**Описание услуг банка и обмена валют в проекте: Разработка бэкенда банковского микросервиса**

**1. Услуги банка:**

* Открытие счета (Open Account):
* Описание: Клиент может открыть новый банковский счет.
* Реализация: Метод POST /bank-services/open-account с параметрами клиента. Возвращается информация о новом счете.
* Выдача кредита (Issue Loan):
* Описание: Банк предоставляет услугу по выдаче кредита клиенту.
* Реализация: Метод POST /bank-services/issue-loan с параметрами клиента и суммой кредита. Возвращается информация о кредите.
* Конвертация валюты (Currency Conversion):
* Описание: Клиент может выполнить операцию по конвертации валюты.
* Реализация: Метод POST /bank-services/currency-conversion с указанием исходной и целевой валют, а также суммы. Возвращается сконвертированная сумма.
* Перевод средств (Transfer Funds):
* Описание: Клиент может осуществлять перевод средств между своими счетами или на счета других клиентов.
* Реализация: Метод POST /bank-services/transfer-funds с указанием счетов отправителя и получателя, а также суммы. Возвращается подтверждение операции.

**2. Обмен валюты:**

* Запрос текущих курсов валют (Get Exchange Rates):
* Описание: Пользователь может получить текущие курсы обмена для различных валют.
* Реализация: Метод GET /currency-exchange/exchange-rates. Возвращает список текущих курсов валют.
* История обменов (Exchange History):
* Описание: Пользователь может просмотреть историю своих операций по обмену валюты.
* Реализация: Метод GET /currency-exchange/exchange-history/{userId}. Возвращает список прошлых операций обмена для конкретного пользователя.
* Операция обмена валюты (Currency Exchange Operation):
* Описание: Пользователь может провести операцию обмена валюты.
* Реализация: Метод POST /currency-exchange/exchange с указанием валют, суммы и направления обмена. Возвращает подтверждение операции.
* Статистика по обмену валюты (Exchange Statistics):
* Описание: Пользователь может получить статистическую информацию о своих операциях по обмену валюты.
* Реализация: Метод GET /currency-exchange/exchange-statistics/{userId}. Возвращает статистику обмена для конкретного пользователя.

**3. Услуги кассы:**

* Выдача наличных (Cash Withdrawal):
* Описание: Клиент имеет возможность снять наличные с банковского счета.
* Реализация: Метод POST /bank-services/cash-withdrawal с указанием счета клиента и суммы для снятия. Возвращает подтверждение операции.
* Внесение наличных (Cash Deposit):
* Описание: Клиент может внести наличные на свой банковский счет.
* Реализация: Метод POST /bank-services/cash-deposit с указанием счета клиента и суммы для внесения. Возвращает подтверждение операции.
* Обмен валюты в кассе (Currency Exchange at Counter):
* Описание: Предоставление услуги по обмену валюты через банковскую кассу.
* Реализация: Метод POST /bank-services/currency-exchange-at-counter с указанием валют, суммы и направления обмена в кассе. Возвращает подтверждение операции.
* Запрос баланса в кассе (Check Balance at Counter):
* Описание: Клиент может запросить баланс своего счета в банковской кассе.
* Реализация: Метод GET /bank-services/check-balance-at-counter/{accountId}. Возвращает текущий баланс счета.
* Платежи через кассу (Counter Payments):
* Описание: Возможность клиенту производить различные платежи через банковскую кассу.
* Реализация: Метод POST /bank-services/counter-payments с указанием деталей платежа. Возвращает подтверждение операции.
* Прочее:
* Кассовые операции в истории (Counter Operations History):
* Описание: Возможность просмотра истории кассовых операций клиента.
* Реализация: Метод GET /bank-services/counter-operations-history/{clientId}. Возвращает список прошлых кассовых операций для конкретного клиента.

**Успешное завершение проекта:**

* Все реализованные банковские услуги функционируют корректно, предоставляя клиентам широкий спектр банковских операций.
* Механизм обмена валюты обеспечивает удобство и прозрачность для пользователей.
* Реализованы меры безопасности для защиты данных клиентов.
* Весь функционал документирован с использованием Swagger, обеспечивая удобство для разработчиков и будущих тестировщиков.
* Проект покрыт тестами с использованием Mockito и JaCoCo для измерения охвата кода.
* Применены принципы AOP для реализации логирования и других поперечно-резцовых задач.
* Использованы технологии Hibernate и JPA для эффективного взаимодействия с базой данных MySQL.
* Вся функциональность прошла тщательное тестирование, и проект готов к развертыванию в продакшене.