

1. Задание на знание Spring

Межпроцессный обмен в микросервисной архитектуре.

Используемый стек:

SpringBoot, Spring MVC, Spring Security, Spring JPA

RabbitMQ, DB H2 или же обычная коллекция в памяти

1) Клиент может выполнять два метода: http post для сохранения клиента и http get для получения информации по клиентам.

2) Первое приложение реализует два метода:

a) Создание клиента HTTP POST

<http://localhost:8181/addClient>

```
{
```

```
  "status": active,
```

```
  "FIO": "Ivanov Ivan Ivanovich",
```

```
}
```

b) Получение всех клиентов HTTP GET:

<http://localhost:8181/getClients>

2) После получения информации, десериализуем данные в объект и отправляем его через асинхронную очередь второму приложению.

3) Второе приложение принимает сообщение через асинхронную очередь и сохраняет в базу данных или же получает данные из БД(в зависимости от сообщения сохраняем клиента или же получаем всех клиентов)

4) Приложение спроектировать таким образом, чтобы можно было заменить базу данных на коллекцию объектов.

5) Вернуть полученный результат клиенту.

6) Тип асинхронной очереди - Request/response

7) Для взаимодействия с rabbitmq можно использовать spring amqp или же spring cloud connector(на усмотрение)

8) Использовать http basic authentication

Проект выложить на github

Схема



Проект выложить на github

2. Задание на знание ООП

Реализовать консольное приложение маркетплейса банка.

Ограничения:

- Применить шаблоны программирования GOF: абстрактная фабрика и стратегия.
- Язык программирования Java
- Фреймворк Spring boot

- Среда разработки IntelliJIdea
- Проект выложить на github

Вводная часть задания.

Клиент банка, он же пользователь маркетплейса.

Каждый клиент содержит атрибуты: ID, ФИО, тип привязанной к нему карты, признак пенсионера.

Есть два типа карт: Visa или Elcart и два типа платежных процессинга: МПЦ и Way4.

Каждая карта содержит атрибуты: ID, номер карты, тип карты.

Все транзакции оплаты пенсионеров(клиенты банка) в независимости от типа карты проходят в МПЦ.

Все остальные транзакции оплаты клиентов банка в независимости от типа карты проходят в Way4.

Алгоритм работы программы:

1. Клиенту необходимо представится(авторизоваться), т.е. ввести ID пользователя, например 77
2. В системе есть коллекция пользователей.
3. Необходимо проверить данного пользователя в коллекции, если его нет, зарегистрировать нового пользователя.
4. Если клиент есть в коллекции, его авторизуем и даем на выбор список товаров.
5. Список товаров - это коллекция объектов типа телефон: Iphone 7, 8 ,9,10,11, 12
6. Клиент выбирает телефон
7. После этого он должен выбрать тип карты, которой он хочет оплатить товар.
8. После оплаты, в зависимости от того какую он карту выбрал и от признака пенсионер он или нет, выполняется метод оплата в платежный процессинг.

3. Задание на знание структур данных и алгоритмов

Реализовать SQL анализатор.

Ограничение:

- Spring boot mvc
- Подключить к проекту swagger
- проект выложить на github
- к проекту приложить postman проект для тестирования приложения

Вводная часть.

Необходимо проанализировать SQL текст и выделить столбцы из sql выражения. SQL выражение может содержать любое количество столбцов.

Выделенные значения столбцов необходимо записать в коллекцию HashMap, где в качестве ключей будут колонки из SQL запросов, а в качестве значений будут текстовые коллекции вида ["int", "string", "double"]

Например есть SQL текст.

```
select
  t.t_partyid as id,
  t2.t_name1,
  t.t_shortcode as Name
from dparty_dbt t
join dpersn_dbt t2 on t2.t_personid = t.t_partyid
where t.t_partyid = 77
```

Мы должны получить текстовую коллекцию: [id, t_name, Name] и вернуть HashMap коллекцию вида: [<id,["int", "string", "double"]>, <t_name,["int", "string", "double"]>, <Name,["int", "string", "double"]>]

Задание:

Необходимо реализовать контроллер, который на вход принимает SQL строку методом POST и возвращает коллекцию, где каждой колонке из sql запроса, соответствуют различные типы данных.