

Project Definition: We would like you to implement a basic monitoring system backend.

This system will not use any 3rd party tools, just pure your code. Project should be implemented in Golang. This system will be the central part of the monitoring system. And there will be small clients which will send monitoring events to this central server from deployed remote servers (you can simulate this by virtual machines or just localhost can work fine. But make sure clients run on different threads). Clients will send random events for project (just for simulating real world event sending process). Please at least have 4-5 different keyed events. We expect monitoring central system will allow clients to

- 1) There should be an HTTP endpoint that clients can send specific types of monitoring events. There will be 2 types of events. One type is cumulative events, in which the data will be stored as added to the previous ones. Second one is time-series. So all events must be stored individually. Event data must contain the following data as JSON format

```
{
  "client_id" : "...", // unique id for all clients (preferably UUID)
  "event_id" : "...", // unique id for all events (preferably UUID)
  "event_timestamp" : "..." // epoch timestamp of the event
  "event_type" : "..." // CUMULATIVE or TIME-SERIES
  "event_key" : "..." // type key of the event
  "event_value" : "..." // value of the event
}
```

- 2) Store events for clients in memory (implement a simple in-memory database)
- 3) Have another endpoint, when called dump the monitoring data for client with given ID. The result must be in JSON format. The result should be group by event key. For CUMULATIVE events there will be only one event data and it will contain the cumulative value. For TIME-SERIES events there should be an array of events.
- 4) Server should avoid duplicate events.

The duration of the project is a week.

Your project will be evaluated on these terms:

- Clear, reusable, modular code and architecture.
- Mastery of concepts, abstractions, separation of concerns.
- Mastery backend development.
- Codebase architecture respecting features requested

We will be expecting to receive

- Source code (please do not minify or obfuscate your code)
- Documentation if present
- A description of how we will execute your project, any dependencies etc.