

## РАЗРАБОТКА РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ОБМЕНА УВЕДОМЛЕНИЯМИ НА ОСНОВЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ

К.Н. Цебренько, канд. техн. наук, доцент

Академия маркетинга и социально-информационных технологий – ИМСИТ  
(Россия, г. Краснодар)

DOI:10.24412/2500-1000-2021-8-1-119-122

**Аннотация.** Цель работы состоит в разработке распределённой системы для отправки уведомлений. В ходе выполнения работы разработаны алгоритмы работы микросервисов распределенной системы. Выполнена реализация микросервисов и их тестирование. Оценена эффективность работы системы в условиях предприятия.

**Ключевые слова:** обмен уведомлениями, микросервисы, распределенная система, Golang, алгоритм, реализация.

Актуальность темы обуславливается необходимостью автоматизировать рутинные действия по рассылке email и sms-уведомлений пользователям [1]. На данный момент каждый сервис самостоятельно отправляет уведомления, что приводит к задержкам в работе пользователя и усложняет обслуживание рассылок администраторам, так как настраивать контент уведомлений приходится отдельно для каждого сервиса. В работе [2] разработана концепция распределённой системы для отправки уведомлений на основе микросервисной архитектуры распределенной системы. Предложена общая схема работы распределённой системы. Обоснован выбор технологий и средств реализации разработки.

Цель работы состоит в разработке распределённой системы для отправки уведомлений на языке Golang. Задачи решаемые в ходе выполнения работы: разработка алгоритмов работы программ, реализация разработки, анализ эффективности разработки. Предметом исследования является разработка распределённой системы отправки уведомлений на языке Go.

В системе необходимо реализовать сле-

дующие микросервисы: scheduler, builder, email-sender, sms-sender [2].

В процессе проектирования микросервисов необходимо разработать модели работы их программной части [3]. Можно использовать высокоуровневое моделирование как в работе [4] и построить гибридную модель модулей подсистемы [5]. Однако, для получения модели сложной подсистемы [6] с точностью до действия удобнее воспользоваться низкоуровневыми методами моделирования. Для составления блок-схем, отражающих работу разрабатываемых алгоритмов, использованы инструменты для создания диаграмм и блок-схем (Microsoft Visio).

Как было сказано выше, scheduler – это основная точка входа в систему и маршрутизатор сообщений по очередям других микросервисов. Он консолидирует всю информацию о уведомлениях, решает какому микросервису дать обработку уведомления в определённом состоянии и записывает информацию о обработке уведомлений в базу данных.

На рисунке 1 представлена блок схема алгоритма работы микросервиса scheduler.

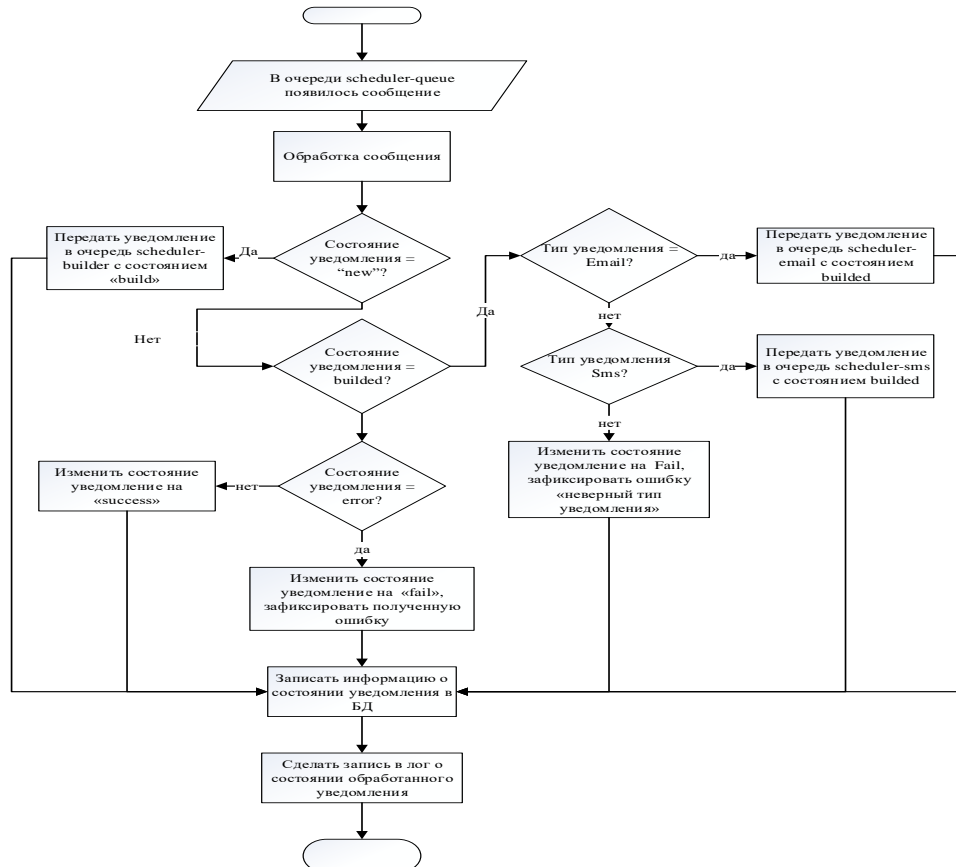


Рис. 1. Блок-схема работы микросервиса scheduler

Микросервис builder предназначен для сборки уведомления из готовых шаблонов, путём замены специальных маркеров на передаваемые данные. Шаблоны email отличаются от версии для sms наличием HTML-верстки, тогда как для sms используется обычный текст, соответственно для разных типов уведомлений используются разные файлы шаблонов, было принято добавлять в конце имени sms шаблона суффикс sms.

Передаваемые параметры сериализуются в тип данных Json, если при десериализации параметров произойдёт ошибка, микросервис вернёт уведомление обратно в очередь scheduler с состоянием "fail".

При успешной сборке шаблона, он будет передан в очередь scheduler с состоянием "built".

Микросервис email-sender предназначен для отправки собранного уведомления электронной почты в smtp-шлюз. В роли шлюза можно использовать SMTP-сервер, бесплатно предоставляемый компанией Яндекс, зарекомендовавший себя как ста-

бильный и довольно производительный сервер отправки электронной почты. Микросервис производит соединение с smtp-сервером в зашифрованном TLS-режиме и отправляет в него подготовленное уведомление

Каждое действие микросервиса записывается в лог файл, при ошибках уведомление возвращается в очередь scheduler-queue в состоянии "fail", при успехе в состоянии "success".

Микросервис sms-sender предназначен для отправки собранного sms уведомления в http-шлюз. В роли шлюза использован российский сервис отправки смс сообщений SMSRU.

Каждое действие микросервиса записывается в лог файл, при ошибках уведомление возвращается в очередь scheduler-queue в состоянии "fail", при успешной обработке в состоянии "success".

При реализации алгоритмов использован компилятор языка go 1.15 с официального сайта проекта <https://golang.org/doc/install#download>. Для

создания микросервисов системы, реализован способ конфигурирования запускаемых приложений. Для этого разработан пакет с именем `config`, в котором реализованы функции конфигурирования микросервисов. Для определения и хранения конфигурационных определены переменными окружения, которые хранятся в `env`-файле. В корне приложения файл с именем «`env`» содержит все необходимые параметры: параметры подключения к базе данных, параметры для соединения с брокером `rabbitmq`, имена очередей, необходимых для работы микросервису, параметры соединения с `smtp`-сервером, параметры соединения с сервисом отправки `sms`. На серверах предприятия данный файл наполняется автоматически из хранилищ в соответствии с запускаемым окружением.

После реализации исходного кода описанных сервисов выполнена разработка автоматических тестов. Язык `Go` содержит в себе встроенный пакет и набор утилит

для автоматических тестов. Чтобы разработать тест функций программы, необходимо создать файл с именем тестируемого пакета и суффиксом «`_test`». Код этих файлов не используется при сборке приложения, тесты запускаются по команде `go test`. Исходный код автоматических тестов позволил выполнить проверку работоспособности приложения.

В работе реализована разработка микросервисов, составляющих распределённую систему отправки уведомлений. Проведён анализ эффективности внедрения результата работы на примере предприятия. Система решает задачи по отправке рассылок уведомлений в виде электронной почты и `sms` сообщений на предприятии и минимизируют сбои в работе сервисов компании, использующих рассылку уведомлений. Система показала высокую эффективность готова к внедрению на предприятиях различного профиля в качестве средства автоматизации рассылки уведомлений пользователям.

#### **Библиографический список**

1. Фролов Р.Н. Информатизация современного российского общества: социально-экономические и правовые аспекты // Информационные ресурсы России. – 2020. – №4 (176). – С. 26-29.
2. Цебренько К.Н. Распределенная система обмена уведомлениями // Вестник ИМСИТ. – 2021. – №2 (86). – С. 39-42.
3. Саакян Р.Р. Моделирование электронной информационно-образовательной среды образовательной организации// Информационные ресурсы России. – 2019. – № 5. – С. 25-29.
4. Цебренько К.Н. Моделирование подсистемы автоматизации технологических процессов и учета в информационных системах медицинских лабораторий // Вестник ИМСИТ. – 2017. – №3 (71). – С. 31-34.
5. Цебренько К.Н. Гибридное моделирование информационной системы агентства недвижимости // Естественно-гуманитарные исследования. – 2017. – № 17 (3). – С. 28-34.
6. Саакян Р.Р. Проектирование информационной системы мониторинга и распределения заказов курьерской службы на основе машинного обучения // Информационные ресурсы России. – 2020. – №3. – С. 34-38.

**ANALYSIS OF ISSUES OF SECURITY OF INFORMATION SYSTEMS ON THE BASIS OF APPLICATION OF HIGH PERFORMANCE COMPUTER SYSTEMS**

**K.N. Tsebrenko**, *Candidate of Technical Sciences, Associate Professor*  
**Academy of Marketing and Social Information Technologies – IMSIT**  
**(Russia, Krasnodar)**

***Abstract.** The purpose of the work is to develop a distributed system for sending notifications. In the course of the work, algorithms for the operation of microservices of a distributed system were developed. Implementation of microservices and their testing has been completed. The efficiency of the system operation in an enterprise environment is estimated*

***Keywords:** exchange of notifications, microservices, distributed system, GoLang, algorithm, implementation.*