### 3.8.1 (2024-12-05)

* Deprecated Monolog\DateTimeImmutable in favor of Monolog\JsonSerializableDateTimeImmutable (#1928)
* Fixed gelf keys not being valid when context/extra data keys have spaces in them (#1927)
* Fixed empty lines appearing in the stack traces when a custom formatter returned null (#1925)

### 3.8.0 (2024-11-12)

* Added $fileOpenMode param to StreamHandler to define a custom fopen mode to open the log file (#1913)
* Fixed PHP 8.4 deprecation notices (#1903)
* Added ability to extend/override IntrospectionProcessor (#1899)
* Added $timeout param to ProcessHandler to configure the stream\_select() timeout to avoid blocking too long (default is 1.0 sec) (#1916)
* Fixed JsonFormatter batch handling to normalize records individually to make sure they look the same as if they were handled one by one (#1906)
* Fixed StreamHandler handling of write failures so that it now closes/reopens the stream and retries the write once before failing (#1882)
* Fixed StreamHandler error handler causing issues if a stream handler triggers an error (#1866)
* Fixed StreamHandler::reset not closing the stream, so that it would fail to write in some cases with long running processes (#1862)
* Fixed RotatingFileHandler issue where rotation does not happen in some long running processes (#1905)
* Fixed JsonFormatter handling of incomplete classes (#1834)
* Fixed RotatingFileHandler bug where rotation could sometimes not happen correctly (#1905)

### 3.7.0 (2024-06-28)

* Added NormalizerFormatter->setBasePath(...) (and JsonFormatter by extension) that allows removing the project's path from the stack trace output (47e301d3e)
* Fixed JsonFormatter handling of incomplete classes (#1834)
* Fixed private error handlers causing problems with custom StreamHandler implementations (#1866)

### 3.6.0 (2024-04-12)

* Added LineFormatter->setBasePath(...) that allows removing the project's path from the stack trace output (#1873)
* Added $includeExtra option in PsrHandler to also use extra data to replace placeholder values in the message (#1852)
* Added ability to customize what is a duplicated message by extending the DeduplicationHandler (#1879)
* Added handling for using GelfMessageFormatter together with the AmqpHandler (#1869)
* Added ability to extend GoogleCloudLoggingFormatter (#1859)
* Fixed \_\_toString failures in context data crashing the normalization process (#1868)
* Fixed PHP 8.4 deprecation warnings (#1874)

### 3.5.0 (2023-10-27)

* Added ability to indent stack traces in LineFormatter via e.g. indentStacktraces(' ') (#1835)
* Added ability to configure a max level name length in LineFormatter via e.g. setMaxLevelNameLength(3) (#1850)
* Added support for indexed arrays (i.e. [] and not {} arrays once json serialized) containing inline linebreaks in LineFormatter (#1818)
* Added WithMonologChannel attribute for integrators to use to configure autowiring (#1847)
* Fixed log record extra data leaking between handlers that have handler-specific processors set (#1819)
* Fixed LogglyHandler issue with record level filtering (#1841)
* Fixed display\_errors parsing in ErrorHandler which did not support string values (#1804)
* Fixed bug where the previous error handler would not be restored in some cases where StreamHandler fails (#1815)
* Fixed normalization error when normalizing incomplete classes (#1833)

### 3.4.0 (2023-06-21)

* Added LoadAverageProcessor to track one of the 1, 5 or 15min load averages (#1803)
* Added support for priority to the AsMonologProcessor attribute (#1797)
* Added TelegramBotHandler topic/message\_thread\_id support (#1802)
* Fixed FingersCrossedHandler passthruLevel checking (#1801)
* Fixed support of yearly and monthly rotation log file to rotate only once a month/year (#1805)
* Fixed TestHandler method docs (#1794)
* Fixed handling of falsey display\_errors string values (#1804)

### 3.3.1 (2023-02-06)

* Fixed Logger not being serializable anymore (#1792)

### 3.3.0 (2023-02-06)

* Deprecated FlowdockHandler & Formatter as the flowdock service was shutdown (#1748)
* Added ClosureContextProcessor to allow delaying the creation of context data by setting a Closure in context which is called when the log record is used (#1745)
* Added an ElasticsearchHandler option to set the op\_type to create instead of the default index (#1766)
* Added support for enum context values in PsrLogMessageProcessor (#1773)
* Added graylog2/gelf-php 2.x support (#1747)
* Improved BrowserConsoleHandler logging to use more appropriate methods than just console.log in the browser (#1739)
* Fixed GitProcessor not filtering correctly based on Level (#1749)
* Fixed WhatFailureGroupHandler not catching errors happening inside close() (#1791)
* Fixed datetime field in GoogleCloudLoggingFormatter (#1758)
* Fixed infinite loop detection within Fibers (#1753)
* Fixed AmqpHandler->setExtraAttributes not working with buffering handler wrappers (#1781)

### 3.2.0 (2022-07-24)

* Deprecated CubeHandler and PHPConsoleHandler as both projects are abandoned and those should not be used anymore (#1734)
* Marked Logger @final as it should not be extended, prefer composition or talk to us if you are missing something
* Added RFC 5424 level (7 to 0) support to Logger::log and Logger::addRecord to increase interoperability (#1723)
* Added SyslogFormatter to output syslog-like files which can be consumed by tools like [lnav](https://lnav.org/) (#1689)
* Added support for \_\_toString for objects which are not json serializable in JsonFormatter (#1733)
* Added GoogleCloudLoggingFormatter (#1719)
* Added support for Predis 2.x (#1732)
* Added AmqpHandler->setExtraAttributes to allow configuring attributes when using an AMQPExchange (#1724)
* Fixed serialization/unserialization of handlers to make sure private properties are included (#1727)
* Fixed allowInlineLineBreaks in LineFormatter causing issues with windows paths containing \n or \r sequences (#1720)
* Fixed max normalization depth not being taken into account when formatting exceptions with a deep chain of previous exceptions (#1726)
* Fixed PHP 8.2 deprecation warnings (#1722)
* Fixed rare race condition or filesystem issue where StreamHandler is unable to create the directory the log should go into yet it exists already (#1678)

### 3.1.0 (2022-06-09)

* Added $datetime parameter to Logger::addRecord as low level API to allow logging into the past or future (#1682)
* Added Logger::useLoggingLoopDetection to allow disabling cyclic logging detection in concurrent frameworks (#1681)
* Fixed handling of fatal errors if callPrevious is disabled in ErrorHandler (#1670)
* Fixed interop issue by removing the need for a return type in ProcessorInterface (#1680)
* Marked the reusable Monolog\Test\TestCase class as @internal to make sure PHPStorm does not show it above PHPUnit, you may still use it to test your own handlers/etc though (#1677)
* Fixed RotatingFileHandler issue when the date format contained slashes (#1671)

### 3.0.0 (2022-05-10)

Changes from RC1

* The Monolog\LevelName enum does not exist anymore, use Monolog\Level->getName() instead.

### 3.0.0-RC1 (2022-05-08)

This is mostly a cleanup release offering stronger type guarantees for integrators with the array->object/enum changes, but there is no big new feature for end users.

See [UPGRADE notes](http://upgrade.md#300) for details on all breaking changes especially if you are extending/implementing Monolog classes/interfaces.

Noteworthy BC Breaks:

* The minimum supported PHP version is now 8.1.0.
* Log records have been converted from an array to a [Monolog\LogRecord object](http://src/Monolog/LogRecord.php) with public (and mostly readonly) properties. e.g. instead of doing $record['context'] use $record->context. In formatters or handlers if you rather need an array to work with you can use $record->toArray() to get back a Monolog 1/2 style record array. This will contain the enum values instead of enum cases in the level and level\_name keys to be more backwards compatible and use simpler data types.
* FormatterInterface, HandlerInterface, ProcessorInterface, etc. changed to contain LogRecord $record instead of array $record parameter types. If you want to support multiple Monolog versions this should be possible by type-hinting nothing, or array|LogRecord if you support PHP 8.0+. You can then code against the $record using Monolog 2 style as LogRecord implements ArrayAccess for BC. The interfaces do not require a LogRecord return type even where it would be applicable, but if you only support Monolog 3 in integration code I would recommend you use LogRecord return types wherever fitting to ensure forward compatibility as it may be added in Monolog 4.
* Log levels are now enums [Monolog\Level](http://src/Monolog/Level.php) and [Monolog\LevelName](http://src/Monolog/LevelName.php)
* Removed deprecated SwiftMailerHandler, migrate to SymfonyMailerHandler instead.
* ResettableInterface::reset() now requires a void return type.
* All properties have had types added, which may require you to do so as well if you extended a Monolog class and declared the same property.

New deprecations:

* Logger::DEBUG, Logger::ERROR, etc. are now deprecated in favor of the Monolog\Level enum. e.g. instead of Logger::WARNING use Level::Warning if you need to pass the enum case to Monolog or one of its handlers, or Level::Warning->value if you need the integer value equal to what Logger::WARNING was giving you.
* Logger::getLevelName() is now deprecated.

### 2.10.0 (2024-11-12)

* Added $fileOpenMode to StreamHandler to define a custom fopen mode to open the log file (#1913)
* Fixed StreamHandler handling of write failures so that it now closes/reopens the stream and retries the write once before failing (#1882)
* Fixed StreamHandler error handler causing issues if a stream handler triggers an error (#1866)
* Fixed JsonFormatter handling of incomplete classes (#1834)
* Fixed RotatingFileHandler bug where rotation could sometimes not happen correctly (#1905)

### 2.9.3 (2024-04-12)

* Fixed PHP 8.4 deprecation warnings (#1874)

### 2.9.2 (2023-10-27)

* Fixed display\_errors parsing in ErrorHandler which did not support string values (#1804)
* Fixed bug where the previous error handler would not be restored in some cases where StreamHandler fails (#1815)
* Fixed normalization error when normalizing incomplete classes (#1833)

### 2.9.1 (2023-02-06)

* Fixed Logger not being serializable anymore (#1792)

### 2.9.0 (2023-02-05)

* Deprecated FlowdockHandler & Formatter as the flowdock service was shutdown (#1748)
* Added support for enum context values in PsrLogMessageProcessor (#1773)
* Added graylog2/gelf-php 2.x support (#1747)
* Improved BrowserConsoleHandler logging to use more appropriate methods than just console.log in the browser (#1739)
* Fixed WhatFailureGroupHandler not catching errors happening inside close() (#1791)
* Fixed datetime field in GoogleCloudLoggingFormatter (#1758)
* Fixed infinite loop detection within Fibers (#1753)
* Fixed AmqpHandler->setExtraAttributes not working with buffering handler wrappers (#1781)

### 2.8.0 (2022-07-24)

* Deprecated CubeHandler and PHPConsoleHandler as both projects are abandoned and those should not be used anymore (#1734)
* Added RFC 5424 level (7 to 0) support to Logger::log and Logger::addRecord to increase interoperability (#1723)
* Added support for \_\_toString for objects which are not json serializable in JsonFormatter (#1733)
* Added GoogleCloudLoggingFormatter (#1719)
* Added support for Predis 2.x (#1732)
* Added AmqpHandler->setExtraAttributes to allow configuring attributes when using an AMQPExchange (#1724)
* Fixed serialization/unserialization of handlers to make sure private properties are included (#1727)
* Fixed allowInlineLineBreaks in LineFormatter causing issues with windows paths containing \n or \r sequences (#1720)
* Fixed max normalization depth not being taken into account when formatting exceptions with a deep chain of previous exceptions (#1726)
* Fixed PHP 8.2 deprecation warnings (#1722)
* Fixed rare race condition or filesystem issue where StreamHandler is unable to create the directory the log should go into yet it exists already (#1678)

### 2.7.0 (2022-06-09)

* Added $datetime parameter to Logger::addRecord as low level API to allow logging into the past or future (#1682)
* Added Logger::useLoggingLoopDetection to allow disabling cyclic logging detection in concurrent frameworks (#1681)
* Fixed handling of fatal errors if callPrevious is disabled in ErrorHandler (#1670)
* Marked the reusable Monolog\Test\TestCase class as @internal to make sure PHPStorm does not show it above PHPUnit, you may still use it to test your own handlers/etc though (#1677)
* Fixed RotatingFileHandler issue when the date format contained slashes (#1671)

### 2.6.0 (2022-05-10)

* Deprecated SwiftMailerHandler, use SymfonyMailerHandler instead
* Added SymfonyMailerHandler (#1663)
* Added ElasticSearch 8.x support to the ElasticsearchHandler (#1662)
* Added a way to filter/modify stack traces in LineFormatter (#1665)
* Fixed UdpSocket not being able to reopen/reconnect after close()
* Fixed infinite loops if a Handler is triggering logging while handling log records

### 2.5.0 (2022-04-08)

* Added callType to IntrospectionProcessor (#1612)
* Fixed AsMonologProcessor syntax to be compatible with PHP 7.2 (#1651)

### 2.4.0 (2022-03-14)

* Added [Monolog\LogRecord](http://src/Monolog/LogRecord.php) interface that can be used to type-hint records like array|\Monolog\LogRecord $record to be forward compatible with the upcoming Monolog 3 changes
* Added includeStacktraces constructor params to LineFormatter & JsonFormatter (#1603)
* Added persistent, timeout, writingTimeout, connectionTimeout, chunkSize constructor params to SocketHandler and derivatives (#1600)
* Added AsMonologProcessor PHP attribute which can help autowiring / autoconfiguration of processors if frameworks / integrations decide to make use of it. This is useless when used purely with Monolog (#1637)
* Added support for keeping native BSON types as is in MongoDBFormatter (#1620)
* Added support for a user\_agent key in WebProcessor, disabled by default but you can use it by configuring the $extraFields you want (#1613)
* Added support for username/userIcon in SlackWebhookHandler (#1617)
* Added extension points to BrowserConsoleHandler (#1593)
* Added record message/context/extra info to exceptions thrown when a StreamHandler cannot open its stream to avoid completely losing the data logged (#1630)
* Fixed error handler signature to accept a null $context which happens with internal PHP errors (#1614)
* Fixed a few setter methods not returning self (#1609)
* Fixed handling of records going over the max Telegram message length (#1616)

### 2.3.5 (2021-10-01)

* Fixed regression in StreamHandler since 2.3.3 on systems with the memory\_limit set to >=20GB (#1592)

### 2.3.4 (2021-09-15)

* Fixed support for psr/log 3.x (#1589)

### 2.3.3 (2021-09-14)

* Fixed memory usage when using StreamHandler and calling stream\_get\_contents on the resource you passed to it (#1578, #1577)
* Fixed support for psr/log 2.x (#1587)
* Fixed some type annotations

### 2.3.2 (2021-07-23)

* Fixed compatibility with PHP 7.2 - 7.4 when experiencing PCRE errors (#1568)

### 2.3.1 (2021-07-14)

* Fixed Utils::getClass handling of anonymous classes not being fully compatible with PHP 8 (#1563)
* Fixed some @inheritDoc annotations having the wrong case

### 2.3.0 (2021-07-05)

* Added a ton of PHPStan type annotations as well as type aliases on Monolog\Logger for Record, Level and LevelName that you can import (#1557)
* Added ability to customize date format when using JsonFormatter (#1561)
* Fixed FilterHandler not calling reset on its internal handler when reset() is called on it (#1531)
* Fixed SyslogUdpHandler not setting the timezone correctly on DateTimeImmutable instances (#1540)
* Fixed StreamHandler thread safety - chunk size set to 2GB now to avoid interlacing when doing concurrent writes (#1553)

### 2.2.0 (2020-12-14)

* Added JSON\_PARTIAL\_OUTPUT\_ON\_ERROR to default json encoding flags, to avoid dropping entire context data or even records due to an invalid subset of it somewhere
* Added setDateFormat to NormalizerFormatter (and Line/Json formatters by extension) to allow changing this after object creation
* Added RedisPubSubHandler to log records to a Redis channel using PUBLISH
* Added support for Elastica 7, and deprecated the $type argument of ElasticaFormatter which is not in use anymore as of Elastica 7
* Added support for millisecond write timeouts in SocketHandler, you can now pass floats to setWritingTimeout, e.g. 0.2 is 200ms
* Added support for unix sockets in SyslogUdpHandler (set $port to 0 to make the $host a unix socket)
* Added handleBatch support for TelegramBotHandler
* Added RFC5424e extended date format including milliseconds to SyslogUdpHandler
* Added support for configuring handlers with numeric level values in strings (coming from e.g. env vars)
* Fixed Wildfire/FirePHP/ChromePHP handling of unicode characters
* Fixed PHP 8 issues in SyslogUdpHandler
* Fixed internal type error when mbstring is missing

### 2.1.1 (2020-07-23)

* Fixed removing of json encoding options
* Fixed type hint of $level not accepting strings in SendGridHandler and OverflowHandler
* Fixed SwiftMailerHandler not accepting email templates with an empty subject
* Fixed array access on null in RavenHandler
* Fixed unique\_id in WebProcessor not being disableable

### 2.1.0 (2020-05-22)

* Added JSON\_INVALID\_UTF8\_SUBSTITUTE to default json flags, so that invalid UTF8 characters now get converted to [�](https://en.wikipedia.org/wiki/Specials_(Unicode_block)#Replacement_character) instead of being converted from ISO-8859-15 to UTF8 as it was before, which was hardly a comprehensive solution
* Added $ignoreEmptyContextAndExtra option to JsonFormatter to skip empty context/extra entirely from the output
* Added $parseMode, $disableWebPagePreview and $disableNotification options to TelegramBotHandler
* Added tentative support for PHP 8
* NormalizerFormatter::addJsonEncodeOption and removeJsonEncodeOption are now public to allow modifying default json flags
* Fixed GitProcessor type error when there is no git repo present
* Fixed normalization of SoapFault objects containing deeply nested objects as "detail"
* Fixed support for relative paths in RotatingFileHandler

### 2.0.2 (2019-12-20)

* Fixed ElasticsearchHandler swallowing exceptions details when failing to index log records
* Fixed normalization of SoapFault objects containing non-strings as "detail" in LineFormatter
* Fixed formatting of resources in JsonFormatter
* Fixed RedisHandler failing to use MULTI properly when passed a proxied Redis instance (e.g. in Symfony with lazy services)
* Fixed FilterHandler triggering a notice when handleBatch was filtering all records passed to it
* Fixed Turkish locale messing up the conversion of level names to their constant values

### 2.0.1 (2019-11-13)

* Fixed normalization of Traversables to avoid traversing them as not all of them are rewindable
* Fixed setFormatter/getFormatter to forward to the nested handler in FilterHandler, FingersCrossedHandler, BufferHandler, OverflowHandler and SamplingHandler
* Fixed BrowserConsoleHandler formatting when using multiple styles
* Fixed normalization of exception codes to be always integers even for PDOException which have them as numeric strings
* Fixed normalization of SoapFault objects containing non-strings as "detail"
* Fixed json encoding across all handlers to always attempt recovery of non-UTF-8 strings instead of failing the whole encoding
* Fixed ChromePHPHandler to avoid sending more data than latest Chrome versions allow in headers (4KB down from 256KB).
* Fixed type error in BrowserConsoleHandler when the context array of log records was not associative.

### 2.0.0 (2019-08-30)

* BC Break: This is a major release, see [UPGRADE.md](http://upgrade.md) for details if you are coming from a 1.x release
* BC Break: Logger methods log/debug/info/notice/warning/error/critical/alert/emergency now have explicit void return types
* Added FallbackGroupHandler which works like the WhatFailureGroupHandler but stops dispatching log records as soon as one handler accepted it
* Fixed support for UTF-8 when cutting strings to avoid cutting a multibyte-character in half
* Fixed normalizers handling of exception backtraces to avoid serializing arguments in some cases
* Fixed date timezone handling in SyslogUdpHandler

### 2.0.0-beta2 (2019-07-06)

* BC Break: This is a major release, see [UPGRADE.md](http://upgrade.md) for details if you are coming from a 1.x release
* BC Break: PHP 7.2 is now the minimum required PHP version.
* BC Break: Removed SlackbotHandler, RavenHandler and HipChatHandler, see [UPGRADE.md](http://upgrade.md) for details
* Added OverflowHandler which will only flush log records to its nested handler when reaching a certain amount of logs (i.e. only pass through when things go really bad)
* Added TelegramBotHandler to log records to a [Telegram](https://core.telegram.org/bots/api) bot account
* Added support for JsonSerializable when normalizing exceptions
* Added support for RFC3164 (outdated BSD syslog protocol) to SyslogUdpHandler
* Added SoapFault details to formatted exceptions
* Fixed DeduplicationHandler silently failing to start when file could not be opened
* Fixed issue in GroupHandler and WhatFailureGroupHandler where setting multiple processors would duplicate records
* Fixed GelfFormatter losing some data when one attachment was too long
* Fixed issue in SignalHandler restarting syscalls functionality
* Improved performance of LogglyHandler when sending multiple logs in a single request

### 2.0.0-beta1 (2018-12-08)

* BC Break: This is a major release, see [UPGRADE.md](http://upgrade.md) for details if you are coming from a 1.x release
* BC Break: PHP 7.1 is now the minimum required PHP version.
* BC Break: Quite a few interface changes, only relevant if you implemented your own handlers/processors/formatters
* BC Break: Removed non-PSR-3 methods to add records, all the add\* (e.g. addWarning) methods as well as emerg, crit, err and warn
* BC Break: The record timezone is now set per Logger instance and not statically anymore
* BC Break: There is no more default handler configured on empty Logger instances
* BC Break: ElasticSearchHandler renamed to ElasticaHandler
* BC Break: Various handler-specific breaks, see [UPGRADE.md](http://upgrade.md) for details
* Added scalar type hints and return hints in all the places it was possible. Switched strict\_types on for more reliability.
* Added DateTimeImmutable support, all record datetime are now immutable, and will toString/json serialize with the correct date format, including microseconds (unless disabled)
* Added timezone and microseconds to the default date format
* Added SendGridHandler to use the SendGrid API to send emails
* Added LogmaticHandler to use the Logmatic.io API to store log records
* Added SqsHandler to send log records to an AWS SQS queue
* Added ElasticsearchHandler to send records via the official ES library. Elastica users should now use ElasticaHandler instead of ElasticSearchHandler
* Added NoopHandler which is similar to the NullHandle but does not prevent the bubbling of log records to handlers further down the configuration, useful for temporarily disabling a handler in configuration files
* Added ProcessHandler to write log output to the STDIN of a given process
* Added HostnameProcessor that adds the machine's hostname to log records
* Added a $dateFormat option to the PsrLogMessageProcessor which lets you format DateTime instances nicely
* Added support for the PHP 7.x mongodb extension in the MongoDBHandler
* Fixed many minor issues in various handlers, and probably added a few regressions too

### 1.26.1 (2021-05-28)

* Fixed PHP 8.1 deprecation warning

### 1.26.0 (2020-12-14)

* Added $dateFormat and $removeUsedContextFields arguments to PsrLogMessageProcessor (backport from 2.x)

### 1.25.5 (2020-07-23)

* Fixed array access on null in RavenHandler
* Fixed unique\_id in WebProcessor not being disableable

### 1.25.4 (2020-05-22)

* Fixed GitProcessor type error when there is no git repo present
* Fixed normalization of SoapFault objects containing deeply nested objects as "detail"
* Fixed support for relative paths in RotatingFileHandler

### 1.25.3 (2019-12-20)

* Fixed formatting of resources in JsonFormatter
* Fixed RedisHandler failing to use MULTI properly when passed a proxied Redis instance (e.g. in Symfony with lazy services)
* Fixed FilterHandler triggering a notice when handleBatch was filtering all records passed to it
* Fixed Turkish locale messing up the conversion of level names to their constant values

### 1.25.2 (2019-11-13)

* Fixed normalization of Traversables to avoid traversing them as not all of them are rewindable
* Fixed setFormatter/getFormatter to forward to the nested handler in FilterHandler, FingersCrossedHandler, BufferHandler and SamplingHandler
* Fixed BrowserConsoleHandler formatting when using multiple styles
* Fixed normalization of exception codes to be always integers even for PDOException which have them as numeric strings
* Fixed normalization of SoapFault objects containing non-strings as "detail"
* Fixed json encoding across all handlers to always attempt recovery of non-UTF-8 strings instead of failing the whole encoding

### 1.25.1 (2019-09-06)

* Fixed forward-compatible interfaces to be compatible with Monolog 1.x too.

### 1.25.0 (2019-09-06)

* Deprecated SlackbotHandler, use SlackWebhookHandler or SlackHandler instead
* Deprecated RavenHandler, use sentry/sentry 2.x and their Sentry\Monolog\Handler instead
* Deprecated HipChatHandler, migrate to Slack and use SlackWebhookHandler or SlackHandler instead
* Added forward-compatible interfaces and traits FormattableHandlerInterface, FormattableHandlerTrait, ProcessableHandlerInterface, ProcessableHandlerTrait. If you use modern PHP and want to make code compatible with Monolog 1 and 2 this can help. You will have to require at least Monolog 1.25 though.
* Added support for RFC3164 (outdated BSD syslog protocol) to SyslogUdpHandler
* Fixed issue in GroupHandler and WhatFailureGroupHandler where setting multiple processors would duplicate records
* Fixed issue in SignalHandler restarting syscalls functionality
* Fixed normalizers handling of exception backtraces to avoid serializing arguments in some cases
* Fixed ZendMonitorHandler to work with the latest Zend Server versions
* Fixed ChromePHPHandler to avoid sending more data than latest Chrome versions allow in headers (4KB down from 256KB).

### 1.24.0 (2018-11-05)

* BC Notice: If you are extending any of the Monolog's Formatters' normalize method, make sure you add the new $depth = 0 argument to your function signature to avoid strict PHP warnings.
* Added a ResettableInterface in order to reset/reset/clear/flush handlers and processors
* Added a ProcessorInterface as an optional way to label a class as being a processor (mostly useful for autowiring dependency containers)
* Added a way to log signals being received using Monolog\SignalHandler
* Added ability to customize error handling at the Logger level using Logger::setExceptionHandler
* Added InsightOpsHandler to migrate users of the LogEntriesHandler
* Added protection to NormalizerFormatter against circular and very deep structures, it now stops normalizing at a depth of 9
* Added capture of stack traces to ErrorHandler when logging PHP errors
* Added RavenHandler support for a contexts context or extra key to forward that to Sentry's contexts
* Added forwarding of context info to FluentdFormatter
* Added SocketHandler::setChunkSize to override the default chunk size in case you must send large log lines to rsyslog for example
* Added ability to extend/override BrowserConsoleHandler
* Added SlackWebhookHandler::getWebhookUrl and SlackHandler::getToken to enable class extensibility
* Added SwiftMailerHandler::getSubjectFormatter to enable class extensibility
* Dropped official support for HHVM in test builds
* Fixed normalization of exception traces when call\_user\_func is used to avoid serializing objects and the data they contain
* Fixed naming of fields in Slack handler, all field names are now capitalized in all cases
* Fixed HipChatHandler bug where slack dropped messages randomly
* Fixed normalization of objects in Slack handlers
* Fixed support for PHP7's Throwable in NewRelicHandler
* Fixed race bug when StreamHandler sometimes incorrectly reported it failed to create a directory
* Fixed table row styling issues in HtmlFormatter
* Fixed RavenHandler dropping the message when logging exception
* Fixed WhatFailureGroupHandler skipping processors when using handleBatch and implement it where possible
* Fixed display of anonymous class names

### 1.23.0 (2017-06-19)

* Improved SyslogUdpHandler's support for RFC5424 and added optional $ident argument
* Fixed GelfHandler truncation to be per field and not per message
* Fixed compatibility issue with PHP <5.3.6
* Fixed support for headless Chrome in ChromePHPHandler
* Fixed support for latest Aws SDK in DynamoDbHandler
* Fixed support for SwiftMailer 6.0+ in SwiftMailerHandler

### 1.22.1 (2017-03-13)

* Fixed lots of minor issues in the new Slack integrations
* Fixed support for allowInlineLineBreaks in LineFormatter when formatting exception backtraces

### 1.22.0 (2016-11-26)

* Added SlackbotHandler and SlackWebhookHandler to set up Slack integration more easily
* Added MercurialProcessor to add mercurial revision and branch names to log records
* Added support for AWS SDK v3 in DynamoDbHandler
* Fixed fatal errors occurring when normalizing generators that have been fully consumed
* Fixed RollbarHandler to include a level (rollbar level), monolog\_level (original name), channel and datetime (unix)
* Fixed RollbarHandler not flushing records automatically, calling close() explicitly is not necessary anymore
* Fixed SyslogUdpHandler to avoid sending empty frames
* Fixed a few PHP 7.0 and 7.1 compatibility issues

### 1.21.0 (2016-07-29)

* Break: Reverted the addition of $context when the ErrorHandler handles regular php errors from 1.20.0 as it was causing issues
* Added support for more formats in RotatingFileHandler::setFilenameFormat as long as they have Y, m and d in order
* Added ability to format the main line of text the SlackHandler sends by explicitly setting a formatter on the handler
* Added information about SoapFault instances in NormalizerFormatter
* Added $handleOnlyReportedErrors option on ErrorHandler::registerErrorHandler (default true) to allow logging of all errors no matter the error\_reporting level

### 1.20.0 (2016-07-02)

* Added FingersCrossedHandler::activate() to manually trigger the handler regardless of the activation policy
* Added StreamHandler::getUrl to retrieve the stream's URL
* Added ability to override addRow/addTitle in HtmlFormatter
* Added the $context to context information when the ErrorHandler handles a regular php error
* Deprecated RotatingFileHandler::setFilenameFormat to only support 3 formats: Y, Y-m and Y-m-d
* Fixed WhatFailureGroupHandler to work with PHP7 throwables
* Fixed a few minor bugs

### 1.19.0 (2016-04-12)

* Break: StreamHandler will not close streams automatically that it does not own. If you pass in a stream (not a path/url), then it will not close it for you. You can retrieve those using getStream() if needed
* Added DeduplicationHandler to remove duplicate records from notifications across multiple requests, useful for email or other notifications on errors
* Added ability to use %message% and other LineFormatter replacements in the subject line of emails sent with NativeMailHandler and SwiftMailerHandler
* Fixed HipChatHandler handling of long messages

### 1.18.2 (2016-04-02)

* Fixed ElasticaFormatter to use more precise dates
* Fixed GelfMessageFormatter sending too long messages

### 1.18.1 (2016-03-13)

* Fixed SlackHandler bug where slack dropped messages randomly
* Fixed RedisHandler issue when using with the PHPRedis extension
* Fixed AmqpHandler content-type being incorrectly set when using with the AMQP extension
* Fixed BrowserConsoleHandler regression

### 1.18.0 (2016-03-01)

* Added optional reduction of timestamp precision via Logger->useMicrosecondTimestamps(false), disabling it gets you a bit of performance boost but reduces the precision to the second instead of microsecond
* Added possibility to skip some extra stack frames in IntrospectionProcessor if you have some library wrapping Monolog that is always adding frames
* Added Logger->withName to clone a logger (keeping all handlers) with a new name
* Added FluentdFormatter for the Fluentd unix socket protocol
* Added HandlerWrapper base class to ease the creation of handler wrappers, just extend it and override as needed
* Added support for replacing context sub-keys using %context.\*% in LineFormatter
* Added support for payload context value in RollbarHandler
* Added setRelease to RavenHandler to describe the application version, sent with every log
* Added support for fingerprint context value in RavenHandler
* Fixed JSON encoding errors that would gobble up the whole log record, we now handle those more gracefully by dropping chars as needed
* Fixed write timeouts in SocketHandler and derivatives, set to 10sec by default, lower it with setWritingTimeout()
* Fixed PHP7 compatibility with regard to Exception/Throwable handling in a few places

### 1.17.2 (2015-10-14)

* Fixed ErrorHandler compatibility with non-Monolog PSR-3 loggers
* Fixed SlackHandler handling to use slack functionalities better
* Fixed SwiftMailerHandler bug when sending multiple emails they all had the same id
* Fixed 5.3 compatibility regression

### 1.17.1 (2015-08-31)

* Fixed RollbarHandler triggering PHP notices

### 1.17.0 (2015-08-30)

* Added support for checksum and release context/extra values in RavenHandler
* Added better support for exceptions in RollbarHandler
* Added UidProcessor::getUid
* Added support for showing the resource type in NormalizedFormatter
* Fixed IntrospectionProcessor triggering PHP notices

### 1.16.0 (2015-08-09)

* Added IFTTTHandler to notify ifttt.com triggers
* Added Logger::setHandlers() to allow setting/replacing all handlers
* Added $capSize in RedisHandler to cap the log size
* Fixed StreamHandler creation of directory to only trigger when the first log write happens
* Fixed bug in the handling of curl failures
* Fixed duplicate logging of fatal errors when both error and fatal error handlers are registered in monolog's ErrorHandler
* Fixed missing fatal errors records with handlers that need to be closed to flush log records
* Fixed TagProcessor::addTags support for associative arrays

### 1.15.0 (2015-07-12)

* Added addTags and setTags methods to change a TagProcessor
* Added automatic creation of directories if they are missing for a StreamHandler to open a log file
* Added retry functionality to Loggly, Cube and Mandrill handlers so they retry up to 5 times in case of network failure
* Fixed process exit code being incorrectly reset to 0 if ErrorHandler::registerExceptionHandler was used
* Fixed HTML/JS escaping in BrowserConsoleHandler
* Fixed JSON encoding errors being silently suppressed (PHP 5.5+ only)

### 1.14.0 (2015-06-19)

* Added PHPConsoleHandler to send record to Chrome's PHP Console extension and library
* Added support for objects implementing \_\_toString in the NormalizerFormatter
* Added support for HipChat's v2 API in HipChatHandler
* Added Logger::setTimezone() to initialize the timezone monolog should use in case date.timezone isn't correct for your app
* Added an option to send formatted message instead of the raw record on PushoverHandler via ->useFormattedMessage(true)
* Fixed curl errors being silently suppressed

### 1.13.1 (2015-03-09)

* Fixed regression in HipChat requiring a new token to be created

### 1.13.0 (2015-03-05)

* Added Registry::hasLogger to check for the presence of a logger instance
* Added context.user support to RavenHandler
* Added HipChat API v2 support in the HipChatHandler
* Added NativeMailerHandler::addParameter to pass params to the mail() process
* Added context data to SlackHandler when $includeContextAndExtra is true
* Added ability to customize the Swift\_Message per-email in SwiftMailerHandler
* Fixed SwiftMailerHandler to lazily create message instances if a callback is provided
* Fixed serialization of INF and NaN values in Normalizer and LineFormatter

### 1.12.0 (2014-12-29)

* Break: HandlerInterface::isHandling now receives a partial record containing only a level key. This was always the intent and does not break any Monolog handler but is strictly speaking a BC break and you should check if you relied on any other field in your own handlers.
* Added PsrHandler to forward records to another PSR-3 logger
* Added SamplingHandler to wrap around a handler and include only every Nth record
* Added MongoDBFormatter to support better storage with MongoDBHandler (it must be enabled manually for now)
* Added exception codes in the output of most formatters
* Added LineFormatter::includeStacktraces to enable exception stack traces in logs (uses more than one line)
* Added $useShortAttachment to SlackHandler to minify attachment size and $includeExtra to append extra data
* Added $host to HipChatHandler for users of private instances
* Added $transactionName to NewRelicHandler and support for a transaction\_name context value
* Fixed MandrillHandler to avoid outputting API call responses
* Fixed some non-standard behaviors in SyslogUdpHandler

### 1.11.0 (2014-09-30)

* Break: The NewRelicHandler extra and context data are now prefixed with extra\_ and context\_ to avoid clashes. Watch out if you have scripts reading those from the API and rely on names
* Added WhatFailureGroupHandler to suppress any exception coming from the wrapped handlers and avoid chain failures if a logging service fails
* Added MandrillHandler to send emails via the Mandrillapp.com API
* Added SlackHandler to log records to a Slack.com account
* Added FleepHookHandler to log records to a Fleep.io account
* Added LogglyHandler::addTag to allow adding tags to an existing handler
* Added $ignoreEmptyContextAndExtra to LineFormatter to avoid empty [] at the end
* Added $useLocking to StreamHandler and RotatingFileHandler to enable flock() while writing
* Added support for PhpAmqpLib in the AmqpHandler
* Added FingersCrossedHandler::clear and BufferHandler::clear to reset them between batches in long running jobs
* Added support for adding extra fields from $\_SERVER in the WebProcessor
* Fixed support for non-string values in PrsLogMessageProcessor
* Fixed SwiftMailer messages being sent with the wrong date in long running scripts
* Fixed minor PHP 5.6 compatibility issues
* Fixed BufferHandler::close being called twice

### 1.10.0 (2014-06-04)

* Added Logger::getHandlers() and Logger::getProcessors() methods
* Added $passthruLevel argument to FingersCrossedHandler to let it always pass some records through even if the trigger level is not reached
* Added support for extra data in NewRelicHandler
* Added $expandNewlines flag to the ErrorLogHandler to create multiple log entries when a message has multiple lines

### 1.9.1 (2014-04-24)

* Fixed regression in RotatingFileHandler file permissions
* Fixed initialization of the BufferHandler to make sure it gets flushed after receiving records
* Fixed ChromePHPHandler and FirePHPHandler's activation strategies to be more conservative

### 1.9.0 (2014-04-20)

* Added LogEntriesHandler to send logs to a LogEntries account
* Added $filePermissions to tweak file mode on StreamHandler and RotatingFileHandler
* Added $useFormatting flag to MemoryProcessor to make it send raw data in bytes
* Added support for table formatting in FirePHPHandler via the table context key
* Added a TagProcessor to add tags to records, and support for tags in RavenHandler
* Added $appendNewline flag to the JsonFormatter to enable using it when logging to files
* Added sound support to the PushoverHandler
* Fixed multi-threading support in StreamHandler
* Fixed empty headers issue when ChromePHPHandler received no records
* Fixed default format of the ErrorLogHandler

### 1.8.0 (2014-03-23)

* Break: the LineFormatter now strips newlines by default because this was a bug, set $allowInlineLineBreaks to true if you need them
* Added BrowserConsoleHandler to send logs to any browser's console via console.log() injection in the output
* Added FilterHandler to filter records and only allow those of a given list of levels through to the wrapped handler
* Added FlowdockHandler to send logs to a Flowdock account
* Added RollbarHandler to send logs to a Rollbar account
* Added HtmlFormatter to send prettier log emails with colors for each log level
* Added GitProcessor to add the current branch/commit to extra record data
* Added a Monolog\Registry class to allow easier global access to pre-configured loggers
* Added support for the new official graylog2/gelf-php lib for GelfHandler, upgrade if you can by replacing the mlehner/gelf-php requirement
* Added support for HHVM
* Added support for Loggly batch uploads
* Added support for tweaking the content type and encoding in NativeMailerHandler
* Added $skipClassesPartials to tweak the ignored classes in the IntrospectionProcessor
* Fixed batch request support in GelfHandler

### 1.7.0 (2013-11-14)

* Added ElasticSearchHandler to send logs to an Elastic Search server
* Added DynamoDbHandler and ScalarFormatter to send logs to Amazon's Dynamo DB
* Added SyslogUdpHandler to send logs to a remote syslogd server
* Added LogglyHandler to send logs to a Loggly account
* Added $level to IntrospectionProcessor so it only adds backtraces when needed
* Added $version to LogstashFormatter to allow using the new v1 Logstash format
* Added $appName to NewRelicHandler
* Added configuration of Pushover notification retries/expiry
* Added $maxColumnWidth to NativeMailerHandler to change the 70 chars default
* Added chainability to most setters for all handlers
* Fixed RavenHandler batch processing so it takes the message from the record with highest priority
* Fixed HipChatHandler batch processing so it sends all messages at once
* Fixed issues with eAccelerator
* Fixed and improved many small things

### 1.6.0 (2013-07-29)

* Added HipChatHandler to send logs to a HipChat chat room
* Added ErrorLogHandler to send logs to PHP's error\_log function
* Added NewRelicHandler to send logs to NewRelic's service
* Added Monolog\ErrorHandler helper class to register a Logger as exception/error/fatal handler
* Added ChannelLevelActivationStrategy for the FingersCrossedHandler to customize levels by channel
* Added stack traces output when normalizing exceptions (json output & co)
* Added Monolog\Logger::API constant (currently 1)
* Added support for ChromePHP's v4.0 extension
* Added support for message priorities in PushoverHandler, see $highPriorityLevel and $emergencyLevel
* Added support for sending messages to multiple users at once with the PushoverHandler
* Fixed RavenHandler's support for batch sending of messages (when behind a Buffer or FingersCrossedHandler)
* Fixed normalization of Traversables with very large data sets, only the first 1000 items are shown now
* Fixed issue in RotatingFileHandler when an open\_basedir restriction is active
* Fixed minor issues in RavenHandler and bumped the API to Raven 0.5.0
* Fixed SyslogHandler issue when many were used concurrently with different facilities

### 1.5.0 (2013-04-23)

* Added ProcessIdProcessor to inject the PID in log records
* Added UidProcessor to inject a unique identifier to all log records of one request/run
* Added support for previous exceptions in the LineFormatter exception serialization
* Added Monolog\Logger::getLevels() to get all available levels
* Fixed ChromePHPHandler so it avoids sending headers larger than Chrome can handle

### 1.4.1 (2013-04-01)

* Fixed exception formatting in the LineFormatter to be more minimalistic
* Fixed RavenHandler's handling of context/extra data, requires Raven client >0.1.0
* Fixed log rotation in RotatingFileHandler to work with long running scripts spanning multiple days
* Fixed WebProcessor array access so it checks for data presence
* Fixed Buffer, Group and FingersCrossed handlers to make use of their processors

### 1.4.0 (2013-02-13)

* Added RedisHandler to log to Redis via the Predis library or the phpredis extension
* Added ZendMonitorHandler to log to the Zend Server monitor
* Added the possibility to pass arrays of handlers and processors directly in the Logger constructor
* Added $useSSL option to the PushoverHandler which is enabled by default
* Fixed ChromePHPHandler and FirePHPHandler issue when multiple instances are used simultaneously
* Fixed header injection capability in the NativeMailHandler

### 1.3.1 (2013-01-11)

* Fixed LogstashFormatter to be usable with stream handlers
* Fixed GelfMessageFormatter levels on Windows

### 1.3.0 (2013-01-08)

* Added PSR-3 compliance, the Monolog\Logger class is now an instance of Psr\Log\LoggerInterface
* Added PsrLogMessageProcessor that you can selectively enable for full PSR-3 compliance
* Added LogstashFormatter (combine with SocketHandler or StreamHandler to send logs to Logstash)
* Added PushoverHandler to send mobile notifications
* Added CouchDBHandler and DoctrineCouchDBHandler
* Added RavenHandler to send data to Sentry servers
* Added support for the new MongoClient class in MongoDBHandler
* Added microsecond precision to log records' timestamps
* Added $flushOnOverflow param to BufferHandler to flush by batches instead of losing the oldest entries
* Fixed normalization of objects with cyclic references

### 1.2.1 (2012-08-29)

* Added new $logopts arg to SyslogHandler to provide custom openlog options
* Fixed fatal error in SyslogHandler

### 1.2.0 (2012-08-18)

* Added AmqpHandler (for use with AMQP servers)
* Added CubeHandler
* Added NativeMailerHandler::addHeader() to send custom headers in mails
* Added the possibility to specify more than one recipient in NativeMailerHandler
* Added the possibility to specify float timeouts in SocketHandler
* Added NOTICE and EMERGENCY levels to conform with RFC 5424
* Fixed the log records to use the php default timezone instead of UTC
* Fixed BufferHandler not being flushed properly on PHP fatal errors
* Fixed normalization of exotic resource types
* Fixed the default format of the SyslogHandler to avoid duplicating datetimes in syslog

### 1.1.0 (2012-04-23)

* Added Monolog\Logger::isHandling() to check if a handler will handle the given log level
* Added ChromePHPHandler
* Added MongoDBHandler
* Added GelfHandler (for use with Graylog2 servers)
* Added SocketHandler (for use with syslog-ng for example)
* Added NormalizerFormatter
* Added the possibility to change the activation strategy of the FingersCrossedHandler
* Added possibility to show microseconds in logs
* Added server and referer to WebProcessor output

### 1.0.2 (2011-10-24)

* Fixed bug in IE with large response headers and FirePHPHandler

### 1.0.1 (2011-08-25)

* Added MemoryPeakUsageProcessor and MemoryUsageProcessor
* Added Monolog\Logger::getName() to get a logger's channel name

### 1.0.0 (2011-07-06)

* Added IntrospectionProcessor to get info from where the logger was called
* Fixed WebProcessor in CLI

### 1.0.0-RC1 (2011-07-01)

* Initial release