_range(  
  "rate(http_requests_total[5m])",  
  1660989814,  
  1661076214,  
  timeout = "60s"  
)  
  
## End(Not run)
```

# Index

`build_url`, 2

`cast_timestamp`, 2

`query_range`, 3