

Функциональная спецификация программного обеспечения «Система управления промокодами»

1. Описание

Настоящий документ содержит описание функциональных характеристик программного обеспечения «Система управления промокодами» (далее «Система»). Система предназначена для полного цикла управления промокодами: от создания кампаний до отслеживания активаций и генерации аналитики.

2. Среда функционирования продукта

Система функционирует в любой среде, поддерживающей контейнерную виртуализацию. Предпочтительной средой являются системы оркестрации Docker Swarm или Kubernetes. Система может быть развернута как standalone-сервис с подключением к базе данных PostgreSQL.

3. Функциональные требования:

Система предназначена для управления жизненным циклом промокодов: создание кампаний, генерация промокодов по шаблонам, валидация и активация кодов, отслеживание статистики использования. Система обеспечивает безопасное и масштабируемое управление промо-акциями для бизнеса любого масштаба. Система представляет следующую функциональность:

3.1. Управление кампаниями промокодов

- Создание, редактирование, просмотр и удаление кампаний
- Настройка временных рамок активности кампаний
- Указание шаблонов генерации промокодов с поддержкой макросов
- Управление метаданными кампаний в формате JSON

3.2. Гибкая генерация промокодов

- Массовая генерация промокодов по шаблону с заданным количеством
- Поддержка сложных шаблонов генерации:
 - {N:5} - генерация 5 цифр
 - {A:3} - генерация 3 букв
 - {HEX:8} - генерация 8 HEX-символов
 - {UUID} - генерация UUID
 - %DATE%, %YEAR%, %MONTH%, %DAY% - макросы дат
 - X - случайный символ (буква или цифра)
- Автоматическая обработка коллизий (дублирование кодов)
- Ручное создание отдельных промокодов

3.3. Операции с промо-кодами

- Активация (redeem) промокодов с привязкой к внешним идентификаторам
- Валидация промокодов (проверка доступности)
- CRUD операции над промо-кодами (создание, чтение, обновление, удаление)
- Настройка лимитов использования для каждого кода
- Управление статусом активности промо-кодов

3.4. Мониторинг и аналитика

- Полное аудиторирование всех операций в системе
- Отслеживание истории использования каждого промо-кода

- Статистика по активациям с метаданными
- Мониторинг доступности системы через health-check эндпоинты
- Детальное логирование всех операций с контекстом

3.5. Безопасность и контроль доступа

- Разделение API на публичные и приватные эндпоинты
- Защита приватных методов ключом доступа (ACCESS_KEY)
- Транзакционная обработка критичных операций
- Предотвращение гонок при активации промо-кодов через блокировки БД

3.6. Интеграционные возможности

- JSON-RPC API для интеграции с внешними системами
- Поддержка метаданных в формате JSON для всех сущностей
- Возможность привязки активаций к внешним идентификаторам
- Гибкая система шаблонов для адаптации к различным бизнес-процессам

3.7. Масштабируемость

- Поддержка горизонтального масштабирования
- Настраиваемый пул соединений с базой данных
- Автоматическое восстановление при сбоях
- Поддержка работы в распределенных средах

4. Системные требования к ПО

Минимальные аппаратные требования:

- Операционная система, способная запускать контейнеры. Предпочтительно Linux.
- Система управления контейнерной виртуализацией (опционально)
- База данных PostgreSQL 16+
- Количество логических ядер процессора: 2
- Семейство процессоров: x86/x64
- Частота процессора: 2.0 ГГц
- Объем установленной памяти: 4 Гб
- Дисковое пространство: 10 Гб

4.1. Минимальные требования к сторонним компонентам и/или системам, необходимым для установки и работы ПО

- PostgreSQL 16+ (Открытая лицензия PostgreSQL)
- Debian 11+ или аналогичная ОС (Открытая лицензия GNU)
- Docker 20.10+ (open-source community edition, опционально)
- Prometheus + Grafana для мониторинга (опционально)

4.2. Языки программирования

При разработке Системы был использован язык программирования GoLang 1.20+ (Открытая лицензия BSD)

5. Архитектурные особенности

5.1. API интерфейс

- JSON-RPC протокол версии 2.0
- Разделение на публичные (/public) и приватные (/) эндпоинты
- Поддержка пакетных запросов
- Стандартизированные форматы ошибок

5.2. Обработка данных

- Транзакционная модель для критичных операций
- Оптимистичные блокировки для предотвращения гонок
- Кэширование часто запрашиваемых данных
- Асинхронная обработка фоновых задач

5.3. Надежность

- Автоматическое восстановление при сбоях
- Резервное копирование данных
- Мониторинг состояния системы в реальном времени
- Детальное логирование для отладки

5.4. Производительность

- Поддержка высоких нагрузок (до 1k запросов в секунду)
- Горизонтальное масштабирование
- Оптимизированные запросы к БД
- Минимальные задержки при обработке запросов