

# Package ‘ApacheLogProcessor’

May 6, 2026

**Type** Package

**Title** Process the Apache Web Server Log Files

**Version** 0.2.3

**Date** 2018-07-18

**Author** Diogo Silveira Mendonca

**Maintainer** Diogo Silveira Mendonca <diogosmendonca@gmail.com>

**Description** Provides capabilities to process Apache HTTPD Log files. The main functionalities are to extract data from access and error log files to data frames.

**License** LGPL-3 | file LICENSE

**URL** <https://github.com/diogosmendonca/ApacheLogProcessor>

**BugReports** <https://github.com/diogosmendonca/ApacheLogProcessor/issues>

**Imports** foreach, parallel, doParallel, utils, stringr

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-07-19 07:30:08 UTC

## Contents

access_log_combined . . . . .	2
access_log_common . . . . .	2
clear.urls . . . . .	2
get.url.params . . . . .	4
parse.php.msgs . . . . .	4
read.apache.access.log . . . . .	5
read.apache.error.log . . . . .	7
read.multiple.apache.access.log . . . . .	7
read.multiple.apache.error.log . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

access\_log\_combined     *Apache log combined file example.*

---

**Description**

A set of 12 log lines in Apache Log Combined Format

**Format**

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined

**Source**

<http://www.infinance.com.br/>

---

access\_log\_common     *Apache log common file example.*

---

**Description**

A set of 12 log lines in Apache Log Common Format

**Format**

LogFormat "%h %l %u %t \"%r\" %>s %b\" common

**Source**

<http://www.infinance.com.br/>

---

clear.urls     *Clear a list of URLs according parameters.*

---

**Description**

Clear a list of URLs according parameters.

**Usage**

```
clear.urls(urls, remove_http_method = TRUE, remove_http_version = TRUE,  
           remove_params_inside_url = TRUE, remove_query_string = TRUE)
```

**Arguments**

`urls` list of URLs

`remove_http_method`  
boolean. If the http method will be removed from the urls.

`remove_http_version`  
boolean. If the http version will be removed from the urls.

`remove_params_inside_url`  
boolean. If the parameters inside the URL, commonly used in REST web services, will be removed from the urls.

`remove_query_string`  
boolean. If the query string will be removed from the urls.

**Value**

a vector with the urls cleaned

**Author(s)**

Diogo Silveira Mendonca

**Examples**

```
#Load the path to the log file
path_combined = system.file("examples", "access_log_combined.txt", package = "ApacheLogProcessor")

#Read a log file with combined format and return it in a data frame
df1 = read.apache.access.log(path_combined)

#Clear the urls
urls <- clear.urls(df1$url)

#Clear the urls but do not remove query strings
urlsWithQS <- clear.urls(df1$url, remove_query_string = FALSE)

#Load a log which the urls have parameters inside
path2 = system.file("examples",
"access_log_with_params_inside_url.txt", package = "ApacheLogProcessor")

#Read a log file with combined format and return it in a data frame
df2 = read.apache.access.log(path2, format = "common")

#Clear the urls with parameters inside
urls2 <- clear.urls(df2$url)
```

`get.url.params`      *Extract from the data frame with the access log the urls query strings parameters and values.*

---

### **Description**

The function supports multivalued parameters, but does not support parameters inside urls yet.

### **Usage**

```
get.url.params(dfLog)
```

### **Arguments**

`dfLog`      a dataframe with the access log. Can be load with `read.apache.access.log` or `read.multiple.apache.access.log`.

### **Value**

a structure of data frames with query strings parameters for each url of the log

### **Author(s)**

Diogo Silveira Mendonca

### **Examples**

```
#Load a log which the urls have query strings
path = system.file("examples", "access_log_with_query_string.log", package = "ApacheLogProcessor")

#Read a log file with combined format and return it in a data frame
df = read.apache.access.log(path, format = "common")

#Clear the urls with parameters inside
params <- get.url.params(df)
```

---

`parse.php.msgs`      *Parses PHP messages and store its parts in a data frame that contains level, message, file, line number and referer.*

---

### **Description**

Parses PHP messages and store its parts in a data frame that contains level, message, file, line number and referer.

**Usage**

```
parse.php.msgs(dfErrorLog)
```

**Arguments**

dfErrorLog      Error log load with the read.apache.error.log or read.multiple.apache.error.log functions.

**Value**

a data frame with PHP error message split in parts.

**Examples**

```
#Loads the path of the erro log
path <- system.file("examples", "error_log.log", package = "ApacheLogProcessor")

#Loads the error log to a data frame
dfELog <- read.apache.error.log(path)

dfPHPMsgs <- parse.php.msgs(dfELog)
```

---

```
read.apache.access.log
      read.apache.log
```

---

**Description**

Reads the Apache Log Common or Combined Format and return a data frame with the log data.

**Usage**

```
read.apache.access.log(file, format = "combined", url_includes = "",
  url_excludes = "", columns = c("ip", "datetime", "url", "httpcode",
  "size", "referer", "useragent"), num_cores = 1, fields_have_quotes = TRUE)
```

**Arguments**

file            string. Full path to the log file.

format         string. Values "common" or "combined" to set the input log format. The default value is the combined.

url\_includes   regex. If passed only the urls that matches with the regular expression passed will be returned.

url\_excludes   regex. If passed only the urls that don't matches with the regular expression passed will be returned.

columns	list. List of columns names that will be included in data frame output. All columns is the default value. c("ip", "datetime", "url", "httpcode", "size", "referrer", "useragent")
num_cores	number. Number of cores for parallel execution, if not passed 1 core is assumed. Used only to convert datetime form string to datetime type.
fields_have_quotes	boolean. If passed as true search and remove the quotes inside the all text fields.

### Details

The functions receives a full path to the log file and process the default log in common or combined format of Apache. LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined LogFormat "%h %l %u %t \"%r\" %>s %b\" common

### Value

a data frame with the apache log file information.

### Author(s)

Diogo Silveira Mendonca

### See Also

<http://httpd.apache.org/docs/1.3/logs.html>

### Examples

```
path_combined = system.file("examples", "access_log_combined.txt", package = "ApacheLogProcessor")
path_common = system.file("examples", "access_log_common.txt", package = "ApacheLogProcessor")

#Read a log file with combined format and return it in a data frame
df1 = read.apache.access.log(path_combined)

#Read a log file with common format and return it in a data frame
df2 = read.apache.access.log(path_common, format="common")

#Read only the lines that url matches with the pattern passed
df3 = read.apache.access.log(path_combined, url_includes="infinance")

#Read only the lines that url matches with the pattern passed, but do not matche the exclude pattern
df4 = read.apache.access.log(path_combined,
url_includes="infinance", url_excludes="infinanceclient")

#Return only the ip, url and datetime columns
df5 = read.apache.access.log(path_combined, columns=c("ip", "url", "datetime"))

#Process using 2 cores in parallel for speed up.
df6 = read.apache.access.log(path_combined, num_cores=2)
```

---

read.apache.error.log *Read the apache error log file and loads it to a data frame.*

---

**Description**

Read the apache error log file and loads it to a data frame.

**Usage**

```
read.apache.error.log(file, columns = c("datetime", "logLevel", "pid",  
  "ip_port", "msg"))
```

**Arguments**

file	path to the error log file
columns	which columns should be loaded. Default value is all columns. c("datetime", "logLevel", "pid", "ip_port", "msg")

**Value**

a data frame with the error log data

**Author(s)**

Diogo Silveira Mendonca

**Examples**

```
#Loads the path of the error log  
path <- system.file("examples", "error_log.log", package = "ApacheLogProcessor")  
  
#Loads the error log to a data frame  
dfELog <- read.apache.error.log(path)
```

---

read.multiple.apache.access.log  
*Reads multiple files of apache web server.*

---

**Description**

The files can be gzipped or not. If the files are gzipped they are extracted once at time, processed and after only the extracted file is deleted.

**Usage**

```
read.multiple.apache.access.log(path, prefix, verbose = TRUE, ...)
```

**Arguments**

path	path where the files are located
prefix	the prefix that identify the logs files
verbose	if prints messages during the processing
...	parameter to be passed to read.apache.access.log function

**Value**

a data frame with the apache log files information.

**Author(s)**

Diogo Silveira Mendonca

**Examples**

```
path <- system.file("examples", package="ApacheLogProcessor")
path <- paste(path, "/", sep="")

#read multiple gzipped logs with the prefix m_access_log_combined_
dfLog <- read.multiple.apache.access.log(path, "m_access_log_combined_")
```

---

read.multiple.apache.error.log

*Reads multiple apache error log files and loads them to a data frame.*

---

**Description**

Reads multiple apache error log files and loads them to a data frame.

**Usage**

```
read.multiple.apache.error.log(path, prefix, verbose = TRUE, ...)
```

**Arguments**

path	path to the folder that contains the error log files
prefix	prefix for all error log files that will be loaded
verbose	if the function prints messages during the logs processing
...	parameters to be passed to read.apache.error.log function

**Value**

a data frame with the error log data

**Examples**

```
path <- system.file("examples", package="ApacheLogProcessor")
path <- paste(path, "/", sep="")

#read multiple gzipped logs with the prefix m_access_log_combined_
dfELog <- read.multiple.apache.error.log(path, "m_error_log_")
```

# Index

`access_log_combined`, [2](#)

`access_log_common`, [2](#)

`clear.urls`, [2](#)

`get.url.params`, [4](#)

`parse.php.msgs`, [4](#)

`read.apache.access.log`, [5](#)

`read.apache.error.log`, [7](#)

`read.multiple.apache.access.log`, [7](#)

`read.multiple.apache.error.log`, [8](#)