Rocket.Chat

Тестовая стратегия

Версия 1.0

История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 12.07.2023 | 1.0 | Создание тестовой стратегии | Попкова Ю.В. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Оглавление

# Введение 4

# 2. Типы проводимого тестирования 5

2.1 Функциональное тестирование 5

2.2 UI тестирование 5

2.3 Тестирование удобства пользования 6

* 1. Тестирование производительности 6

2.5 Тестирование безопасности 7

2.6 Тестирование совместимости 8

2.7 Тестирование доступности 8

## 2.8 Конфигурационное 8

# 3. Описание модулей, протестированные в Rocket.Chat 9

## 3.1 Форма авторизации и Учетная запись пользователя 9

## 3.2 Главная страница 9

## 3.3 Поиск 9

## 3.4 Каталог 9

## 3.5 Внешний вид списка чатов 9

## 3.6 Создать 9

## 3.7 Панель администрирования 9

## 3.8 Приложения 10

## 3.9 Аудит 10

1. Окружение для работы 11
2. Виды тестовой документации 11
3. Время проведения тестирования 12
4. **Введение**

**Rocket.Chat** — корпоративная платформа обмена сообщениями, которая позволяет командам общаться и сотрудничать в режиме реального времени.

Это платформа с открытым исходным кодом. Благодаря этому мессенджер может быть настроен в соответствии с потребностями различных компаний, с возможностью локального, облачного или гибридного развертывания.

Особенности Rocket.Chat:

# Легкий обмен файлами

# Поддержка обмена аудиофайлами

# Видеоконференцсвязь (например, с помощью интеграции приложения [Jit.si](http://Jit.si))

# Отдельные каналы (частный и публичный варианты)

# Поддержка сквозного шифрования

# Неограниченная история сообщений

# Поддержка интеграции нескольких сторонних приложений

# Поддержка push-уведомлений и тд

Преимущества Rocket.Chat:

# Возможность установки Rocket.Chat на собственные серверы, что делает его бесплатным и дает инструменты для тонкой настройки.

# Конфиденциальность. Rocket.Chat не предоставляет данные и сообщения третьим лицам и не хранит удаленные сообщения.

# Защищенность. E2E шифрование, сертифицирован ISO 27001 и соответствует GDPR, CCPA, LGPD и HIPAA.

# Открытость Rocket.Chat. Будучи open source проектом, Rocket.Chat позволяет самостоятельно исправлять ошибки и публиковать обновления.

# Поддержка всех базовых функций. Rocket.Chat обеспечивает стабильную работу всех основных функций, используемых мессенджерами. Этим закрываются потребности подавляющего большинства пользователей.

# Возможность модификации для расширения функционала.

# Постоянное развитие. Rocket.Chat регулярно обновляется, становясь все удобнее и приятнее в использовании.

Состав команды: sunnycro

Работа с предыдущими проектами:

# Функциональное тестирование СберМаркета и TestIT

# Тестирование интерфейса СберСтрахования

# Тестирование мобильного приложения tutu

Работа с программами и приложениями:

* Test IT
* Notion
* Postman
* DevTools
* прошла обучение Linux-командам и SQL запросов.

1. **Типы проводимого тестирования**

2.1 Функциональное тестирование

Функциональное тестирование — это процесс обеспечения качества (QA) в рамках цикла разработки программного обеспечения, необходимый для проверки реализуемости функциональных требований, согласно спецификации тестируемого программного обеспечения. Функциональное тестирование проводится для оценки соответствия системы или компонента заданным функциональным требованиям.

Такое тестирование проводится по принципу черного ящика, в связи с чем функциональность ПО можно протестировать, не зная принципа его внутренней работы. Это снижает требования к тестировщикам в части знания языков программирования или конкретных аспектов реализации программного обеспечения.

2.2 UI тестирование

UI-тестирование — это этап в процессе тестирования ПО, проверяющий интерфейс. Тестирование интерфейса помогает проверить функции приложения, имитируя действия пользователей, т.е. в процессе тестирования выполняется проверка элементов, проверка на их корректность с помощью ввода данных в приложение через устройства ввода или средств автоматизированного UI-тестирования. Это можно делать руками или с помощью автоматизации.

2.3 Тестирование удобства пользования

Тестирование удобства пользования — это нефункциональный вид тестирования программного обеспечения, являющийся подмножеством тестирования пользовательского опыта - UX. В целом оно подразделяется на понятность, обучаемость, работоспособность, привлекательность и соответствие. Такое тестирование предназначено для определения того, насколько программный продукт понятен, легок в освоении, прост в эксплуатации и привлекателен для пользователей при определенных условиях и требованиях. Этот тип тестирования обычно выполняется реальными пользователями.

2.4 Тестирование производительности

Тестирование производительности — это нефункциональный вид тестирования программного обеспечения, используемый для проверки скорости, времени отклика, стабильности, надежности, масштабируемости и использования ресурсов программного приложения при определенной рабочей нагрузке, обычно регрессионным образом, когда в приложение ежедневно или еженедельно вносятся небольшие изменения. Основная цель тестирования производительности - выявить и устранить узкие места производительности в программном приложении.

Типы тестирования производительности:

2.4.1. Стресс тестирование

Этот тест проводится первым. Нагрузка постепенно увеличивается до тех пор, пока приложение не перестанет работать корректно. В конце теста фиксируется количество пользователей, которое приложение выдерживало, соответствуя требованиям производительности, и сколько выдержать не смогло. Первое значение и будет пределом производительности вашего приложения. Часто этот вид тестирования проводится, если заказчик предвидит резкое увеличение нагрузки на систему.

2.4.2. Нагрузочный тест

Нагрузка на систему подается на протяжении 4-8 часов. В это время собираются метрики производительности: количество запросов в секунду, транзакций в секунду, время отклика от сервера, процент ошибок в ответах, утилизация аппаратных ресурсов и т д. Собранные метрики проходят проверку на соответствие заданным требованиям.

Также в результате получаем локализацию узких мест в производительности приложения и дефектов, подробное профилирование всех компонентов системы и утилизацию аппаратных ресурсов под целевой нагрузкой.

2.4.3. Тестирование на больших объемах данных

Данный вид тестирования помогает сделать прогноз относительно работоспособности приложения. Форма подаваемой нагрузки та же, что и при нагрузочном тестировании. Задача теста – узнать, какое влияние окажет увеличение объема данных на систему.

2.4.4. Тестирование отказоустойчивости

Продолжительность нагрузки может варьироваться в зависимости от целей и возможностей проекта, доходя до семи дней и более. В результате получаем представление о том, как изменится производительность системы в течение длительного периода времени под нагрузкой, например, в течение недели. Снизится ли уровень производительности? Способно ли приложение выдерживать стабильную нагрузку без критических сбоев?

2.4.5. Тестирование масштабирования

Профиль нагрузки тот же, что и при нагрузочном тестировании. В результате, получаем ответы на следующие вопросы:

* Увеличится ли производительность приложения, если добавить дополнительные аппаратные ресурсы?
* Увеличится ли производительность пропорционально количеству добавленных аппаратных средств?

2.5 Тестирование безопасности

Это тип тестирования ПО, который выявляет уязвимости, угрозы и риски. Целью тестов безопасности является выявление всех возможных лазеек и слабых мест в ПО, которые могут привести к потере информации, доходов, репутации компании, сотрудников или клиентов. Общая стратегия безопасности основывается на трех основных принципах:

* Конфиденциальность - сокрытие определенных ресурсов или информации;
* Целостность - ресурс может быть изменен только в соответствии с полномочиями пользователя;
* Доступность - ресурсы должны быть доступны только авторизованному пользователю, внутреннему объекту или устройству;

2.6 Тестирование совместимости

Тестирование совместимости измеряет степень того, насколько удовлетворительно элемент тестирования может функционировать параллельно с другими независимыми продуктами в общей среде (сосуществование) и, по мере необходимости, обменивается информацией с другими системами или компонентами (функциональная совместимость).

2.7 Тестирование доступности

Тестирование доступности — это подмножество тестирования удобства. Его цель - убедиться в том, что продукт удобен в использовании людям с различными видами ограничений, инвалидности или особенностями восприятия. Это могут быть проблемы со зрением, слухом или ограничения в подвижности рук.

2.8 Конфигурационное

Конфигурационное тестирование - специальный вид тестирования, направленный на проверку работы ПО при различных аппаратных и программных конфигурациях системы (заявленных платформах, поддерживаемых драйверах, при различных конфигурациях компьютеров и т. д.).

1. **Описание модулей, протестированные в Rocket.Chat**

3.1 Форма авторизации и Учетная запись пользователя

Проверка, редактирование, обновление информации в профиле на панели пользователя, настройки интерфейса и безопасностью.

3.2 Главная страница

Просмотр документации, переход и установка мобильного и веб-приложений, переход в разделы «Пользователи» и «Комнаты», создание канала.

3.3 Поиск

Работа с поиском каналов и пользователей.

3.4 Каталог

Просмотр существующих каналов, команд и пользователей.

3.5 Внешний вид списка чатов

Настройка внешнего вида списков чатов.

3.6 Создать

3.6.1. Создание каналов, команд, обсуждений и личных переписок

3.6.2. Настройка параметров каналов, команд, обсуждений и личных переписок

3.6.3. Отправка, редактирование и удаление текстовых и аудио-сообщений, файлов и смайлов

3.6.4. Создание звонков

3.6.5. Цитирование, добавление реакции, переотправка сообщений, переход в треды и обсуждения, профиля пользователя

* 1. Панель администрирования:

3.7.1. Информация: основная информация о рабочем пространстве.

3.7.2. Импорт: импорт данных из других сервисов в рабочем пространстве.

3.7.3. Пользователи: управление учетными записями пользователей.

3.7.4. Комнаты: создание комнат и управление ими.

3.7.5. Приглашения: просмотр и удаление ссылок на приглашения, созданные для приглашения пользователей в команду/каналы.

3.7.6. Регистрация: регистрация дает доступ к подключению собственного рабочего места к облаку Rocket.Chat.

3.7.7. Просмотр журналов: проверка журналов рабочей области.

3.7.8. Панель управления федерацией: проверка статистики о подключении федерации на сервере.

3.7.9. Ящики электронной почты: управление почтовыми ящиками или учетными записями электронной почты для каналов электронной почты.

3.7.10. Пользовательские эмодзи: настройка и управление пользовательскими эмодзи в рабочем пространстве.

3.7.11. Интеграция: интеграция инструментов и услуг для облегчения рабочих процессов.

3.7.12. Приложения OAuth: настройка пользовательских приложений для открытой аутентификации.

3.7.13. Отправка почты: отправка электронной почты пользователям в рабочей области.

3.7.14. Статус пользователя: настройка статусов пользователей в рабочей области.

3.7.15. Разрешения: выдача разрешений пользователям на выполнение определенных действий.

3.7.16. Настройки: настройки рабочего пространства.

3.7.17. Панель взаимодействия: просмотр деятельность пользователей на каналах рабочего пространства.

3.7.18. Управление устройствами: управление устройствами, связанные рабочим пространством.

3.8 Приложения

Каталог приложений, которые устанавливаются и настраиваются под рабочее пространство.

3.9 Аудит

Просмотр и выгрузка сообщений.

1. **Окружение для работы**

macOS версия 12.1

Rocket.Chat версия 6.2.8

Safari версия 15.2

Google Crome версия 114.0.5735.198

1. **Виды тестовой документации**

Для тестирования Rocket.Chat использовались следующие методы:

5.1. Тест-план — это документ, описывающий цели, подходы, ресурсы и график запланированных тестовых активностей.

5.2. Отчет о тестировании - периодический отчет, в котором документируется подробная информация о выполнении теста и его результате.

5.3. Тест-кейсы — это чёткое описание входных данных, условий и процедуры тестирования, а также ожидаемых результатов.

5.4. Чек-лист — это список проверок для тестирования продукта.

5.5. Баг-репорт — это документ, содержащий информацию о найденном баге.

5.6. Сценарии тестирования — это подробные описания или записи о том, как пользователь будет взаимодействовать с приложением во время тестирования программного обеспечения.

Проводились следующие техники тест-дизайна:

* Попарное тестирование
* Прогнозирование ошибок
* Классы эквивалентности
* Таблица принятия решений

1. **Время проведения тестирования**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Вид работы** | **Начало** | **Окончание** |
| 1 | Создание тест-плана, сценариев и чек-листов | 12.07.2023 | 15.07.2023 |
| 2 | Проведение функционального тестирования, фиксация результатов | 15.07.2023 | 16.07.2023 |
| 3 | Проведение тестирования безопасности, фиксация результатов | 16.07.2023 | 17.07.2023 |
| 4 | Проведение тестирования производительности, фиксация результатов | 17.07.2023 | 18.07.2023 |
| 5 | Проведение UI-тестирования, фиксация результатов | 18.07.2023 | 19.07.2023 |
| 6 | Проведение тестирования удобства использования, фиксация результатов | 19.07.2023 | 20.07.2023 |
| 7 | Оформление отчета о тестировании | 20.07.2023 | 22.07.2023 |