Отчет о проведении юнит-тестирования сервисного слоя

Проект: Модуль обработки запросов и чатов

1. Введение

Данный отчет представляет результаты разработки и выполнения юнит-тестов для сервисов модуля обработки запросов и чатов. Основной целью являлось обеспечение качества и надежности ключевых компонентов системы путем их изолированного тестирования. Были разработаны тесты, покрывающие как основные сценарии использования, так и граничные случаи и обработку ошибок.

Разработка тестов проводилась с использованием фреймворка JUnit 5 и библиотеки Mockito, что позволило эффективно изолировать тестируемые сервисы от их зависимостей (репозиториев, других сервисов, внешних API) и сосредоточиться на проверке их внутренней логики.

2. Объем тестирования

Комплект юнит-тестов был разработан для следующих классов сервисного слоя:

- `FileProcessorServiceImpl`
- `TextProcessingService`
- `ExternalMessagingServiceImpl`
- `AutoResponderServiceImpl`
- `ClientServiceImpl`
- `LeastBusyAssignmentService`
- `NotificationServiceImpl`
- `UserServiceImpl`
- `SubscriptionServiceImpl`
- `SubscriptionPriceCalculateServiceImpl`

3. Методология и инструменты

- **Фреймворк тестирования:** JUnit 5 Jupiter
- **Библиотека мокирования:** Mockito
- **Подход:** Юнит-тестирование "белого ящика", основанное на анализе исходного кода сервисов.
- **Основные проверяемые аспекты:**
    - Позитивные сценарии выполнения основной бизнес-логики.
    - Обработка некорректных, `null` и граничных входных данных.
    - Корректное взаимодействие с зависимостями (проверка вызовов методов мок-объектов и передаваемых им аргументов).
    - Обработка исключений, как ожидаемых (бизнес-логика), так и неожидаемых (ошибки зависимостей).
    - Проверка состояний объектов и возвращаемых значений.

4. Обзор тестового покрытия по сервисам

- **FileProcessorServiceImpl**:
  - Тесты покрывают метод processFileUpload:
    - Успешная загрузка с перезаписью (overwriteExisting = true) и без нее.
    - Корректное взаимодействие с JobLauncher и Job (Spring Batch).
    - Обработка различных статусов завершения JobExecution (COMPLETED, FAILED).
    - Извлечение статистики (processedCount, duplicatesCount, rowErrors) и ошибок (globalErrors) из ExecutionContext.
    - Проверка отказоустойчивости при null статистике (getSafeInteger, getSafeMap).
    - Обработка ошибок на этапе определения типа файла (detectFileType) и запуска Job.
  - Отдельно протестирована логика метода detectFileType, включая обработку исключений от FileTypeDetector и двойной вызов детектора.
- **TextProcessingService**:
  - Протестирована работа метода processQuery для всех типов операций (CORRECTION, REWRITE, CORRECTION_THEN_REWRITE).
  - Реализована проверка корректности формируемого запроса (GenerationRequest), передаваемого в TextProcessingApiClient, с помощью ArgumentCaptor (проверяется промпт, параметры из MLParamsConfig, флаг isTextGeneration).
  - Покрыты сценарии обработки невалидных входных данных: null/пустой query (IllegalArgumentException), null generationType (NullPointerException).
  - Протестирована обработка ошибки валидации запроса (ConstraintViolationException), включая перехват и оборачивание в MLException со статусом 400 и проверку вызова validator.
  - Проверена логика возврата исходного текста запроса (fallbackText) при получении null ответа или ответа с null/пустым текстом от apiClient.
  - Протестирована обработка исключений (MLException, включая обернутые IOException и др.) от apiClient с проверкой механизма повторных попыток (ожидается 3 вызова apiClient.generateText).
- **ExternalMessagingServiceImpl**:
  - Протестирована отправка сообщений для каналов Telegram и Email.
  - Реализована проверка содержимого команды SendMessageCommand, помещаемой в очередь, с помощью ArgumentCaptor (проверяются канал, ID чата, контент, ID Telegram, адреса Email, тема).
  - Проверена корректная обработка пустого или null сообщения (отправка не происходит).
  - Покрыты сценарии ошибок: чат не найден (ChatNotFoundException), у чата отсутствует клиент (ResourceNotFoundException), у чата отсутствует

компания (ResourceNotFoundException), у чата не определен канал (ExternalMessagingException).
    - Протестирована обработка неподдерживаемых каналов (VK, WhatsApp), включая проверку оборачивания внутреннего исключения во внешнее ExternalMessagingException.
    - Протестированы случаи, когда конфигурация для Telegram или Email не найдена в репозитории (проверяется выброс ExternalMessagingException с причиной ResourceNotFoundException).
    - Проверен сценарий сбоя при помещении команды в очередь (InterruptedException, оборачивается в ExternalMessagingException).
    - Проверены граничные случаи: null имя у клиента при отправке Email (команда отправляется с null toEmailAddress), null ID/адрес в конфигурационных объектах (команда отправляется с null значениями).
- **AutoResponderServiceImpl**:
    - Протестирован метод processNewPendingChat:
        - Успешный сценарий: нахождение чата, первого сообщения, маппинг, косвенная проверка вызова processIncomingMessage через verify его зависимостей.
        - Ошибки: чат не найден (AutoResponderException), ошибка маппера (RuntimeException оборачивается).
        - Граничные случаи: чат не в статусе PENDING_AUTO_RESPONDER (выход), нет первого сообщения (выход).
    - Протестирован метод processIncomingMessage:
        - Успешный сценарий: коррекция запроса, поиск ответа, рерайт ответа, сохранение внутреннего сообщения (processAndSaveMessage) и отправка внешнего (sendMessageToExternal). Проверено содержимое отправляемого сообщения и senderId внутреннего сообщения (равен ID клиента).
        - Сценарий с ошибкой рерайта: используется оригинальный текст ответа, ошибка логируется, исключение не пробрасывается.
        - Сценарий без найденных ответов: публикация события эскалации (AutoResponderEscalationEvent), проверка содержимого события (chatId, clientId).
        - Сценарии ошибок: исключения от TextProcessingService (MLException) или AnswerSearchService (AnswerSearchException) приводят к эскалации, отправке сообщения об ошибке клиенту и выбросу AutoResponderException. Неожиданные Exception обрабатываются аналогично.
        - Граничные случаи: статус чата не PENDING_AUTO_RESPONDER или отправитель не CLIENT (выход), пустой запрос клиента (выход).
        - Проверена отказоустойчивость метода sendAutoResponderMessage при

ошибках processAndSaveMessage или sendMessageToExternal (ошибк и логируются, не пробрасываются).
    - Протестирован метод stopForChat (проверка отсутствия ошибок, т.к. метод пуст).
- **ClientServiceImpl**:
    - Покрыты методы поиска findById, findByName, findDtoById, findClientEntityByTelegramUsern ame, включая случаи, когда сущность не найдена, и передачу null/пустых строк в findByName.
    - Протестирован метод findClientEntityByEmail (возвращает Optional.empty()).
    - Протестирован метод getClientsByCompany, учитывая, что репозиторий возвращает Optional<Client> (тесты проверяют возврат списка из 0 или 1 DTO).
    - Протестирован метод createClient (с пользователем и без), включая проверку всех полей сохраняемой сущности (name, company, user, typeClient, createdAt, updatedAt) с помощью ArgumentCaptor.
    - Проверена обработка ошибок при создании: компания не найдена (ResourceNotFoundException), пользователь не найден (ResourceNotFoundException), ошибка сохранения в репозитории (DataAccessException).
- **LeastBusyAssignmentService**:
    - Протестирован метод findLeastBusyOperator: нахождение оператора (возвращается первый из списка), отсутствие операторов (возвращается Optional.empty()), ошибка репозитория (DataAccessException). Проверен вызов корректного метода репозитория (findLeastBusyUser с двумя аргументами).
    - Протестирован метод assignOperator: успешное назначение, случаи с null чатом или null компанией (возвращается Optional.empty()), ошибка репозитория при поиске оператора.
- **NotificationServiceImpl**:
    - Протестирован метод createNotification: проверка сохраняемой сущности (Notification) и аргументов вызова WebSocketMessagingService.sendToUser (ID пользователя, топик, DTO) с помощью ArgumentCaptor.
    - Проверена обработка ошибок при создании: null User (NullPointerException), ошибка сохранения, ошибка маппинга, ошибка отправки WebSocket. Проверен сценарий с null Chat (уведомление создается).
    - Протестированы методы getUnreadNotifications и getAllNotifications, включая случаи отсутствия уведомлений и ошибки репозитория/маппера.
    - Протестирован метод markNotificationAsRead: успешное обновление (проверка флага isRead в сохраняемой сущности, проверка аргументов sendToUser), случай, когда уведомление уже прочитано, случай,

когда уведомление не найдено, обработка ошибок сохранения/маппинга/отправки WebSocket.
    - Протестирован нереализованный метод `sendNotificationToUser` (проверка `UnsupportedOperationException`).
- **`UserServiceImpl`**:
    - Протестированы методы `findById` и `findDtoById`, включая случаи ненахождения пользователя и ошибки репозитория/маппера.
    - Протестирован метод `getAllUsers`: **подтверждено тестами, что метод использует `userRepository.findAll()` и игнорирует переданный `companyId`**, возвращая всех пользователей. Проверен случай отсутствия пользователей и ошибки репозитория/маппера.
    - Протестирован нереализованный метод `updateOnlineStatus` (проверка `UnsupportedOperationException`).
- **`SubscriptionServiceImpl`**:
    - Протестирован метод `subscribe`:
        - Сценарий нового пользователя: проверка вызова `companyService.createCompany` (с DTO с пустым именем и email), `userRepository.updateByCompanyIdAndEmail`, `roleService.addRole` (дважды), `subscribeDataMapper.toSubscription`, `subscriptionRepository.save`, `mapperDto.toSubscriptionDto`. Проверка аргументов через `ArgumentCaptor`, где это возможно.
        - Сценарий пользователя с существующей ролью: проверка выброса `AlreadyInCompanyException`.
        - (Отмечена необходимость тестов на ошибки зависимостей).
    - Протестирован метод `getSubscription`: успешный сценарий, случаи `UserNotFoundException` (пользователь не найден или без компании), `NotFoundSubscriptionException`, обработка ошибок репозиториев/маппера.
    - Протестирован метод `countPrice`: проверка делегирования вызова `SubscriptionPriceCalculateService`.
    - Протестированы методы `addOperatorCount` и `subtractOperatorCount`: успешное изменение счетчика (проверка через `ArgumentCaptor`), проверка выброса исключений при достижении лимита/минимума (`MaxOperatorsCountException`, `SubtractOperatorException`), проверка `NotFoundSubscriptionException`, обработка ошибок сохранения.
    - **Замечание:** Тесты для метода `cancel` не были реализованы, так как метод в текущем виде вызывает `disbandCompany`, отсутствующий в интерфейсе `CompanyService`.
- **`SubscriptionPriceCalculateServiceImpl`**:
    - Реализовано исчерпывающее тестирование с помощью параметризованных тестов (`@ParameterizedTest`, `@CsvSource`).
    - Проверены методы `calculateDiscountMonths` и `calculateDiscountPeople` на различных значениях, включая граничные (`<= 1`) и значения выше порога максимальной скидки (0.25), подтверждена линейная формула расчета.

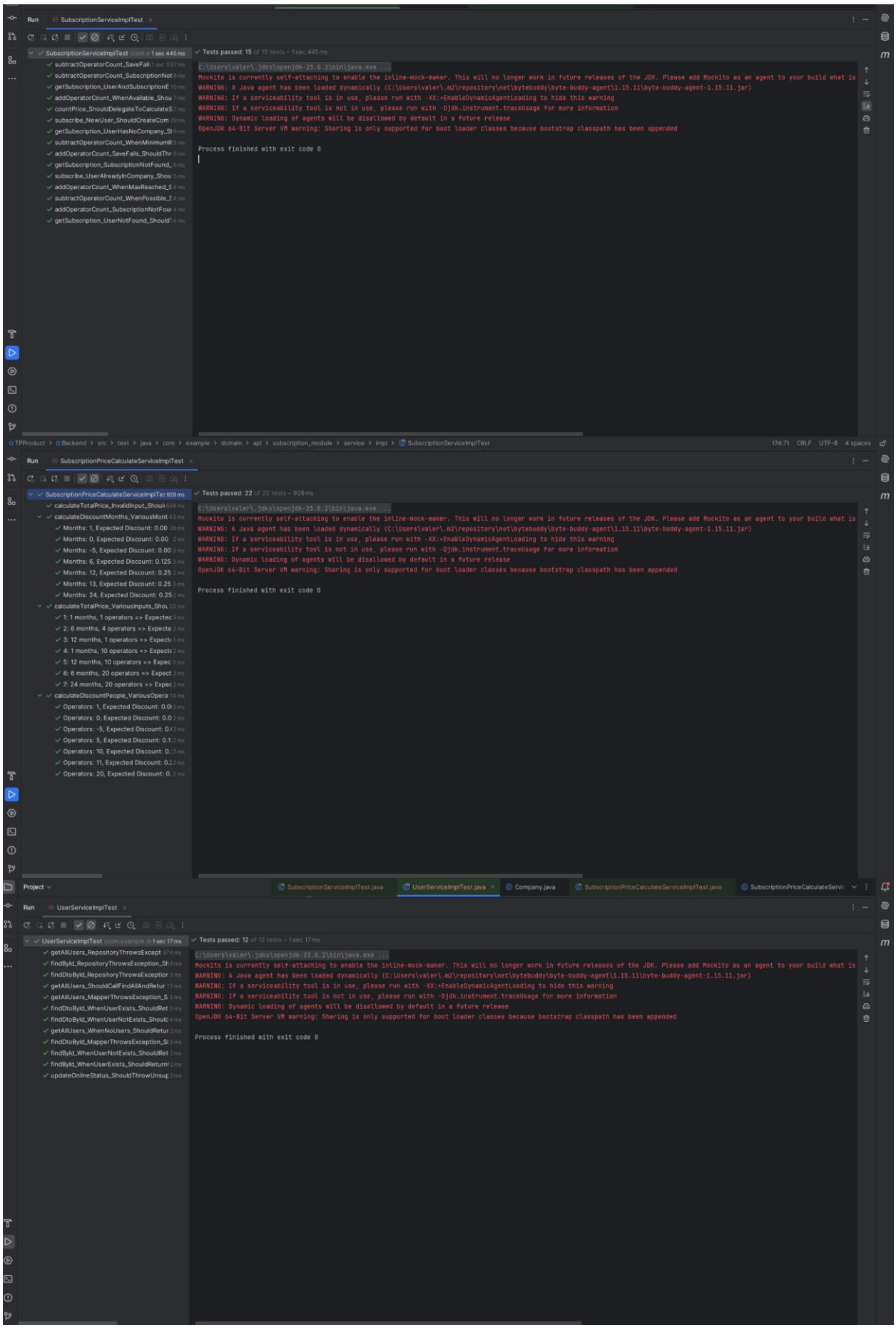
- Проверен метод `calculateTotalPrice` на различных комбинациях месяцев и операторов, включая случаи с нулевой, частичной и максимальной (50%) суммарной скидкой. Ожидаемые значения были рассчитаны согласно формуле сервиса.
- Протестирована обработка некорректных входных данных: `null` значения в DTO приводят к `NullPointerException`, значения `<= 0` обрабатываются без ошибок (приводят к нулевой скидке).

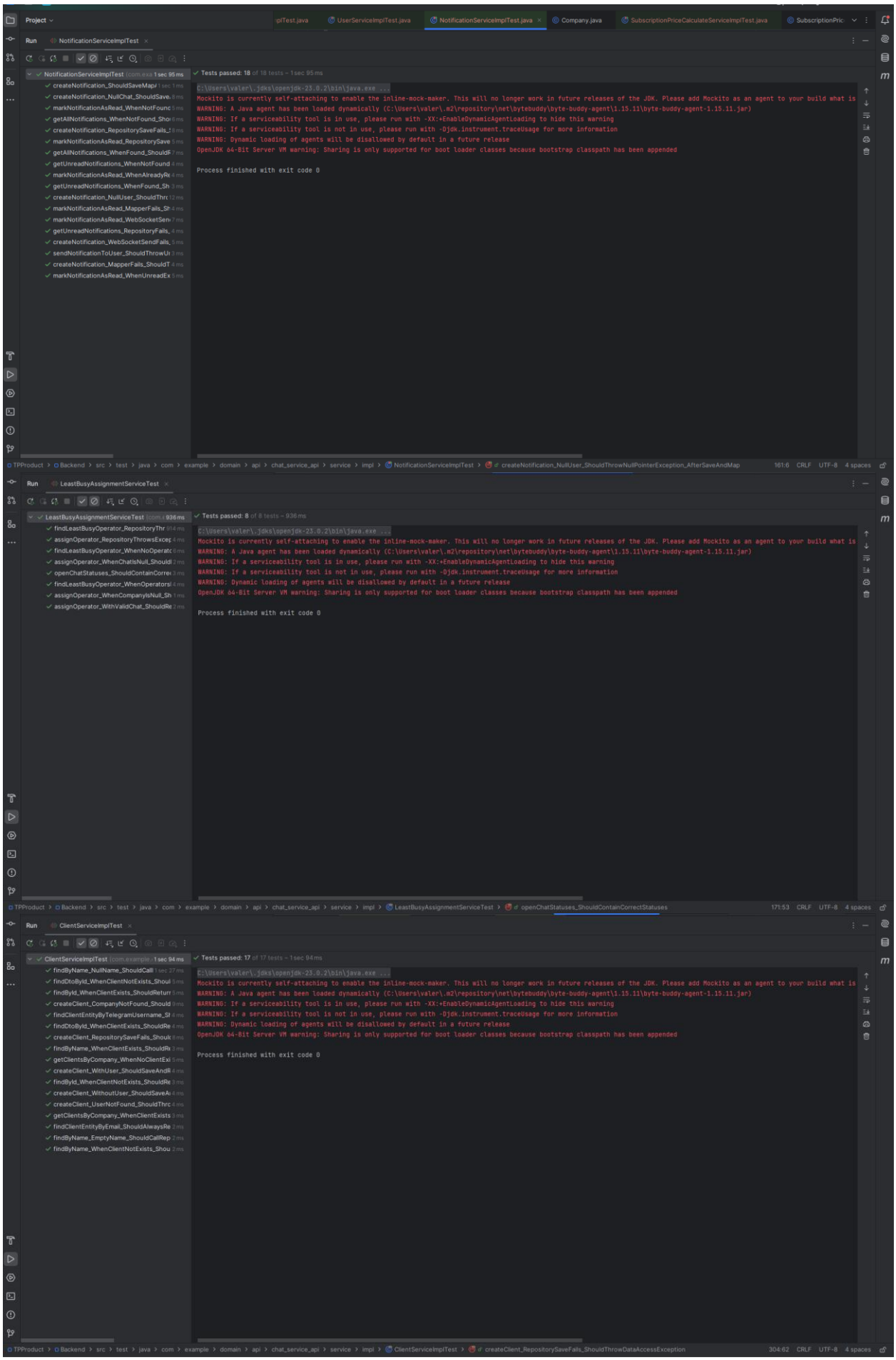
5. Результаты тестирования

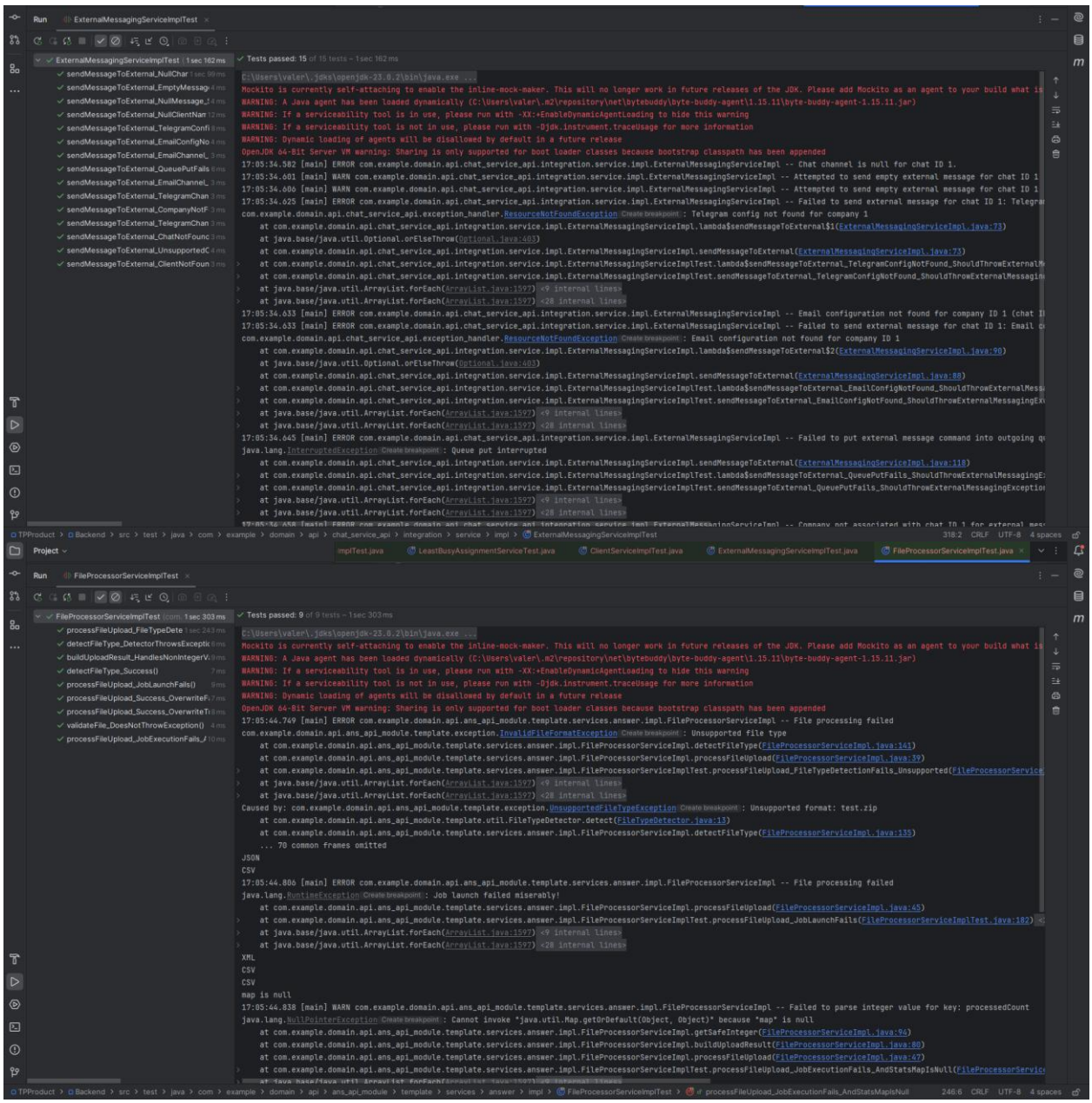
Все разработанные юнит-тесты для перечисленных сервисов успешно выполняются. Комплект тестов обеспечивает высокое покрытие кода и проверяет разнообразные сценарии использования и обработки ошибок.

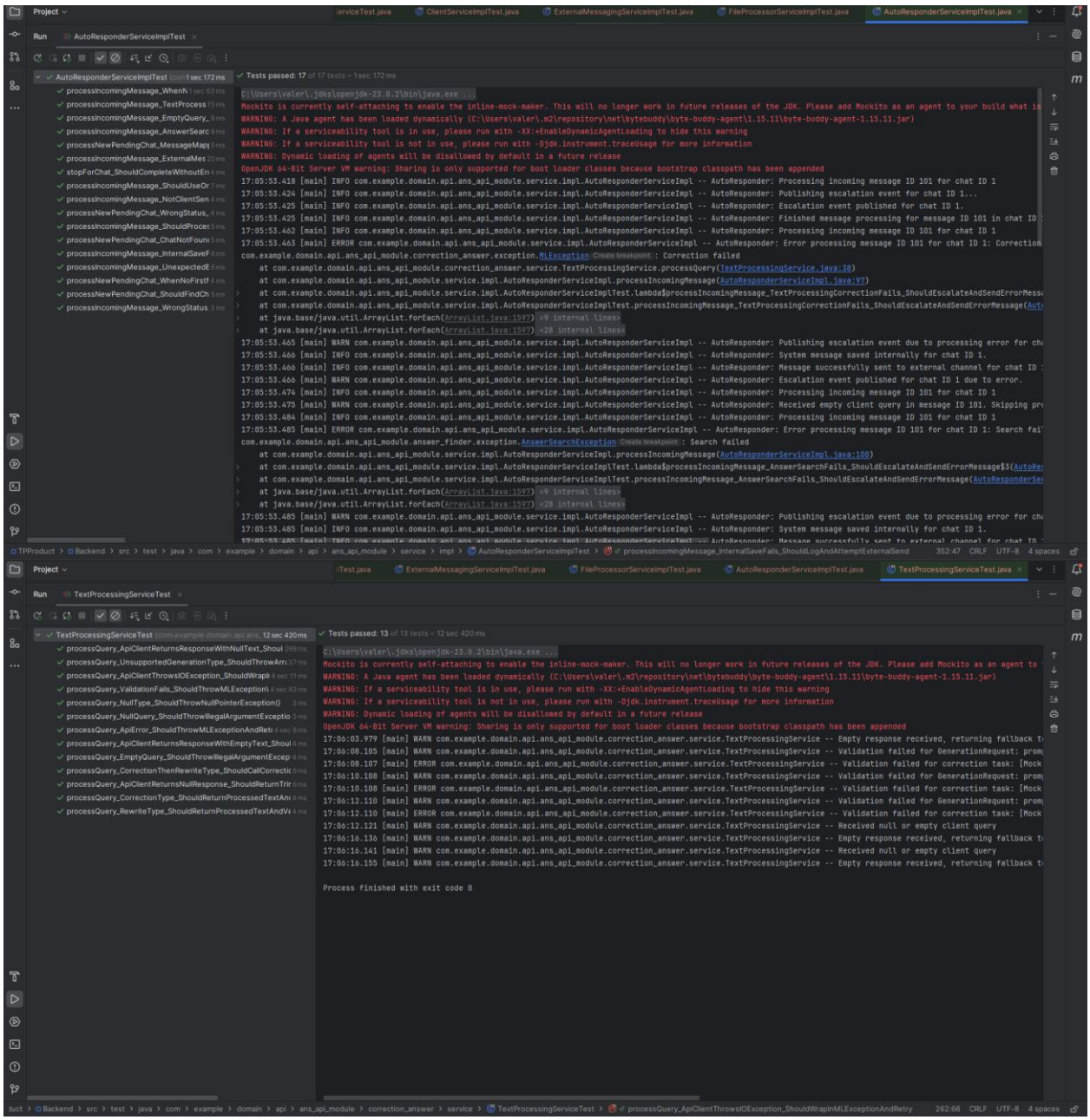
6. Скриншоты выполнения тестов

- **[Общий результат выполнения тестов]**


Run
SubscriptionServiceImplTest
Tests passed: 15 of 15 tests – 1 sec 445 ms
C:\Users\valer\.jdks\openjdk-23.0.2\bin\java.exe ...
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is
WARNING: A Java agent has been loaded dynamically (C:\Users\valer\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Process finished with exit code 0
Run
SubscriptionPriceCalculateServiceImplTest
Tests passed: 22 of 22 tests – 928 ms
C:\Users\valer\.jdks\openjdk-23.0.2\bin\java.exe ...
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is
WARNING: A Java agent has been loaded dynamically (C:\Users\valer\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Process finished with exit code 0
Project
SubscriptionServiceImplTest.java
UserServiceImplTest.java
Company.java
SubscriptionPriceCalculateServiceImplTest.java
Run
UserServiceImplTest
Tests passed: 12 of 12 tests – 1 sec 17 ms
C:\Users\valer\.jdks\openjdk-23.0.2\bin\java.exe ...
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is
WARNING: A Java agent has been loaded dynamically (C:\Users\valer\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Process finished with exit code 0


Project
NotificationServiceImplTest.java
UserServiceImplTest.java
NotificationServiceImplTest.java
Company.java
SubscriptionPriceCalculateServiceImplTest.java
Run
NotificationServiceImplTest
Tests passed: 18 of 18 tests – 1 sec 95 ms
C:\Users\valer\.jdks\openjdk-23.0.2\bin\java.exe ...
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is
WARNING: A Java agent has been loaded dynamically (C:\Users\valer\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Process finished with exit code 0
createNotification_ShouldSaveMap
createNotification_NullChat_ShouldSave
markNotificationAsRead_WhenNotFound
getAllNotifications_WhenNotFound_Sho
createNotification_RepositorySaveFails_
markNotificationAsRead_RepositorySave
getAllNotifications_WhenFound_ShouldR
getUnreadNotifications_WhenNotFound
markNotificationAsRead_WhenAlreadyRe
getUnreadNotifications_WhenFound_Sh
createNotification_NullUser_ShouldThr
markNotificationAsRead_MapperFails_Sh
markNotificationAsRead_WebSocketSen
getUnreadNotifications_RepositoryFails_
createNotification_WebSocketSendFails_
sendNotificationToUser_ShouldThrowUn
createNotification_MapperFails_ShouldT
markNotificationAsRead_WhenUnreadEx
TPProduct › Backend › src › test › java › com › example › domain › api › chat_service_api › service › impl › NotificationServiceImplTest › createNotification_NullUser_ShouldThrowNullPointerException_AfterSaveAndMap
161:6 CRLF UTF-8 4 spaces
Run
LeastBusyAssignmentServiceTest
Tests passed: 8 of 8 tests – 936 ms
C:\Users\valer\.jdks\openjdk-23.0.2\bin\java.exe ...
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is
WARNING: A Java agent has been loaded dynamically (C:\Users\valer\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Process finished with exit code 0
findLeastBusyOperator_RepositoryThr
assignOperator_RepositoryThrowsExcep
findLeastBusyOperator_WhenNoOperato
assignOperator_WhenChatIsNull_Should
openChatStatuses_ShouldContainCorre
findLeastBusyOperator_WhenOperatorsI
assignOperator_WhenCompanyIsNull_Sh
assignOperator_WithValidChat_ShouldRe
TPProduct › Backend › src › test › java › com › example › domain › api › chat_service_api › service › impl › LeastBusyAssignmentServiceTest › openChatStatuses_ShouldContainCorrectStatuses
171:53 CRLF UTF-8 4 spaces
Run
ClientServiceImplTest
Tests passed: 17 of 17 tests – 1 sec 94 ms
C:\Users\valer\.jdks\openjdk-23.0.2\bin\java.exe ...
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is
WARNING: A Java agent has been loaded dynamically (C:\Users\valer\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Process finished with exit code 0
findByName_NullName_ShouldCall
findDtoById_WhenClientNotExists_Shoul
findById_WhenClientExists_ShouldReturn
createClient_CompanyNotFound_Should
findClientEntityByTelegramUsername_Sh
findDtoById_WhenClientExists_ShouldRe
createClient_RepositorySaveFails_Shoul
findByName_WhenClientExists_ShouldRe
getClientsByCompany_WhenNoClientExi
createClient_WithUser_ShouldSaveAndR
findById_WhenClientNotExists_ShouldRe
createClient_WithoutUser_ShouldSaveA
createClient_UserNotFound_ShouldThro
getClientsByCompany_WhenClientExists
findClientEntityByEmail_ShouldAlwaysRe
findByName_EmptyName_ShouldCallRep
findByName_WhenClientNotExists_Shou
TPProduct › Backend › src › test › java › com › example › domain › api › chat_service_api › service › impl › ClientServiceImplTest › createClient_RepositorySaveFails_ShouldThrowDataAccessException
304:62 CRLF UTF-8 4 spaces


Run
ExternalMessagingServiceImplTest
Tests passed: 15 of 15 tests – 1 sec 162 ms
C:\Users\valer\.jdks\openjdk-23.0.2\bin\java.exe ...
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is
WARNING: A Java agent has been loaded dynamically (C:\Users\valer\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
17:05:34.582 [main] ERROR com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl -- Chat channel is null for chat ID 1.
17:05:34.601 [main] WARN com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl -- Attempted to send empty external message for chat ID 1
17:05:34.606 [main] WARN com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl -- Attempted to send empty external message for chat ID 1
17:05:34.625 [main] ERROR com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl -- Failed to send external message for chat ID 1: Telegra
com.example.domain.api.chat_service_api.exception_handler.ResourceNotFoundException Create breakpoint : Telegram config not found for company 1
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl.lambda$sendMessageToExternal$1(ExternalMessagingServiceImpl.java:73)
at java.base/java.util.Optional.orElseThrow(Optional.java:403)
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl.sendMessageToExternal(ExternalMessagingServiceImpl.java:73)
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImplTest.lambda$sendMessageToExternal_TelegramConfigNotFound_ShouldThrowExternalM
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImplTest.sendMessageToExternal_TelegramConfigNotFound_ShouldThrowExternalMessagin
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <9 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <28 internal lines>
17:05:34.633 [main] ERROR com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl -- Email configuration not found for company ID 1 (chat I
17:05:34.633 [main] ERROR com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl -- Failed to send external message for chat ID 1: Email c
com.example.domain.api.chat_service_api.exception_handler.ResourceNotFoundException Create breakpoint : Email configuration not found for company ID 1
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl.lambda$sendMessageToExternal$2(ExternalMessagingServiceImpl.java:90)
at java.base/java.util.Optional.orElseThrow(Optional.java:403)
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl.sendMessageToExternal(ExternalMessagingServiceImpl.java:88)
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImplTest.lambda$sendMessageToExternal_EmailConfigNotFound_ShouldThrowExternalMessa
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImplTest.sendMessageToExternal_EmailConfigNotFound_ShouldThrowExternalMessagingEx
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <9 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <28 internal lines>
17:05:34.645 [main] ERROR com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl -- Failed to put external message command into outgoing q
java.lang.InterruptedException Create breakpoint : Queue put interrupted
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImpl.sendMessageToExternal(ExternalMessagingServiceImpl.java:118)
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImplTest.lambda$sendMessageToExternal_QueuePutFails_ShouldThrowExternalMessagingE
at com.example.domain.api.chat_service_api.integration.service.impl.ExternalMessagingServiceImplTest.sendMessageToExternal_QueuePutFails_ShouldThrowExternalMessagingExceptio
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <9 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <28 internal lines>
TPProduct › Backend › src › test › java › com › example › domain › api › chat_service_api › integration › service › impl › ExternalMessagingServiceImplTest
318:2 CRLF UTF-8 4 spaces
Project
ClientServiceImplTest.java
ExternalMessagingServiceImplTest.java
FileProcessorServiceImplTest.java
Run
FileProcessorServiceImplTest
Tests passed: 9 of 9 tests – 1 sec 303 ms
C:\Users\valer\.jdks\openjdk-23.0.2\bin\java.exe ...
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is
WARNING: A Java agent has been loaded dynamically (C:\Users\valer\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
17:05:44.749 [main] ERROR com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl -- File processing failed
com.example.domain.api.ans_api_module.template.exception.InvalidFileFormatException Create breakpoint : Unsupported file type
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl.detectFileType(FileProcessorServiceImpl.java:141)
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl.processFileUpload(FileProcessorServiceImpl.java:39)
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImplTest.processFileUpload_FileTypeDetectionFails_Unsupported(FileProcessorServic
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <9 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <28 internal lines>
Caused by: com.example.domain.api.ans_api_module.template.exception.UnsupportedFileTypeException Create breakpoint : Unsupported format: test.zip
at com.example.domain.api.ans_api_module.template.util.FileTypeDetector.detect(FileTypeDetector.java:13)
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl.detectFileType(FileProcessorServiceImpl.java:135)
... 70 common frames omitted
JSON
CSV
17:05:44.806 [main] ERROR com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl -- File processing failed
java.lang.RuntimeException Create breakpoint : Job launch failed miserably!
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl.processFileUpload(FileProcessorServiceImpl.java:45)
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImplTest.processFileUpload_JobLaunchFails(FileProcessorServiceImplTest.java:182)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <9 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <28 internal lines>
XML
CSV
CSV
map is null
17:05:44.838 [main] WARN com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl -- Failed to parse integer value for key: processedCount
java.lang.NullPointerException Create breakpoint : Cannot invoke "java.util.Map.getOrDefault(Object, Object)" because "map" is null
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl.getSafeInteger(FileProcessorServiceImpl.java:96)
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl.buildUploadResult(FileProcessorServiceImpl.java:60)
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImpl.processFileUpload(FileProcessorServiceImpl.java:47)
at com.example.domain.api.ans_api_module.template.services.answer.impl.FileProcessorServiceImplTest.processFileUpload_JobExecutionFails_AndStatsMapIsNull(FileProcessorServic
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <9 internal lines>
TPProduct › Backend › src › test › java › com › example › domain › api › ans_api_module › template › services › answer › impl › FileProcessorServiceImplTest › processFileUpload_JobExecutionFails_AndStatsMapIsNull
246:6 CRLF UTF-8 4 spaces



7. Заключение

Разработанный набор юнит-тестов позволяет верифицировать корректность работы отдельных сервисов системы в изолированной среде. Успешное прохождение тестов свидетельствует о качестве и надежности реализованной бизнес-логики, а также корректной обработке граничных случаев и ошибок. Данный тестовый набор является важным инструментом для поддержки и дальнейшего развития проекта, обеспечивая возможность безопасного рефакторинга и внесения изменений.