

О проекте

Проект представляет собой платежную систему, позволяющая в зависимости от типа карты использовать соответствующие процессинговые центры для выпуска карт, пополнения и списания средств.

Стек технологий

Язык программирования: Java 11

Фреймворки и библиотеки:

1. Spring - Фреймворк для создания приложений на Java
2. Spring boot 2.7.9 - Версия фреймворка Spring Boot, облегчающего создание приложений Spring
3. Spring Data - Часть Spring, предназначенная для доступа к данным и интеграции с различными источниками данных.
4. PostgreSQL - Система управления реляционными базами данных, с которой ваш проект, вероятно, взаимодействует
5. Flyway - Библиотека для управления версиями базы данных, часто используется с PostgreSQL.
6. Mockito - Библиотека для создания моков (подделок) в тестировании.
7. Lombok - Библиотека, которая упрощает написание кода, добавляя аннотации для генерации методов и других конструкций.
8. Spring Validation – Библиотека для валидации данных

Архитектура

Контроллеры (Controllers): В этом каталоге находятся классы, отвечающие за обработку HTTP запросов

Сущности (Entities): В этом каталоге находятся классы, описывающие наши сущности

Репозитории (Repositories): В этом каталоге находятся интерфейсы, описывающие методы для взаимодействия с базой данных

Сервисы (Services): В этом каталоге находятся классы, отвечающие за бизнес-логику приложения

Мappers: В этом каталоге находятся классы, отвечающие за преобразование (Например с DTO в сущность или с сущности в DTO)

Исключения (exceptions): В этом каталоге описываются различные виды исключений

Обработчики исключений(exceptionHandlers): В этом каталоге находятся классы, отвечающие за обработку исключений

API Endpoints

POST <http://localhost:8080/api/cards/issue>

Content-Type: application/json

Описание: Выпускает новую карту

Пример запроса:

```
{  
  "passportNumber": "2386015",  
  "paymentSystem": "visa",  
  "phoneNumber": "996770230704",  
  "firstName": "Artur",  
  "lastName": "Aidarov"  
}
```

POST <http://localhost:8080/api/cards/replenish>

Content-Type: application/json

Описание: Пополняет баланс карты

Пример запроса:

```
{  
  "paymentSystem": "visa",  
  "cardNumber": "4169585306046785",  
  "amount": 100  
}
```

POST <http://localhost:8080/api/cards/withdraw>

Content-Type: application/json

Описание: Снимает средства с карты

Пример запроса:

```
{  
  "paymentSystem": "visa",  
  "cardNumber": "4169585306046785",  
  "pinCode": "8295",  
  "amount": 100  
}
```

GET <http://localhost:8080/api/cards?phoneNumber=996770230704>

Описание: получает все карты, зарегистрированные на номер телефона

Пример запроса:

<http://localhost:8080/api/clients?phoneNumber=996770230704>

Ответ:

```
{  
  {  
    "id": 3,  
    "clientId": 1,  
    "phoneNumber": "996770230704",  
    "cardNumber": "4169585366581179",  
    "expirationDate": "2028-10-22",  
    "balance": 0.0,  
    "paymentSystemName": "Visa"  
  },  
  {  
    "id": 4,  
    "clientId": 1,  
    "phoneNumber": "996770230704",  
    "cardNumber": "2221272075892416",  
    "expirationDate": "2026-10-22",  
    "balance": 0.0,  
    "paymentSystemName": "MasterCard"  
  }  
}
```

GET <http://localhost:8080/api/clients?passportNumber=2386015>

Описание: получает информацию о клиенте по его номеру паспорта

Пример запроса: <http://localhost:8080/api/clients?passportNumber=111111>

Ответ:

```
[{"id": 1,
  "passportNumber": "2386015",
  "firstName": "Artur",
  "lastName": "Aidarov",
  "cards": [
    {
      "id": 1,
      "clientId": 1,
      "phoneNumber": "996770230704",
      "cardNumber": "4169585307410514",
      "expirationDate": "2028-10-23",
      "balance": 0.0,
      "paymentSystemName": "Visa"
    },
    {
      "id": 2,
      "clientId": 1,
      "phoneNumber": "996770230704",
      "cardNumber": "2221272000159722",
      "expirationDate": "2026-10-23",
      "balance": 0.0,
      "paymentSystemName": "MasterCard"
    },
    {
      "id": 3,
      "clientId": 1,
      "phoneNumber": "996770111111",
      "cardNumber": "4169585390853576",
      "expirationDate": "2028-10-23",
      "balance": 0.0,
      "paymentSystemName": "Visa"
    }
  ]
}]
```

Расширение

На текущее число (21.10.2023) сервис умеет работать только с картами VISA и MasterCard.

Как расширяться, если нужно будет работать с новыми типами карт?

1. Если карта имеет новые свойства (допустим логин клиента), то нужно будет создать новый класс для этого типа карты, унаследоваться от класса Card и затем дополнить таблицу Cards новым полем
2. Для нового типа карт нужно будет создать свой процессинговый центр в каталоге Services. В этом процессинговом центре имплементируем интерфейс PaymentProcessingStrategy и затем определяем логику создания карты, пополнения и снятия средств
3. Возможно придется слегка изменить DTO и для метод для преобразования в классе CardMapper в каталоге Mappers добавив туда новое поле