

Проектная задача Bank API

Требуется реализовать веб-сервис, реализующий логику работы клиентов с банковскими счетами.

API Физического лица (клиента), должно позволять выполнять следующие действия по счетам:

- 1) Выпуск новой карты по счету
- 2) Просмотр списка карт
- 3) Внесение вклада
- 4) Проверка баланса

Структура приложения, структура схем таблиц базы данных, состав моделей и наименования точек входа не регламентируется. Необходимо самостоятельно решить, какие данные могут потребоваться для корректной работы приложения.

Технические требования:

Приложение должно быть написано на Java 8 с использованием Spring* (или Spring-boot) в любой конфигурации. Приложение должно собираться в *.jar и быть исполняемым из командной строки. Желательно должна быть применена архитектура MVC (Model View Controller). Стилль кода должен соответствовать java code conventions.

В качестве базы данных допускается использование любой известной БД, но для простоты разработки и тестирования желательно использовать H2 Embedded с инициализацией схемы и предзаполнением БД при старте приложения. При использовании предустановленной БД, в проект необходимо добавить скрипты создания схемы БД. Взаимодействие с БД из приложения может быть реализовано как с использованием фреймворков (hibernate, jooq), так и на чистом JDBC. Не допускается использование SpringData. Схема БД должна быть нормализована, иметь корректные связи между сущностями и должны быть созданы индексы, где они необходимы.

Приложение должно принимать на вход и отдавать данные в формате JSON.

Например возвращаемые данные для запроса списка карт:

```
[
  {
    "id": 1,
    "number": "1111 2222 3333 4444"
  },
  {
    "id": 2,
    "number": "1111 2222 3333 4444"
  },
  {
    "id": 3,
    "number": "1111 2222 3333 4444"
  }
]
```

На все точки входа должны быть написаны интеграционные тесты, так же должны быть написаны юнит тесты.

*При отсутствии навыков работы со Spring, допускается разработка приложения любым известным способом.

Приложение должно быть реализовано поэтапно.

- 1) БД, DAO слой.
- 2) Сервисный слой
- 3) Слой сетевого взаимодействия

После каждого этапа необходимо написать тесты.

Бонус 1

После реализации основных действий, можно перейти к реализации работы с контрагентами.

API Контрагентов должно предоставлять следующие функции:

- 1) Добавление контрагента
- 2) Просмотр контрагентов
- 3) Перевод средств контрагенту

Бонус 2

Помимо разработки клиентской части приложения, можно добавить вторую точку входа для сотрудника банка.

API Банковского сотрудника должно предоставлять следующие функции:

- 1) Добавление нового физ-лица
- 2) Открытие счета
- 3) Подтверждение выпуска карты
- 4) Подтверждение операции

После добавления новых функций, необходимо добавить соответствующие проверки на стороне клиента (например исключение неподтвержденной карты из списка или изменение баланса карты только после подтверждения банком)