
log4Esp Documentation

Release 1.0.0

log4Esp

Feb 18, 2019

Contents

1	Classes
----------	----------------

3

log4Esp is a light-wight logging library for ESP8266.

namespace log4Esp

Variables

Logger **LOG** = *Logger*()

class Appender

#include <Appender.h> Appenders represent a logging destination.

Subclassed by *log4Esp::RollingFileAppender*, *log4Esp::SerialAppender*

class RollingFileAppender : public log4Esp::Appender

#include <RollingFileAppender.h> *RollingFileAppender* logs data to a rolling file.

class SerialAppender : public log4Esp::Appender

#include <SerialAppender.h> *SerialAppender* logs data to Serial.

class `Logger`

Public Functions

Logger (**const** `char` **name* = "default", `bool` *addDefaultSerialAppender* = true)

Logger represents the central interface to all log operations.

Parameters

- *name*: labels internal *Logger* name
- *addDefaultSerialAppender*: adds a *SerialAppender* optional

template <**class** `T`, `typename`... *Args*>

`void fatal` (`T` *msg*, `Args`... *args*)

Print a log entry in [fatal] mode to all associated Appenders.

Parameters

- *msg*: log entry with optional format instructions and placeholders
- *args*: arguments replacing placeholders

template <**class** `T`, `typename`... *Args*>

`void error` (`T` *msg*, `Args`... *args*)

Print a log entry in [error] mode to all associated Appenders.

Parameters

- *msg*: log entry with optional format instructions and placeholders
- *args*: arguments replacing placeholders

template <**class** `T`, `typename`... *Args*>

void **warning** (T *msg*, Args... *args*)
Print a log entry in [warning] mode to all associated Appenders.

Parameters

- *msg*: log entry with optional format instructions and placeholders
- *args*: arguments replacing placeholders

template <class T, typename... Args>
void **verbose** (T *msg*, Args... *args*)
Print a log entry in [verbose] mode to all associated Appenders.

Parameters

- *msg*: log entry with optional format instructions and placeholders
- *args*: arguments replacing placeholders

template <class T, typename... Args>
void **trace** (T *msg*, Args... *args*)
Print a log entry in [trace] mode to all associated Appenders.

Parameters

- *msg*: log entry with optional format instructions and placeholders
- *args*: arguments replacing placeholders

const char ***getName** ()

Return internal *Logger* name

std::vector<*Appender* *> &**getAppender** ()

Return list of associated Appenders

void **addFormatterToAll** (*Appender*::FormatterFunction *formatterFunction*)

Parameters

- *add*: same FormatterFunction to all associated Appenders

void **addFilterToAll** (*Appender*::FilterFunction *filterFunction*)

Parameters

- *add*: same Filter to all associated Appenders

void **addLevelToAll** (*Appender*::Level *level*)

Parameters

- *add*: same Level to all associated Appenders

template <class T>
void **print** (*Appender*::Level *level*, T *msg*, ...)
Print a log entry to all associated Appenders.

Parameters

- *level*: represents logging level of entry

- `msg`: log entry with optional format instructions and placeholders
- `args`: arguments replacing placeholders

class Appender

Appenders represent a logging destination.

Subclassed by *log4Esp::RollingFileAppender*, *log4Esp::SerialAppender*

Public Functions

virtual Print &**getOutput** () = 0

Return output reference

*Appender::*FormatterFunction **getFormatter** ()

Return associated Formatter

void **setFormatter** (FormatterFunction *formatterFunction*)

Parameters

- `set`: Formatter

std::vector<*Appender::*FilterFunction> &**getFilter** ()

Return all associated Filters

void **addFilter** (FilterFunction *filterFunction*)

Parameters

- `add`: a Filter

void **setLevel** (Level *level*)

Parameters

- `set`: a certain logging Level

void **print** (Level *level*, const char **msg*, va_list **args*)

Print a log entry.

Parameters

- `level`: represents logging level of entry
- `msg`: log entry with optional format instructions and placeholders
- `args`: arguments replacing placeholders

Public Static Functions

*Appender::*FormatterFunction **getDefaultFormatter** ()

Return default Formatter instance

const __FlashStringHelper ***toString** (Level *level*, bool *shortName* = true)

Return Level as String representation

L

- log4Esp (C++ type), 1
- log4Esp::Appender (C++ class), 1, 5
- log4Esp::Appender::addFilter (C++ function), 5
- log4Esp::Appender::getDefaultFormatter (C++ function), 5
- log4Esp::Appender::getFilter (C++ function), 5
- log4Esp::Appender::getFormatter (C++ function), 5
- log4Esp::Appender::getOutput (C++ function), 5
- log4Esp::Appender::print (C++ function), 5
- log4Esp::Appender::setFormatter (C++ function), 5
- log4Esp::Appender::setLevel (C++ function), 5
- log4Esp::Appender::toString (C++ function), 5
- log4Esp::LOG (C++ member), 1
- log4Esp::Logger (C++ class), 3
- log4Esp::Logger::addFilterToAll (C++ function), 4
- log4Esp::Logger::addFormatterToAll (C++ function), 4
- log4Esp::Logger::addLevelToAll (C++ function), 4
- log4Esp::Logger::error (C++ function), 3
- log4Esp::Logger::fatal (C++ function), 3
- log4Esp::Logger::getAppender (C++ function), 4
- log4Esp::Logger::getName (C++ function), 4
- log4Esp::Logger::Logger (C++ function), 3
- log4Esp::Logger::print (C++ function), 4
- log4Esp::Logger::trace (C++ function), 4
- log4Esp::Logger::verbose (C++ function), 4
- log4Esp::Logger::warning (C++ function), 3
- log4Esp::RollingFileAppender (C++ class), 1
- log4Esp::SerialAppender (C++ class), 1