

## Build a simple pager duty application

### Product requirements:

Pager duty is a product that can receive alerts from certain alerting systems like cloudwatch, newrelic etc, and send these alerts to developers in the form of sms, phone calls etc.

We can create teams, and map a set of developers to those teams. When an alert is raised by the alerting system for a team, an sms needs to be sent to one of the developers of that team.

We have 2 entities:

Team: id, name

Developer: id, team\_id, name, phone number

- Create team api - This api takes in a team and a list of developers to be mapped with this team, and is expected to create the corresponding entries in the database.

Sample request: {"team": {"name": "claims"}, "developers": [{"name": "someone", "phone\_number": "9999999999"}, {"name": "somebody", "phone\_number": "9111111111"}]}

- Receive alert api - Whenever an alerting system figures out that an alert is to be sent, this api is called. Team id is sent as an input to this api. We should then choose any developer from the team and send out an sms to the developer by making a dummy api call to a different notification service with a hardcoded message.

Notification service api: POST <https://run.mocky.io/v3/fd99c100-f88a-4d70-aaf7-393dbbd5d99f>

Sample request: {"phone\_number": "9999999999", "message": "Too many 5xx!"}

- Write 1-2 sample unit test cases - no need to be exhaustive - but the ones that are written should be production quality

- Use external database, build on your local machine, take help from internet wherever needed, copy paste code snippets if needed

- Try to be as close as possible to production quality coding conventions

- Handle failures wherever applicable

- Since time is limited, please prioritize tasks in the order they have been mentioned in the problem.

- We are not building this for scale, so let's limit the tech stack to code and external database in the interest of time. We are also not assessing choice of tech stack, so anything that you are comfortable with is fine.