## HandlerInterface + isHandling(array \$record) + handle(array \$record) + handleBatch(array \$records) + close() Handler ResettableInterface + handleBatch(array \$records) + close() reset() \_\_destruct() \_sleep() AbstractHandler # \$level # \$bubble + \_\_construct(\$level =Logger::DEBUG, bool ProcessableHandlerInterface FormattableHandlerInterface \$bubble=true) + isHandling(array pushProcessor(callable + setFormatter(FormatterInterface \$record) \$callback) \$formatter) + setLevel(\$level) popProcessor() + getFormatter() + getLevel() + setBubble(bool \$bubble) + getBubble() + reset() AbstractProcessingHandler reset() write(array \$record) LogglyHandler # const HOST # const ENDPOINT\_SINGLE # const ENDPOINT\_BATCH # \$curlHandlers # \$token # \$tag construct(string \$token, \$level=Logger ::DEBUG, bool \$bubble=true) + setTag(\$tag) + addTag(\$tag) + handleBatch(array \$records) # getCurlHandler(string \$endpoint) # write(array \$record) # send(string \$data,

string \$endpoint)
# getDefaultFormatter()