

План функционального тестирования системы А + В

ООО "Тестовое Задание"

Версия 1.0

2025

Автор: Жуклин Никита Дмитриевич

План функционального тестирования системы А + В

Ссылки

Дата	Версия	Описание	Автор
2025-07-06	1.0	Создание и утверждение документа	Жуклин Н.Д.

Введение

Данный тест-план описывает функциональное тестирование взаимодействия микросервисов А и В, осуществляемого посредством обмена сообщениями через брокер RabbitMQ (протокол AMQP).

Цель тестирования — проверить корректность обработки команд на подключение, отключение и блокировку услуги, а также взаимодействие между сервисами через очередь сообщений (AMQP).

Сервис А принимает команды от клиента и отправляет команды в очередь, дожидаясь ответа от Сервиса В перед отправкой следующей команды.

Тестирование охватывает как позитивные сценарии, так и негативные — включая проблемы сериализации, недоступность очереди, ошибки формата и потерю сообщений.

Объекты тестирования

- REST API интерфейс Сервиса А (подключение, отключение, блокировка услуги)
- Взаимодействие Сервиса А и Сервиса В через RabbitMQ
- Формат AMQP сообщений и соответствие содержимого
- Механизм сериализации: А не отправляет новую команду до получения ответа от В
- Обновление статуса услуги в Сервисе А на основании ответов от В

Проблемы и риски

- Возможная потеря сообщений в очереди AMQP
- Непредсказуемые таймауты при ожидании ответа от сервиса В
- Невозможность протестировать негативные сценарии без контроля над сервисом В
- Ошибки сериализации/десериализации JSON
- Задержки или тайм-ауты при доставке сообщений через RabbitMQ

План функционального тестирования системы А + В

Функции, которые нужно протестировать

- Обработка команд: /connect, /disconnect, /block
- Очередность команд и ожидание ответа от В
- Обработка таймаута или недоступности В
- Обновление статуса услуги на основании подтверждения
- Проверка повторной отправки при сбое
- Проверка логирования

Функции, которые НЕ нужно тестировать

- Авторизация REST API
- Интерфейс клиента
- Визуализация статуса услуги
- Нагрузочное тестирование очереди
- Тестирование внутренней логики RabbitMQ

Подходы

- Функциональное тестирование на уровне API и обмена сообщениями
- Использование заглушек для имитации ответов от Сервиса В
- Нагрузочное моделирование последовательных вызовов
- Негативное тестирование с обрывом связи, пустым ответом, некорректным сообщением
- Проверка времени ответа и сериализации (следующая команда отправляется только после подтверждения)

Критерии прохождения тестов

- Все команды возвращают корректные HTTP-ответы
- Ответы от Сервиса В корректно обрабатываются, и статус обновляется
- При ошибках взаимодействия с В статус не меняется
- Все команды выполняются строго последовательно
- Структура всех сообщений соответствует документации

План функционального тестирования системы А + В

- В логах отражены все этапы запроса и ответа

Критерии начала тестирования

- Деплой обоих сервисов А и В
- Доступ к тестовому REST API Сервиса А
- Документация на структуру AMQP-сообщений
- Возможность мониторинга сообщений между А и В
- Доступ к журналам событий

Критерии окончания тестирования

- Все запланированные тест-кейсы выполнены
- Все критические и блокирующие баги устранены
- Все бизнес-функции протестированы и подтверждены логами
- Подготовлен итоговый отчет

Критерии остановки и требования для возобновления тестирования

Остановка тестирования:

- Потеря связи с RabbitMQ или сбой инфраструктуры
- Критические ошибки в логике обработки команд
- Отсутствие документации или ключевых интерфейсов

Возобновление тестирования:

- Восстановление доступности систем и среды
- Предоставление спецификаций и схем обмена сообщениями
- Обновление окружения и тестов

Результаты тестирования

- Протоколы тестов с логами
- Список всех отправленных и полученных сообщений
- Сводка по отклонённым и успешным командам

План функционального тестирования системы А + В

- Зарегистрированные баги и статус их обработки

Оставшиеся задачи тестирования

- Интеграция с другими микросервисами
- Долговременное тестирование обработки очереди
- Проверка отказоустойчивости

Требования среды

- Стабильный деплой сервисов А и В
- Тестовый экземпляр AMQP RabbitMQ
- Набор заглушек для имитации ответов от Сервиса В
- Инструменты: Postman, RabbitMQ, система CI (Jenkins/GitLab CI)

Требования по части кадров и их обучения

- Знание REST API и HTTP-кодов
- Навыки работы с брокерами сообщений (RabbitMQ)
- Ознакомление с бизнес-логикой проекта
- Умение читать и анализировать логи сервисов

Обязанности

Имя	Должность	Обязанности
Жуклин Н.Д.	Руководитель тестирования	Планирование, контроль, итоговый отчёт
Тестировщик 1	Тестировщик	Тестирование REST API и логики команд
Тестировщик 2	Тестировщик	Тестирование взаимодействия с RabbitMQ
Автоматизатор 1	Автоматизатор	Автоматизация тестов, интеграция в CI/CD

Расписание

Этап	Дата начала	Дата окончания	Комментарии
Подготовка инфраструктуры	2025-07-10	2025-07-11	Настройка очередей и сервисов
Разработка тест-кейсов	2025-07-12	2025-07-13	Сценарии + тестовые события

План функционального тестирования системы А + В

Выполнение тестирования	2025-07-14	2025-07-16	Ручное, автоматизированное тестирование
Анализ результатов	2025-07-13	2025-07-14	Формирование отчётов
Закрытие проекта	2025-07-15	2025-07-15	Согласование результатов

Планирование рисков и непредвиденных обстоятельств

- При потере связи с Сервисом В — использовать мок-сервер или заглушку
- При блокировке AMQP очереди — очистка и перезапуск брокера
- При кадровых рисках — распределение задач между тестировщиками
- При изменениях требований — корректировка тест-кейсов и документации

Утверждение

Документ утвержден руководителем тестирования: Жуклиным Никитой Дмитриевичем.

Дата утверждения: 09-07-2025

Глоссарий

RabbitMQ — брокер сообщений, реализующий шаблон pub/sub и очереди

AMQP — протокол обмена сообщениями между сервисами

REST API — интерфейс взаимодействия клиента и сервера по HTTP

Команды — операции /connect, /disconnect, /block

Сервис А — отправитель AMQP-сообщений и REST API

Сервис В — исполнитель команд и получатель AMQP

Mock — эмулятор очереди, используемый в тестировании