**Short summary:**

Build an Instant Payment API service that allows users to send money instantly using REST.

**Focus:**

High Availability, Transactional Processing, Error Handling

**Task requirements:**

Use Git as a version control system, and upload your code into a public GitHub repository.

As a framework, use Spring Boot with Maven or Gradle, and your preferred Java version.

No UI implementation is required for this task.

Containerize your application, to prepare it for future cloud deployment.

**The API should:**

- Perform balance checks before processing.

- Handle concurrency (e.g., prevent double spending / double notifications).

- Save the transactions into a PostgreSQL table

- Use Kafka to asynchronously send a notification of the transaction to the recipient.

**Implementation extras:**

- Implement a fault tolerant system

- Tests to ensure functionality

- Document the REST API

**Bonus for Senior Candidates:**

How would you architect an instant payment system to ensure high availability and fault tolerance?

*(covering database partitioning, microservices, and resilience patterns).*

Prepare to present this designed architecture, this would be presented at a later stage of the interview process.

The task is designed to be completed within 6 hours. Please focus on delivering a working solution that demonstrates your understanding of a highly available service in a cloud environment, ensuring proper resiliency and transactional integrity for a financial system. If you are unable to fully complete all aspects, or run out of time, please document your approach and next steps.

When you are finished, please share your GitHub repository link with us.