

Поисковый модуль

1. Доступность

1.1. При возникновении ошибки соединения с БД во время исполнения функции API модулем поиска

- Пользователю немедленно возвращается ответ с HTTP-статусом 500 (Internal Server Error) и понятным описанием ошибки (например, "Сервис временно недоступен").
- Ошибка с деталями (временная метка, ID запроса, причина сбоя) записывается в централизованный лог.
- Счетчик ошибок в системе мониторинга увеличивается с тегами `url="/api/search"`, `status="5xx"`, `error_type="db_connection"`.
- Поисковый модуль продолжает обрабатывать последующие запросы, пытаясь переподключиться к БД по заданному алгоритму (с экспоненциальной задержкой).

1.2 При потере соединения с брокером событий во время обработки запросов поиска

- Модуль автоматически переключается на использование кэшированных данных на период до 5 минут
- Пользователю возвращается ответ со статусом 200, но с пометкой "Данные могут быть неактуальны"
- Система начинает попытки восстановления соединения с экспоненциальной задержкой
- Ошибка и параметры подключения логируются в централизованную систему логирования
- В метриках мониторинга увеличивается счетчик ошибок в разрезе `component="search"`, `error_type="broker_connection"`

Метрика:

- Система продолжает обрабатывать запросы без превышения времени ответа 200 ms
- Уровень доступности сохраняется на уровне 99.999% в течение времени простоя брокера до 5 минут

2. Модифицируемость

2.1. Необходимость изменить конфигурационные файлы поискового модуля для детальной отладки проблемного запроса.

- Администратор через файл конфигурации изменяет параметр `logging.level.com.example.search` с INFO на DEBUG.

- Изменение конфигурации автоматически применяется ко всем экземплярам поискового модуля без их перезагрузки (используется механизм hot-reload).
- Модуль немедленно начинает записывать в лог детальную отладочную информацию для указанного пакета/запросов.
- Оказывает влияние на производительность или обработку текущих запросов не более 50%.

3. Развертываемость

3.1. Готовность новой версии поискового модуля для исправления критической уязвимости.

- Установщик подготавливает окружение.
- Проверка целостности артефактов.
- Нагрузка постепенно переводится на новые экземпляры.
- После успешного развертывания и проверки здоровья старые экземпляры выводятся из эксплуатации.
- В случае обнаружения проблем на новом этапе (например, failed health-check) запускается автоматический откат к предыдущей стабильной версии.

3.2 Обновление поискового модуля в production среде с соблюдением высокой доступности

Ответ:

- Обновление выполняется по стратегии blue-green с использованием Ansible
- Новая версия модуля развертывается параллельно с текущей, после чего трафик плавно переключается
- Во время обновления обеспечивается доступность сервиса на уровне 99.999%
- Все необходимые зависимости проверяются и устанавливаются автоматически
- После успешного развертывания система автоматически запускает набор smoke-тестов
- В случае обнаружения критических ошибок автоматически запускается процедура отката

Метрика:

- Время обновления не превышает 25 минут
- Отсутствие ощутимого для пользователей простоя
- Сохранение уровня доступности 99.999% во время обновления
- Успешное выполнение всех smoke-тестов после развертывания

Модуль синхронизации

1.Доступность

1.1 При потере соединения с брокером сообщений мастер-системы во время синхронизации данных:

- модуль должен автоматически attempt to reconnect с экспоненциальной задержкой;
- ошибка должна быть залогирована в централизованную систему логирования;
- администратор должен быть уведомлен через систему мониторинга;
- модуль должен продолжить синхронизацию данных после восстановления соединения без потери данных.

1.2 При получении события с некорректными данными из мастер-системы:

- событие должно быть помещено в карантин (dead-letter queue);
- модуль должен продолжить обработку следующих событий без остановки;
- ошибка должна быть залогирована с указанием ID события и причины ошибки;
- администратор должен получить оповещение о наличии некорректных данных для ручной обработки.

1.3 При поступлении некорректных данных из мастер-системы во время синхронизации

Ответ:

- Система помещает проблемное событие в очередь "на проверку" без остановки обработки других событий
- Пользователю/администратору отправляется уведомление о некорректных данных
- Ошибка и содержание события логируются в централизованную систему с пометкой "validation_error"
- В метриках мониторинга увеличивается счетчик ошибок в разрезе component="sync", error_type="validation"
- Система продолжает обработку корректных событий, соблюдая максимальную задержку не более 5 секунд

Метрика:

- Продолжение работы модуля без прерывания основного потока данных
- Уровень доступности сохраняется на уровне 99.99%
- Обработка корректных данных продолжается с соблюдением требования по задержке

2. Модифицируемость

2.1 Администратору требуется изменить логику обработки входящих событий из мастер-системы. Администратор вносит изменения в конфигурацию модуля и перезапускает его. Ожидается, что:

- модуль запускается с новой конфигурацией;
- изменения вступают в силу без необходимости переписывания кода;
- время простоя модуля составило не более 5 минут.

2.2 Необходимо изменить конфигурацию модуля (например, адрес брокера сообщений или уровень логирования) в runtime

- Администратор вносит изменения в централизованное хранилище конфигураций.
- Модуль синхронизации обнаруживает изменение конфигурации (через - механизм опроса или получение уведомления).
- Модуль применяет новую конфигурацию без необходимости перезагрузки всего процесса.
- Изменение вступает в силу.
- Факт изменения конфигурации регистрируется в логах.

2.3 При добавлении нового типа события из мастер-системы:

- разработчик должен иметь возможность добавить новый обработчик через API модуля;
- изменение должно требовать только добавления нового класса-обработчика и конфигурации;
- существующая логика не должна нарушаться;
- тестирование и развертывание изменения должно занимать не более 4 часов.

2.4 Обновление версии библиотеки обработки событий в модуле синхронизации

Ответ:

- Администратор через централизованную систему конфигурации указывает новую версию библиотеки
- Система автоматически загружает новую версию библиотеки и проверяет совместимость
- Модуль перезапускается с новой версией библиотеки с использованием zero-downtime стратегии
- Во время перезапуска обработка событий временно замедляется, но не останавливается

- После успешного перезапуска система проверяет корректность обработки событий

Метрика:

- Время простоя экземпляра модуля составило не более 10 минут
- Система продолжает обрабатывать события с соблюдением максимальной задержки 5 секунд
- Обеспечен уровень доступности 99.99% во время обновления
- Соответствие требованию по сроку внедрения (не более 2 недель)

3.Развертываемость

3.1 При возникновении ошибок после развертывания новой версии модуля синхронизации в продуктивной среде администратор запускает процедуру отката.

Ожидается:

- процедура отката выполнена успешно;
- версия модуля откатена на предыдущую стабильную;
- модуль запущен и работает с предыдущими настройками;
- процедура отката заняла не более 10 минут.

3.2 При обновлении модуля в условиях высокой доступности (99,99%):

- обновление должно производиться по схеме blue-green или rolling update;
- downtime во время обновления должен быть нулевым;
- система должна автоматически проверять работоспособность новой версии перед полным переключением;
- откат на предыдущую версию должен занимать не более 5 минут.

3.3 При подготовке окружения для установки модуля:

- все зависимости (Java, Docker, библиотеки) должны устанавливаться автоматически через Ansible-плейбук;
- проверка готовности окружения должна выполняться скриптом перед установкой;
- время установки и настройки окружения не должно превышать 20 минут;
- в случае ошибки установки скрипт должен предоставить детальный отчет для исправления.

Модуль миграции

1.Доступность

1.1 При возникновении ошибки чтения данных из мастер-системы во время процесса миграции:

- модуль должен приостановить миграцию и залогировать ошибку;
- администратор должен быть уведомлен о сбое;
- после устранения ошибки модуль должен возобновить миграцию с точки останова;
- общее время простоя миграции не должно превышать 1 часа.

1.2 При обнаружении некорректных данных в процессе миграции:

- модуль должен автоматически помещать некорректные записи в карантин (отдельная таблица или файл);
- миграция корректных данных должна продолжаться без остановки;
- ошибка должна фиксироваться в логах с указанием типа ошибки и идентификатора записи;
- администратор должен получать уведомление о наличии некорректных данных для последующего анализа.

1.3 При нехватке ресурсов (память, диск) во время миграции:

- модуль должен приостановить процесс и уведомить администратора;
- в логах должны быть указаны точные параметры нехватки ресурсов;
- после решения проблемы модуль должен продолжить работу с последней контрольной точки;
- система должна предусматривать возможность очистки временных ресурсов автоматически.

1.4 При одновременном запуске нескольких экземпляров миграции:

- модуль должен обеспечивать блокировки для предотвращения конфликтов;
- только один экземпляр должен быть активным для каждой задачи миграции;
- ошибки конфликта должны логироваться и уведомлять администратора;
- система должна поддерживать режим работы "только чтение" для источников во избежание повреждения данных.

2.Модифицируемость

2.1 Разработчику требуется добавить поддержку миграции данных из новой мастер-системы. Разработчик добавляет новый коннектор в код модуля.

Ожидается, что:

- изменение будет внесено в код на этапе разработки;
- внесение и тестирование изменений займёт менее 8 часов;
- никаких побочных эффектов на существующую функциональность не возникнет.

2.2 При необходимости изменить логику преобразования данных:

- изменение должно вноситься через конфигурационные файлы (JSON/YAML) без изменения кода;
- новая логика должна применяться без перезапуска модуля (hot-reload);
- система должна проверять валидность новой конфигурации перед применением;
- время на внедрение изменения не должно превышать 2 часов.

2.3 При добавлении нового источника данных для миграции:

- разработчик должен иметь возможность добавить новый коннектор через API модуля;
- изменение должно требовать только добавления нового класса-коннектора и конфигурации;
- существующая логика миграции не должна нарушаться;
- тестирование и развертывание изменения должно занимать не более 1 рабочего дня.

2.4 При изменении схемы данных целевой базы:

- модуль должен поддерживать возможность трансформации данных на лету;
- изменения в схеме должны описываться в конфигурации без переписывания кода;
- система должна обеспечивать обратную совместимость со старыми схемами данных;
- обновление конфигурации не должно требовать остановки миграции.

2.5 При необходимости добавить новую метрику мониторинга процесса миграции:

- разработчик должен иметь возможность добавить метрику через конфигурационный файл;
- система должна автоматически начать сбор и экспорт новой метрики;
- изменение не должно требовать перекомпиляции или перезапуска модуля;
- новая метрика должна стать доступной в системе мониторинга в течение 5 минут.

2.6 Требуется изменение бизнес-логики преобразования данных абонентов во время миграции

Ответ:

- Администратор загружает новый конфигурационный файл с обновленными правилами преобразования через централизованную систему конфигурации
- Модуль динамически подгружает новые правила без перезапуска процесса миграции
- Новые правила применяются к данным, начиная с текущей позиции миграции
- Система сохраняет статистику по обработке данных с разбивкой по старым и новым правилам
- В логах фиксируется время и детали изменения конфигурации

Метрика:

- Время внедрения изменений не превышает 10 минут
- Отсутствие простоя в работе модуля
- Скорость миграции сохраняется на уровне не менее 20 млн. абонентов в день
- Соответствие требованию по сроку внедрения новой функциональности (не более 2 недель)

3.Развертываемость

3.1 При развертывании новой версии модуля миграции в продуктивной среде с помощью Ansible:

- установка и настройка модуля должны выполняться автоматически;
- время развертывания не должно превышать 15 минут;
- модуль должен быть готов к работе без ручного вмешательства;
- после развертывания модуль должен корректно работать в заданном окружении.

3.2 После развертывания новой версии модуля миграции в prod процесс начал завершаться с ошибкой OutOfMemoryError и не может продолжить работу

- Администратор запускает подготовленный Ansible-плейбук для отката (rollback.yml).
- Плейбук останавливает текущую версию модуля, развертывает предыдущую стабильную версию из артефактов, применяет предыдущую версию конфигураций и запускает модуль.
- Система мониторинга показывает, что модуль перешел в состояние "Работает".

3.3 При автоматической установке модуля с помощью Ansible:

- плейбук должен проверять наличие всех зависимостей (Java, Docker, библиотеки);

- в случае отсутствия зависимостей - автоматически устанавливать их;
- время полной установки и настройки не должно превышать 30 минут;
- после установки система должна автоматически запускаться и проходить самодиагностику.

3.4 При конфигурировании модуля в различных окружениях (dev/test/prod):

- все настройки должны управляться через централизованную систему (Ansible, Consul);
- конфигурация должна подгружаться без перезапуска модуля;
- система должна поддерживать различные профили конфигурации;
- изменение настроек должно занимать не более 5 минут.

3.5 При развертывании модуля в containerized среде (Docker/Kubernetes):

- модуль должен быть упакован в Docker-образ с минимальными зависимостями;
- образ должен automatically build и push в registry при каждом коммите;
- развертывание в Kubernetes должно выполняться через Helm-чарт;
- время развертывания нового экземпляра не должно превышать 3 минут.

Сценарии атрибутов качества - Примеры

500 пользователей инициируют 2000 запросов в течение 30 секунд в нормальном режиме работы. Система обрабатывает все запросы со средней задержкой в две секунды

Сервер в серверной ферме вышел из строя во время нормальной работы, система информирует оператора и продолжает работать без простоя

Разработчик хочет изменить пользовательский интерфейс. Изменение будет внесено в код на этапе проектирования. Внесение и тестирование займёт менее трёх часов, и никаких побочных эффектов не возникнет

Источник стимула Стимул Окружение Arteфакт Реакция Мера реакции

Примеры

Доступность

При возникновении ошибки соединения с БД во время исполнения функции API модулем поиска

- пользователю должен быть возвращен ответ со статусом 500, содержащий описание ошибки;
- ошибка и параметры вызова функции должны быть залогированы;
- в метриках мониторинга должен быть увеличен счетчик ошибок в разрезе url="<URL функции>", status="5xx".
- сервис должен продолжить работать в обычном режиме.

Модифицируемость

Администратору требуется обновить версию JVM, используемую модулем синхронизации в проме. Администратор устанавливает новую версию JVM на сервер, переконфигурирует и перезапускает экземпляр модуля синхронизации. Ожидается, что

- экземпляр модуля запущен и использует новую версию JVM
- downtime экземпляра модуля составил не более 15 минут.

Развертываемость

При возникновении ошибок при переходе на новую версию модуля миграции в проме администратор запускает процедуру отката версии в установщике модуля. Ожидается:

- процедура отката выполнена успешно
- версия модуля откатена на предыдущую
- модуль запущен и работает с предыдущими настройками и окружением.
- процедура отката заняла не более 30 минут.