Backend developer coding task "Social tournament service"

As a gaming website we want to implement a tournament service with a feature called "Back a friend".

Each player holds certain amount of bonus points. Website funds its players with bonus points based on all kind of activity. Bonus points can traded to goods and represent value like real money.

One of the social products class is a social tournament. This is a competition between players in a multi-player game like poker, bingo, etc)

Entering a tournament requires a player to deposit certain amount of entry fee in bonus points. If a player has not enough point he can ask other players to back him and get a part the prize in case of a win.

In case of multiple backers, they submit equal part of the deposit and share the winning money in the same ration.

From a technical side, the following REST service with 5 endpoints should be implemented

#1 Take and fund player account

/take?playerId=P1&points=300   takes 300 points from player P1 account

/fund?playerId=P2&points=300   funds player P2 with 300 points. If no player exist should create new player

#2 Announce tournament specifying the entry deposit

/announceTournament?tournamentId=1&deposit=1000

#3 Join player into a tournament and is he backed by a set of backers

/joinTournament?tournamentId=1&playerId=P1&backerId=P2&backerId=P3

Backing is not mandatory and a player can be play on his own money

#4 Result tournament winners and prizes

/resultTournament with a POST document in format

{"tournamentId": "1", "winners": [{"playerId": "P1", "prize": 500}]}

#5 Player balance

/balance?playerId=P1

Example response: {"playerId": "P1", "balance": 456.00}

#6 Reset DB.

/reset

Should reset DB to initial state

Full use case example:

Prepare initial balances

/fund?playerId=P1&points=300
/fund?playerId=P2&points=300
/fund?playerId=P3&points=300
/fund?playerId=P4&points=500
/fund?playerId=P5&points=1000

Tournament deposit is 1000 points

/announceTournament?tournamentId=1&deposit=1000

P5 joins on his own

/joinTournament?tournamentId=1&playerId=P5

P1 joins backed by P2, P3, P4

/joinTournament?tournamentId=1&playerId=P1&backerId=P2&backerId=P3&backerId=P4

All of them P1, P2, P3, P4 contribute 250 points each.

P1 wins the tournament and his prize is 2000. P2 P3 P4 they all get 25% of the prize.

/resultTournament with a POST

```
{"winners": [{"playerId": "P1", "prize": 2000}]}
```

After tournament result is processed the balances for players must be as specified below

P1, P2, P3 - 550

P4 - 750

P5 - 0

Implementation must guarantee that

- no player balance ever goes to zero
- no point is lost due to service outage

Endpoints 1-4 must return HTTP status codes only like 2xx, 4xx, 5xx

Endpoints 5 must return json document in the format on the example above

Your solution must be delivered as a source code on github / bitbucket and a docker compose script (use 1 free repository at hub.docker.com to deploy container image)

You can use whatever programming language and any open source data storage.

Feel free to ask for details.