

Написать бэкенд сервис, по обработке транзакций игрока.  
Каждый запрос имеет token, без которого запрос не действительный.  
Пользователи должны храниться в кэше с помощью map[uint64]\*User  
Пользователи, которые подверглись изменению, должны сохраняться в базу с периодичностью в 10 секунд.  
Статистика(depositCount, depositSum, ....) должны считаться отдельно(не в структуре пользователя) и в реалтайме, и тоже держаться в кэше.  
Метод GetUser не должен взаимодействовать с базой, а только с кэшем.

### 1. AddUser

Создает пользователя в системе.

POST "/user/create"

Принимаемые данные:

```
{ "id": 1, "balance": 0.0, "token": "testtask" }
```

Необходимо:

Проверка на уникальность ID пользователя

Успешные ответ:

```
{ "error": "" }
```

Неуспешный:

```
{ "error": "...." }
```

### 2. GetUser

Возвращает данные пользователя

POST "/user/get"

Принимаемые данные:

```
{ "id": 1, "token": "testtask" }
```

Успешный ответ:

```
{ "id": 1, "balance": 100.5, "depositCount": 1, "depositSum": 50, "betCount": 1, "betSum": 50, "winCount": 6, "winSum": 100 }
```

Неуспешный ответ:

```
{ "error": "... " }
```

### 3. AddDeposit

Начисляет деньги пользователю за совершенный депозит

POST "/user/deposit"

Принимаемые данные:

```
{“userId”:1, “depositId”:1209, “amount”:50, “token”:“testtask”}
```

Где:

userId - ID пользователя

depositId - ID депозита

amount - сумма депозита

Необходимо записывать каждый депозит пользователя отдельной коллекцией, где будет храниться баланс пользователя до депозита и баланс пользователя после, время начисления.

Успешный ответ:

```
{“error”:“”, “balance”:50}
```

Где balance - баланс пользователя после начисления депозита.

Неуспешный ответ:

```
{“error”:“...”}
```

#### **4.Transaction**

Совершение игровой транзакции.

POST “/transaction”

Принимаемые данные:

```
{“userId”:1, “transactionId”:501, “type”:“Win”, “amount”:50.5, “token”:“testtask”}
```

Где:

type - может быть Win(результат ставки) и Bet(совершение ставки)

amount - сумма изменения баланса игрока

transactionId - ID транзакции

Необходимо:

Проверять можно ли игроку совершить ставку(баланс не может быть отрицательным)

Записывать транзакции отдельной коллекцией, где будет ID ставки, сумма изменения, баланс до транзакции, баланс после транзакции, время.

Успешный ответ:

```
{“error”:“”, “balance”:100.5}
```

Где balance - баланс пользователя после обработки транзакции

Неуспешный ответ:

```
{“error”:“...”}
```

