

Backend Engineering Study

Description

XXX is a casual mobile game played by millions of users around the world. As backend software engineers, we are providing a Rest API to keep users' progress. Our systems have to be fast and secure to provide a better user experience. Our backend services are written in Spring Boot with Java.

1. User Progress

1.1 Features

- New user starts with 5,000 coins from level 1
- User progress (level, coins) is kept in MySQL Database which is our main persistent storage
- After each level, the user increments the level and earns 25 coins

1.2 Flow and Requests

- *CreateUserRequest* -> Creates user and returns unique user id, level, and coin
- *UpdateLevelRequest* -> Client sends after each level win. Updates level and coin of the user. Returns updated progress data

2. Tournament

We will have a time-limited tournament to give users a chance to compete with each other and earn rewards.

2.1 Features

- Each tournament starts at 00:00 (UTC) and ends at 20:00 (UTC) daily
- When a tournament finishes, a new one is started automatically
- Users should be at least level 20 and have to pay 1,000 coins to enter
- Users are grouped into 20 people by their levels at the tournament entrance. 20-100 level users can only be in the same group. It goes like this for every 100 levels. 101-200, 201-300 etc.
- Users compete in their own groups until the tournament is over.
- Users earn rewards by their ranks
- User cannot enter a new tournament if he/she didn't claim last entered tournament
- The tournament score is incremented one for each level update

2.2 Leaderboard

- Each tournament group has one leaderboard
- It is sorted from highest score to lowest
- Leaderboard data include user id, username, and tournament score
- Provided leaderboard data should be real-time

2.3 Rewards

- 1st user wins 10,000 coins
- 2nd user wins 5,000 coins
- 3rd user wins 3,000 coins
- 4th - 10th user wins 1,000 coins

2.4 Flow and Requests

- EnterTournamentRequest -> Enters current tournament and returns current group leaderboard
- ClaimRewardRequest -> Claims reward of a specific tournament and returns updated progress data
- GetRankRequest -> Gets rank of a specific player for a specific tournament
- GetLeaderboardRequest -> Gets full leaderboard data of a tournament group

Notes

We will pay attention to:

- The design and code structure
- Readability and comprehensibility of code (Clean code)
- Consideration of security, concurrency, and performance issues at high load
- Welcome to use any external data sources with MySQL if it helps to have better performance
 - Unit tests

Submission should include short documentation about your implementation and design decisions. Please feel free to contact us if you have any further questions.