

Требования к проекту Java Currency Converter Bot

Введение	2
Основные функции:	2
Общие требования	3
Управление версиями и сотрудничество	3
Целевые артефакты	3
Функциональные требования	4
Аутентификация пользователей	4
Настройки	4
Конвертер	4
История	5
Уведомления	5
Математические операции с валютами	6
Источник данных для курсов валют	6
Нефункциональные требования	8
Производительность	8
Надежность	8
Безопасность	8
Требования к технологическому стеку	10
Требования к документации	11
Требования к тестированию	12
Руководство по сдаче проекта	13
Требования	13
Архитектура	13
Полный проект	13
Глоссарий	14
Основные термины	14
Функциональные модули	14
Команды бота	17
Технические термины	15
Нефункциональные требования	Ошибка! Закладка не определена.
Процесс разработки	16
Форматы данных	17
Статусы операций	17

1 Введение

Целью проекта является разработка Telegram-бота для конвертации валют, который позволяет пользователям получать актуальные курсы валют, конвертировать суммы, настраивать уведомления и просматривать историю запросов.

1.1 Основные функции:

- Установка домашней валюты и валютной пары по умолчанию и валютной пары по умолчанию.
- Конвертация валют по запросу пользователя.
- Просмотр истории запросов за указанный период.
- Настройка уведомлений о изменении курса валют.
- Математические операции с конвертацией валют.

2 Общие требования

2.1 Управление версиями и сотрудничество

- Весь код должен быть размещен в репозитории GitHub.
- Репозиторий должен содержать файл `README.md` с подробным руководством по:
 - Настройке проекта локально.
 - Сборке и запуску приложения.
 - Развертыванию приложения с использованием Docker.
 - Взаимодействию с Telegram-ботом.
- Docker-образ приложения должен быть размещён на Docker Hub.

2.2 Целевые артефакты

- Приложение должно создавать исполняемый "fat JAR" как основной артефакт.
- Приложение должно быть развертываемо как Docker-контейнер.
- Использовать инструмент сборки Gradle для управления зависимостями, сборки и упаковки приложения.

3 Функциональные требования

Пользователь должен иметь возможность:

3.1 Аутентификация пользователей

- Идентифицироваться по Chat ID. Регистрация не требуется.

3.2 Настройки

- устанавливать домашнюю валюту.
- устанавливать валютную пару по умолчанию.

3.3 Конвертер

- Запрашивать курс обмена:
 - Если пользователь ничего не указал, то выводится курс валютной пары по умолчанию.
 - Если пользователь указал одну валюту, то производить вывод курса валютной пары домашней валюты и введенной пользователем.
 - Если указаны две валюты, то показывать курс обмена.
- Конвертировать указанные суммы по правилам курса обмена.

3.3.1 Форматы ввода

- Валюта задаётся в формате ISO 4217 (3 буквенных кода): USD (доллар США), EUR (евро), RUB (рубль) и т.д. Полный список поддерживаемых кодов должен быть доступен по команде /codes.

- Сумма для конвертации:
 - Целое число (100) или число с плавающей точкой (50.25).
 - Разделитель дробной части — точка (.).
 - Отрицательные значения не допускаются.

Примеры корректных запросов:

`/convert 100 USD to EUR`
`/convert 75.5 GBP to JPY`

3.4 История

- Просматривать все запросы за указанный период.
- Просматривать запросы для конкретной валюты/пары за период.
- Хранение данных 30 дней. Автоматическое удаление старых записей каждый день.
- Экспорт истории в формате CSV по запросу
 - Если не указана валютная пара, то выводится история по всем параметрам валютным парам.
 - Если указан timestamp “От”, но не указана timestamp “До”, то выводится вся история с timestamp “От”.
 - Если не указан период, то выводится вся история.

3.4.1 Форматы дат и времени:

- Дата в формате YYYY-MM-DD (например, 2024-01-31).
- Если время не указано, подразумевается 00:00:00 UTC.
- Пример: `/export 2024-01-01 2024-01-31`

3.5 Уведомления

Получать уведомления:

- Для желаемых курсов обмена.
- Если курс обмена значительно изменился.

Порог изменения курса: пользователь задаёт абсолютное значение или процент (например, +5%, -0.1).

- Команды для добавления уведомления, удаления уведомления по его ID и получения списка всех уведомлений

3.5.1 Форматы условий для уведомлений:

- Абсолютное значение: USD/EUR > 1.2 (курс больше 1.2).
- Процентное изменение: USD/EUR +5% (увеличение на 5% от текущего курса).

- Примеры:

```
/alert_add USD/EUR > 1.2
```

```
/alert_add GBP/RUB -2.5%
```

3.6 Математические операции с валютами

- Производить математические операции с конвертацией валют.
- Разрешать вычисления с разными валютами.
- Поддерживаются операции +, -, *, / с приоритетом.
- Конвертация всех сумм в домашнюю валюту по текущему курсу.

3.6.1 Форматы выражений:

- Для вычислений величины приводятся к валюте которая указана после to
- Операции: +, -, *, / с стандартным приоритетом.
- Скобки () для изменения порядка вычислений.

- Пример:

```
/calc (100 USD + 50 EUR) * 2 to RUB
```

3.7 Источник данных для курсов валют

- Использовать Open Exchange Rates (требуется API-ключ, бесплатный тариф до 1000 запросов/месяц).
- Производить запрос актуальных данных с Open Exchange Rates минимум раз в 1 час.

4 Общие требования

4.1 Производительность

- Максимальное время ответа — 300 мс (исключая задержки внешнего API).
- Приложение должно поддерживать до 100000 одновременных пользователей.

4.2 Надежность

- Повторные попытки при ошибках внешнего API (3 попытки с интервалом 1 сек).

4.3 Безопасность

- Шифрование API-ключа Open Exchange Rates и данных подключения к БД в переменных окружения.
- Валидация ввода пользователя (регулярные выражения для валют и сумм).

4.4 Роли пользователей:

- Пользователь (User) - им является любой пользователь, которого можно идентифицировать по chat_id.
 - Отправляет запросы через telegram бота.
 - Стандартные права доступа,
- Администратор (Admin) - им является пользователь, который может идентифицировать себя по API-ключу.
 - API-ключ позволяет использовать эндпоинт Users
 - расширенные права доступа
 - доступ к эндпоинту /healthcheck

4.5 Авторизация:

- Поддержка авторизации через Telegram по Chat ID.
- Поддержка авторизации через HTTP (для API)
 - Ключи доступа должны храниться в зашифрованном виде в БД
 - Используется только группой пользователей Admin.

4.6 Эндпоинт Healthcheck

- Должен быть доступен без авторизации по HTTP.
- Должен возвращать:

- Статус сервера
- Список авторов (студентов)

4.7 Эндпоинт Users

- Должен быть доступен только для группы пользователей Admin.
- Должен предоставлять список всех пользователей.
- Генерация/отзыв API-ключей.
- Должен позволять принудительное обновление курсов валют.
- Должен предоставлять возможность рассылки глобальных уведомлений.

5 Требования к технологическому стеку

- **Язык программирования:** Java 23.
- **Фреймворк:** Spring 6.2.3 (WebFlux, JPA, Modulith).
- **Telegram API:** Приложение должно интегрироваться с Telegram API для предоставления функциональности бота.
- **Брокер сообщений:** RabbitMQ.
- **База данных:** PostgreSQL.
- **Контейнеризация:** Docker, Docker Compose.
- **Логирование:** SLF4J, Log4j
- **Тестирование:** JUnit, Mockito

6 Требования к документации

Должна быть написана документация для API с использованием Spring Docs.

7 Требования к тестированию

- Должны быть реализованы unit-тесты и интеграционные тесты для всех модулей.
- Обеспечить покрытие кода тестами не менее 60%.

8 Руководство по сдаче проекта

8.1 Требования

Предоставить документ с требованиями, который включает:

- Цель и функциональность приложения.
- Ключевые функции Telegram-бота.
- Нефункциональные требования.
- Любые предположения или ограничения.

8.2 Архитектура

Предоставить документ по архитектуре, который включает:

- Высокоуровневый обзор дизайна системы.
- Диаграммы (например, компонентные диаграммы, диаграммы последовательностей или блок-схемы) для иллюстрации архитектуры.
- Обоснование выбранного технологического стека.
- Описание того, как приложение будет строиться, развертываться и запускаться.

8.3 Полный проект

Предоставить полный проект в виде:

- Ссылки на репозиторий GitHub с полной кодовой базой.
- Ссылки на Docker Hub, где опубликован Docker-образ.
- Ссылки или имя пользователя Telegram-бота для тестирования и взаимодействия.

9 Глоссарий

9.1 Основные термины

Telegram	Многофункциональный мессенджер, который позволяет пользователям обмениваться сообщениями.
Telegram-бот	Программа в мессенджере Telegram, автоматически обрабатывающая команды пользователей
Chat ID	Уникальный идентификатор пользователя/чата в Telegram, используемый для аутентификации
Домашняя валюта	Основная валюта пользователя, в которой отображаются результаты операций (например, RUB)
Валютная пара	Пара валют для обмена (например, USD/EUR)
Курс обмена	Отношение стоимости одной валюты к другой (например, 1 USD = 0.92 EUR)

9.2 Функциональные модули

Конвертер валют	Модуль для перевода суммы из одной валюты в другую по текущему курсу
История операций	Журнал всех запросов пользователя с возможностью фильтрации по дате и валютам
Уведомления	Система оповещений о достижении заданного курса или его изменении
Математические операции	Калькулятор для вычислений с разными валютами с автоматической конвертацией

9.3 Технические термины

Open Exchange Rates	Внешний API для получения актуальных курсов валют
Fat JAR	Исполняемый JAR-файл со всеми зависимостями
API-ключ	Секретный уникальный идентификатор, используемый для аутентификации и авторизации пользователя, разработчика или вызывающей программы в API
API	Набор способов и правил, по которым различные программы общаются между собой и обмениваются данными
Валидация	Процесс проверки и подтверждения того, что продукт, система или процесс будут функционировать должным образом и удовлетворять ожидания пользователей.
Авторизация	Процесс предоставления определённому лицу или группе лиц прав на выполнение определённых действий, а также процесс проверки (подтверждения) этих прав при попытке выполнения этих действий.
Эндпоинт	Конечная точка веб-сервиса, к которой клиентское приложение обращается для выполнения определённых операций или получения данных.
HTTP	Протокол передачи гипертекста. Это набор правил, по которым данные в интернете передаются между разными источниками, обычно между компьютерами и серверами.
БД	Набор структурированных данных, предназначенный для хранения, обработки и изменения большого количества информации.
Логирование	Процесс записи и хранения информации о событиях, действиях и состояниях системы, приложений или пользователей.

Фреймворк	Набор инструментов, библиотек и правил, который помогает разработчику быстро создать продукт: сайт, приложение
Telegram API	Набор инструментов, который позволяет разработчикам программно взаимодействовать с платформой Telegram.
Брокер сообщений	Программное обеспечение, которое обеспечивает связь и передачу сообщений между различными компонентами или приложениями в распределённой системе.
Spring WebFlux	Реактивный фреймворк для обработки HTTP-запросов
PostgreSQL	Реляционная база данных для хранения истории операций
RabbitMQ	Брокер сообщений

9.4 Процесс разработки

Gradle	Система сборки проекта
Docker	Платформа для контейнеризации приложения
GitHub	это веб-сервис для хостинга IT-проектов и их совместной разработки.
Контейнеризация	Метод, с помощью которого программный код упаковывается в единый исполняемый файл вместе с библиотеками и зависимостями, чтобы обеспечить его корректный запуск.
Docker-образ	Шаблон (физически — исполняемый пакет), из которого создаются Docker-контейнеры.
Docker Hub	Облачная платформа для публикации, хранения и распространения Docker-образов.
Docker-контейнер	Стандартизированный, изолированный и портативный пакет программного обеспечения, который включает в себя всё необходимое для запуска приложения.

JUnit/Mockito Фреймворки для модульного тестирования

9.5 Форматы данных

CSV Формат экспорта истории операций (значения, разделенные запятыми)

JSON Формат обмена данными с внешними API

TIMESTAMP Метка времени в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС

Регулярные выражения для проверки:

Валюты: `^[A-Z]{3}$` (3 заглавные буквы).

Суммы: `^\d+(\.\d{1,2})?$` (положительное число с 2 знаками после точки).

Даты: `^\d{4}-\d{2}-\d{2}$`.

Времени: `:\d{4}:\d{2}:\d{2}$`

9.6 Тестирование

Тестирование Проверка на соответствие заявленной спецификации

Unit-тесты Проверка отдельных частей кода на корректность работы. В программировании под словом «юнит» чаще понимают функцию, метод или класс в исходном коде.

Интеграционные тесты Проверка отдельных модулей или компонентов приложения на совместимость друг с другом

9.7 Команды бота

Команда	Формат	Пример	Описание
/start	/start	/start	Инициализация бота
Настройки			
/sethome	/sethome <ВАЛЮТА>	/sethome RUB	Установка домашней валюты
/setpair	/setpair <ВАЛЮТА1> /<ВАЛЮТА2>	/setpair USD/EUR	Установка пары по умолчанию
/settings	/settings	/settings	Получение текущей домашней валюты и пары по умолчанию
Конвертер валют			
/currencies	/currencies	/currencies	Список поддерживаемых валют
/rate	/rate [ВАЛЮТА1]/[ВАЛЮТА2]	/rate EUR/USD	Курс для пары
	/rate [ВАЛЮТА]	/rate EUR	Курс для пары домашней валюты и указанной
	/rate	/rate	Выводится курс валютной пары по умолчанию
/convert	/convert <СУММА> <ИЗ> <В>	/convert 100 USD RUB	Конвертация с указанием суммы
	/convert <СУММА> <ИЗ>	/convert 100 USD	Конвертация суммы с использованием домашней валюты и указанной валюты
	/convert <СУММА>	/convert 100	Конвертация суммы с использованием валютной пары по умолчанию
История операций			

/export	/export [ОТ] [ДО] [ВАЛЮТА]	/export 2024-03-01 2024-03-20 EUR	Экспорт истории в CSV-файле по датам и валюте
	/export [ОТ] [ВАЛЮТА]	/export 2024-03-01 EUR	Экспорт истории в CSV-файле начиная с даты и валюте
	/export [ОТ] [ДО]	/export 2024-03-01 2024-03-20	Экспорт истории в CSV-файле по датам
	/export [ОТ]	/export 2024-03-01	Экспорт истории в CSV-файле начиная с даты
	/export [ВАЛЮТА]	/export EUR	Экспорт всей истории в CSV-файле по валюте
	/export	/export	Экспорт всей истории в CSV файле
/clear_history	/clear_history	/clear_history	Удаление персональной истории
Уведомления			
/alert_add	/alert_add <ПАРА> <УСЛОВИЕ>	/alert_add USD/RUB > 90	Создание уведомления с абсолютным значением
		/alert_add EUR/USD +5%	Создание уведомления с относительным значением
/alert_list	/alert_list	/alert_list	Получение списка активных уведомлений
/alert_remove	/alert_remove <ID>	/alert_remove 42	Удаление уведомления
Математические операции			
/calc	/calc <ВЫРАЖЕНИЕ> to <ВАЛЮТА>	/calc 100 USD + 50 EUR * 2 to RUB	Расчет с конвертацией
/math_help	/math_help	/math_help	Справка по

			операторам
--	--	--	------------

10 Исключительные ситуации

10.1 Пользовательский ввод

- Некорректные команды
 - `InvalidCommandException`: Несуществующая команда (например, `/conv 100 USD`)
 - `MalformedCurrencyPairException`: Неверный формат валютной пары (`USDEUR` вместо `USD/EUR`)
- Ошибки валидации
 - `InvalidCurrencyCodeException`: Несуществующий код валюты (`/convert 100 USD to XXXX`)
 - `NegativeAmountException`: Отрицательная сумма (`/convert -100 USD to EUR`)
 - `UnsupportedMathOperationException`: Недопустимая операция (`/calc 100 USD ^ 2`)

Исключение	Действие	Пример ответа пользователю
<code>InvalidCommandException</code>	Отправить справку по командам	"Неизвестная команда. Доступные команды: <code>/convert</code> , <code>/alert_add ...</code> "
<code>MalformedCurrencyPairException</code>	Показать пример формата	"Используйте формат <code>USD/EUR</code> . Пример: <code>/setpair USD/RUB</code> "
<code>InvalidCurrencyCodeException</code>	Отправить список поддерживаемых валют	"Валюта <code>XXX</code> не найдена. Список валют: <code>/currencies</code> "
<code>NegativeAmountException</code>	Указать требование к сумме	"Сумма должна быть положительной. Пример: <code>/convert 100 USD to EUR</code> "
<code>InvalidCurrencyCodeException</code>	Показать допустимые операторы	"Поддерживаются только <code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> . Пример: <code>/calc 100 USD + 50 EUR</code> "

10.2 Работа с внешними API

- Open Exchange Rates

- RateLimitExceededException: Превышен лимит запросов (1000/месяц)
- ExchangeRateUnavailableException: API недоступен (HTTP 503)
- InvalidApiKeyException: Неверный/просроченный API-ключ
- Telegram API
 - TelegramApiTimeoutException: Таймаут при отправке сообщения
 - ChatNotFoundException: Чат с пользователем не существует (удален/заблокирован)

Исключение	Пример ответа пользователю	Реакция системы
RateLimitExceededException	"Данные временно из кэша (актуальны на {дата})"	в healthcheck указать, что превышен лимит запросов
ExchangeRateUnavailableException	"Сервис курсов недоступен."	Автоматический retry через 1 сек, всего не более 3-х попыток
InvalidApiKeyException	"Данные временно из кэша (актуальны на {дата})"	в healthcheck указать, что ApiKey вышел из строя
TelegramApiTimeoutException	-	Повторная попытка через 5 сек
ChatNotFoundException	-	Деактивация уведомлений для чата. Очистка истории

10.3 Бизнес-логика

- Конвертация валют
 - CurrencyPairNotSupportedException: Пара не поддерживается (например, BTC/RUB)
- Уведомления
 - AlertThresholdExceededException: Порог уведомления не может быть >1000%
 - DuplicateAlertException: Уже существует уведомление для этой пары с таким же порогом
- История операций
 - HistoryPurgeException: Ошибка при автоматическом удалении старых записей
 - ExportFailedException: Не удалось сформировать CSV-файл

Исключение	Пример ответа пользователю	Реакция системы
CurrencyPairNotSupportedException	"Пара USD/BTC не поддерживается. Доступные валюты: можно посмотреть по команде \currencies"	-
AlertThresholdExceededException	"Данные временно из кэша (актуальны на {дата})"	-
DuplicateAlertException	"У вас уже есть уведомление для USD/EUR > 1.2 (ID: 42)"	-
HistoryPurgeException	-	Повтор через 1 час
ExportFailedException	"Не удалось создать CSV:"	Логирование ошибки

10.4 Системные ошибки

- База данных
 - DatabaseConnectionException: Потеря соединения с PostgreSQL
- Очередь сообщений
 - RabbitMQPublishException: Ошибка публикации сообщения в RabbitMQ

Исключение	Пример ответа пользователю	Реакция системы
DatabaseConnectionException	"Телеграм бот временно недоступен"	Попытка восстановить соединение
RabbitMQPublishException	-	Не более 5 повторных попыток отправки сообщения в очередь с ожиданием между попытками в 20 секунд.

10.5 Безопасность

- `InvalidApiKeyFormatException`: Неверный формат API-ключа в HTTP-заголовке
- `UnauthorizedAccessException`: Попытка доступа к `/users` без прав администратора

Исключение	Пример ответа пользователю	Реакция системы
<code>InvalidApiKeyFormatException</code>	<code>{"error":"Invalid API key format","valid_example":"Bearer abc123-def456"}</code>	HTTP 400 с описанием ошибки
<code>UnauthorizedAccessException</code>	<code>{"error":"Forbidden"}</code>	HTTP 403 + логирование попытки

11 Пользовательские сценарии

11.1 Аутентификация пользователя

Действия пользователя:

1. Открывает чат с ботом в Telegram.
2. Отправляет команду /start.

Действия системы:

3. Система автоматически сохраняет Chat ID.
4. Система отвечает приветственным сообщением.

11.2 Установка настроек пользователем

11.2.1 Установка домашней валюты

Последовательность действий:

1. Пользователь вводит команду: /sethome RUB.
2. Бот проверяет валидность кода валюты.
3. Бот сохраняет настройку в профиле.

11.2.2 Установка пары по умолчанию

Последовательность действий:

1. Пользователь вводит команду: /setpair USD/EUR.
2. Бот проверяет форматы и существование валют.
3. Бот сохраняет настройку в профиле.

11.3 Конвертер

11.3.1 Простая конвертация

Последовательность действий:

1. Пользователь вводит команду: /convert 100 USD to RUB.
2. Парсинг суммы и валют из команды ботом.
3. Получение актуального курса валют.
4. Расчет: $100 \times 92.50 = 9250 \text{ RUB}$

11.3.2 Автодополнение

Последовательность действий:

1. Пользователь вводит команду: /convert 100 USD.
2. Парсинг суммы и валюты из команды ботом.
3. Получение из настроек пользователя домашнего курса валют

4. Получение актуального курса валют.
5. Расчет: $100 \times 92.50 = 9250$ RUB.

11.4 История операций

Последовательность действий:

1. Пользователь вводит команду: `/export 2024-03-01 2024-03-15 EUR`.
2. Проверяется формат дат и валюты.
3. Производится фильтрация операций по промежутку времени и валюте.
4. Производится генерация CSV-файла.
5. Бот отправляет файл документом в чат.

11.5 Уведомления

11.5.1 Создание уведомления

Последовательность действий:

1. Пользователь вводит команду: `/alert_add USD/RUB > 95`.
2. Проверяется существования пары.
3. Производится сохранение порога.
4. Пользователю отправляется сообщение, что уведомление создано.

11.5.2 Срабатывание уведомления

Последовательность действий:

1. Происходит обновление курса валют в системе.
2. Производится сравнение новых валют с порогами.
3. Отправка уведомления в чат, что порог превышен.

11.6 Математические операции

Последовательность действий:

1. Пользователь вводит команду: `/calc (100 USD + 50 EUR) * 2 to JPY`.
2. Производится разбор выражения.
3. Производится конвертация валют по текущему курсу в JPY.
4. Производятся вычисления.
5. Результат возвращается пользователю.

12 Инструкция пользователя

1. Начало работы

Настройте параметры:

- Установите домашнюю валюту (Изначально стоит RUB):

Пример:

`/sethome RUB` (рубли).

- Задайте валютную пару по умолчанию (Изначально стоит USD/RUB):

Пример: `/setpair USD/EUR`

2. Основные команды

- Конвертация валют

- Команда: `/convert`

- Примеры:

- `/convert 100 USD to EUR` → конвертация 100 долларов в евро.

- `/convert 500` → конвертация 500 единиц валюты с использованием валютной пары по умолчанию.

- `/convert 200 EUR` → конвертация 200 евро в домашнюю валюту.

- История операций

- Команда: `/export`

- Экспорт в CSV:

`/export 2024-01-01 2024-01-31` → операции с 1 по 31 января 2024 сохранит историю в файл.

- Уведомления

- Команда: `/alert`

- Добавить уведомление:

- `/alert_add USD/EUR > 1.2` → при превышении курса 1.2.

- `/alert_add GBP/RUB -5%` → при снижении на 5%.

- Список активных уведомлений:

`/alert_list`.

- Удалить уведомление:

`/alert_remove 3` → удалит уведомление с ID=3.

- Математические операции

- Команда: `/calc`

- Примеры:

- `/calc 100 USD + 50 EUR to RUB` → сумма в рублях.

- `/calc (200 JPY * 3) / 2 to EUR` → вычисления со скобками.

- Дополнительные команды
 - Список кодов валют:
/codes → отобразит все поддерживаемые валюты.
 - Помощь:
/help → примеры корректных запросов.

3. Форматы ввода

- Валюты: 3 заглавные буквы (ISO 4217).
Примеры: USD, EUR, RUB.
- Суммы: Положительные числа с точкой.
Примеры: 100, 75.5.
- Даты: YYYY-MM-DD.
Пример: 2024-03-15.
- Условия для уведомлений:
Абсолютное значение: USD/EUR > 1.2.
Процентное изменение: GBP/RUB +3%.

4. Важно

- Курсы обновляются каждый час.
- История хранится 30 дней.
- Недопустимые форматы:
 - Пробелы в парах: USD/EUR (верно), USD / EUR (ошибка).
 - Отрицательные суммы.