Overview

Design and implement **Simple Casino** consisting of two microservices

- `Game Service`
- `Wallet Service`

System should allow to create player's wallet, deposit funds and make bets.

Functional requirements

Wallet Service API

Register

- In
  - playerId
- Out
  - OK, Balance(0)
  - KO - player already registered

Deposit

- In
  - playerId
  - amount
- Out
  - OK, Balance(..)
  - KO, Balance(..) - playerId not found

Withdraw

- In
  - playerId
  - amount
- Out
  - OK, Balance(..)
  - KO, Balance(..) - playerId not found or insufficient funds

Balance

- In
  - playerId
- Out

    - OK, Balance(..)
    - KO - player not found

Game Service API

PlaceBet

- In
    - playerId
    - gameId
    - amount
- Out
    - OK, Balance(..)
    - KO - incorrect game or wallet error

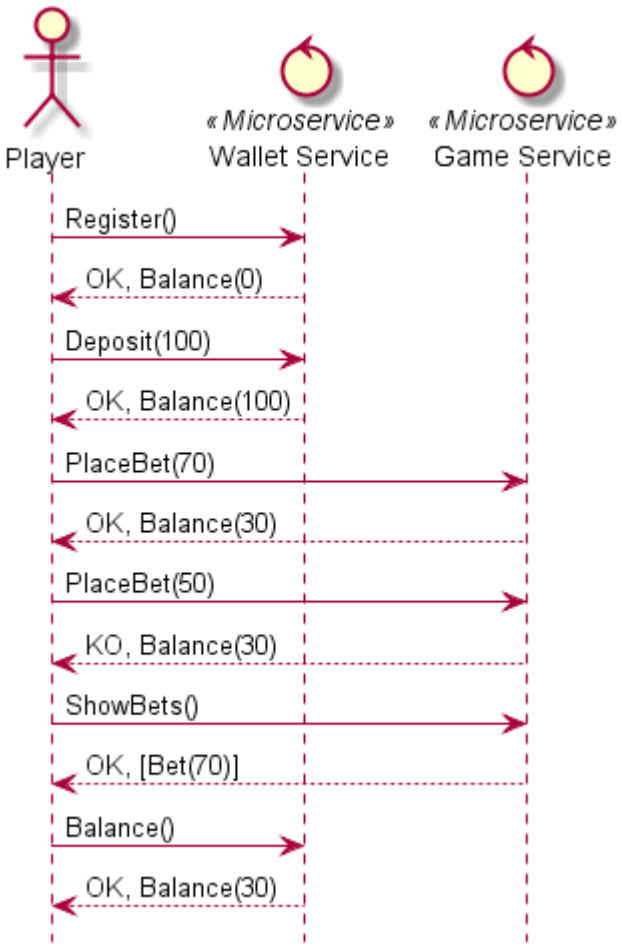
ShowBets

- In
    - playerId
- Out
    - OK, List<Bet>
    - KO, player not found

Non functional requirements

1. One of following languages (Java, Scala, Groovy) should be used
2. Pick up frameworks or libraries according to your needs
3. Persistence layer (SQL, NoSQL, etc) is mandatory for `Wallet Service`
4. System should support simple scaling scenario, i.e. several instances of `Game Service` can work with one `Wallet Service`
5. Automated tests are highly appreciated

Functional test scenario



Deliverables

1. Source code for both services
2. Brief guide how to build services
3. Scripts and guides how to start the whole system
4. **Simple Casino** should be started as
    - One instance of `Wallet Service`
    - Two instances of `Game Service`
5. Source code for functional test scenario
6. Brief guide how to execute functional test scenario against working system