TEST TASK

Intro

Our mission at Lyte — to feel every seat. If you get sick or change plans, you should be able to return your tickets, and Lyte should help find a buyer for these tickets.

Often fans reserve tickets with us while we don't have them. As soon as we get them, we deliver them. Process of attaching tickets to fan's reservations is "fulfilling".

Task

You need to write a service that will assign tickets to reservations and show reservation total cost.

Models

Tier - ticket type. For example, tickets can be regular or VIP. Tier has the following structure:

```
{
   “tier_id”: 1,                         - int, tier id
   “name”: “GA”,                         - str, tier name
   “price”: 10,                          - decimal, the cost of the each ticket in this tier
   “transfer_fee_percent”: 0.05,         - decimal, Lyte reward percentage for ticket sale
}
```

Ticket - a ticket that allows a fan to attend the concert. It has the following structure:

```
{
   “ticket_id”: 1,        - int, ticket id
   “tier_id”: 1,          - int, tier id,
   “status”: “free”,      - str, ticket status. One status out of two - “free” or “sold”.
}
```

TicketRequest - request to buy a ticket, fans send a request for the ticket, and as soon as it becomes available, we will give this ticket to the fan. It has the following structure:

```
{
   “ticket_request_id”: 1,  - int, ticket request id
   “tier_id”: 1,            - int, tier id
   “ticket_id”: 1           - int (could be None), real ticket id that attached to this ticket reservation
   “reservation_id”: 1      - int, user reservation id
}
```

Reservation - reservation that user created

```
{
   “reservation_id”: 1                      - int, reservation id,
   “email”: “example@gmail.com”             - str, client email
}
```

Service specification

The service must have 2 endpoints.

1. /api/v1/ticket-requests/ - POST - Accepts a list of Reservations with TicketRequests.
   Returns a list of TicketRequests filled with Tickets and prices with fees.
   You can attach ticket to TicketRequest only if the ticket status is “free”
   Tickets attached to TicketRequests should change their status to “sold”.
   If there are not enough tickets, an error should be returned.
   Reservation cost is the sum of ticket prices (tier.price).
   Reservation fee is the sum of tickets fees (tier.price*tier.transfer_fee_percent).

   Body example:

```
[
        {
                “reservation_id” : 1,
                “ticket_requests” : [
                        {
                                “ticket_request_id”: 1,
                                “tier_id”: 1,
                                “ticket_id”: null
                        },
                        {
                                “ticket_request_id”: 2,
                                “tier_id”: 2,
                                “ticket_id”: null
                        }
                ]
        },
        {
                “reservation_id”: 2,
                “ticket_requests”: [
                        {
                                “ticket_request_id”: 3,
                                “tier_id”: 2,
                                “ticket_id”: null
                        }
                ]
        }
]
```

```
Response example:
[
        {
                "reservation_id": 1,
                "reservation_cost": 12,  # cost without fees
                "reservation_fee": 1.2,
                "ticket_requests" : [
                        {
                                "ticket_request_id": 1,
                                "tier_id": 1,
                                "ticket_id": 1
                        },
                        {
                                "ticket_request_id": 2,
                                "tier_id": 2,
                                "ticket_id": 2
                        }
                ]
        },
        {
                "reservation_id": 2,
                "reservation_cost": 10,
                "reservation_fee": 1,
                "ticket_requests": [
                        {
                                "ticket_request_id": 3,
                                "tier_id": 2,
                                "ticket_id": 3
                        }
                ]
        }
]
```

2. /api/v1/performance/ - GET - Returns average request processing time (in milliseconds) and requests count. Statistics should be from process startup.
   Response example:
```
{
        "avarage_process_time": 0.3241,
        "requests_count" : 11
}
```

Test assessment criteria

1. Code quality - readability, correctness of the service, project organization
2. Completeness of the solution - handling various situations, tests