

# Package: druid (via r-universe)

September 1, 2024

**Title** Logger for 'Ambiorix'

**Version** 0.0.0.9000

**Description** A logger for 'Ambiorix' applications to log various events.

**License** GPL (>= 2)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Imports** R6, log, cli

**Suggests** ambiorix

**Repository** <https://devopifex.r-universe.dev>

**RemoteUrl** <https://github.com/devOpifex/druid>

**RemoteRef** HEAD

**RemoteSha** 175b8d87bc1b6b48c713ccd1747ffaa76c9b89fb

## Contents

Druid . . . . .	1
middleware . . . . .	3
<b>Index</b>	<b>4</b>

---

Druid	<i>Druid</i>
-------	--------------

---

## Description

A soft wrapper around [log::Logger](#) to easily log messages for ambiorix applications.

## Super class

[log::Logger](#) -> Druid

## Methods

### Public methods:

- [Druid\\$new\(\)](#)
- [Druid\\$split\(\)](#)
- [Druid\\$log\(\)](#)
- [Druid\\$clone\(\)](#)

### Method `new()`:

*Usage:*

```
Druid$new(  
  prefix = ">",  
  http_accept = FALSE,  
  http_accept_encoding = FALSE,  
  http_accept_language = FALSE,  
  http_cache_control = FALSE,  
  http_connection = FALSE,  
  http_cookie = FALSE,  
  http_host = FALSE,  
  http_sec_fetch_dest = FALSE,  
  http_sec_fetch_mode = FALSE,  
  http_sec_fetch_site = FALSE,  
  http_sec_fetch_user = FALSE,  
  http_upgrade_insecure_requests = FALSE,  
  http_user_agent = FALSE,  
  path_info = FALSE,  
  query_string = FALSE,  
  remote_addr = FALSE,  
  remote_port = FALSE,  
  request_method = FALSE,  
  script_name = FALSE,  
  server_name = FALSE,  
  server_port = FALSE,  
  content_length = FALSE,  
  content_type = FALSE,  
  http_referer = FALSE  
)
```

*Arguments:*

`prefix` Prefix of the log messages.

`http_accept`, `http_accept_encoding`, `http_accept_language`, `http_cache_control`, `http_connection`, `http_cookie`, `http_host`, `http_sec_fetch_dest`, `http_sec_fetch_mode`, `http_sec_fetch_site`, `http_sec_fetch_user`, `http_upgrade_insecure_requests`, `http_user_agent`, `path_info`, `query_string`, `remote_addr`, `remote_port`, `request_method`, `script_name`, `server_name`, `server_port`, `content_length`, `content_type`, `http_referer`  
Whether to log these headers.

*Details:* Constructor

### Method `split()`:

*Usage:*

```
Druid$split()
```

*Details:* Split every item on its own log line

**Method** `log()`:

*Usage:*

`Druid$log(req)`

*Arguments:*

`req` A request, see [ambiorix::Request](#).

*Details:* Log from request

**Method** `clone()`: The objects of this class are cloneable with this method.

*Usage:*

`Druid$clone(deep = FALSE)`

*Arguments:*

`deep` Whether to make a deep clone.

---

middleware

*Middleware*

---

## Description

Druid logger for ambiorix applications.

## Usage

```
druid(...)
```

## Arguments

... Arguments passed to [Druid](#).

# Index

`ambiorix::Request`, 3

`Druid`, 1, 3

`druid (middleware)`, 3

`log::Logger`, 1

`middleware`, 3