

Engineering @ AURA - API Design

REQUIREMENTS

The team at AURA wants to create an **on-demand emergency response platform**— similar conceptually to how you might request a ride-hailing service and be matched with the closest available taxi in real-time.

In coming up with this idea, the AURA engineering team coined the phrase: '**Never miss a panic**' as a guiding principle.

You have been tasked with creating a **highly fault tolerant** panic API. You are required to implement the following:

1. **POST /panic**
2. **GET /panic/:id**

In line with the above, you are required to ensure that an integrator consuming this API is able to **query a panic even if the creation has not yet completed**.

The API's performance should be observable and a rough rate of 1000 req/sec was mentioned in planning.

CONSTRAINTS

The team you are working with uses AWS and typescript in-house. They have asked that you make use of whichever frameworks you want outside of those two fixed requirements.

TIMEFRAME

You have a week to build this API following which you will need to present your solution to the AURA team. You can decide how deeply you go into the solution within that time constraint.

We will want to review the source code during the demo. Make sure you have thought about how to share and present this during the session.

Good luck, and we look forward to seeing what you come up with.