

Overview

The purpose of the assignment is to assess object oriented analysis and modelling skills, Java coding skills, and code structuring. Take your time on the task, but don't get too carried away. If you submit a solution that is in any way incomplete, the parts that you decided to focus on are relevant.

Keeping the objective in mind, you are free to use whatever tools, libraries, frameworks at your disposal. Artificial Intelligence (AI) usage is encouraged. Please include a README in any format on how to run and use the program.

Requirements

Your task is to write a backend application that will receive a bet and process it for (1) jackpot pool contribution and (2) evaluate the bet for jackpot reward. You need to implement the following:

- **An API endpoint to publish a bet to Kafka.**
- **A Kafka consumer that listens to `jackpot-bets` Kafka named-topic.**
- **Each bet must contribute to a matching jackpot pool.**
- **Each bet must be evaluated for jackpot reward.**
- **An API endpoint to evaluate if a bet wins a jackpot reward.**

The details of these Use Cases can be found in the next page. There are no further defined requirements for the API, it is up to you to design and implement the necessary code in order for the API to support the mentioned operations above. This refers to the entire flow starting from the API, down to the persistence layer.

It is expected for you to spend around 90 minutes to complete the exercise.

Delivery

- The solution needs to be 100% executable
- Provide a link to the GitHub repository where your solution is committed

- Please make sure it is not private or restricted
- Provide a README file (documentation) how to run and use the solution

Jackpot Service

We want to launch a new backend service that will manage jackpot contributions and rewards. Here you have the relevant Use Cases to support:

1. An API endpoint to publish a bet to Kafka.

- a. This endpoint should publish a bet to Kafka. A bet is represented by:
 - i. Bet ID
 - ii. User ID
 - iii. Jackpot ID
 - iv. Bet Amount
- b. The topic should be `jackpot-bets` as commented before.

2. A Kafka consumer that listens to `jackpot-bets` Kafka named-topic.

3. Each bet must contribute to a matching jackpot.

- a. Each jackpot will start with a configurable initial pool value. Each Jackpot can have a different configuration for contribution. In the beginning we want to support just two options, but have the option to add more configurations in future.
 - i. Fixed contribution as a percentage of the Bet Amount.
 - ii. Variable contribution as a percentage of the Bet Amount. In the beginning the contribution is bigger and over time it becomes lower at fixed rate as the jackpot pool increases.
- b. The System checks for a matching jackpot based on Jackpot ID. If there is such a jackpot, the system should contribute to the jackpot pool according to its configuration.
- c. A Jackpot Contribution in database is composed by:
 - i. Bet ID
 - ii. User ID

- iii. Jackpot ID
- iv. Stake Amount
- v. Contribution Amount
- vi. Current Jackpot Amount
- vii. Created At Date

4. An API end point to evaluate if a bet wins the jackpot reward.

- a. This endpoint should check if a contributing bet wins the jackpot reward and return the reward as a response.
- b. Each Jackpot can have a different configuration for rewards. In the beginning we want to support just two options, but have the option to add more configurations in future.
- c. Fixed chance for a reward as a percentage.
- d. Variable chance as a percentage. In the beginning the chance for reward is smaller and over time it becomes bigger as the jackpot pool increases. If the jackpot pool hits a limit, then the chance becomes 100%.
- e. When a jackpot is rewarded, it should be reset to the initial pool value.
- f. A Jackpot Reward in database is composed by:
 - i. Bet ID
 - ii. User ID
 - iii. Jackpot ID
 - iv. Jackpot Reward Amount
 - v. Created At Date

Conditions:

- If the Kafka setup is too complex, use mocks for the Kafka producer. Just log the payload.
 - Use an in-memory database for the bets and jackpots.
-