

Coding Challenge

Telemetry package

Description

This package should offer a variety of features for meaningful logging. The goal is to remove the need to run, operate and maintain multiple agents/collectors in order to support telemetry data, like logs or transactions. By that the code using this package never has to be adjusted when changing the logging technology.

Requirements

- Every log needs a timestamp
- The library should offer several log level:
 - Debug, Info, Warning, Error
- Logs can have meta information in form of tags / attributes. Example:

origin: http customerId: 123

- The library supports multiple implementations (“drivers”) which process the logs. Drivers could be:
 - A direct output via CLI
 - JSON written as a file
 - Simple text file
 - etc.
- Transaction-styled logs¹, consisting of:
 - Transaction ID
 - Attributes

Boundary conditions

- **Usability:** Reasonable default configuration, supports different drivers and runs out of the box.
- **Configurable:** Customizable by a configuration file without touching the core code or the code which uses this package.
- **Extensible:** Without touching code the package should be extensible by new drivers.
- Programming languages have to be PHP or Go.
- The code has to be unit tested.
- How you design and implement this is up to you. Because of that this project is openly described on purpose.

¹ A transaction is basically the processing of a job by the application. Example: Request arrives, service processes something, saves something to the DB, email sent, end. A transaction is therefore a kind of grouping element of several logs that are all written in this process to handle this one job/request.

Deliverables

- UML diagrams of the software design
- GitHub repository with the source code