



Санкт-Петербургский государственный университет
Кафедра системного программирования

Разработка серверной части платформы для управления автоматизациями

Александр Андреевич Федькин, группа 20.Б11-мм

Научный руководитель: С.Ю. Сартасов, старший преподаватель кафедры системного программирования
Консультант: А.В. Полозенко, руководитель группы в ООО «Яндекс.Технологии»

Санкт-Петербург
2023

- Автоматизация (automation) — автоматизированный процесс, который сокращает вмешательство человека в процессы получения, преобразования и передачи информации
- Примеры:
 - ▶ Системы восстановления пароля
 - ▶ Автоматическая рассылка почты
 - ▶ Чат-боты

- Текущие автоматизации в компании сложно поддерживать:
 - ▶ Конфигурации чат-ботов поддержки (YAML файлы)
 - ▶ Рассылки писем (groovy скрипты)
 - ▶ Маршрутизация обращений пользователей (groovy скрипты)
- Существующие решения для управления автоматизациями не обладают всей необходимой функциональностью одновременно:
 - ▶ Нет возможности реализовать необходимые автоматизации в рамках одного инструмента
- Возникает необходимость создания нового инструмента

Целью данной производственной практики является разработка серверной части платформы для управления автоматизациями

Задачи:

- Сформулировать требования к создаваемому продукту
- Произвести обзор существующих инструментов для управления автоматизациями
- Спроектировать архитектуру системы
- Реализовать компоненты системы
- Произвести апробацию системы

- Создание, редактирование, версионирование автоматизаций
- Возможность наличия циклов в графах автоматизаций
- Возможность приостановить автоматизацию и ожидать сообщение
- Настройка условий автозапуска (триггеров)
- Возможность создавать автоматизации без опыта в программировании

Нефункциональные требования

- Язык разработки — Java или Kotlin
- Кеширование графов
- Неблокирующее взаимодействие с сервисами по HTTP
- Слабая связность компонентов платформы
- Легкая расширяемость

Критерий	Nirvana	Airflow	NiFi	PLynx
Версионирование графов	+	±	+	-
Цикличность графов	-	-	-	-
Ожидание сообщения	-	-	-	-
Необходимость опыта в программировании	±	+	±	+

Таблица 1: Сравнение существующих решений

Используемые технологии:

- Основа системы:
 - ▶ Kotlin
 - ▶ Spring boot
 - ▶ kotlinx.coroutines
- Хранилище данных:
 - ▶ Redis + Redisson
 - ▶ PostgreSQL + jOOQ + Flyway
- Документация:
 - ▶ Swagger

Архитектура

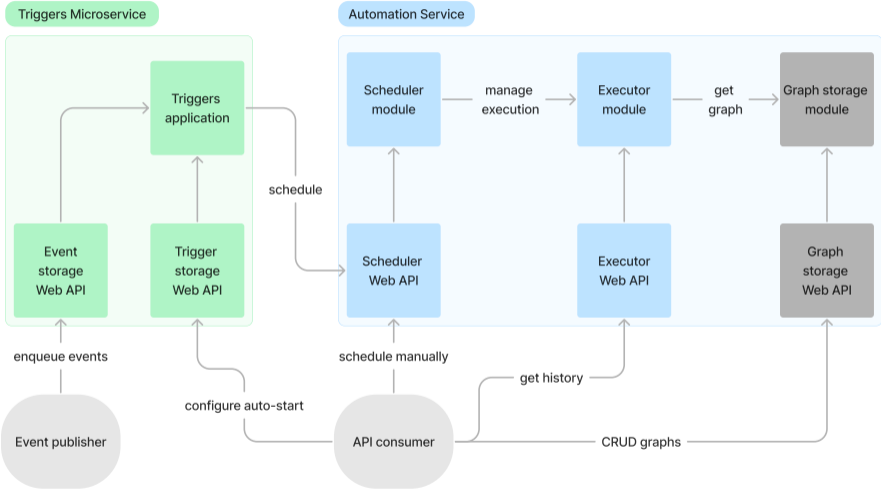


Рис. 1: Диаграмма коммуникации компонентов

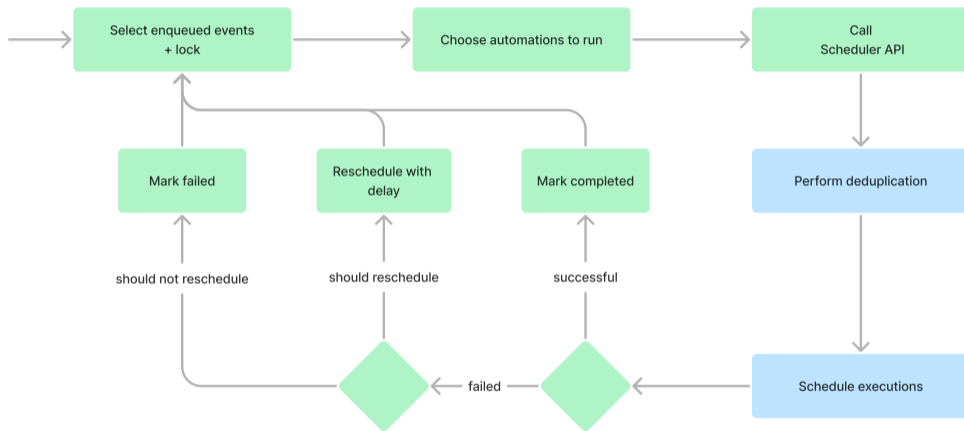


Рис. 2: Принцип работы микросеви́са триггеров

Реализация

- read-modify-write производится с оптимистической блокировкой
update executions set ... where id = ... and revision = ...
- acquired – исполнение назначено на конкретный экземпляр приложения
- released – исполнение не назначено

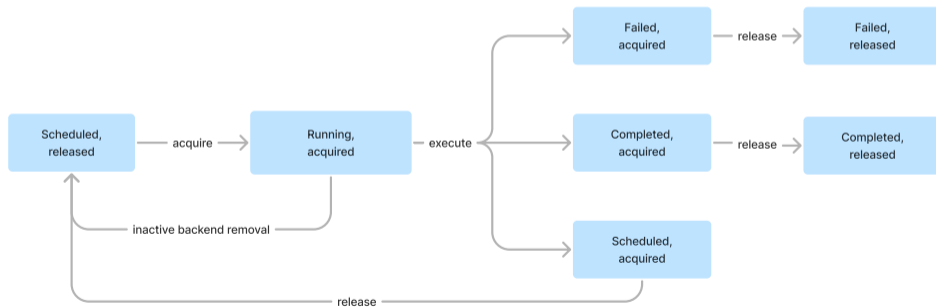


Рис. 3: Жизненный цикл исполнения

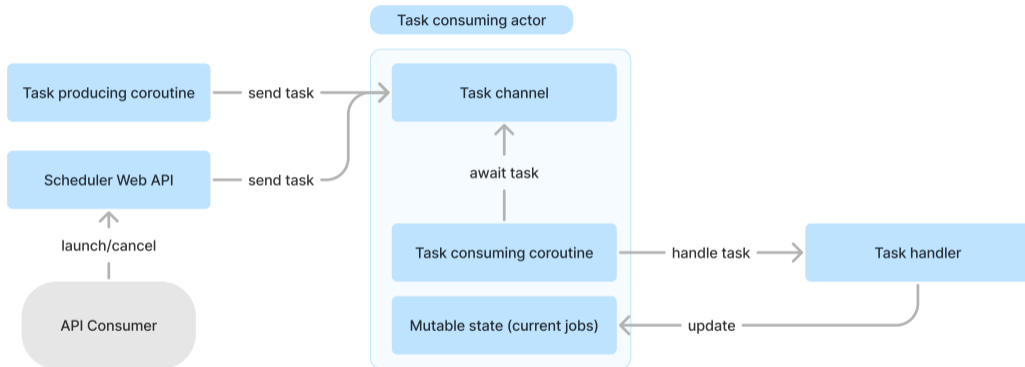


Рис. 4: Принцип работы модуля планирования

- Настроены процессы CI/CD
- Настроены окружения для тестирования и промышленной эксплуатации
- Настроены системы мониторинга, оповещения о нештатных ситуациях
- Активный перенос существующих автоматизаций:
 - ▶ Рассылка почты
 - ▶ Маршрутизация обращений

В ходе работы были выполнены следующие задачи:

- Сформулированы требования к создаваемому продукту
- Произведен обзор существующих инструментов для управления автоматизациями
- Спроектирована архитектура системы
- Реализованы компоненты системы
- Произведена апробация системы