Coding challenge

This coding challenge requires the creation of an endpoint that saves/retrieves historical events.

Saving Events:

The event should have this fields:

- ID (this id should be autogenerated by the activity service, you can pick the format and explain why you pick this kind of id and the policy to generate it)
- CreatedAt timestamp
- Email string
- Environment string
- Component string
- Message string
- Data payload JSON

For example:

- ID: #123
- CreatedAt: 1526123095
- Email: diego@chefhero.com
- Environment: production
- Component: orders
- Message “the buyer #123456 has placed an order successfully”
- Data: { “order_id”: 123, “amount”:300, “created_at”:1526123095, …} This can have different schema, so the system should be able to store a random JSON payload.

Retrieving Events:

For retrieving events, the endpoint should be able to return a list of historical activities based on specified filters given to the endpoint. The following filters can be used:

- By Email
- By Environment
- By Component
- By some text on the message
- From a given date

For example:

- I want to retrieve the activity of diego@chefhero.com from **1-1-2018** on the environment “**production**” on the component “**inventory**” that contains the text “**error**”

What We’re Looking For

- We’re looking for you to split the responsibilities of the system into different layers.

- We would like you to specify what you would test and how in a README.md file.

- This service does not have so much business logic, but you should indicate on the code where the business logic will be located in case this service become more complex in the future.

- We would also like you to indicate where you would specify any future business logic additions to the code.

Nice to have but not necessary:

- Another interesting extra would be how the system would manage hundreds of activity per second, would you use any kind of cache, another kind of strategy to write the data on the database
- Will be nice if you can do on Golang, and using gRPC; but is not necessary

If you have doubts please do not hesitate to ask for clarification.