**EarthQuake API Caching & Rate Limiting**

**Objective:**  
Create a small backend service that wraps a free Earthquake API. You will build two endpoints to retrieve earthquake data and demonstrate basic caching, rate limiting, and load testing. Use the USGS Earthquake API (<https://earthquake.usgs.gov/fdsnws/event/1/> ) for the data. Feel free to browse the internet or use any AI tool you like. You are not expected to have any prior experience with the given earthquake API.

Your two endpoints will do the following:

1. **GET /earthquakes** – Accept optional parameters (for example, startTime, endTime) to fetch matching data from the external Earthquake API. Cache these results so repeated requests for the same data return instantly. Enforce a simple rate limit (for example, a maximum number of requests per minute).
2. **GET /earthquakes/:id** – Return a single earthquake’s details by its unique identifier. If the data was recently fetched, serve it from the cache.

**Requirements:**

1. Build a minimal backend service with the above endpoints.
2. Set up **caching** so repeated requests for the same data do not always call the Earthquake API.
3. Implement a basic **rate limiting** mechanism to prevent excessive requests.
4. Use load testing (any method you prefer) to confirm that your rate limiting is effective.
5. Test your final endpoints with Postman to ensure they work.
6. Create a public GitHub repository containing your solution.
7. Record a **2-minute** demo on Authcast showing how to use your endpoints.
8. Share your **GitHub** link and demo recording once you’re done.

**Tech Stack**

* **Backend**: Open to all languages/frameworks

earthquake-api/

├── app.js

├── routes/

│ └── earthquakeRoutes.js

├── services/

│ └── earthquakeService.js

├── cache/

│ └── cache.js

├── utils/

│ └── rateLimiter.js

├── package.json