**Site Reliability Engineering A4A Craft Demo**

**Problem Statement**

Time series data is a collection of observations obtained through [repeated measurements over time](https://www.influxdata.com/blog/what-is-time-series-data-and-why-should-you-care/). Plot the points on a graph, and one of your axes would always be time.

Some important Time series data that we use day-to-day would be important metrics like Requests Per Second (also known as Transactions Per Second), Latency etc.

As part of this project, your goal is to create a Restful Webservice in Python/Go/Java (your choice). This service should expose two API endpoints and implement GET and POST HTTP Verbs. You are welcome to use any libraries/frameworks that you would like to complete this problem.

**API Design:**

GET /v1/<service-id>/info --- Should display Aggregated information about the service.

Fields that should be returned:

* Service Name
* Service id - unique id , UUID or any id ??
* Total Requests served
* Max Latency observed
* Total Http\_200
* Total Http\_400
* Total Http\_502

POST /v1/<service-id>/data --- Should accept a payload for a particular service-id and add a corresponding record to the datastore.

Fields that should come as part of the payload:

* Service Name
* Service id
* Some random payload

Fields that should be inferred before storing in the choice of backend:

* Epoch Time - time when the response was sent back to the client.
* Latency Observed - Time taken to calculate the factorial of length of payload.

Responses will be based on below criteria:

Calculate the length of the payload and Calculate the factorial of that number.

If the calculation is taking more than 4s to calculate the factorial then respond with 5xx http status code and increment the http\_502 counter.

Otherwise respond with the factorial and increment the http\_200 counter and 200 status code.

If the payload is missing in POST, then increment the http\_400 counter.

For example - if the length of payload is 50 then calculate the factorial(50) and record the time it takes to execute the factorial and that would be latency observed.

**Essentials:**

Please have multiple service-ids as part of the dataset that can be queried by GET API call.

**Backend:**

It is okay to have a file-based backend for this or any other choice of database you would like to use. Our questions will revolve around how you approached the problem.

**Bonus (Totally Optional):**

This part is entirely optional and will not be part of the assessment.

Allow Datamodel to aggregate the data with 5 mins interval and visualize the data using charts as part of GET API and simple HTML interface (all we want to see are charts)

This would be a new GET API that can take aggregation time as a parameter and provide the aggregated data.

**Super Bonus (Totally and Totally Optional):**

Dockerize the whole stuff.