

24H Junior Backend case study

Context

Your team is building a global monitoring system for natural disasters which ingests real-time earthquake data, stores it efficiently, and makes it available for visualization and downstream analytics.

Your mission

Implement a basic backend service that ingests earthquake data from a public source, stores it in a PostgreSQL database, and exposes it via a RESTful API.

Requirements

- Use Python (FastAPI preferred) or Node.js
- Ingest earthquake data from USGS or use mock data
- Store data in PostgreSQL with a schema including:
 - Location
 - Magnitude
 - Depth
 - Time
- Implement RESTful API endpoints:
 - GET /earthquakes: list recent earthquakes
 - GET /earthquakes/{id}: fetch details
- Include basic error handling and logging
- Write a README with setup instructions

Deliverables

GitHub repo with:

- Source code
- Docker setup (compose and file)
- README with Setup instructions and design decisions
- Sample data or ingestion script
- API documentation (e.g., inline or OpenAPI)
- Brief notes on design decisions
- Short document with your assumptions and any limitations

Bonus (Optional)

- Add filtering by magnitude or date
- Simulate real-time updates (e.g., periodic refresh each 10 seconds)
- Include basic unit tests
- Add simple monitoring/logging (e.g., log to file)

Evaluation Criteria

- Code quality and structure
- Efficient schema and indexing
- RESTful design, filtering, pagination
- Docker, tests, logging, real-time simulation
- Documentation and ease of setup

We understand this case study requires a meaningful time investment, especially during a busy week. While we don't expect perfection, the goal is to get a deeper sense of how you approach problems, structure your work, and make technical decisions under realistic constraints. Do your best and prioritize on the elements you think are most important in the given time – we then discuss them together.

Happy coding,

Your Beyond Gravity Downstream team!