



Система управления сценариями обработки
событий

Notification System

**Установка, настройка и обновление
компонентов**

Версия 2.17

Руководство администратора

Содержание

О компании Bercut.....	6
Назначение документа.....	8
Термины и определения.....	9
1. Общие сведения.....	12
1.1. Внешняя система.....	13
1.2. Сценарий.....	13
1.3. Правило.....	14
1.4. Критерий.....	15
1.5. Действие.....	16
1.6. Микросервисы.....	16
1.7. Макроподстановки.....	17
1.8. Группа значений.....	17
1.9. Ролевая модель доступа.....	18
2. Архитектура системы.....	20
3. Интеграция с внешними системами.....	24
3.1. Интеграция с Apache Kafka.....	24
3.2. Интеграция с RabbitMQ.....	26
3.3. Отправка HTTP-сообщений.....	26
3.4. Интеграция с Scheduler Subsystem.....	27
4. Установка и настройка компонентов.....	28
4.1. Ядро системы.....	28
4.1.1. Настройка NS SLES EM.....	28
4.1.2. Настройка LW SA Container EM BO.....	29
4.1.3. Настройка LW SA Container.....	36
4.1.4. Установка и настройка схем базы данных.....	37
4.2. Настройка AuditDeliveryReport.....	43
4.3. Установка BlackList.....	43
4.3.1. Установка и настройка SA NS BlackList.....	44

4.3.2. Установка и настройка схемы данных DB BLACKLIST.....	44
4.3.3. Настройка Service Gateway.....	44
4.4. Настройка Audit API.....	45
4.5. Настройка System Principal.....	45
4.6. Настройка интеграции с Apache Kafka.....	46
4.7. Установка и настройка HTTP-адаптера.....	58
4.8. Установка и настройка RabbitMQ-адаптера.....	61
4.9. Установка и настройка адаптера для БД.....	63
4.10. Установка и настройка SA NS GroupValues.....	68
4.11. Установка и настройка SA NS ScheduledScenario.....	69
4.12. Установка и настройка компонентов системы BRE MicroServices.....	73
4.13. Установка и настройка SA NS AuditAggregator.....	76
4.14. Установка и настройка SA KeysValuesStorage.....	77
4.15. Установка и настройка SA NS KVStorageManagement.....	79
4.16. Настройка кеширования ответов на внешние вызовы.....	80
5. Обновление системы.....	82
6. Логирование.....	83
7. Мониторинг системы.....	84
7.1. Основные показатели.....	84
7.2. Мониторинг приложений Unica.....	85
7.3. Время отклика.....	86
Приложение 1. Работа с отложенными сценариями.....	87
П 1.1. Создание микросервиса.....	87
П 1.2. Настройка отложенного исполнения сценариев.....	89
П 1.3. Вариант настройки отложенного исполнения сценариев.....	92
П 1.3.1. Создание микросервиса.....	93
П 1.3.2. Настройка сценария А.....	94
П 1.3.3. Настройка сценария Б.....	98
Приложение 2. Создание событий.....	103
П 2.1. Пользовательское событие.....	103
П 2.1.1. Создание микросервиса.....	103

П 2.1.2. Загрузка микросервисов в систему.....	111
П 2.1.3. Просмотр и заполнение справочников.....	112
П 2.1.4. Создание и настройка сценария.....	117
П 2.1.5. Обновление микросервиса и сценария.....	123
П 2.2. Событие для отложенного сценария.....	126
П 2.3. Событие с поддержкой транзакционности.....	126
П 2.4. События для интеграции с Apache Kafka.....	127
П 2.4.1. Событие для обработки SA NS Kafka Consumer.....	127
П 2.4.2. Событие для обработки SA NS Kafka Producer.....	134
П 2.5. События для SA NS HttpProducer.....	139
Приложение 3. Правила преобразования сообщений адаптерами Consumer...	142
П 3.1. Преобразование с помощью файла маппинга.....	142
П 3.1.1. XML-преобразование.....	145
П 3.1.2. JSON-преобразование.....	147
П 3.2. Преобразование с помощью плоской структуры.....	149
Приложение 4. Правила преобразования сообщений адаптерами Producer.....	150
П 4.1. Преобразование с помощью файла маппинга.....	150
П 4.1.1. Синтаксис XML-преобразования.....	151
П 4.1.2. Синтаксис JSON-преобразования.....	153
П 4.1.3. Синтаксис преобразования в формат plain text.....	156
П 4.2. Преобразование с помощью плоской структуры.....	156
Приложение 5. Реализация обработки событий.....	159
П 5.1. Критерии.....	159
П 5.1.1. Требования к выделению критериев.....	160
П 5.1.2. Требования к описанию критериев.....	161
П 5.2. Обработка событий.....	161
П 5.3. Получение значений критериев (Getter).....	162
П 5.4. Справочники.....	163
П 5.5. Действия (Action).....	164
П 5.5.1. Исключения.....	164
П 5.6. Макроподстановки.....	164

П 5.7. Реализация микросервисов.....	165
П 5.8. Требования к наименованию артефактов и расширению интерфейсов..	165
Источники информации.....	167
История изменений.....	168

О компании Bercut

Bercut — мировой поставщик решений в области ИТ, который предлагает уникальный подход к развитию и управлению услугами совместно с оператором и абонентом.

Техническая поддержка

Компания Bercut предлагает заказчикам полную техническую поддержку продуктов.

Bercut осуществляет гарантийное и послегарантийное сопровождение поставляемых комплексов по отдельному договору.

При возникновении в процессе эксплуатации ситуаций, не указанных в пакете эксплуатационной документации, пользователь может обратиться в группу технической поддержки компании Bercut одним из указанных ниже способов:

- на сайте <https://support.bercut.com> создать заявку (раздел **Заявки**);
- отправить электронное письмо на адрес support@bercut.com;
- позвонить по телефону +7 (812) 327-3231.

Уведомление об авторских правах

Компания Bercut обладает исключительным правом на данные материалы.

Не допускается полностью или частично воспроизводить или передавать данный документ в какой-либо форме, любым способом и в любом формате, электронными или механическими средствами, включая фотокопирование, запись и хранение в системе базы данных, не получив предварительное согласие в письменном виде от компании Bercut.

Обратная связь

Уважаемый читатель!

Наша цель — улучшение документации с точки зрения удобства ее использования, полноты и понятности изложенного материала. Свои вопросы, предложения, замечания об ошибках, неясности в изложении, нехватке примеров вы можете передать одним из указанных ниже способов:

- на сайте <https://support.bercut.com> создать заявку (раздел **Заявки**);
- отправить электронное письмо на адрес techwriters@bercut.com.

Пожалуйста, укажите:

- версию системы;
- название документа;
- номер версии документа;
- по возможности — главу, раздел и страницу, к которым относятся ваши замечания.

После исправления текста по замечаниям мы известим вас о выходе новой версии документа.

i Примечание. В соответствии с положениями политики конфиденциальности мы принимаем обратную связь от компаний, с которыми установлены соответствующие договорные обязательства. Если вы являетесь третьей стороной, пожалуйста, обратитесь к представителям компании, с которой у вас заключен договор.

Назначение документа

Документ содержит:

- общие сведения о системе Notification System: назначение, описание специфичных терминов и общего принципа работы;
- описание архитектуры и предполагаемой схемы развертывания;
- описание порядка установки, настройки и обновления компонентов системы.

Руководство предназначено для специалистов заказчика, которые выполняют установку и настройку системы Notification System.

Термины и определения

BO Product Catalog

Веб-приложение для управления продуктами из системы Product Catalog.

BP

Business Process. Сценарий на языке BPEL, выполняемый на Platform v3 для реализации бизнес-задачи. Реализует одну или несколько бизнес-операций.

BPEL

Business Process Execution Language. Язык на основе XML для формального описания бизнес-процессов и протоколов их взаимодействия. Задача BPEL — координация совместной работы веб-служб.

CareM DB

База данных системы CareM SelfCare Tele2.

CRM

Customer Relationship Management. Система управления взаимоотношениями с клиентами.

DWH

Data Warehouse. Информационная база данных, которая получает данные из используемых компанией баз данных и других источников данных, а затем трансформирует их в структуру, подходящую для выполнения бизнес-анализа.

KeysValues Storage

Система хранения данных по ключам, которая работает с резидентной базой данных Tarantool. Система используется для быстрого доступа к данным и кешу *Platform v3*.

LW SA Container

Light Weight Specialized Adapter Container. Контейнер на базе Tomcat, который используется для подключения SA к RTSIB.

MVNO

Mobile Virtual Network Operator. Виртуальный оператор сети мобильной связи.

Notification System

Сокращенно — *NS*. Система управления сценариями обработки событий для определенного продукта. На основании критериев Notification System определяет подходящее правило и выполняет соответствующие действия.

PCRF

Policy and Charging Rules Function. Элемент сети сотовой связи, который определяет правила применения политик обслуживания абонентов, разрешает или запрещает предоставление абонентам определенных сервисов и устанавливает параметры качества обслуживания в соответствии с заданными характеристиками (QoS).

Platform v3

Платформа компании Bercut для разработки приложений (VAS-услуг), реализуемых как набор бизнес-процессов.

PostgreSQL

Объектно-реляционная система управления базами данных. Использует язык SQL.

RabbitMQ

Открытая платформа, которая реализует протокол передачи сообщений между компонентами системы — AMQP. На аппаратном уровне устанавливается на RabbitMQ Server.

RabbitMQ Server

Компонент, выполняющий маршрутизацию и доставку сообщений, а также распределение потоков данных по протоколу AMQP.

RTSIB

Real-Time Service Integration Bus. Интеграционная шина для обмена данными между *бизнес-процессами*, *SA*, *SE* и компонентами Platform v3 в онлайн-режиме.

SA

Specialized Adapter. Специализированный адаптер, развернутый на *LW SA Container*. Обеспечивает взаимодействие внешней системы с уровнем логики *Platform v3* с помощью шины *RTSIB* по веб-интерфейсу.

SCP

Service Control Point. Элемент интеллектуальной сети. Узел управления услугами. Предоставляет возможности для реализации программной среды выполнения логики телекоммуникационных интеллектуальных услуг и взаимодействия с другими элементами сети.

SE

Service Enabler. Логическая группа вспомогательных операций на языке Java. Бизнес-процесс использует SE для решения технологических задач.

Service Gateway

Сокращенно — SG. Шлюз доступа к сервисам Platform v3 для внешних приложений.

Service Profile Management

Система хранения и управления профилями сервисов. В разных источниках используются названия *Service Profile* или *Service Management*. Сокращенно — *SP* или *SPM*.

SLES

Service Logic Execution System. Система выполнения бизнес-процессов на Platform v3.

Unica

Платформа услуг и абонентского сервиса. Предназначена для построения систем гибкого интеллектуального обслуживания абонентов в инфраструктуре оператора мобильной связи.

Welcome SMS

Система информирования абонента при регистрации в *гостевой сети*. Продуктовое решение на базе системы Notification System.

Группа

Объединение значений или групп.

Группа новая

Новая группа, которая не используется в правилах.

Группа релизная

Группа, которая используется в правилах для сценариев.

Группа черновая

Группа, которая доступна для правок. Обязательно должна принадлежать корневой группе в статусе *новая* или *релизная*. Черновая группа используется в правилах для черновых сценариев. При отсутствии черновой группы в черновом сценарии используется релизная группа.

Продукт

Товар или услуга, которую оператор связи предлагает абоненту для приобретения, потребления и использования.

Регион домашний

Регион, в котором абонент заключил договор об оказании услуг и был зарегистрирован в сети оператора сотовой связи.

Секционирование

Разделение хранимых объектов баз данных — таблиц, индексов, материализованных представлений — на отдельные части с разными параметрами физического хранения. От англ. *partitioning*.

Сеть гостевая

Сеть другого оператора связи, в пределах страны или за границей, вне зоны действия *домашней сети*.

Сеть домашняя

Сеть, в которой абонент постоянно зарегистрирован на основании договора об оказании услуг, заключенного с оператором связи.

Привилегия управления продуктами

Привилегии, которые распространяют свое влияние не только на сами продукты, так и в целом на всю систему.

Привилегия управления системой

Привилегии, которые имеют смысл только в глобальном смысле и не влияют на управление отдельными продуктами.

1. Общие сведения

Система Notification System обрабатывает события, приходящие от внешних систем. Каждому событию сопоставляются критерии. Система анализирует значения этих критериев, выбирает подходящее правило и выполняет определенное действие.

Notification System — система управления сценариями обработки событий продукта, реализованная на Platform v3 [1].¹

Общий принцип работы Notification System:

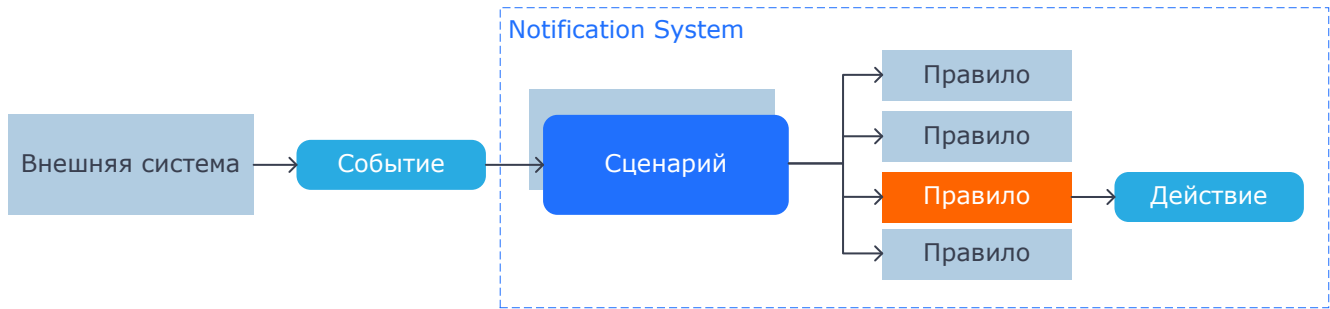


Рис. 1. Обработка событий

1. В Notification System из *внешней системы* поступает *событие*.
2. На основе свойств события — источника события и типа события — система подбирает *сценарий* для его обработки.
3. Система обрабатывает правила в заданном порядке. При этом критерии сопоставляются с заданными значениями. На основании *критериев* система определяет будет ли выполнено текущее *правило* сценария. Для определения значений критериев система вызывает *микросервисы*.
4. Notification System выполняет одно или несколько *действий*, которые соответствуют правилу.

Примечание. 1 правило = 1 действие. У действия может быть несколько операций.

Если в параметрах действия используются *макроподстановки*, система вызывает соответствующие микросервисы для замены наименования макроса на значение — результат вызова микросервиса.

Для управления сценариями обработки событий продукта для пользователя должна быть настроена *ролевая модель доступа*.

Пример:

- внешняя система — Welcome SMS;
- событие «регистрация абонента в сети»;
- критерии «тарифный план абонента», «регион абонента», «регион пребывания»;
- правило:
 - тарифный план = «Гамма»;
 - регион абонента = «Новосибирская область»;
 - регион пребывания = «Ленинградская область».

¹ Другое название системы Notification System — Business Rules Engine (BRE).

- действие «Отправка SMS-сообщения» с текстом *Желаем Вам приятной поездки! Стоимость исходящих вызовов — 3 рубля, входящих — бесплатно, отправка SMS — 2 рубля.*

Для события «регистрация абонента в сети» система выполняет действие, если значения критериев соответствуют условиям правила.

1.1. Внешняя система

Внешняя система передает в Notification System информацию о событии и его параметры.

Внешние системы для Notification System — источники *событий*. Одна внешняя система — один portType.

При наступлении события внешняя система передает:

- идентификатор события,
- входные параметры;
- значения параметров.

Например, система Welcome SMS передает событие «Регистрация абонента в сети». Параметры такого события: MSISDN, VLR, IMSI, HLR и MSC.

1.2. Сценарий

Сценарий определяет действие, которое система Notification System выполняет при наступлении определенного события. Действие зависит от выбранного правила с набором условий (критериев). Сценарии настраивает администратор.

Сценарий — набор *правил* и *действий*, которые определяют реакцию системы Notification System на конкретное событие.

Событие — событие от внешней системы, метод в portType внешней системы.

Примечание. 1 сценарий = 1 событие. У события может быть 1 или несколько сценариев.

Свойства сценария:

- наименование сценария;
- источник события — идентификатор системы;
- *событие* — идентификатор операции;
- входные параметры — параметры события;
- *правила*, по которым система выбирает действие. Правила состоят из логических выражений с *критериями*.
- *действия*, одно из которых система выполняет в результате обработки события.

Выполнение сценария

При поступлении запроса от внешней системы Notification System определяет сценарии по параметрам запроса: источнику события и операции.

Если для одного события предусмотрено несколько сценариев с одинаковыми свойствами — источником и событием, — система исполняет сценарии последовательно.

Выполнение сценария включает:

- подбор подходящего правила по значениям критериев;
- выполнение действия, которое соответствует выбранному правилу.

В системе присутствует возможность *отложенного исполнения сценариев*. Для этого нужно специальным образом настроить два сценария:

- сценарий А, результатом выполнения которого является действие, вызывающее другой сценарий в заданное время и дату.
- сценарий Б, запускаемый в определенное время и дату.

Настройка сценария

Сценарии настраивает администратор в веб-интерфейсе системы.

1.3. Правило

Сценарий включает одно или несколько правил. Правило состоит из логических выражений с критериями. Если результатом вычисления всех логических выражений является «истина», система применяет правило — выполняет действие, соответствующее этому правилу.

*Правило — набор логических выражений (условий), при соблюдении которых Notification System выполняет **действие**, соответствующее правилу. В результате исполнения сценария система применяет только одно — первое подходящее правило.*

Правила настраиваются в таблице правил веб-интерфейса или в Excel-файле. Затем вы можете выгрузить параметры сценария: описание правила, список возможных **действий**, параметров операций, **критериев** и **макроподстановок**.

Правило состоит из одного или нескольких логических выражений с критериями. Правило применяется, если результатом вычисления всех логических выражений, которые входят в правило, является «истина».

Для настройки правила пользователь:

1. Выбирает действие, которое система выполнит при применении правила.
2. Указывает параметры операций, включенных в действие. В значение параметра можно включить макроподстановки сценария.
3. Настраивает условия применения правила — *логические выражения* с критериями. Для этого выбирает логическую операцию и значение, с которым сравнивается критерий. В правиле могут участвовать не все критерии сценария.

Применение правила — выполнение действия, которое ему соответствует. После применения правила сценарий считается выполненным, поиск правил прекращается.

Логическое выражение

Логическое выражение состоит из:

- **критерия**,
- логической операции (оператор),
- значения или набора возможных значений критерия.

<критерий> <логическая операция> <значение>



Рис. 2. Пример логического выражения с критериями

Для определения значения критерия система вызывает микросервис.

Правило

При расчете правила система последовательно вычисляет значения входящих в него логических выражений. Расчет ведется в порядке приоритета критериев.

- Если результатом вычисления очередного логического выражения является «ложь», остальные логические выражения не вычисляются.
- Если результатом вычисления всех логических выражений, входящих в правило, является «истина», система применяет правило.

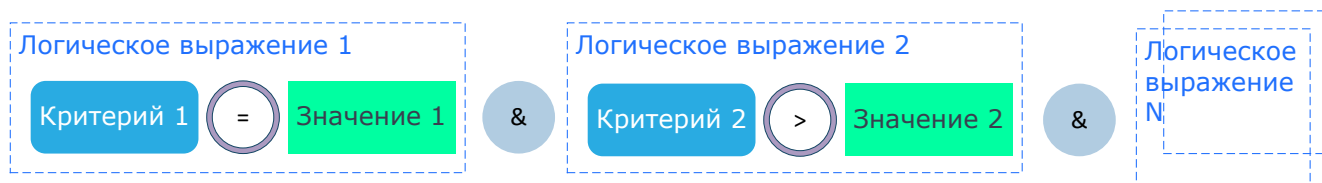


Рис. 3. Правило

1.4. Критерий

Критерии и их значения определяют условия, при которых система выполняет то или иное действие.

Критерий — параметр события или результат вызова [микросервиса](#).

Критерии используются в:

- **Правилах:** система определяет значение критерия и выбирает подходящее правило. В одном правиле могут учитываться значения одного, нескольких критериев или ни одного.
- **Действиях:** система использует критерии в качестве входных параметров операций, которые входят в действие, а также в качестве результата обработки события.
- **Макроподстановках:** администратор указывает критерий, на основе которого определяется значение макроса.

Примеры критериев: «тарифный план абонента», «регион абонента», «регион пребывания».

Критерий характеризуется *приоритетом*. Приоритет влияет на порядок вычисления логического выражения с этим критерием в правиле. Чем больше значение приоритета, тем раньше система вычислит значение логического выражения при поиске правила.

1.5. Действие

Действие включает в себя одну или несколько операций. Эти операции выполняются при соблюдении условий правила.

*Действие — часть **сценария**, которая определяет реакцию системы на событие. В сценарий входит одно или несколько действий.*

В результате применения **правила** выполняется одно действие, которое соответствует правилу.

Свойства действия:

- Наименование действия.
- Список операций. Действие состоит из одной или нескольких *операций*. Действие также может быть пустым. Операция определяет вызов **микросервиса**.
- Выходной параметр, который может передавать во внешнюю систему как результат обработки события после выполнения сценария.

1.6. Микросервисы

Микросервисы используются в сценарии для получения значения критерия и выполнения операций.

*Микросервис — артефакт, предназначенный для получения значений критериев или выполнения бизнес-задачи. Микросервисы вызываются в **сценарии**:*

- для определения значения **критериев**, входящих в **правило** — т.е. **критерии** (Getter);
- при выполнении операций, которые входят в состав **действия**, предусмотренного сценарием — т.е. **операции** (Action);
- для **макроподстановок**.

В микросервисах могут быть вызваны **справочники** (Dictionary).

Особенности микросервисов:

- Являются самодостаточными объектами для интеграции: содержат логику получения, документацию, справочники.
- Делятся по типам: *getter* и *action*.
- Имеют стандартный WSDL-интерфейс и встраиваются в RTSIB. Реализация может быть на BPEL или Java (SA).
- Повторное использование объектов в разных продуктах.

Примеры микросервисов:

- получение **MSISDN** по значению **IMSI**,
- получение домашнего региона абонента по **MSISDN**,
- отправка SMS-сообщения.

Один микросервис возвращает один или несколько критериев. Для получения значения критерия при вызове микросервиса указывается наименование критерия и входные параметры. Например, для получения значения критерия «Домашний регион»:

Домашний регион (IMSI)

Для выполнения действия при вызове микросервиса указывается наименование операции и входные параметры. Например, для отправки SMS-сообщения абоненту:

Отправка SMS (Адрес отправителя, Адрес получателя, IMSI получателя, текст сообщения)

1.7. Макроподстановки

Макроподстановки позволяют заменять в правилах и действиях наименование макроса на значение, вычисленное в результате вызова соответствующего микросервиса.

Макроподстановка используется при настройке правил и действий сценария. В результате макроподстановки система заменит наименование макроса результатом вызова микросервиса, который ему соответствует.

Примеры микросервисов, которые используются для макроподстановок: получение наименования домашнего региона, значение баланса, наименование услуги.

Например, для операции «Отправка SMS» вы можете использовать макроподстановки при задании значений параметра «Текст сообщения». Для этого администратор при настройке сценария добавляет необходимые макроподстановки. Эти макроподстановки пользователь может использовать в тексте сообщений при настройке сценария в Excel-файле.

1.8. Группа значений

Группа значений — набор значений, которые могут использоваться в выражениях.

Группа значений включает в себя другие группы или список значений.

Структура группы:

- Корневая группа — это группа, которая не имеет родителя, не может содержать значений и может иметь дочерние группы. Может быть ассоциирована с бизнес-критерием. К корневой группе можно применить ролевую модель.

Корневая группа может быть в одном из трех статусов:

- **релизная группа**, используется в [правилах](#) для релизных [сценариев](#).
- **новая группа**, не используется в [правилах](#).
- **черновая группа**, обязательно должна принадлежать корневой группе в статусе **новая группа** или **релизная группа**. Используется в [правилах](#) для черновых сценариев. При отсутствии черновой группы в черновом сценарии используется релизная группа.

- Вложенная группа значений (подгруппа). Это группа, которая имеет родителя, содержит значения и не имеет дочерних групп.
- Вложенная группа групп (подгруппа). Это группа, которая имеет родителя, не содержит значения и может иметь дочерние группы. Глубина вложенности неограниченна.

Группа значений связывается с бизнес-критерием. Для одного критерия может быть задана одна группа²

Пример структуры группы:

Корневая группа 1

² в рамках разных продуктов, для критерия могут быть заданы разные группы.

```
Группа 1
  Группа 1.1
    Группа 1.1.1
      Список значений
    Группа 1.N
      Список значений
  Группа N
```

Группы используются для логической группировки других групп и списков значений. Список значений содержит список данных определенного типа:

- одиночное значение — строка или число;
- диапазон [Число А;Число В];
- маска.

В качестве значений группы могут быть использованы значения из справочников.

1.9. Ролевая модель доступа

Для управления доступом пользователей к системе используется ролевая модель.

При настройке управления доступом для каждого из пользователей настраиваются роли. Каждая роль включает в себя привилегии. Привилегия — права для работы с объектами системы.

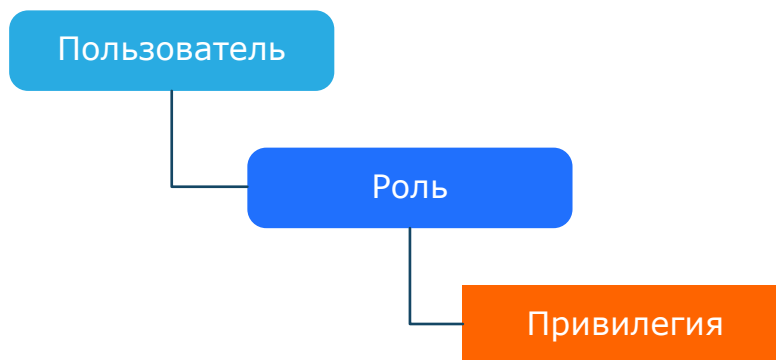


Рис. 4. Ролевая модель пользователей

Для того, чтобы пользователь мог войти в веб-интерфейс системы Notification System, администратор должен предоставить ему доступ — создать учетную запись *пользователя* и назначить ему одну или несколько *ролей*. Роли определяют круг доступных пользователю действий и используются для объединения *привилегий* в наборы, которые можно назначать пользователям.

Каждый связанный с ролью пользователь обладает всеми включенными в нее привилегиями. Если пользователю задано несколько ролей, он получает все привилегии, включенные во все роли.

Роли можно назначить как в целом пользователю, так и отдельным продуктам для пользователя. Роли для продуктов дополняют роли у пользователя, выполняется объединение ролей.

Для успешного создания пользователей предварительно должна быть настроена аутентификация. Описание настройки аутентификации — [4.1.2](#).

і Примечание. Список привилегий фиксирован на уровне системы, расширяется от версии к версии и не может быть изменен пользователем.

Привилегии делятся на два типа:

- Привилегия управления продуктами;
- Привилегия управления системой.

Привилегии управления продуктами распространяют свое влияние на продукты. Роли с этими привилегиями могут быть назначены пользователям:

- глобально — применяется ко всем продуктам;
- на отдельные продукты — применяется только к тому продукту, для которого назначена.

Привилегии управления системой распространяют свое влияние на систему и не имеют отношения к продуктам. Если роль, включающую эту привилегию, назначить на продукт — это не даст какого-либо эффекта. К привилегиям управления системой относят, например, управление *критериями* и *микросервисами*.

2. Архитектура системы

Для подключения к базам данных используются SA. Логику работы системы реализуют SE и микросервисы MS, которые выполняются в среде SLES.

Архитектура Notification System основана на Platform v3. Гибкость настройки системы позволяет разрабатывать решения для разных продуктов.



Рис. 5. Схематическое представление системы Notification System

Примеры продуктовых решений:

- Система Welcome SMS для информирования абонента при регистрации в гостевой сети. Настройка продуктового решения — [11].
- Система T2Group для работы с групповыми скидками. Настройка продукта «Линии» — [14], информация по установке системы — [15].
- Система PromoCode для работы с промокодами. Настройка продукта «Промокоды» — [16], информация по установке системы — [17].

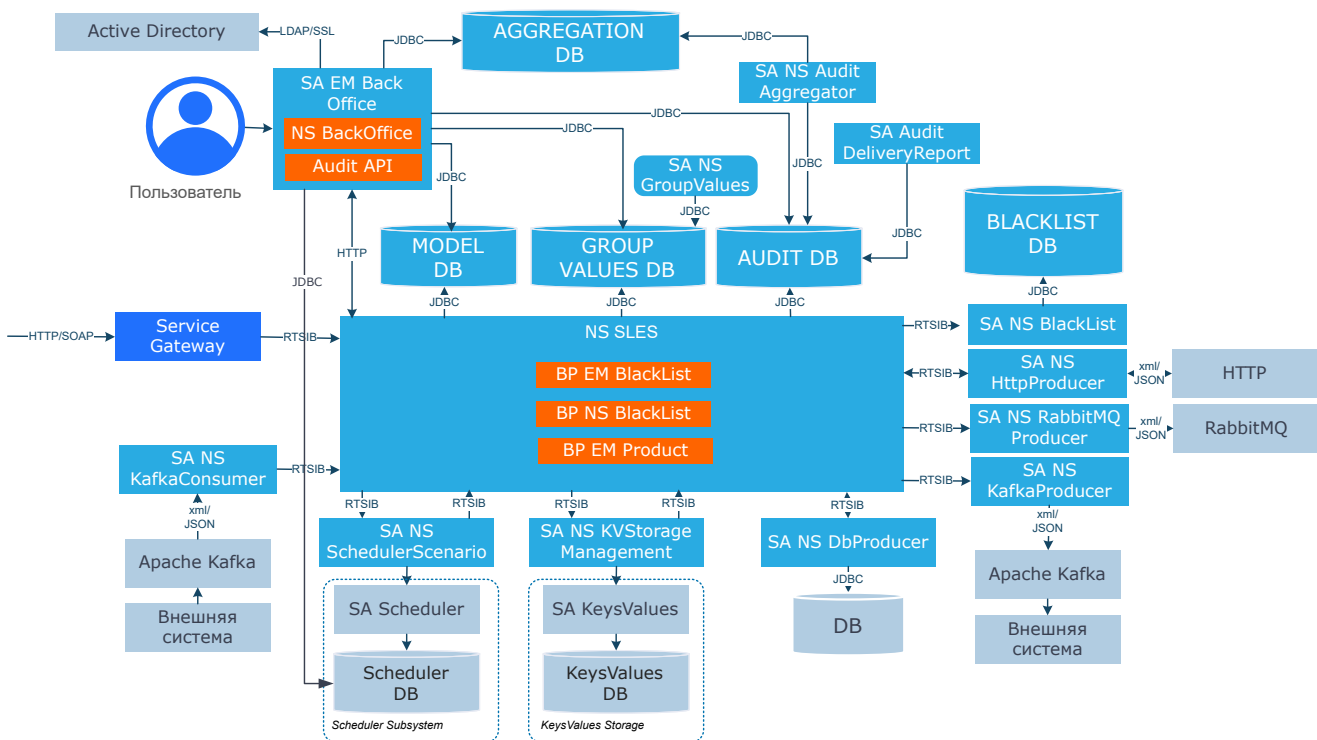


Рис. 6. Архитектура системы Notification System

Название компонента	Описание
Service Gateway	Шлюз доступа к сервисам Platform v3 для внешних приложений [1].

Название компонента	Описание
NS SLES	Компонент, который обрабатывает правила. Базируется на SLES — системе выполнения бизнес-процессов на Platform v3.
SA NS KafkaConsumer	<p>Специализированный компонент для получения и обработки входного потока событий от Apache Kafka [13].</p> <p>! Внимание! Компонент разработан специально для системы Notification System и отличается от SA KafkaConsumer системы Kafka System [12].</p>
SA NS KafkaProducer	<p>Специализированный компонент для формирования и отправки потока событий в Apache Kafka [13].</p> <p>! Внимание! Компонент разработан специально для системы Notification System и отличается от SA KafkaProducer системы Kafka System [12].</p>
SA AuditDeliveryReport	Специализированный компонент, который предоставляет интерфейс для сохранения отчетов о доставке SMS в базе данных AUDIT.
SA EM BackOffice	Специализированный компонент, который предоставляет интерфейс для работы в Nack Office [5]. При авторизации пользователей может взаимодействовать с Active Directory.
SA NS AuditAggregator	Специализированный компонент, который предоставляет интерфейс для работы с агрегированными данными из схем данных AUDIT и AGGREGATION.
SA NS BlackList	Специализированный компонент, который предоставляет интерфейс управления черными списками.
SA NS BackOffice	Специализированный компонент, который предоставляет пользовательский интерфейс управления, составления, загрузки, выгрузки и настройки сценариев системы.
SA NS DbProducer	Специализированный компонент, который предоставляет интерфейс для обращения к базам данных и получения данных из них.
SA NS GroupValues	Специализированный компонент, который предоставляет интерфейс для взаимодействия с базой данных GROUP VALUES. Используется для работы с группами через методы SOAP API.
SA NS HttpProducer	Специализированный компонент, который предоставляет интерфейс для передачи сообщений любого типа из RTSIB-шины по HTTP.
SA NS KVStorage Management	Специализированный компонент, который предоставляет интерфейс для взаимодействия с системой KeysValues Storage.
SA NS RabbitMQProducer	Специализированный компонент, который предоставляет интерфейс для передачи сообщений любого формата от компонентов системы в RabbitMQ.
SA NS ScheduledScenario	Специализированный компонент, который предоставляет интерфейс для взаимодействия с системой Scheduler Subsystem [19].
BP	<p>Бизнес-процессы — сценарии на языке BPEL, которые выполняются на Platform v3 для реализации бизнес-задач. Каждый бизнес-процесс реализует одну или несколько бизнес-операций.</p> <ul style="list-style-type: none"> ▪ Микросервисы BP EM BlackList и BP NS BlackList используются для получения информации о наличии абонента в черном списке.

Название компонента	Описание
	<ul style="list-style-type: none"> BP EM Product используется в качестве справочника для получения списка всех сценариев в веб-интерфейсе Notification System (NS BackOffice).
БД	<ul style="list-style-type: none"> PostgreSQL AUDIT — база данных для организации хранения и доступа к данным аудита, которые необходимы для реализации функций системы Notification System [4]. Oracle BLACKLIST — база данных для хранения и изменения списков номеров пользователей и масок номеров черного и белого списков различных сервисов [4]. PostgreSQL MODEL — база данных для хранения и доступа к данным модели сценариев и информации, необходимой при работе с BO Notification System.[4]. PostgreSQL GROUP VALUES — база данных для хранения и доступа к данным групп значений [4]. PostgreSQL AGGREGATION — база данных для хранения и доступа к агрегированным данным [4].

Используя данную архитектуру, можно построить типовую схему взаимодействия Notification System с другими системами компании Bercut:

- В качестве источника события выступает любая система Bercut N1.
- Микросервисы получают данные от системы-источника N2 или из базы данных.
- Полученные данные отправляются в целевую систему Bercut N3.

Пользователь управляет системой с помощью веб-интерфейса.

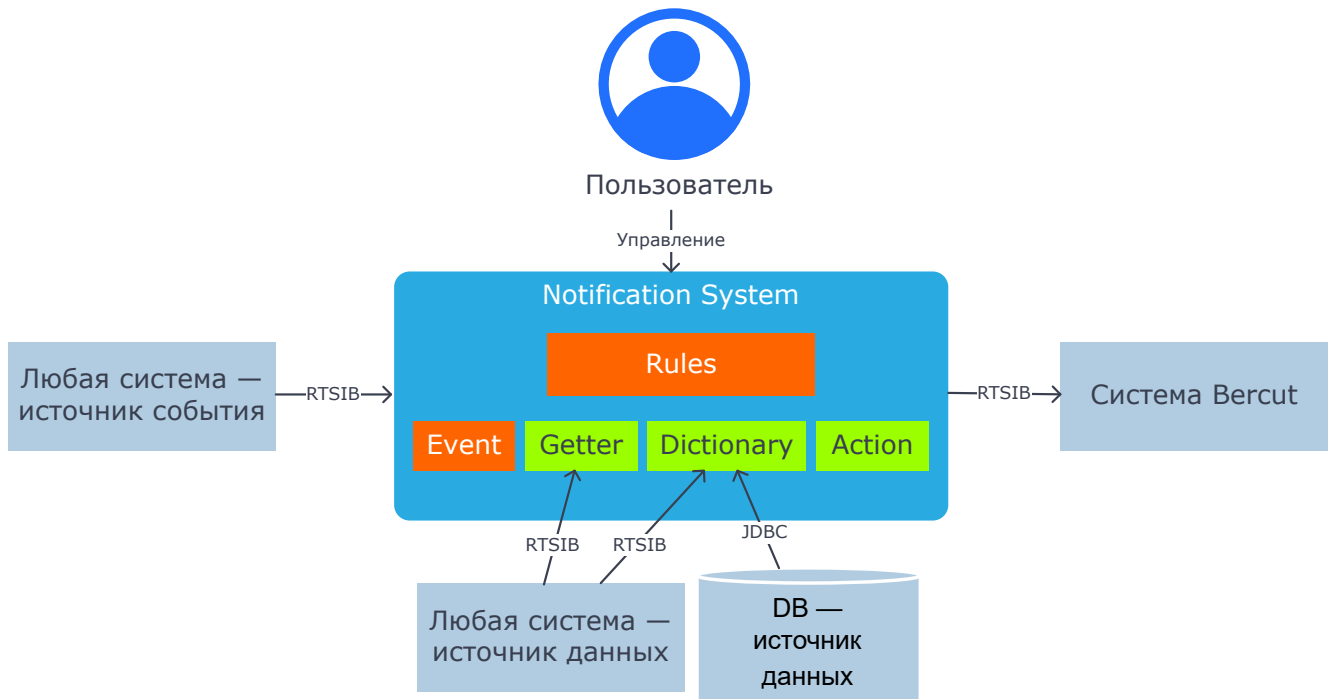


Рис. 7. Взаимодействие системы Notification System с другими системами Bercut

Название сущности	Описание
Rules	Правила сценария. Подробнее о правиле — 1.3 .

Название сущности	Описание
Event	Событие, которое поступает в Notification System из внешней системы — метод в PortType внешней системы. Подробнее — 1.2 .
Микросервисы	<p>При работе сценария вызываются микросервисы:</p> <ul style="list-style-type: none"> ▪ Getter — для определения значения критериев, входящих в правило (критерии); ▪ Action — для выполнения операций, которые входят в состав действия, предусмотренного сценарием (операции). <p>В микросервисах могут быть применены справочники (Dictionary).</p>

3. Интеграция с внешними системами

Система Notification System может взаимодействовать с системами Apache Kafka и RabbitMQ, а также отправлять сообщения на заданные URL-адреса по HTTP.

3.1. Интеграция с Apache Kafka

Для взаимодействия с Apache Kafka система использует специализированные адаптеры SA NS KafkaConsumer и SA NS KafkaProducer.

Система Notification System взаимодействует с внешними системами, используя брокер сообщений Apache Kafka [13]:

- SA NS KafkaConsumer получает сообщения из Apache Kafka в XML- или JSON-формате и преобразует их в формат RTSIB.
- SA NS KafkaProducer отправляет сообщения в Apache Kafka, преобразуя их из формата RTSIB обратно в XML- или JSON-формат.

! Внимание! Компоненты SA NS KafkaConsumer и SA NS KafkaProducer разработаны для системы Notification System и отличаются от специализированных адаптеров системы Kafka System — SA KafkaConsumer и SA KafkaProducer. Подробнее о Kafka System — [12].

Получение событий из Apache Kafka

Для получения событий из Apache Kafka [13] система Notification System использует универсальный адаптер SA NS KafkaConsumer. При преобразовании вы можете задать входной и выходной формат для преобразования.

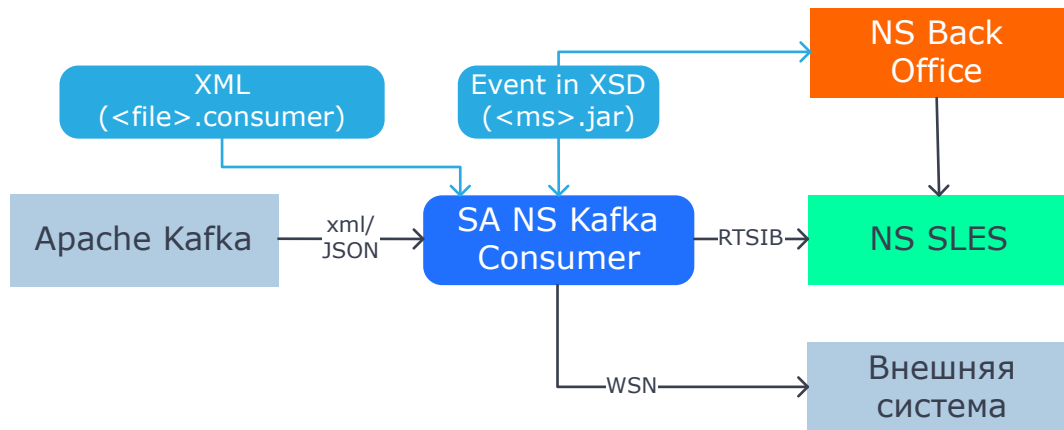


Рис. 8. Получение событий из Apache Kafka

Компонент SA NS KafkaConsumer:

1. Читает сообщения из Apache Kafka несколькими потоками. Каждый из потоков воспринимается в Apache Kafka как отдельный SA NS KafkaConsumer, но все SA NS KafkaConsumer имеют одинаковый groupId.
 - 1.1. Если система Apache Kafka недоступна, потоки периодически будут пытаться подключиться.
 - 1.2. Если получатель сообщений недоступен, SA NS KafkaConsumer приостанавливает передачу сообщений в потоках, пока получатель вновь не

станет доступен. Как только получатель становится доступен, все полученные, но необработанные сообщения, возвращаются обратно в Apache Kafka. Каждый отдельно взятый поток отправляет сообщения получателю по очереди и синхронно дожидается ответа о его обработке.

- Преобразовывает сообщения из входного формата в выходной формат. Формат входных данных описан в XML-файлах. Данные преобразуются на основании специальных XSD-файлов, которые являются частью микросервисов. Эти XSD-файлы описывают входной интерфейс сценария. XSD и XML-файлы находятся в каталоге `em_data`. Если преобразовать сообщение не удалось, оно будет отправлено без преобразования.

Примечание. Подробнее о правилах преобразования и обработке ошибок — Приложение 3

- Отправляет сообщения с помощью одного из интерфейсов:
 - специальный апу-интерфейс в Notification System. Под апу-интерфейсом понимается SOAP-интерфейс, в котором определено конкретное место, куда можно поместить апу-элемент.
 - WSN-интефейс `AOI_WSBaseNotification`, по которому сообщение отправится в компонент, который этот интерфейс имплементирует.

При отправке сообщений применяются следующие принципы:

- Если отправка сообщения не удалась из-за транспортных проблем, это сообщение и все остальные возвращаются в Apache Kafka, а поток ждет некоторое время, чтобы снова прочитать сообщения из Apache Kafka.
- Если отправка сообщения не удалась из-за бизнес-исключения, то данное сообщение будет считаться отправленным.

Отправка событий в Apache Kafka

Для отправки событий в Apache Kafka [13] система Notification System использует SA NS KafkaProducer.

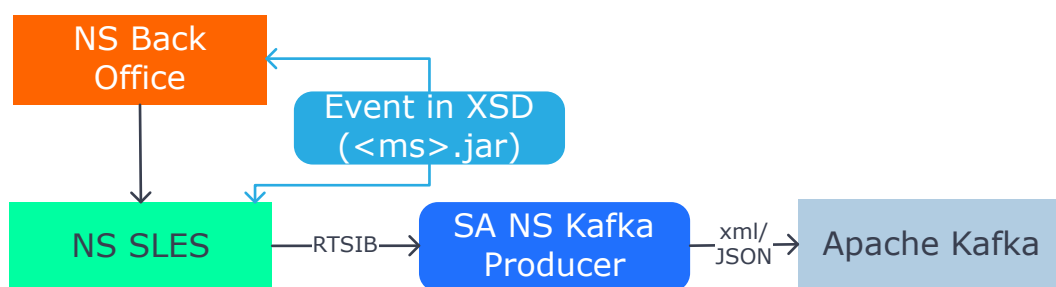


Рис. 9. Отправка событий в Apache Kafka

Компонент SA NS Kafka Producer:

- Получает сообщения через специальный апу-интерфейс. Под апу-интерфейсом понимается специальный SOAP-интерфейс, в котором определено место, куда можно положить апу-элемент.
- Преобразует сообщения из RTSIB-формата в формат JSON или XML в зависимости от настроек.
- Отправляет сообщения в Apache Kafka. Сообщения отправляются в топика, заданные в настройках. Имя топика задано в значении элемента `kafkaTopicName` или в сообщении.

i Примечание. *Имя топика, заданное в сообщении, имеет наивысший приоритет.*

4. При неудачной попытке отправить сообщение в Apache Kafka SA NS KafkaProducer приостанавливает прием сообщений. Сообщения, которые отправить в Apache Kafka не удалось, считаются утерянными.
5. При обнаружении недоступности Apache Kafka запускается поток, который начинает отправлять запросы о доступности Apache Kafka. Как только система Apache Kafka станет доступна, прием и отправка сообщений возобновляются.

3.2. Интеграция с RabbitMQ

Для взаимодействия с RabbitMQ система использует специализированный адаптер SA NS RabbitMQProducer.

Система Notification System взаимодействует с платформой RabbitMQ [22] с помощью компонента SA NS RabbitMQProducer.

SA NS RabbitMQProducer

SA NS RabbitMQProducer выполняет следующие действия:

1. Получает сообщения через WSN или через специальный апу-интерфейс. Адаптер реализует специальный интерфейс, в котором определены элементы для передачи параметра `queueName` и данных апу-типа.
2. Преобразовывает сообщения из RTSIB-формата в JSON или XML, в зависимости от настроек.
3. Отправляет сообщения в RabbitMQ в определенные очереди. Название очереди задается в значении параметра `queueName` из входящего сообщения. Если его нет, то название очереди вычисляется согласно MIB-настройкам.

Доступность RabbitMQ проверяется раз в секунду. Если RabbitMQ становится недоступен, то прием сообщений приостанавливается. Маршруту присваивается статус `unavailable`. Как только RabbitMQ становится доступен, SA NS RabbitMQProducer возобновляет прием сообщений. Маршруту присваивается статус `available`.

При неудачной попытке отправить сообщение в RabbitMQ система проверяет доступность системы RabbitMQ. Сообщения, которые отправить в RabbitMQ не удалось, будут утеряны. Информация об этом отобразится в трейсах.

3.3. Отправка HTTP-сообщений

Для отправки сообщений на URL-адрес система использует специализированный адаптер SA NS HttpProducer.

Система Notification System использует для передачи сообщений по HTTP адаптер SA NS HttpProducer, который выполняет следующие действия:

1. Получает сообщения через специальный апу-интерфейс. Адаптер реализует специальный интерфейс, в котором определен элемент для передачи данных апу-типа.
2. Преобразовывает сообщения из RTSIB-формата в JSON или XML, в зависимости от настроек.
3. Отправляет сообщения методом POST по определенному URL-адресу.

Сообщения, которые отправить не удалось, будут утеряны. Информация об этом отобразится в трейсе.

3.4. Интеграция с Scheduler Subsystem

Для выполнения отложенных сценариев необходима интеграция с Scheduler Subsystem.

В процессе выполнения отложенных сценариев система Notification System взаимодействует с системой Scheduler Subsystem.

! Внимание! Начиная с версии NS 2.10.6 для работы с SA NS ScheduledScenario требуется версия Scheduler Subsystem не ниже 3.2.

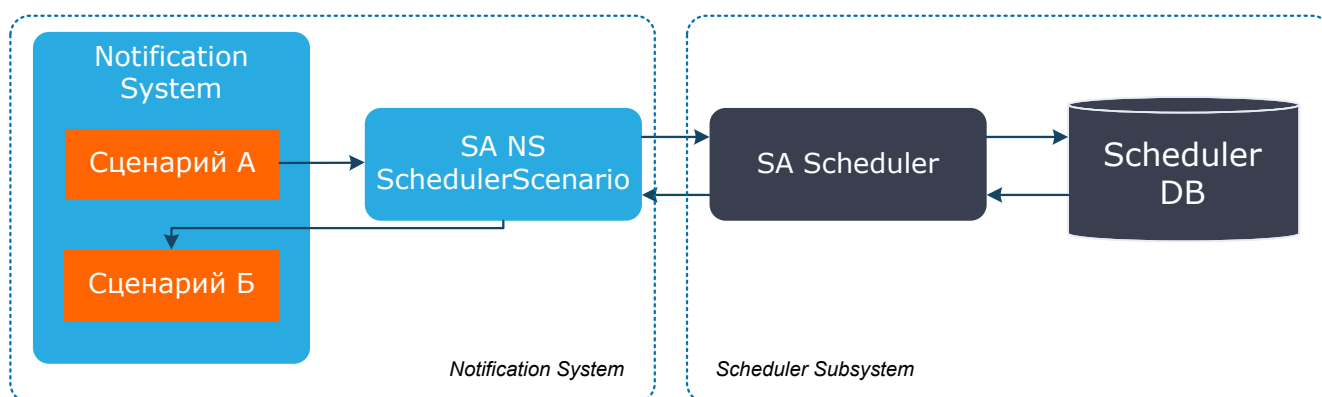


Рис. 10. Взаимодействие с Scheduler Subsystem

При создании отложенных сценариев результатом запуска сценария А является действие, которое вызывает другой сценарий (сценария Б) в заданное время и дату.

Допускается любой набор параметров сценария Б, но в нем должны присутствовать обязательные параметры.

Принцип работы системы при выполнении отложенного сценария:

1. В NS BackOffice для сценария А предварительно *настроено* действие: создание или замена отложенного сценария. При запуске сценария А запрос отправляется в SA NS SchedulerScenario.
2. SA NS SchedulerScenario переадресует запрос в SA Scheduler, передавая параметры запуска сценария.
3. SA Scheduler записывает в Scheduler DB информацию о дате и времени запуска отложенного сценария.
4. Когда наступает установленное время, Scheduler DB инициирует запуск отложенного сценария, параметры которого были ему переданы — Сценария Б.
5. SA Scheduler передает запрос в SA NS SchedulerScenario.

i Примечание. Если на шаге 4 или 5 SA Scheduler или БД Scheduler были недоступны, и вернулись к жизни в течение *MaxLifeTime(4.11)*, то отложенные сценарии будут выполнены.

6. SA NS SchedulerScenario запускает выполнение отложенного сценария Б. Сценарий Б предварительно настроен в NS BackOffice. В качестве инициирующего события для него задано выполнение отложенного сценария.
7. Сценарий Б выполняется в NS BackOffice в соответствии с настроенной таблицей правил.

4. Установка и настройка компонентов

Компоненты системы устанавливаются на серверы под управлением ОС Solaris или дистрибутива Red Hat в соответствии со схемой развертывания.

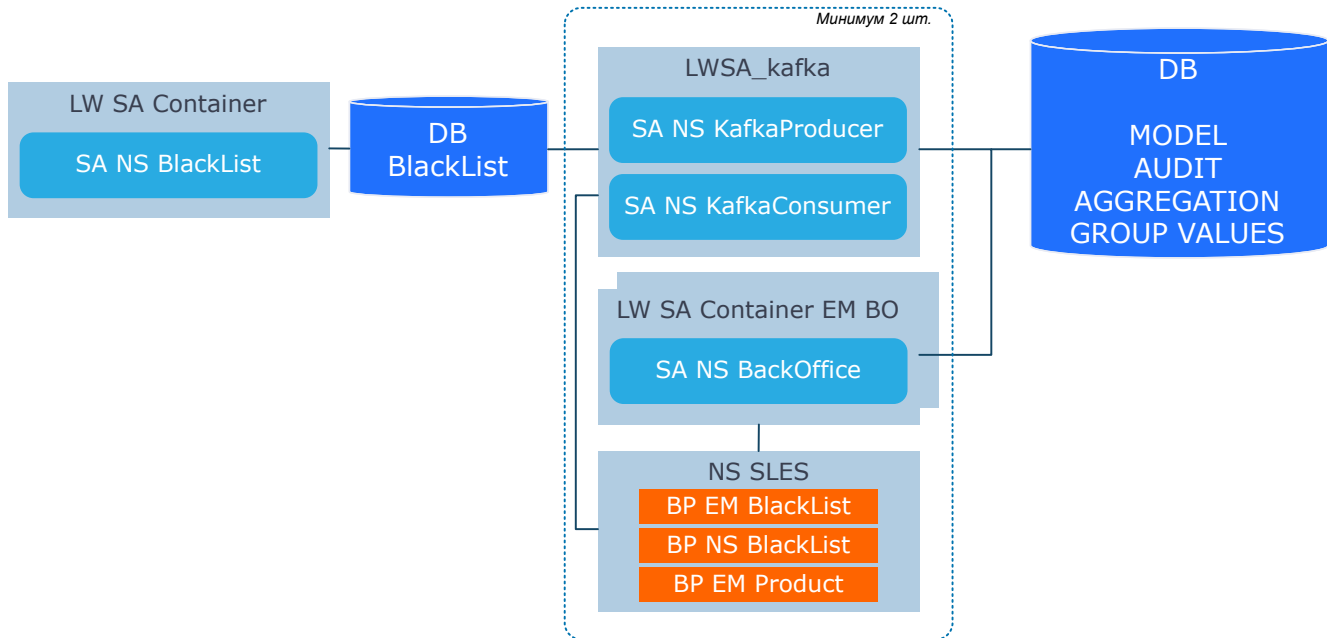


Рис. 11. Вариант типовой схемы развертывания для резервирования 2N

Предварительные действия

1. Установите Oracle JDK 1.13.X. Убедитесь, что прописана переменная окружения `JAVA_HOME`. Путь к Java указан в переменной `PATH`.
2. Установите системы ATLAS Core [9] версии 1.6.10 и выше на все серверы. Описание установки — [1].
3. Задайте в профиле пользователя `bercut` (`.bash_profile/.profile`) переменную окружения:

```
export JAVA_HOME_130=/opt/jdk13
```

4. Перезапустите SSM, чтобы изменения из шага 3 стали доступны запускаемым приложениям через SSM.

Обновление системы

Если у вас уже установлена система, то вы можете обновить ее до желаемой версии. Подробнее — 5.

4.1. Ядро системы

4.1.1. Настройка NS SLES EM

NS SLES 3.5 — модифицированная версия *SLES 3.5*, которая включает в себя движок по обработке правил Notification System.

Порядок настройки:

1. Установите компонент `NotificationSystem/ns-sles-3.5-multiplatform-release.zip` в каталог `/opt/BERCem/nssles`.
2. Задайте в файле `/opt/BERCem/ns_sles/bin/options.sh`:
 - `-Datlas.applicationName = em`
 - `OPTIONS = ${OPTIONS} " -Xms4G"`
 - `OPTIONS = ${OPTIONS} " -Xmx4G"`
3. Все настройки задайте аналогично SLES 3.5.
4. Разверните файлы `BP_*.jar` и `SE_*.jar` из каталога `NotificationSystem/deployable/sles_deployable/`.
5. Убедитесь что настроен локальный маршрут `{http://www.bercut.com/wsd/NS_InterpreterStatus}` одним из способов:
 - по умолчанию, в MIB-группе `/CMP_NsSLES_v3.5/@libs/MessageBus/Core/Configuration/Routing/DefaultScaleTable/`.
 - в списке маршрутов, в MIB-группе `/CMP_NsSLES_v3.5/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/`.

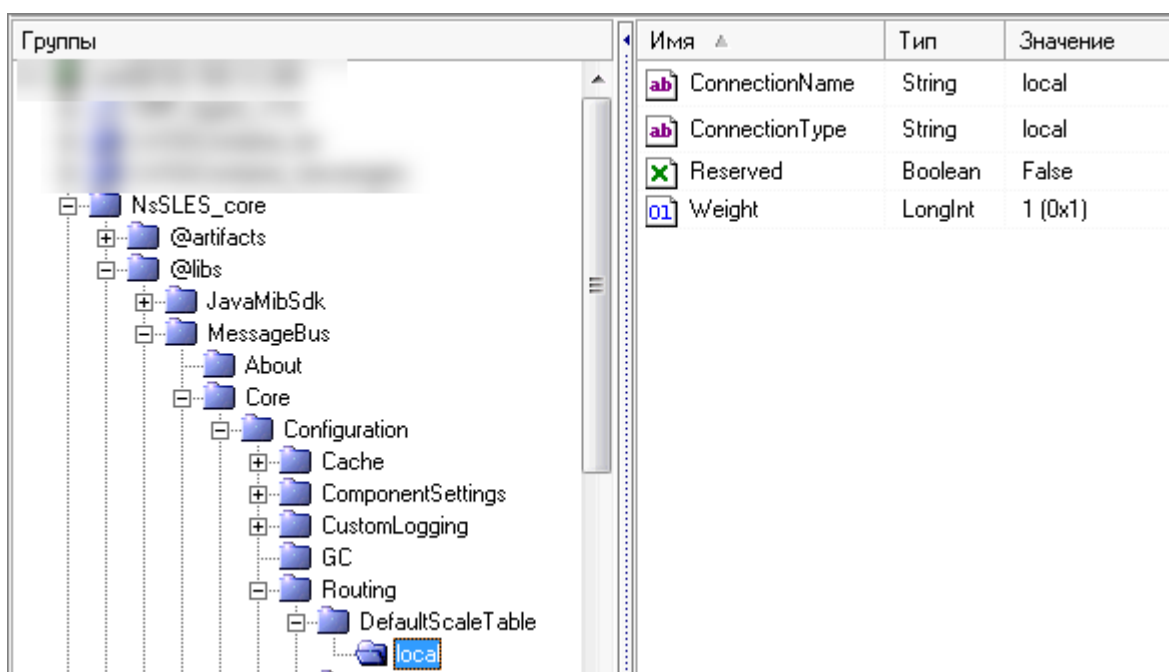


Рис. 12. Вариант настройки маршрута `{http://www.bercut.com/wsd/NS_InterpreterStatus}`

6. При необходимости разверните микросервисы системы BRE Microservices [20] и настройте маршруты — 4.12.
7. При необходимости настройте кеширование ответов на внешние вызовы — 4.16.

4.1.2. Настройка LW SA Container EM BO

Подготовка LSC

На каждый из серверов:

1. Установите компоненты LW SA Container 3.5 в каталог `/opt/BERCem/lsc_embo`.
2. В файле настроек `/opt/BERCem/lsc_embo/bin/setenv.sh`:
 - Укажите номер TCP-порта, по которому будет доступен BO Notification System: `INST_HTTP_PORT = 8100`.
 - Задайте имя контейнера `INST_NAME = embo`.

- **Задайте опции:**

- `CATALINA_OPTS=${CATALINA_OPTS}" -XX:PermSize=128m -XX:MaxPermSize=128m"`
- `CATALINA_OPTS=${CATALINA_OPTS}" -Xms1G -Xmx1G"` или более.

3. Настройте автоматический запуск компонентов ATOMS Start Stop Manager [9].

Примечание. Вы можете воспользоваться упрощенной настройкой опций Java. Используйте файл `options-usr.sh.template` в каталоге `conf`. Переименуйте этот файл в `options-usr.sh` и отредактируйте его. Заданные настройки применятся перед формированием опций запуска Tomcat.

Установка NS BackOffice

1. Скопируйте в каталог `LWSA_EMBO/webapps` файл `ROOT.war` из каталога `NotificationSystem/backoffice/`.

2. Настройте аутентификацию и авторизацию пользователей в группе `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Core/Configuration/` MIB-параметр `AuthMethod` — тип авторизации.

2.1. Для авторизации с использованием Active Directory установите `AuthMethod` в значение `ActiveDirectory`, затем настройте параметры группы `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Core/Configuration/ActiveDirectory` — 4.1.2.1.

Примечание. В случае наличия Active Directory в организации, предпочтительным является именно этот метод взаимодействия.

2.2. Для авторизации с использованием Tomcat установите `AuthMethod` в значение `TomcatRealm`. Далее настройте механизм Realms — 4.1.2.2.

Примечание. Используйте этот метод, если Active Directory отсутствует, либо ее использование невозможно. Данный механизм подразумевает настройку взаимодействия с различными Realms являющимися источниками где указываются пользователи, пароли и роли.

3. Настройте тайм-аут обновления справочников. Задайте настройки параметра `CronSchedule` в группе `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Dictionary/Configuration/`.

4. Настройте URL-префикс для BackOffice в группе `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Core/Configuration/`. MIB-параметр — `UrlPrefix`.

5. Настройте соединение с базой данных Scheduler для построения отчетов об отложенных сценариях:

- Перейдите в MIB-группу `/LWSAContainer_embo/DataSources/ConnectionsList/jdbc/scheduler/ProtocolLayers/Jdbc` и задайте значения параметров:
 - `Url`;
 - `UserName`;
 - `Password`;
 - `ConnectionType` = `postgre`.
- Перейдите в MIB-группу `/LWSAContainer_embo/Core/Configuration` и настройте значение параметра:
 - `DataSourceName` = `jdbc/scheduler`.

! Внимание! Параметр должен иметь строго такое значение, так как обращение к этому ресурсу задано внутри самих отчетов.

6. Убедитесь, что на рабочей станции с BackOffice настроен DNS. Подключение к AD должно происходить по DNS-имени. Например, вы можете добавить в `/etc/hosts` `copr.tele2.ru` и соответствующий IP.

После установки BackOffice будет доступен по адресу `http://<ip:8100>/`. Пароль, логин по умолчанию: `root, bercut`.

! Внимание! Начиная с версии 2.6, после установки BackOffice заново загрузите настройки справочников для критериев в систему. Описание настройки критериев — [5].

Настройка NS BackOffice

1. Задайте маршрут от SLES до LWSA_embo: `{http://www.bercut.com/specs/ns/EmBoProduct}EM_BO_ProductsPortType`.
2. Задайте тайм-аут `PortTypeInvokeTimeout LongInt = 15 (0xF)` в группе `../LIB_MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/{http://www.bercut.com/specs/ns/EmBoProduct}EM_BO_ProductsPortType/`.
3. Задайте маршрут от LWSA до SLES: `{http://www.bercut.com/wsd/ms/EM_Product}EM_Product`.
4. Задайте тайм-аут `PortTypeInvokeTimeout LongInt = 15 (0xF)` в группе `../LIB_MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/{http://www.bercut.com/wsd/ms/EM_Product}EM_Product/`.
5. Перегрузите справочники с помощью MIB-параметра `ReloadCache = true` в группе `/LWSA_EMBO/@artifacts/SA_EmBackOffice/Dictionary/Configuration/`.

4.1.2.1. Настройка аутентификации через Active Directory

Установите `AuthMethod` в значение `ActiveDirectory`, настройте MIB-параметры группы `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Core/Configuration/ActiveDirectory`.

Система Notification System позволяет настраивать аутентификацию с использованием Active Directory (AD) и возможностей самого сервера Tomcat через JNDIRealm, который предоставляет доступ по протоколу LDAP — 4.1.2.2.

Ввиду ограниченных возможностей взаимодействия через Realm (4.1.2.2), в NS Back Office реализована полноценная поддержка взаимодействия с AD, которая используется для:

- для аутентификации пользователей;
- для отображения списка существующих в AD пользователей;
- настройки фильтров по определенными атрибутами.

Для настройки аутентификации через Active Directory:

1. Перейдите в MIB-группу `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Core/Configuration/` и установите `AuthMethod` в значение `ActiveDirectory`.
2. Перейдите в MIB-группу `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Core/Configuration/ActiveDirectory` и настройте значения параметров:
 - `AccountNameAttribute` — значение атрибута `Account name` в AD. Значение по умолчанию: `'sAMAccountName'`.
 - `DisplayNameAttribute` — значение атрибута `Display name` в AD. Значение по умолчанию: `'displayName'`.

- *Domain* — имя контролера AD. Например, `corp.tele2.ru`.
- *EmailAttribute* — Значение атрибута *Email* в AD. Значение по умолчанию: 'userPrincipalName'.
- *MasterPasswordLoginEnabled* — задайте значение 'True', если хотите включить возможность использования мастер пароля. Значение по умолчанию: False.
- *Mode* — режим задания ролей пользователей. Значение по умолчанию: 'BO_ONLY'. Возможные значения:

- BO_ONLY — роли пользователей задаются с помощью BackOffice. В AD задается признак наличия у пользователя доступа к системе Notification System;

i Примечание. В режиме BO_ONLY необходимо для всех пользователей с доступом к системе Notification System уменьшить количество AD-ролей и сократить их до одного AD-признака Sys NS. После этого можно будет упростить фильтры. Данным признаком должны быть обозначены все пользователи, которые имеют доступ к Notification System на уровне AD. Пользователь, у которого не задан признак, не отображается в системе.

- BO_AND_AD — в AD задается признак наличия у пользователя доступа к системе Notification System. Если пользователь задан в настройках BackOffice, то используются глобальные и продуктовые роли, которые заданы в BackOffice. Если пользователь отсутствует в настройках BackOffice, то используются глобальные и продуктовые роли, которые заданы в AD. Для использования ролей из AD задайте все необходимые роли в MIB-параметрах *SearchFilterForAuth* и *SearchFilterForList*;
 - MIXED — система получает роли пользователей и из BackOffice, и из AD. Глобальные и продуктовые роли объединяются.
- *PageSize* — размер страницы для страничного запроса LDAP.
 - *Password* — пароль ;
 - *SearchFilterForAuth* — поисковое выражение фильтра, используемое для аутентификации. Значение по умолчанию: `(&(objectClass=user)(memberOf=CN=Sys NS,OU=Notification System,OU=TestUsers,OU=Accounts,DC=office,DC=bercut,DC=ru)(sAMAccountName={0}))`.
 - *SearchFilterForList* — поисковое выражение фильтра, используемое для отображение списка доступных пользователей. Значение по умолчанию: `(&(objectClass=user)(memberOf=CN=Sys NS,OU=Notification System,OU=TestUsers,OU=Accounts,DC=office,DC=bercut,DC=ru)(| (sAMAccountName={0})(displayName={0})))`.

! Внимание! В значении MIB-параметров `SearchFilterForAuth` и `SearchFilterForList` настройте фильтры LDAP таким образом, чтобы под них попадали все пользователи системы Notification System в Active Directory. Фильтры используются в AD для:

- `SearchFilterForAuth` — фильтр используется в AD для аутентификации. Поиск выполняется только по `sAMAccountName` — идентификатору пользователя. Например:

```
SearchFilterForAuth=(amp(objectClass=user) (|
(memberOf=CN=Sys NS,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
TAdmin,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
BAdmin,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
SubsServ,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
BercSupp,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS User,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
ApprUser,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(sAMAccountName={0}))
```

- `SearchFilterForList` — фильтр используется в AD для поиска и отображения списка пользователей. Поиск выполняется по параметрам:
 - `sAMAccountName` — идентификатору пользователя;
 - `displayName` — по отображаемому имени ФИО.

Например:

```
SearchFilterForList=(amp(objectClass=user) (|
(memberOf=CN=Sys NS,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
TAdmin,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
BAdmin,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
SubsServ,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
BercSupp,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS User,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(memberOf=CN=Sys NS
ApprUser,OU=Applications,OU=Groups,DC=corp,DC=tele2,DC=ru)
(| (sAMAccountName={0}) (displayName={0})))
```

! Внимание! Фильтры не должны содержать переносов на новые строки и/или лишних пробелов, например между скобками — фильтр должен быть одной сплошной строкой. В противном случае от AD будут приходить ошибки в трейсах. Например: `Active Directory error. Explanation: Unbalanced parenthesis. Cause: null. Stacktrace: javax.naming.directory.InvalidSearchFilterException: Unbalanced parenthesis; remaining name 'DC=corp,DC=tele2,DC=ru'.`

- `TotalLimit` — ограничение выдаваемого списка пользователей. Если указано значение 0 — без ограничения. Значение по умолчанию: 50.
- `UseSsl` — установите в значение True, если вы хотите использовать LDAP over SSL. Значение по умолчанию: False.
- `User` — имя пользователя в AD.

3. Перейдите в MIB-группу `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Core/Configuration/ActiveDirectory/SASL` и задайте настройки Simple Authentication and Security Layer. В значении параметра `QOP` укажите качество защиты (Quality Of Protection):

- AUTH — только аутентификация;
- AUTH_INT — аутентификация и защита целостности;
- AUTH_CONF — аутентификация, защита целостности и конфиденциальности.

Значение по умолчанию: AUTH.

4. Для шифрование HTTP-соединений используется механизм SSL. Если вы установили `UseSsl=True`, установите сертификат на сервер и настройте доступ в AD по SSL.

На сервере откройте каталог `LW SA Container` на котором развернута `SA_EmBackOffice` и перейдите в подгруппу `JDK -> JRE -> bin`. Запустите команду импорта сертификата `keytool -import`.

Для определения расположения JDK вы можете использовать переменную окружения `$JAVA_HOME`. Команда будет выглядеть так:

```
<$JAVA_HOME>/jre/bin/keytool -import -storepass <arg> -keystore <keystore> -alias <alias> -file <filename>
```

где:

- `storepass <arg>` — пароль хранилища ключей;
- `keystore <keystore>` — имя хранилища ключей. По умолчанию задайте `<$JAVA_HOME>/jre/lib/security/cacerts`;
- `alias <alias>` — псевдоним;
- `file <filename>` — файл сертификата.

Пример настройки `SearchFilterForAuth` и `SearchFilterForList`

Back Office реализует собственный механизм авторизации и соответственно хранит информацию по ролям пользователей в собственной БД, поэтому нет необходимости хранения этой информации в атрибутах `memberOf` пользователей в Active Directory. Достаточно наличия только одного признака для фильтрации пользователей системы NS — `Sys NS`.

После удаления информации о продуктах и ролях из атрибутов `memberOf`, у каждого пользователя системы будет один атрибут, который относится к системе NS — `memberOf=CN=Sys NS`, и вы можете использовать значения переменных `SearchFilterForAuth` и `SearchFilterForList` по умолчанию:

- `SearchFilterForAuth=(&(objectClass=user)(|(memberOf=CN=Sys NS,OU=Notification System,OU=TestUsers,OU=Accounts,DC=office,DC=bercut,DC=ru))(sAMAccountName={0}))`
- `SearchFilterForList=(&(objectClass=user)(|(memberOf=CN=Sys NS,OU=Notification System,OU=TestUsers,OU=Accounts,DC=office,DC=bercut,DC=ru))(sAMAccountName={0})(displayName={0})))`

Пример настройки доступа в AD по SSL

Например:

- используется `JAVA_HOME=/usr/java/jdk1.7.0_171-amd64`;
- создается псевдоним `certnew`;
- в качестве файла сертификата используется `/home/bercut/certnew.cer`;
- Пароль хранилища ключей никогда не менялся, поэтому используется пароль по умолчанию `changeit`.

Пример команды импорта сертификата:

```
/usr/java/jdk1.7.0_171-amd64/jre/bin/keytool -import -storepass changeit -keystore "/usr/java/jdk1.7.0_171-amd64/jre/lib/security/cacerts"
```

```
-alias certnew -file /home/bercut/certnew.cer
```

Пример команды выгрузки списка установленных сертификатов:

```
<$JAVA_HOME>/jre/bin/keytool -v -list -keystore <keystore>
```

где:

- `keystore <keystore>` — имя хранилища ключей.

По умолчанию задайте `<$JAVA_HOME>/jre/lib/security/cacerts`.

Для выгрузки списка установленных сертификатов в файл `cacerts_list` выполните команду:

```
/usr/java/jdk1.7.0_171/jre/bin/keytool -v -list -keystore "/usr/java/jdk1.7.0_171/jre/lib/security/cacerts" > cacerts_list
```

4.1.2.2. Настройка аутентификации через Tomcat

Установите `AuthMethod` в значение `TomcatRealm`, настройте взаимодействие с различными `Realms`.

BackOffice поддерживает аутентификацию встроенную в сервер приложений Tomcat, которая реализуется через механизм `Realm`. Подробнее о механизме — [24].

Для работы механизма `Realm` настройте взаимодействие с различными `Realms`: укажите пользователей, пароли и роли.

Для настройки аутентификации через Tomcat:

1. Перейдите в MIB-группу `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/Core/Configuration/` и установите `AuthMethod` в значение `TomcatRealm`.
2. Настройте один или несколько `Realm`:

2.1. `UserDatabaseRealm`.

По умолчанию в LW SA Container уже настроен `UserDatabaseRealm`. Он использует ресурс JNDI для хранения пользовательской информации. По умолчанию ресурс JNDI поддерживается XML-файлом `{tomcat_home}/conf/tomcat-user.xml`. `UserDatabaseRealm` не предназначен для использования в продуктиве.

В файле `{tomcat_home}/conf/server.xml` он уже настроен следующим образом:

```
<Realm className="org.apache.catalina.realm.LockOutRealm">
<!-- This Realm uses the UserDatabase configured in the global JNDI
resources under the key "UserDatabase". Any edits
that are performed against this UserDatabase are immediately
available for use by the Realm. -->
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>
</Realm>
```

`UserDatabaseRealm` определяется внутри `LockOutRealm`, который предназначен для ограничения неудачных попыток аутентификации за определенный промежуток времени. Откройте на редактирование файл `{tomcat_home}/conf/tomcat-user.xml` и добавьте пользователей. Например:

```
<user username="test_user1" password="test_password1"/>
<user username="test_user2" password="test_password2"/>
```

После внесения изменений в файл `tomcat-user.xml` перезагрузите LW SA Container EM BO.

2.2. `DataSourceRealm`.

DataSourceRealm используется для поиска пользователей в реляционной базе данных, доступ к которой осуществляется пул соединений JDBC DataSource, что требует его настройки в файлах конфигурации {tomcat_home}/conf/server.xml и {tomcat_home}/conf/context.xml.

В БД создайте таблицу пользователей вида:

```
create table users (  
  user_name      varchar(15) not null primary key,  
  user_pass      varchar(255) not null  
);
```

Название таблицы и полей задаются с помощью атрибутов userTable, userNameCol, userCredCol в элементе Realm. Сама таблица может содержать и другие поля. В файле {tomcat_home}/conf/server.xml настройте **DataSourceRealm** следующим образом:

```
<Realm className="org.apache.catalina.realm.DataSourceRealm"  
  dataSourceName="jdbc/authority"  
  userTable="users" userNameCol="user_name" userCredCol="user_pass"  
  allRolesMode=authOnly/>
```

Задавать атрибуты, определяющие таблицу ролей пользователей (userRoleTable) и соответствующее поле имен ролей (roleNameCol) не требуется, как и создавать в БД саму таблицу с ролями. Notification System не использует простой стандартный механизм авторизации Java Web Specification. Атрибут allRolesMode=authOnly указывает что используется только аутентификация.

2.3. JDBC Database Realm.

JDBC Database Realm, как и **DataSourceRealm**, используется для поиска пользователей в реляционной базе данных. Но в отличие от него не использует пул соединений и не требует его настройки в файлах конфигурации {tomcat_home}/conf/server.xml и {tomcat_home}/conf/context.xml.

! Внимание! Использовать **JDBC Database Realm** не рекомендуется, из-за его ограничений по сравнению с **DataSourceRealm**, в новых версиях Tomcat он не поддерживается.

JDBC Database Realm использует следующие атрибуты, определяющие соединение с БД: connectionName, connectionPassword, connectionURL, driverName.

В файлах {tomcat_home}/conf/server.xml и {tomcat_home}/conf/context.xml настройте **JDBC Database Realm** следующим образом:

```
<Realm className="org.apache.catalina.realm.JDBCRealm"  
  driverName="org.gjt.mm.mysql.Driver"  
  connectionURL="jdbc:mysql://localhost/authority?user=dbuser&password=dbpass"  
  userTable="users" userNameCol="user_name" userCredCol="user_pass"  
  allRolesMode=authOnly/>
```

В остальном настройки **JDBC Database Realm** аналогичны **DataSourceRealm**.

4.1.3. Настройка LW SA Container

Установите LW SA Container, если вы будете использовать необязательные функциональности, например:

- [AuditDeliveryReport](#);
- [SA_NS KafkaConsumer](#) или [SA_NS KafkaProducer](#) для взаимодействия с Apache Kafka;
- [HTTP-адаптер SA_NS HttpProducer](#);

- *RabbitMQ-адаптер*;
- *SA NS GroupValues для работы с группами*;
- *SA NS ScheduledScenario для взаимодействия с подсистемой Scheduler Subsystem*.

i Примечание. Для обеспечения работоспособности рекомендуется устанавливать отдельный LW SA Container для каждой из используемых функциональностей.

Порядок установки и настройки:

1. Установите компоненты LW SA Container 3.5 в каталог /opt/BERCem/<имя LWSA>.
2. В файле настроек /opt/BERCem/<имя LWSA>/conf/setenvuser.sh задайте:
 - Уникальный TCP-порт, по которому будет доступен BO Notification System: `INST_HTTP_PORT = 8100`.
 - `APP_NAME = <имя LWSA>`.
3. Настройте автоматический запуск компонентом ATOMS Start Stop Manager [9].

4.1.4. Установка и настройка схем базы данных

Установите и настройте схемы данных *AUDIT*, *MODEL*, *GROUP VALUES* и *AGGREGATION*.

Установите и настройте схемы данных.

Для установки БД требуется PostgreSQL версии 12 и выше.

1. Установите схему данных. Инструкция по установке базы находится в файле `Readme.txt` в соответствующем разделе каталога `NotificationSystem\db_scripts\<Имя БД>\Scheme\`.
2. Настройте на BackOffice подключение к БД в следующих переменных:
 - 2.1. `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/<Имя БД>/Configuration/DbConnection`.
 - 2.2. `/LWSAContainer_embo/DataSources/ConnectionsList/<jdbc-имя источника данных>/ProtocolLayers/Jdbc/:`
 - 2.2.1. *Url*
 - 2.2.2. *UserName*
 - 2.2.3. *Password*
 - 2.2.4. *ConnectionType* = `postgre`.
3. Настройте подключение к базе на Interpreter:
 - 3.1. `NsSLES_em/@libs/NsInterpreter/<Имя БД>/Configuration/DbConnection/:`
 1. *ApplicationName*
 2. *Login*
 3. *Password*
 4. *URL*
 5. *MaxPoolConnections*.
4. При необходимости настройте резервирование. В MIB-группах укажите настройки подключения:
 - 4.1. `/LWSAContainer_embo/DataSources/ConnectionsList/<jdbc-имя источника данных>/ProtocolLayers/Jdbc/` в значении параметра *Url*.
 - 4.2. `/NsSLES/@libs/NsInterpreter/<Имя БД>/Configuration/DbConnection/` в значении параметра *Url*.

Настройки укажите в формате:

```
jdbc:postgresql://xx.xx.xx.x1:5432,xx.xx.xx.x2:5432/db?targetServerType=primary
```

где:

- xx.xx.xx.x1 — IP-адрес основного сервера;
- xx.xx.xx.x2 — IP-адрес резервного сервера;
- db — имя БД.

Пример:

```
jdbc:postgresql://192.168.0.1:5432,192.168.0.2:5432,192.168.0.3:5432/db_model?  
targetServerType=primary
```

4.1.4.1. Установка и настройка схемы данных DB AUDIT

Для установки БД требуется PostgreSQL версии 12 и выше.

Порядок работы с DB AUDIT:

1. **Установите** схему данных. Каталог установочного дистрибутива для схемы данных NotificationSystem\db_scripts\audit_pg12\Schema\.
 2. Настройте подключение к базе на BackOffice:
 - 2.1. /LWSAContainer_embo/@artifacts/SA_EmBackOffice/Audit/Configuration/DbConnection
 - 2.1.1. *JndiDataSource* = `jdbc/auditdb`
 - 2.2. Присвойте параметру /LWSAContainer_embo/DataSources/ConnectionsList/jdbc&2fauditdb/ProtocolLayers/Jdbc/RemoveAbandonedTimeout значение `300`.
 3. Настройте подключение к базе на Interpreter:
 - 3.1. Присвойте параметру NsSLES_em/@libs/NsInterpreter/Audit/Configuration/DbConnection/MaxPoolConnections значение `20`.
 - 3.2. NsSLES_em/@libs/NsInterpreter/Audit/Configuration
 - 3.2.1. *PersistThreadsCount* = `10`
 - 3.2.2. *AuditRuleIsOn* = `false`.
- ! Внимание!** Количество *PersistThreadsCount* должно быть меньше количества, заданного в параметре *MaxPoolConnections*.
4. При необходимости включите трассировку стека ошибок. Для этого перейдите в MIB-группу NsSLES_em/@libs/NsInterpreter/Audit/Configuration/ и присвойте параметру *AuditStackTraceIsOn* значение 'true'. По умолчанию параметр имеет значение 'false' для оптимизации хранения данных аудита.
 5. При необходимости настройте **резервирование** в группах:
 - /LWSAContainer_embo/DataSources/ConnectionsList/jdbc/auditdb/ProtocolLayers/Jdbc/;
 - /NsSLES/@libs/NsInterpreter/Audit/Configuration/DbConnection/.

4.1.4.1.1. Секционирование таблиц DB AUDIT

Для таблиц базы данных DB AUDIT используется декларативное секционирование.

Секционирование таблиц выполняется по диапазону — PARTITION BY RANGE.

На первом уровне таблицы секционирование выполняется по дням. На втором — секционирование дневных секций выполняется по часам. Структура секций соответствует структуре основной таблицы, для которой эти секции создаются.

i Примечание. Полная информация по секционированию — [4].

Создание секций с помощью функции `manage_partitions()`

Секции создаются и удаляются автоматически с помощью функции `manage_partitions()`, которая запускается по расписанию стандартным планировщиком заданий `cron`. Эта функция использует параметры из таблицы `NS_AUDIT_SETTINGS` схемы данных `AUDIT`:

- `NEXT_DAYS_FOR_CREATE` — количество дней с текущей даты, на которое система пытается заранее создать секции. Значение по умолчанию: '10'.
- `MAX_STORAGE_PERIOD` — количество дней, в течение которых будут храниться записи. Значение по умолчанию: '100'.
- `PREVIOUS_DAYS_FOR_REMOVE` — количество дней после срока хранения, на которое система пытается удалить секции. Значение по умолчанию: '10'.
- `MAIN_TABLESPACE` — табличное пространство для таблиц секций.
- `INDX_TABLESPACE` — табличное пространство для индексов секций.

! Внимание! Для эффективной работы индексов и оптимизатора запросов к секционированным таблицам задайте параметр настройки `CONSTRAINT_EXCLUSION = PARTITION` (актуален для PostgreSQL 10). Данный параметр находится в разделе `QUERY TUNING` конфигурационного файла `postgresql.conf`. Дополнительная информация в открытом доступе: <https://www.postgresql.org/docs/10/static/ddl-partitioning.html#DDL-PARTITIONING-CONSTRAINT-EXCLUSION>.

Путь к файлу `postgresql.conf` можно определить с помощью PSQL-команды:

```
SHOW config_file
```

После изменения конфигурационного файла в `bash` под пользователем `postgres` перезагрузите конфигурацию с помощью команды:

```
pg_ctl reload
```

4.1.4.1.2. Настройка поиска по критериям в DB AUDIT

Пользователь работает с критериями в `BO Notification System`. Первичную настройку критериев выполняет администратор.

Настройка определяют набор критериев, по которым будет работать быстрый поиск (построение индексов) в `BO Notification System`.

Для настройки используйте следующую команду `INSERT` при работе с таблицей `NS_AUDIT_DESCRIPTION` в `DB AUDIT`:

```
INSERT INTO ns_audit_description(
    id, model_event_id, description)
VALUES (1, '1', '{"auditDescription": [{
    "id": "msisdnSet",
    "name": "Поиск по MSISDN",
    "parameters": [{
    "table": "ns_audit_event",
    "column": "request",
    "criteria": "msisdn",
    "displayName": "MSISDN"
    }]
}]})');
```

где:

- id — любое название, отражающие суть фильтра;
- name — название, которое отображается пользователю в Back Office;
- parameteres — список параметров, по которым выполняется фильтрация. Может быть несколько критериев.
- table — ns_audit_event;
- column — request;
- criteria — название критерия в XSD или JSON, для которого нужно делать фильтрацию;
- displayName — название для критерия для пользователя в Back Office.

Примечание. Идентификатор события MODEL_EVENT_ID таблицы NS_AUDIT_DESCRIPTION должен быть равен идентификатору события ID таблицы NS_MODEL_EVENT.

Пример отображения критерия в Back Office:

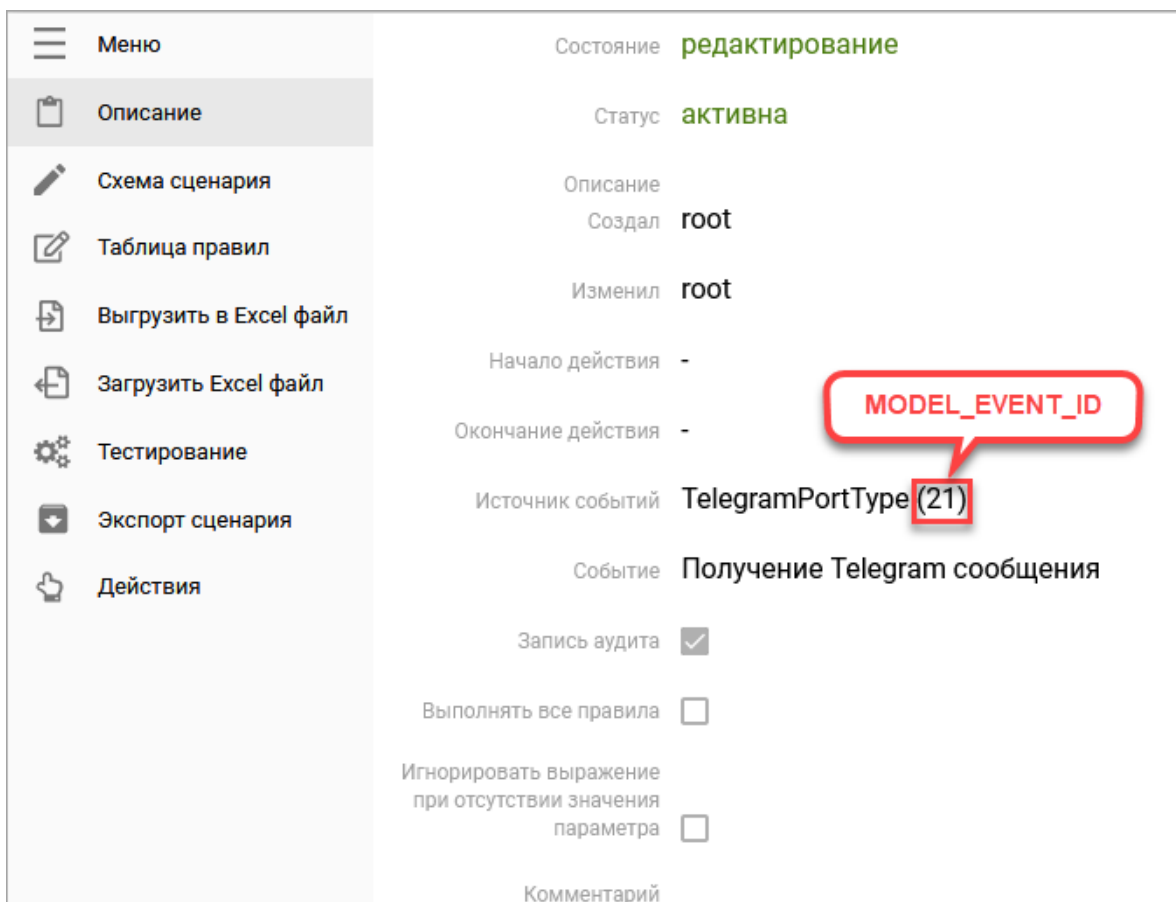


Рис. 13. Отображение MODEL_EVENT_ID для сценария в Back Office

4.1.4.1.3. Настройка расширенного аудита

Для настройки расширенного аудита настройте маршрут и задайте MIB-параметры группы /LWSAContainer_embo/@artifacts/SA_EmBackOffice/Audit/Configuration/.

Для настройки расширенного аудита:

1. Задайте маршрут от LWSA_embo к SA Messaging: {http://www.bercut.com/specs/ao/sal/messaging}MessagingPortType. Это необходимо для отправки отчетов по электронной почте.

2. Перейдите в MIB-группу `/LWSAContainer_embo/@artifacts /SA_EmBackOffice/Audit/Configuration/` и задайте значения параметров:

- `ReportMaxRecordsCountPerFile` — максимальное количества записей в каждом файле отчета аудита. Значение по умолчанию: '100000'. Минимальное значение: '10'. Максимальное значение: '1000000'.
- `ReportQueryExecutorsCount` — количество обработчиков, используемых для чтения очереди заданий аудита. Параметр используется для настройки одновременной обработки нескольких запросов. Значение по умолчанию: '1'. Минимальное значение: '1'. Максимальное значение: '20'.

i Примечание. Для применения изменений перезапустите систему.

- `ReportTasksSchedulerPeriod` — интервал проверки наличия новых заданий в очереди аудита. Задается в секундах. Значение по умолчанию: '60'. Минимальное значение: '100000'.

i Примечание. Для применения изменений перезапустите систему.

- `ReportsNotificationsSenderAddress` — адрес электронной почты, с которого отправляется нотификация о том, что отчет аудита готов. Значение по умолчанию: 'bre@office.bercut.com'.
- `ReportsStoragePeriod` — время хранения отчетов аудита. Задается в днях. Значение по умолчанию: '2'. Минимальное значение: '1'. Максимальное значение: '100'.
- `SaveReportFiles` — сохранение отчетов на диске. Значение по умолчанию: 'false'.
- `SaveReportSqlDetailsFiles` — сохранение информации об SQL-запросах отчетов на диске. Значение по умолчанию: 'false'.

4.1.4.2. Установка и настройка схемы данных DB MODEL

Для установки БД требуется PostgreSQL версии 12 и выше.

Порядок работы с DB MODEL:

1. **Установите** схему данных. Каталог установочного дистрибутива для схемы данных `NotificationSystem\db_scripts\data_model_pg12\Scheme\`.
2. Настройте подключение к базе в BackOffice `/LWSAContainer_embo/@artifacts/SA_EmBackOffice/EMRepo/Configuration/DbConnectionJndiDataSource = jdbc/modeldb`.
3. Настройте подключение к базе в interpreter в группе `NsSLES_em/@libs/NsInterpreter/EMRepo/Configuration/DbConnection/`.
4. При необходимости настройте **резервирование** в группах:
 - `/LWSAContainer_embo/DataSources/ConnectionsList/jdbc/modeldb/ProtocolLayers/Jdbc/;`
 - `/NsSLES/@libs/NsInterpreter/EMRepo/Configuration/DbConnection/.`

4.1.4.3. Установка и настройка схемы данных DB GROUP VALUES

Для установки БД требуется PostgreSQL версии 12 и выше.

Порядок работы с DB GROUP VALUES:

1. **Установите** схему данных. Каталог установочного дистрибутива для схемы данных `NotificationSystem\db_scripts\group_values_pg12\Scheme\`.

2. Настройте подключение к базе в BackOffice /LWSAContainer_embo/@artifacts/SA_EmBackOffice/GroupValues/Configuration/DbConnectionJndiDataSource = `jdbc/groupsdb`
3. Настройте подключение к базе в interpreter в группе NsSLES_em/@libs/NsInterpreter/GroupValues/Configuration/DbConnection/.
4. При необходимости настройте *резервирование* в группах:
 - /LWSAContainer_embo/DataSources/ConnectionsList/jdbc/groupsdb/ProtocolLayers/Jdbc/
 - /NsSLES/@libs/NsInterpreter/GroupValues/Configuration/DbConnection/;

При настройке обратите внимание на значения параметров:

- NsSLES_em/@libs/NsInterpreter/GroupValues/Configuration/DbConnection/:
 - *MaxLifeTime* — время обновления соединения при подключении к DB GROUP VALUES. Если одна из БД выходит из строя, то повторное переключению к ней зависит от этого значения.
 - *MaxPoolConnections* — при большом количестве экземпляров DB GROUP VALUES рекомендуется увеличить это значение. Данный параметр ограничивает максимальное количество соединений к БД.
 - *URL* — URL может быть как единичный, так и множественный, если группа БД работают по принципу *master-slave*. Пример использования нескольких БД:

```
jdbc:postgresql://192.168.1.1:5432,192.168.1.2:5432,192.168.1.3:5432/  
db_groupvaluesloaddb?readOnly=true
```

Настройка `readOnly=true` необязательна.

- NsSLES_em/@libs/NsInterpreter/GroupValues/Configuration/:
 - *GroupValuesQueueSize* — очередь обработки запросов к DB GROUP VALUES. При достижении 80 % устанавливает `hwm` для маршрута входящего `portType`. При понижении до 60 % — снова устанавливает в `lwm`.
 - *GroupValuesThreadCount* — количество потоков, обрабатывающих запросы к DB GROUP VALUES.

4.1.4.4. Установка и настройка схемы данных DB AGGREGATION

Для установки БД требуется PostgreSQL версии 12 и выше.

Порядок работы с DB AGGREGATION:

1. *Установите* схему данных. Каталог установочного дистрибутива для схемы данных NotificationSystem\db_scripts\audit_aggregation_pg12\Schema\.
2. Настройте подключение к базе данных в Back Office
 - 2.1. /LWSAContainer_embo/@artifacts/SA_EmBackOffice/Aggregator/Configuration/DbConnection
 - 2.1.1. *JndiDataSource* = `jdbc/aggr`.
 - 2.2. /LWSAContainer_embo/DataSources/ConnectionsList/jdbc/aggr/ProtocolLayers/Jdbc
 - 2.2.1. *InitSql* = установка значения '0' для `set idle_in_transaction_session_timeout`.
3. При необходимости настройте *резервирование* в группе:
 - /LWSAContainer_embo/DataSources/ConnectionsList/jdbc/ns_audit_aggregation/ProtocolLayers/Jdbc/.
4. Установите инструкции для отчетов:

4.1. Перейдите в каталог `NotificationSystem\db_scripts\aggr_instructions` и в файле `install.ini` задайте параметры подключения к базе данных.

4.2. Для установки инструкций используйте один из следующих вариантов:

4.2.1. Из каталога `NotificationSystem\db_scripts\aggr_instructions` выполните команду, в зависимости от используемой ОС: `./install_all.sh` или `./install_all.bat`.

4.2.2. Из каталога `NotificationSystem\db_scripts\aggr_instructions\installer` выполните команды:

- `./install.sh ../scenario.xml`
- `./install.sh ../product.xml`
- `./install.sh ../micro_service.xml`
- `./install.sh ../event.xml`
- `./install.sh ../action.xml`
- `./install.sh ../scheduled_event.xml`

При необходимости настройте отчеты для отображения агрегированных данных в NS Back Office — [4.13](#).

4.2. Настройка AuditDeliveryReport

Компонент `AuditDeliveryReport` используется для сохранения в DB AUDIT отчетов о доставке SMS-сообщений. Данная установка не является обязательным требованием для функционирования системы `Notification System`.

Порядок настройки:

1. Скопируйте файл `SA_AuditDeliveryReport_v1.war` в каталог `webapps - NotificationSystem/deployable/lwsa_deployable/`.

2. Настройте подключение к базе на `AuditDeliveryReport`:

2.1. `/LWSAContainer/@artifacts/SA_AuditDeliveryReport/Audit/Configuration/DbConnection` (или задать `JndiDataSource`)

2.1.1. *URL*

2.1.2. *Login*

2.1.3. *Password*

2.2. `/LWSAContainer/@artifacts/SA_AuditDeliveryReport/Audit/Configuration/DbConnection`

2.2.1. *MaxPoolConnections* = `20`

2.3. `/LWSAContainer/@artifacts/SA_AuditDeliveryReport/Audit/Configuration`

2.3.1. *PersistThreadsCount* = `10`

4.3. Установка BlackList

Компоненты используются для получения информации о наличии абонента в черном списке.

Данная установка не является обязательным требованием для функционирования системы `Notification System`.

4.3.1. Установка и настройка SA NS BlackList

Установка JAR-файлов

1. Разверните:

- 1.1. BP_NS_BlackList_v1.jar на SLES NotificationSystem/deployable/sles_deployable/.
- 1.2. BP_EM_BlackList_v1.jar на SLES NotificationSystem/deployable/sles_deployable/.

2. Скопируйте в каталог LWSA_embo/em_data файл NotificationSystem/deployable/nsbo_deployable/BP_EM_BlackList_v1.jar.

Установка SA

Скопируйте в каталог LWSA/webapps файл NotificationSystem/deployable/lwsa_deployable/SA_NS_BlackList_v1.war.

4.3.2. Установка и настройка схемы данных DB BLACKLIST

Для установки БД требуется Oracle версии 12 и выше.

Порядок работы с DB BLACKLIST:

1. Заполните параметры в файле db_scripts\black_list\oracle\config.sql.
2. Запустите db_scripts\black_list\oracle\install.bat.
3. Заполните параметры в файле NotificationSystem\db_scripts\black_list\oracle\config.sql.
4. Запустите файл install.bat из каталога NotificationSystem\db_scripts\black_list\oracle\.
5. Настройте подключение к базе на lsc_em:
 - 5.1. /LWSAContainer/@artifacts/SA_NsBlackList_v1.0/DataBaseConnections/ConnectionsList/blacklist/ProtocolLayers/JDBC/
 - 5.1.1. *DataSourceName* = jdbc/blacklist
 - 5.2. /LWSAContainer/DataSources/ConnectionsList/jdbc&2fblacklist/ProtocolLayers/Jdbc
 1. *JndiDataSource*
 2. *UserName*
 3. *Password*
 4. *ConnectionType* = oracle

4.3.3. Настройка Service Gateway

Для доступа к API BlackList используется Service Gateway. Установите этот компонент или используйте существующий.

1. Опубликуйте следующие интерфейсы:

- {http://www.bercut.com/wsd/EM_BL}EM_BL_Call
- {http://www.bercut.com/wsd/EM_BL}EM_BL_Admin

2. Настройте маршруты для указанных PT до NsSLES.

3. Добавьте маршрут от LWSA_embo до SA NS BlackList: {http://www.bercut.com/wsd/DBBlackList-1}BlPkg.

4.4. Настройка Audit API

Настройка получения аудита через API. Используется для получения данных из аудита для внешних систем. Данная установка не является обязательным требованием для функционирования системы Notification System.

Вы можете получить данные аудита через веб-интерфейс на вкладке «Аудит» или через SOAP-запрос.

Задайте время ожидания для запроса в MIB-параметре `OperationInvokeTimeout` = 310 группы `/NsSLES_em/@libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/{http://www.bercut.com/specs/ns/NsAudit}NsAuditPortType/OperationsTimeout/findAuditEvents/`.

4.5. Настройка System Principal

Настройка System Principal используется при вызове внешних партнеров.

Когда внешняя система вызывает Notification System, она передает параметр **System Principal** = <наименование партнера>. Notification System запускает ряд микросервисов под полученным System Principal. Микросервисы, в свою очередь, обращаются в процессе работы к другим внешним системам — например, Product Catalog и IN@Voice.

Чтобы не настраивать для каждого нового партнера System Principal во внешних системах, вы можете настроить в системе Notification System подмену System Principal.

Для настройки:

1. Перейдите в MIB-группу `/NsSLES/@libs/NsInterpreter/Core/Configuration/ForceSystemPrincipal/`.
2. Создайте новую подгруппу. В качестве имени группы укажите `portType`, для которого будете настраивать подмену System Principal. Например: `{http://www.bercut.com/wsdl/ms/AOI_NS_Communication}NsCommunicationSendKafkaPortType/`.
3. В значении параметра `SystemPrincipal` укажите наименование параметра System Principal.

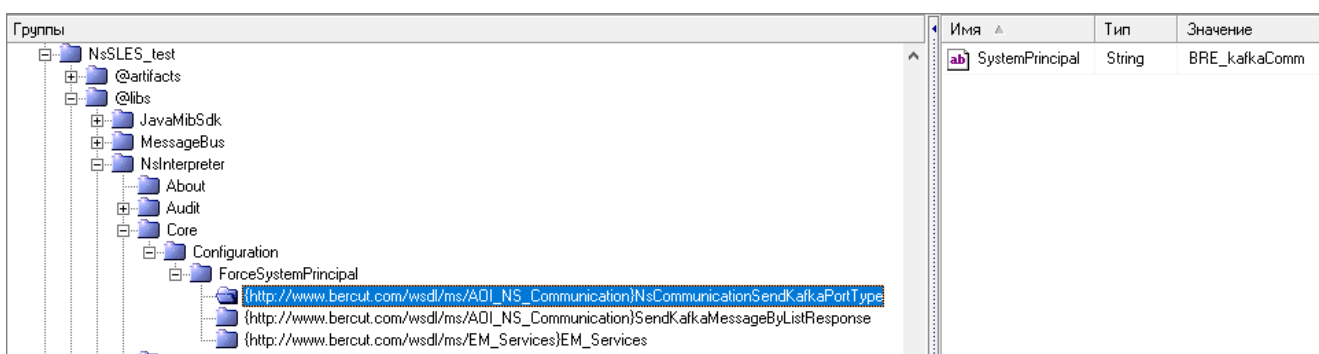


Рис. 14. Вариант настройки System Principal

4.6. Настройка интеграции с Apache Kafka

Настройка `LWSA_kafka` не является обязательным требованием для функционирования системы Notification System.

Компоненты используются для:

- получения событий из Apache Kafka;
- передачи в Notification System;
- прием после обработки со стороны Notification System;
- отправки в Apache Kafka.

Подготовка LWSA_kafka

1. Установите контейнер LWSA 3.5 в каталог `/opt/BERCem/lsc_kafka`.
2. Откройте для редактирования файл `/opt/BERCem/lsc_kafka/conf/setenvuser.sh` и задайте:
 - уникальный порт `INST_HTTP_PORT`;
 - имя контейнера `APP_NAME = kafka`;
 - опцию `CATALINA_OPTS=${CATALINA_OPTS} " -XX:PermSize=128m -XX:MaxPermSize=128m"`;
 - опцию `CATALINA_OPTS=${CATALINA_OPTS} " -Xms1G -Xmx1G"` (или более).

Установка SA

Установите SA NS KafkaConsumer и SA NS KafkaProducer. Для этого скопируйте из дистрибутива системы NotificationSystem/deployable/lwsa_deployable/ файлы `SA_NS_KafkaConsumer_v1.war` и `SA_NS_KafkaProducer_v1.war` в каталог на сервере `/LWSAContainer_kafka/webapps`.

Настройка SA NS KafkaConsumer

1. На сервере создайте каталог `/LWSAContainer_kafka/em_data`.
2. Перейдите в MIB-группу `/LWSAContainer_kafka/@artifacts/SA_NsKafkaConsumer/Core/Configuration/` и задайте значения параметров:

Таблица 1. Обязательные параметры SA NS Kafka Consumer

Параметр	Тип	Описание
<code>KafkaBootstrapServers</code>	String	Адреса серверов Apache Kafka в формате <code><IP-адрес>:<порт></code> . Обязательная переменная. Разделитель — точка с запятой. После применения будет выполнен перезапуск всех потоков. Пример: <code>127.0.0.1:9092;127.0.0.2:9092;127.0.0.3:9092</code> .
<code>KafkaGroupId</code>	String	Название группы Apache Kafka, в которую будет входить SA NS KafkaConsumer. Настройка для Kafka — <code>group.id</code> [13]. После применения будет выполнен перезапуск всех потоков.
<code>KafkaTopics</code>	String	Список топиков Apache Kafka, сообщения из которых будут обрабатываться. Разделитель —

Примечание. Значение должно совпадать для всех SA NS KafkaConsumer.

Параметр	Тип	Описание
		<p>точка с запятой. Пример: 'Topic1;Topic2;TopicN'.</p> <div style="border: 1px solid black; padding: 5px;"> <p>i Примечание. После применения будут перезапущены все потоки.</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>i Примечание. При работе системы список топиков, заданный в значении <code>KafkaTopics</code>, объединяется со списком топиков группы <code>/SA_NsKafkaConsumer/Core/Configuration/Topics/</code>. Если топик присутствует в обоих списках, то для него применяются более точные настройки из MIB-группы <code>/SA_NsKafkaConsumer/Core/Configuration/Topics/<Имя топика></code>.</p> </div>

При необходимости настройте параметры:

Таблица 2. Необязательные параметры SA NS Kafka Consumer

Параметр	Тип	Описание
<code>BreErrorBehavior</code>	String	<p>Поведение при получении от Notification System ошибок выполнения. Задается в формате: <code><ошибка>=<тип поведения></code>.</p> <p>Разделитель между ключом и значением — знак равно. Разделитель между записями — точка с запятой или перенос строки. Применяется без перезагрузки.</p> <p>Пример значений:</p> <pre>NOT_SUPPORTED_MESSAGE=SUSPEND lang.IllegalArgumentException=IGNORE</pre> <p>По умолчанию срабатывает поведение ALARM.</p> <p>Допустимые типы ошибок:</p> <ul style="list-style-type: none"> NOT_SUPPORTED_MESSAGE — ошибка «неподдерживаемое сообщение», может быть получена от Notification System, если туда отправлено сообщение, для обработки которого не задеплоено ни одного сценария. DEFAULT — для всех остальных ошибок. Пользовательские ошибки. Для настройки ошибки укажите строку, содержащую окончание названия исключения. Например: если нужно обработать ошибку <code>java.lang.IllegalArgumentException</code>, укажите: <code>IllegalArgumentException=IGNORE</code>. <p>Допустимые типы поведения:</p> <ul style="list-style-type: none"> IGNORE — игнорирование ошибки, сообщение в трейсы выводится на уровне debug; ALARM — сообщение в трейсы выводится на уровне alarm;

Параметр	Тип	Описание
		<ul style="list-style-type: none"> ▪ SUSPEND — сообщение в трейсы выводится на уровне alarm, выполняется приостановка работы на определенное время.
<i>ConversionErrorBehavior</i>	String	<p>Описание поведения при ошибках конвертации. Задается в формате: <ошибка>=<тип поведения>.</p> <p>Разделитель между ключом и значением — знак равно. Разделитель между записями — точка с запятой или перенос строки. Применяется без перезагрузки.</p> <p>Пример значений:</p> <pre>MS_NOT_FOUND=IGNORE MAPPING=SUSPEND lang.IllegalArgumentException=IGNORE</pre> <p>По-умолчанию срабатывает поведение ALARM.</p> <p>Допустимые типы ошибок:</p> <ul style="list-style-type: none"> ▪ INVALID_MAPPING_FILE — некорректные значения в файле маппинга. ▪ NO_MATCH — входящее сообщение не было распознано как xml или json; ▪ MAPPING — ошибки маппинга. Например, если входящее сообщение имеет неожиданную структуру, или вычисление xPath или jsonPath приводит к ошибке. ▪ MS_NOT_FOUND — не найден микросервис, ассоциированный с файлом маппинга или с сообщением. ▪ DEFAULT — для всех остальных ошибок. ▪ Пользовательские ошибки. Для настройки ошибки укажите строку, содержащую окончание названия исключения. Например: если нужно обработать ошибку <code>java.lang.IllegalArgumentException</code>, укажите: <code>IllegalArgumentException=IGNORE</code>. <p>Допустимые типы поведения:</p> <ul style="list-style-type: none"> ▪ IGNORE — игнорирование ошибки, сообщение в трейсы выводится на уровне debug; ▪ ALARM — сообщение в трейсы выводится на уровне alarm; ▪ SUSPEND — сообщение в трейсы выводится на уровне alarm, выполняется приостановка работы на определенное время.
<i>ConversionMappingFiles</i>	String	<p>Список файлов маппинга для преобразования, названия файлов должны оканчиваться на <code>.consumer.xml</code>. Если не заполнено, то будет срабатывать маппинг по-умолчанию (7.3.2)). Разделитель — точка с запятой, применяется без перезагрузки. Пример значения: <code>'mapping1.consumer.xml;mapping2.consumer.xml'</code>.</p>

Параметр	Тип	Описание
<i>DefaultEncoding</i>	String	Кодировка для конвертации по умолчанию, если файл маппинга не найден.
<i>DefaultMappingEnabled</i>	Boolean	Флаг включения маппинга по-умолчанию, если не найден файл маппинга.
<i>KafkaMaxPartitionFetch Bytes</i>	LongInt	Настройка для Apache Kafka - <code>max.partition.fetch.bytes</code> [26]. Максимальный объем данных на раздел, который вернет сервер Apache Kafka, в байтах. Если значение 0, то будет использовано значение по-умолчанию из Apache Kafka.
<i>KafkaProperties</i>	String	Произвольный набор настроек для Consumer брокера Apache Kafka [26]. Разделитель — точка с запятой или перенос строки. После применения будут перезапущены все потоки. Пример значения: <code>'key1=value1;key2=value2;key3=value3'</code> . ! Внимание! Настройки имеют максимальный приоритет применения.
<i>PollerThreadTimeout</i>	LongInt	Промежуток времени в миллисекундах, через который будут осуществляться попытки прочесть сообщения из Apache Kafka. Применяется без перезагрузки. Значение по умолчанию: 200. Диапазон допустимых значений: 0 — 2000000000.
<i>ReloadFiles</i>	Boolean	Перезагрузить файлы маппинга и микросервисы из каталога <code>em_data</code> . Значение по умолчанию: True. Для перезагрузки измените значение параметра на True и примените изменение.
<i>RtsibErrorBehavior</i>	String	Поведение при ошибках опраки сообщения по RTSIB. Задается в формате: <code><ошибка>=<тип поведения></code> . Разделитель между ключом и значением — знак равно. Разделитель между записями — точка с запятой или перенос строки. Применяется без перезагрузки. Пример значений: <pre>TIMEOUT=IGNORE ROUTE_UNAVAILABLE=SUSPEND lang.IllegalArgumentException=IGNORE</pre> По-умолчанию срабатывает поведение ALARM. Допустимые типы ошибок: <ul style="list-style-type: none"> TIMEOUT — тайм-аут при отправке сообщения в Notification System. ROUTE_UNAVAILABLE — маршрут не настроен. UNKNOWN_COMPONENT — маршрут настроен, но на той стороне не нашлось компонента. COMPONENT_UNAVAILABLE — компонент недоступен. DEFAULT — для всех остальных ошибок.

Параметр	Тип	Описание
		<ul style="list-style-type: none"> Пользовательские ошибки. Для настройки ошибки укажите строку, содержащую окончание названия исключения. Например: если нужно обработать ошибку <code>java.lang.IllegalArgumentException</code>, укажите: <code>IllegalArgumentException=IGNORE</code>. <p>Допустимые типы поведения:</p> <ul style="list-style-type: none"> IGNORE — игнорирование ошибки, сообщение в трейсы выводится на уровне debug; ALARM — сообщение в трейсы выводится на уровне alarm; SUSPEND — сообщение в трейсы выводится на уровне alarm, выполняется приостановка работы на определенное время.
<i>Slf4jAtlasLoggerEnabled</i>	Boolean	Флаг включения вывода дополнительных логов от сторонних библиотек. Значение по умолчанию: False.
<i>SuspenseRetryTimeout</i>	LongInt	Время ожидания для возобновления работы, если сработало поведение SUSPEND. Применяется без перезагрузки. Значение по умолчанию: 60. Диапазон допустимых значений: 1 — 10000.
<i>TopicThreadCount</i>	LongInt	<p>Количество потоков для топика, которые будут опрашивать Apache Kafka. Применяется без перезагрузки. Значение по умолчанию: 4. Диапазон допустимых значений: 1 — 4096.</p> <div style="border: 1px solid orange; padding: 5px;"> <p>! Внимание! Общее количество потоков на всех SA NS Kafka Consumer не должно превышать количество секций Apache Kafka — значение параметра <code>partition</code> [26].</p> </div>
<i>TraceBlockEnabled</i>	Boolean	<p>Флаг чтения блока trace из входного JSON-сообщения:</p> <ul style="list-style-type: none"> True — SA NS KafkaConsumer вычитывает блок trace из входного JSON-сообщения и перекладывает его в EventContext. False — не вычитывает. <p>Значение по умолчанию: False.</p>
<i>UseWSN</i>	Boolean	<p>Флаг использования WSN PortType для передачи сообщений:</p> <ul style="list-style-type: none"> True — используется portType {http://docs.oasis-open.org/wsn/bw-2}NotificationConsumer, по которому сообщение отправится в компонент, который этот portType имплементирует. False — используется portType {http://www.bercut.com/wsdl/ns/kafkaconsumer} NS_KafkaConsumer для передачи сообщения в Notification System.

Параметр	Тип	Описание
		Применяется без перезагрузки. Значение по умолчанию: False.

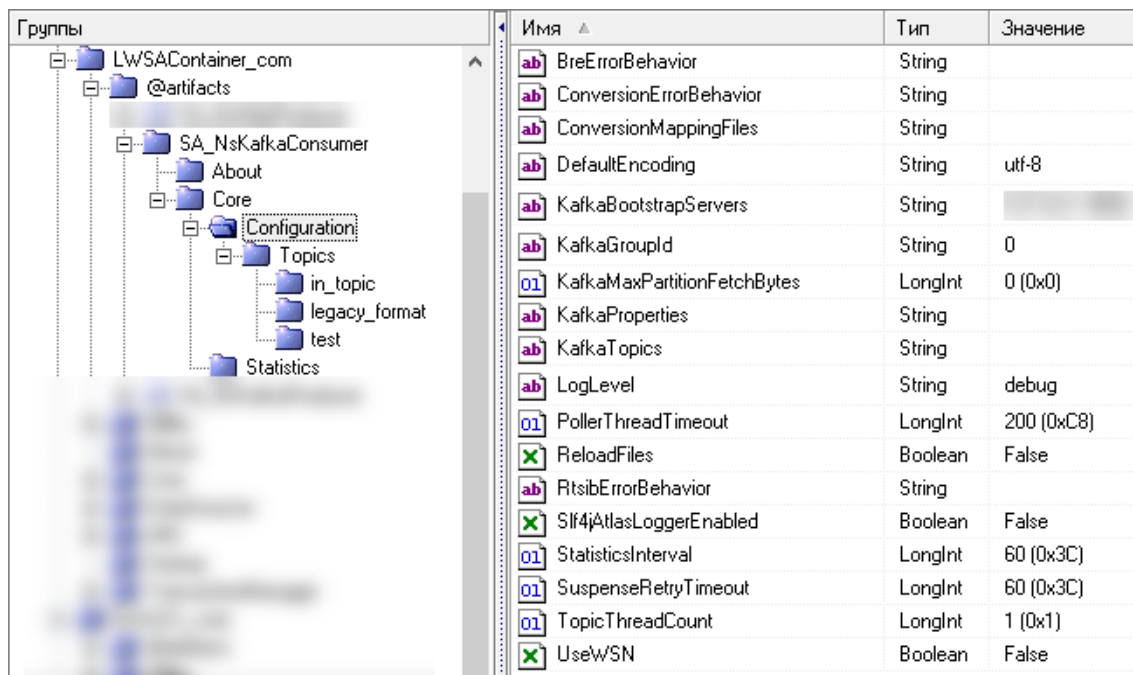


Рис. 15. Вариант настройки SA NS KafkaConsumer

- При необходимости вы можете настроить поведение системы при работе с каждым из топиков. Для этого перейдите в группу /SA_NsKafkaConsumer/Core/Configuration/Topics/ и создайте новую группу. Имя группы — имя топика Apache Kafka.

i Примечание. При работе системы список топиков, заданный в значении *KafkaTopics*, объединяется со списком топиков группы /SA_NsKafkaConsumer/Core/Configuration/Topics/. Если топик присутствует в обоих списках, то для него применяются более точные настройки из MIB-группы /SA_NsKafkaConsumer/Core/Configuration/Topics/<Имя топика>.

Таблица 3. Параметры для настройки отдельных топиков

Параметр	Тип	Описание
<i>BreErrorBehavior</i>	String	<p>Поведение при получении от Notification System ошибок выполнения. Задается в формате: <ошибка>=<тип поведения>.</p> <p>Разделитель между ключом и значением — знак равно. Разделитель между записями — точка с запятой или перенос строки. Применяется без перезагрузки.</p> <p>Пример значений:</p> <pre>NOT_SUPPORTED_MESSAGE=SUSPEND lang.IllegalArgumentException=IGNORE</pre> <p>По умолчанию срабатывает поведение ALARM.</p> <p>Допустимые типы ошибок:</p>

Параметр	Тип	Описание
		<ul style="list-style-type: none"> ▪ NOT_SUPPORTED_MESSAGE — ошибка «неподдерживаемое сообщение», может быть получена от Notification System, если туда отправлено сообщение, для обработки которого не задеплоено ни одного сценария. ▪ DEFAULT — для всех остальных ошибок. ▪ Пользовательские ошибки. Для настройки ошибки укажите строку, содержащую окончание названия исключения. Например: если нужно обработать ошибку <code>java.lang.IllegalArgumentException</code>, укажите: <code>IllegalArgumentException=IGNORE</code>. <p>Допустимые типы поведения:</p> <ul style="list-style-type: none"> ▪ IGNORE — игнорирование ошибки, сообщение в трейсы выводится на уровне debug; ▪ ALARM — сообщение в трейсы выводится на уровне alarm; ▪ SUSPEND — сообщение в трейсы выводится на уровне alarm, выполняется приостановка работы на определенное время.
<i>ConversionErrorBehavior</i>	String	<p>Описание поведения при ошибках конвертации. Задается в формате: <ошибка>=<тип поведения>.</p> <p>Разделитель между ключом и значением — знак равно. Разделитель между записями — точка с запятой или перенос строки. Применяется без перезагрузки.</p> <p>Пример значений:</p> <pre style="background-color: #f0f0f0; padding: 5px;">MS_NOT_FOUND=IGNORE MAPPING=SUSPEND lang.IllegalArgumentException=IGNORE</pre> <p>По-умолчанию срабатывает поведение ALARM.</p> <p>Допустимые типы ошибок:</p> <ul style="list-style-type: none"> ▪ INVALID_MAPPING_FILE — некорректные значения в файле маппинга. ▪ NO_MATCH — входящее сообщение не было распознано как xml или json; ▪ MAPPING — ошибки маппинга. Например, если входящее сообщение имеет неожиданную структуру, или вычисление xPath или jsonPath приводит к ошибке. ▪ MS_NOT_FOUND — не найден микросервис, ассоциированный с файлом маппинга или с сообщением. ▪ DEFAULT — для всех остальных ошибок. ▪ Пользовательские ошибки. Для настройки ошибки укажите строку, содержащую окончание названия исключения. Например: если нужно обработать ошибку <code>java.lang.IllegalArgumentException</code>, укажите: <code>IllegalArgumentException=IGNORE</code>.

Параметр	Тип	Описание
		<p>Допустимые типы поведения:</p> <ul style="list-style-type: none"> ▪ IGNORE — игнорирование ошибки, сообщение в трейсы выводится на уровне debug; ▪ ALARM — сообщение в трейсы выводится на уровне alarm; ▪ SUSPEND — сообщение в трейсы выводится на уровне alarm, выполняется приостановка работы на определенное время.
<i>ConversionMappingFiles</i>	String	<p>Список файлов маппинга для преобразования, названия файлов должны оканчиваться на <code>.consumer.xml</code>. Если не заполнено, то будет срабатывать маппинг по-умолчанию (п. 3.2)). Разделитель — точка с запятой, применяется без перезагрузки. Пример значения: <code>'mapping1.consumer.xml;mapping2.consumer.xml'</code>.</p>
<i>CustomDateTimeFormats</i>	String	<p>Дополнительный список форматов при преобразовании из строки в дату и время. Применяются по очереди. Используются только в случае, если не удалось распознать дату и время по-умолчанию. Не обязательно к заполнению. Разделитель — точка с запятой, применяется без перезагрузки. Пример: <code>yyyy-MM-dd;dd MM yyyy</code>.</p>
<i>DefaultEncoding</i>	String	<p>Кодировка для конвертации по умолчанию, если файл маппинга не найден.</p>
<i>DefaultMappingEnabled</i>	Boolean	<p>Флаг использования маппинга по умолчанию, если не найден файл маппинга.</p> <ul style="list-style-type: none"> ▪ True — использовать; ▪ False — не использовать. <p>Значение по умолчанию: True.</p>
<i>RtsibErrorBehavior</i>	String	<p>Поведение при ошибках опрарки сообщения по RTSIB. Задается в формате: <code><ошибка>=<тип поведения></code>.</p> <p>Разделитель между ключом и значением — знак равно. Разделитель между записями — точка с запятой или перенос строки. Применяется без перезагрузки.</p> <p>Пример значений:</p> <pre>TIMEOUT=IGNORE ROUTE_UNAVAILABLE=SUSPEND lang.IllegalArgumentException=IGNORE</pre> <p>По-умолчанию срабатывает поведение ALARM.</p> <p>Допустимые типы ошибок:</p> <ul style="list-style-type: none"> ▪ TIMEOUT — тайм-аут при отправке сообщения в Notification System. ▪ ROUTE_UNAVAILABLE — маршрут не настроен. ▪ UNKNOWN_COMPONENT — маршрут настроен, но на той стороне не нашлось компонента.

Параметр	Тип	Описание
		<ul style="list-style-type: none"> COMPONENT_UNAVAILABLE — компонент недоступен. DEFAULT — для всех остальных ошибок. Пользовательские ошибки. Для настройки ошибки укажите строку, содержащую окончание названия исключения. Например: если нужно обработать ошибку <code>java.lang.IllegalArgumentException</code>, укажите: <code>IllegalArgumentException=IGNORE</code>. <p>Допустимые типы поведения:</p> <ul style="list-style-type: none"> IGNORE — игнорирование ошибки, сообщение в трейсы выводится на уровне debug; ALARM — сообщение в трейсы выводится на уровне alarm; SUSPEND — сообщение в трейсы выводится на уровне alarm, выполняется приостановка работы на определенное время.
<i>SuspenseRetryTimeout</i>		Время ожидания для возобновления работы, если сработало поведение SUSPEND. Применяется без перезагрузки. Значение по умолчанию: 60. Диапазон допустимых значений: 1 — 10000.
<i>TopicThreadCount</i>	LongInt	Количество потоков для топика, которые будут опрашивать Apache Kafka. Применяется без перезагрузки. Значение по умолчанию: 4. Диапазон допустимых значений: 1 — 4096.

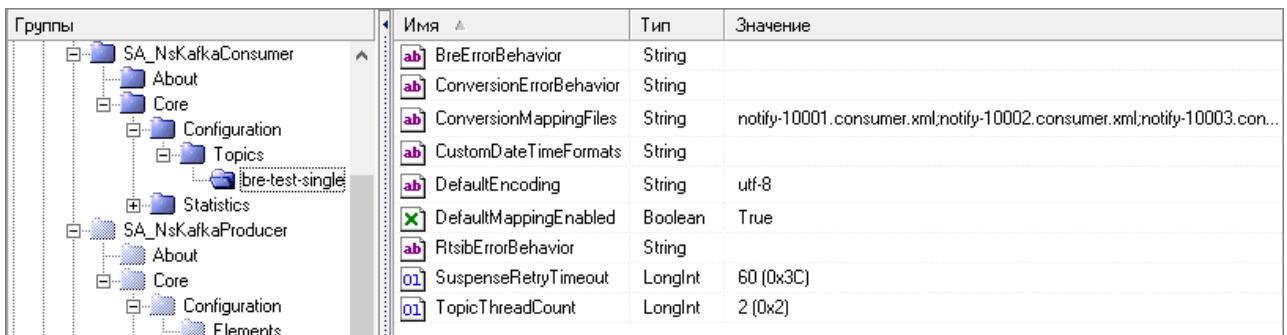


Рис. 16. Вариант настройки группы /SA_NsKafkaConsumer/Core/Configuration/Topics/<Имя топика>

! Внимание! Количество потоков со всех SA NS Kafka Consumer не должно превышать количества разделов в Apache Kafka на топик. В противном случае потоки не будут получать сообщения.

Настройка SA NS KafkaProducer

1. Перейдите в MIB-группу `/LWSAContainer_kafka/@artifacts/SA_NsKafkaProducer/Core/Configuration/` и задайте значения параметров:

Таблица 4. Обязательные параметры SA NS Kafka Producer

Параметр	Тип	Описание
<i>DefaultTopic</i>	String	Название топика для отправки сообщения по умолчанию. Заполнение обязательно, применяется без перезагрузки.
<i>DefaultFormat</i>	String	Формат кодирования сообщений для отправки по умолчанию. Заполнение обязательно, применяется без перезагрузки. Допустимые значения: 'json', 'xml'. Значение по умолчанию: 'xml'.

2. При необходимости настройте необязательные MIB-параметры:

Таблица 5. Необязательные параметры SA NS Kafka Producer

Параметр	Тип	Описание
<i>ConversionMappingFiles</i>	String	Список файлов маппинга для преобразования. Названия файлов должны оканчиваться на .producer.xml . Если не заполнено, то будет срабатывать маппинг по умолчанию. Разделитель — точка с запятой, применяется без перезагрузки. Пример: <code>mapping1.producer.xml; mapping2.producer.xml</code> .
<i>DefaultEncoding</i>	String	Кодировка по умолчанию. Значение по умолчанию: 'utf-8'.
<i>KafkaBootstrapServers</i>	String	Адреса серверов Apache Kafka в формате <code><IP-адрес>:<порт></code> . Заполнение необязательно, разделитель — точка с запятой. После применения будет выполнено переподключение к Apache Kafka. Пример: <code>127.0.0.1:9092;127.0.0.2:9092;127.0.0.3:9092</code> .
<i>KafkaProperties</i>	String	Произвольный набор настроек для Kafka Producer (https://kafka.apache.org/documentation/#producerconfigs). Заполнение необязательно. Разделитель — точка с запятой и/или перенос строки. После применения произойдет переподключение к Apache Kafka. Настройки имеют максимальный приоритет применения. Пример значения: <pre>key1=value1 key2=value2 key3=value3</pre>
<i>ReloadFiles</i>	Boolean	Перезагрузить файлы маппинга и микросервисы из каталога <code>em_data</code> . Значение по умолчанию: 'True'. Для перезагрузки измените значение параметра на 'True' и примените изменение.

Параметр	Тип	Описание
<i>Slf4jAtlasLoggerEnabled</i>	Boolean	Флаг включения вывода дополнительных логов от сторонних библиотек. Значение по умолчанию: 'False'.

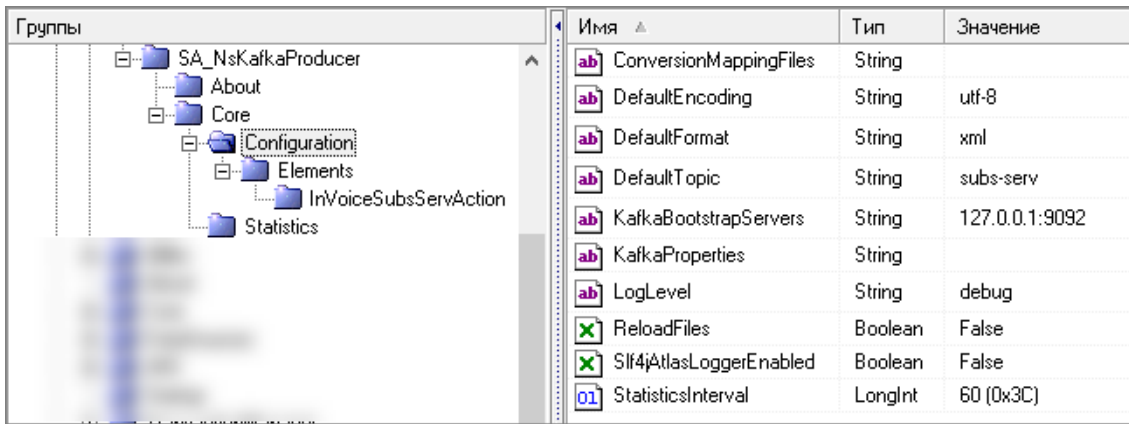


Рис. 17. Вариант настройки SA NS KafkaProducer

3. При необходимости вы можете настроить поведение системы при работе с каждым из элементов. Для этого перейдите в группу *SA_NsKafkaProducer/Core/Configuration/Elements/* и создайте новую группу. Имя группы — название XSD-элемента.

Примечание. В качестве ключа выступает название корневого элемента.

Таблица 6. Параметры для настройки отдельных элементов

Параметр	Тип	Описание
<i>ConversionMappingFiles</i>	String	Список файлов маппинга для преобразования. Названия файлов должны оканчиваться на .producer.xml . Если не заполнено, то будет срабатывать маппинг по умолчанию. Разделитель — точка с запятой, применяется без перезагрузки. Пример: <code>mapping1.producer.xml; mapping2.producer.xml</code> .
<i>DefaultJsonMethod</i>	String	Метод конвертации значений в JSON-формат. Допустимые значения: <ul style="list-style-type: none"> types — конвертация с учетом XSD-типов; string — конвертация в строки.
<i>DefaultEncoding</i>	String	Кодировка по умолчанию. Значение по умолчанию: 'utf-8'.
<i>DefaultFormat</i>	String	Формат кодирования сообщений для отправки по умолчанию. Заполнение обязательно, применяется без перезагрузки. Допустимые значения: 'json', 'xml'. Значение по умолчанию: 'xml'.
<i>DefaultPartitionKey</i>	String	Значение ключа по умолчанию, используется если ключ не найден по набору полей.
<i>FieldPartitionKey</i>	String	Набор полей, по которым может быть сформирован ключ. Разделитель между наборами — точка с запятой. Разделитель между полями в наборе — знак плюс. Применяется без перезагрузки.

Параметр	Тип	Описание
Topic	String	Имя топика для отправки сообщения в Apache Kafka.

Имя	Тип	Значение
ConversionMappingFile	String	InVoiceSubsServAction-json.producer.xml
DefaultEncoding	String	utf-8
DefaultFormat	String	xml
DefaultPartitionKey	String	
FieldPartitionKey	String	
Topic	String	subs-serv

Рис. 18. Вариант настройки группы SA_NsKafkaProducer/Core/Configuration/Elements

Пример настройки группы SA_NsKafkaProducer/Core/Configuration/Elements для элементов:

Рис. 19. Пример настройки группы SA_NsKafkaProducer/Core/Configuration/Elements

- AddComment;

ConversionMappingFile	
DefaultEncoding	utf-8
DefaultFormat	json
Topic	add_comment
DefaultPartitionKey	
FieldPartitionKey	

- ChangeStatus;

ConversionMappingFile	
DefaultEncoding	utf-8
DefaultFormat	json
Topic	ch_abon_status
DefaultPartitionKey	
FieldPartitionKey	

- toDSuite.

ConversionMappingFile	
-----------------------	--

<i>DefaultEncoding</i>	utf-8
<i>DefaultFormat</i>	json
<i>Topic</i>	BREtoDSUITE
<i>DefaultPartitionKey</i>	
<i>FieldPartitionKey</i>	

Настройка маршрутов

1. Убедитесь что настроены маршруты между LWSA_kafka и SLES:

1.1. От LWSA_kafka к SLES:

```
{http://www.bercut.com/wsdl/ns/kafkaconsumer}NS_KafkaConsumer
```

1.2. От SLES к LWSA_kafka:

```
{http://www.bercut.com/wsdl/ns/kafkaproducer}NS_KafkaProducer
```

2. Убедитесь, что настроены маршруты между NS SLES EM и Communication Layer [18].

4.7. Установка и настройка HTTP-адаптера

Настройка HTTP-адаптера не является обязательным требованием для функционирования системы Notification System.

Данный компонент используется для возможности отправки нотификации по HTTP.

Установка SA NS HttpProducer

Установите SA NS HttpProducer. Для этого скопируйте из дистрибутива системы NotificationSystem/deployable/lwsa_deployable/ файл SA_NS_HttpProducer_v1.war в каталог на сервере LWSA/webapps.

Настройка SA NS HttpProducer

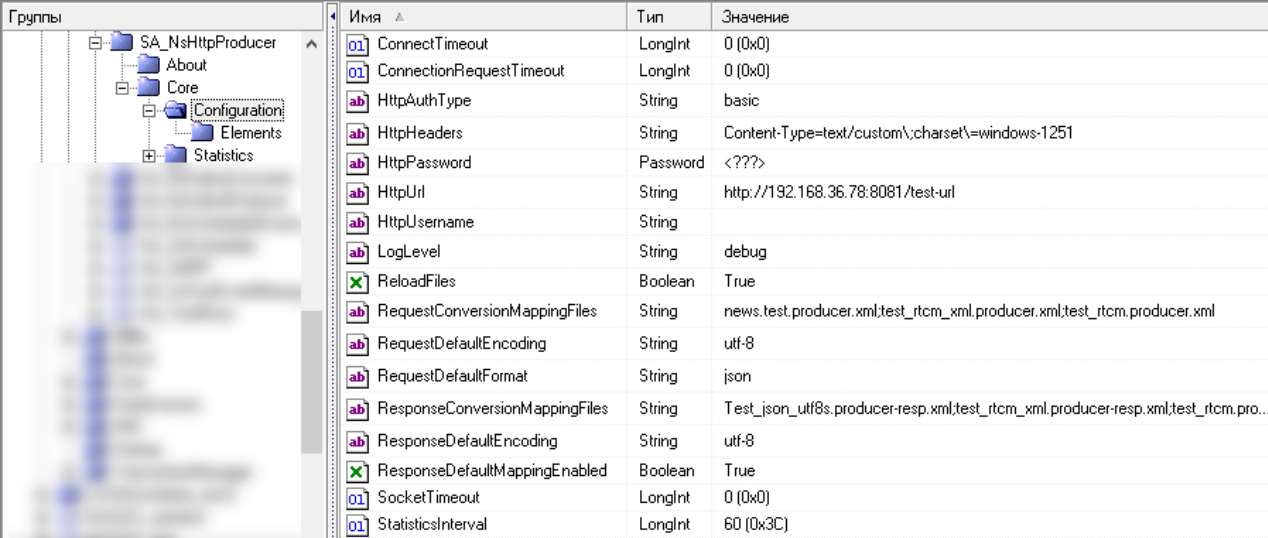
1. Перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NsHttpProducer/Core/Configuration/` и задайте значения параметров:

- *ConnectTimeout* — задает время ожидания до установления соединения, в миллисекундах. Применяется без перезагрузки. Значение по умолчанию: '0' (время ожидания не ограничено).
- *ConnectionRequestTimeout* — задает тайм-аут, используемый при запросе соединения из диспетчера соединений, в миллисекундах. Применяется без перезагрузки. Значение по умолчанию — '0' (время ожидания не ограничено).
- *HttpAuthType* — тип аутентификации. Допустимые значения:
 - basic;
 - digest.

Значение по умолчанию: 'basic'. Заполнение необязательно, применяется без перезагрузки.

- *HttpHeaders* — произвольный набор заголовков для HTTP. Заполнение необязательно, разделитель — точка с запятой и (или) перенос строки. После применения будут перезапущены все потоки. Настройки имеют максимальный приоритет применения. Пример: `key1=value1;key2=value2;key3=value3`.

- *HttpPassword* — пароль. Заполнение необязательно, применяется без перезагрузки.
- *HttpUrl* — URL-адрес запроса. Заполнение обязательно, применяется без перезагрузки.
- *HttpUsername* — имя пользователя. Заполнение необязательно, применяется без перезагрузки.
- *ReloadFiles* — флаг перезагрузки файлов маппинга и микросервисов из каталога `em_data` на сервере.
- *RequestConversionMappingFiles* — список файлов маппинга для преобразования запроса. Названия файлов должны оканчиваться на `.producer.xml`. Если не заполнено, то будет срабатывать маппинг по умолчанию. Разделитель — точка с запятой, применяется без перезагрузки. Пример: `mapping1.producer.xml;mapping2.producer.xml`.
- *RequestDefaultEncoding* — кодировка запроса по умолчанию. Значение по умолчанию: `'utf-8'`.
- *RequestDefaultFormat* — формат конвертирования запроса по умолчанию. Значение по умолчанию: `'xml'`.
- *ResponseConversionMappingFiles* — список файлов маппинга для преобразования ответа. Названия файлов должны оканчиваться на `.producer-resp.xml`. Если не заполнено, то будет срабатывать маппинг по умолчанию. Разделитель — точка с запятой, применяется без перезагрузки. Пример: `mapping1.producer-resp.xml;mapping2.producer-resp.xml`.
- *ResponseDefaultEncoding* — кодировка для конвертации ответа по умолчанию. Значение по умолчанию: `'utf-8'`.
- *ResponseDefaultMappingEnabled* — флаг использования маппинга ответа по умолчанию, если не найден файл маппинга. Значение по умолчанию: `'True'`.
- *SocketTimeout* — максимальный период бездействия между двумя последовательными пакетами данных. Определяет тайм-аут сокета (тайм-аутом ожидания данных), в миллисекундах. Значение `'0'` — бесконечное время ожидания. Применяется без перезагрузки.



Имя	Тип	Значение
01 ConnectTimeout	LongInt	0 (0x0)
01 ConnectionRequestTimeout	LongInt	0 (0x0)
ab HttpAuthType	String	basic
ab HttpHeaders	String	Content-Type=text/custom\,charset=windows-1251
ab HttpPassword	Password	<???
ab HttpUrl	String	http://192.168.36.78:8081/test-url
ab HttpUsername	String	
ab LogLevel	String	debug
X ReloadFiles	Boolean	True
ab RequestConversionMappingFiles	String	news.test.producer.xml;test_rtcml.producer.xml;test_rtcml.producer.xml
ab RequestDefaultEncoding	String	utf-8
ab RequestDefaultFormat	String	json
ab ResponseConversionMappingFiles	String	Test_json_utf8s.producer-resp.xml;test_rtcml.producer-resp.xml;test_rtcml.pro...
ab ResponseDefaultEncoding	String	utf-8
X ResponseDefaultMappingEnabled	Boolean	True
01 SocketTimeout	LongInt	0 (0x0)
01 StatisticsInterval	LongInt	60 (0x3C)

Рис. 20. Пример настройки SA NS HttpProducer

2. Если вам требуется настроить таблицу маппинга элементов запроса, перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NsHttpProducer/Core/Configuration/Elements/` и создайте новую группу. Имя группы — имя элемента в теге `<rtsib-element>` в файле маппинга. Задайте значения параметров:

- *RequestConversionMappingFile* — название файла маппинга для преобразования. Файл должен оканчиваться на `.producer.xml`. Если не заполнено, то будет

срабатывать маппинг по умолчанию. Применяется без перезагрузки. Пример значения: `mapping1.producer.xml`.

- *RequestDefaultFormat* — формат конвертирования по умолчанию. Значение по умолчанию: `xml`.
- *RequestDefaultJsonMethod* — метод конвертирования по умолчанию для JSON-формата. Значение по умолчанию: `strings`.
- *RequestDefaultEncoding* — кодировка по умолчанию. Значение по умолчанию: `utf-8`.
- *ResponseConversionMappingFile* — название файла маппинга для преобразования. Должно оканчиваться на `.consumer.xml`. Если не заполнено, то будет срабатывать маппинг по умолчанию. Применяется без перезагрузки. Пример значения: `'mapping1.consumer.xml'`.
- *ResponseDefaultEncoding* — кодировка для конвертации ответа по умолчанию. Значение по умолчанию: `'utf-8'`.
- *ResponseDefaultMappingEnabled* — флаг использования маппинга ответа по умолчанию, если не найден файл маппинга. Значение по умолчанию: `'True'`.
- *ResponseCustomDateTimeFormats* — дополнительный список форматов при преобразовании из строки в дату и время. Применяются по очереди. Используются только в случае, если не удалось распознать дату и время по умолчанию. Не обязательно к заполнению. Разделитель — точка с запятой, применяется без перезагрузки. Пример значения: `'yyyy-MM-dd;dd MM yyyy'`.

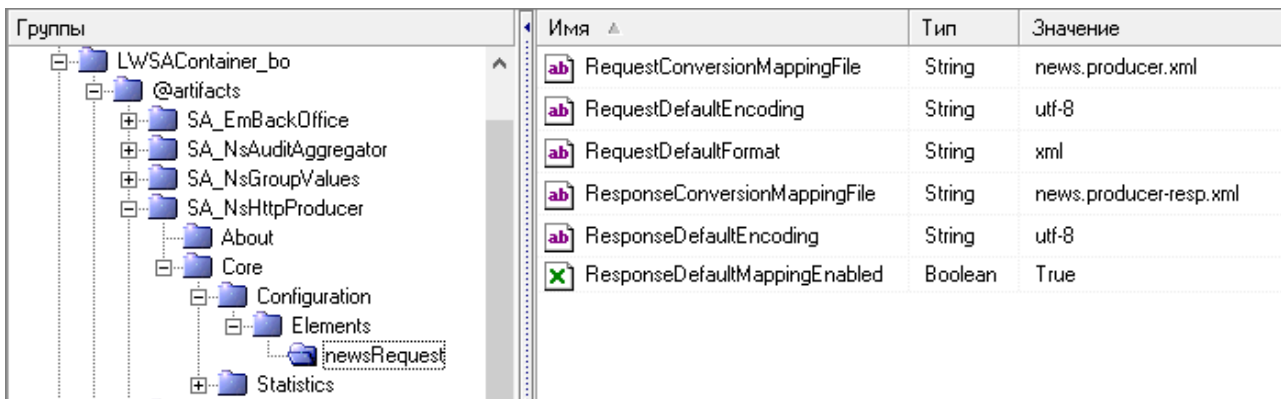


Рис. 21. Пример настройки группы SA_NsHttpProducer/Core/Configuration/Elements/

3. Убедитесь, что настроено соединение от NS SLES EM до LSC.

SA NS HttpProducer использует интерфейс `{http://www.bercut.com/wsd/ns/httpproducer}NS_HttpProducer`.

О работе SA NS HttpProducer в MIB-группе `/LWSAContainer/@artifacts/SA_NsHttpProducer/Core/Statistics/Performance/General` выводит статистику работы:

- *InRequestsCnt* — количество входящих сообщений.
- *InRequestsMax* — максимальное количество входящих сообщений.
- *InRequestsMin* — минимальное количество входящих сообщений.
- *InRequestsAvg* — среднее количество входящих сообщений.
- *OutSuccessfulCnt* — количество успешных исходящих сообщений.
- *OutSuccessfulMax* — максимальное количество успешных исходящих сообщений.
- *OutSuccessfulMin* — минимальное количество успешных исходящих сообщений.
- *OutSuccessfulAvg* — среднее количество успешных исходящих сообщений.
- *OutFailedCnt* — количество неуспешных исходящих сообщений.
- *OutFailedMax* — максимальное количество неуспешных исходящих сообщений.
- *OutFailedMin* — минимальное количество неуспешных исходящих сообщений.
- *OutFailedAvg* — среднее количество неуспешных исходящих сообщений.

- *CurrentLoadTPS* — текущая нагрузка, транзакций в секунду в течение интервала обновления.

4.8. Установка и настройка RabbitMQ-адаптера

Настройка RabbitMQ-адаптера не является обязательным требованием для функционирования системы Notification System.

Компонент SA NS RabbitMQProducer используется для предоставления возможности отправки нотификации в систему RabbitMQ.

Установка SA NS RabbitMQProducer

Установите SA NS RabbitMQProducer. Для этого скопируйте из дистрибутива системы NotificationSystem/deployable/lwsa_deployable/ файл SA_NS_RabbitMQProducer_v1.war в каталог на сервере LWSA/webapps.

Настройка SA NS RabbitMQProducer

1. Перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NsRabbitMQProducer/Core/Configuration/` и задайте значения параметров:

- *RabbitMQHost* — IP-адрес RabbitMQ. Заполнение обязательно, применяется без перезагрузки.
- *RabbitMQPort* — порт RabbitMQ. Заполнение обязательно, применяется без перезагрузки.
- *RabbitMQUsername* — имя пользователя. Заполнение обязательно, применяется без перезагрузки.
- *RabbitMQPassword* — пароль. Заполнение обязательно, применяется без перезагрузки.
- *RabbitMQProperties* — произвольный набор заголовков для RabbitMQ. Заполнение необязательно, разделитель — точка с запятой и (или) перенос строки. После применения будут перезапущены все потоки. Настройки имеют максимальный приоритет применения. Пример: `key1=value1;key2=value2;key3=value3`.
- *QueueNameDefault* — название очереди по умолчанию. Заполнение обязательно, применяется без перезагрузки. Значение по умолчанию: `defaultQueue`.
- *QueueToNotificationMapping* — маппинг очереди назначения по наименованию нотификации. Заполнение необязательно. Разделитель записей — точка с запятой и (или) перенос строки. Разделитель ключа и значения — знак равно, разделитель значений — запятая, применяется без перезагрузки. Пример:

```
Message1=notify-100,notify-200,notify-300;Message2=notify-150,notify-250,notify-301
topic-yyy=notify-111
topic-zzz=notify-222
```

- *QueueEncodeDefault* — формат кодирования сообщений для отправки. Допустимые значения:
 - json;
 - xml.

Значение по умолчанию: 'xml'.

- *QueueEncodeJson* — переопределяет формат кодирования на JSON для списка очередей. Заполнение необязательно. Разделитель — точка с запятой. Применяется без перезагрузки. Пример: `Queue1;Queue2;QueueN`.
- *QueueEncodeXml* — переопределяет формат кодирования на XML для списка очередей. Настройка более приоритетна, чем *QueueEncodeJson*. Заполнение

необязательно, разделитель — точка с запятой, применяется без перезагрузки.
 Пример: Queue1;Queue2;QueueN.

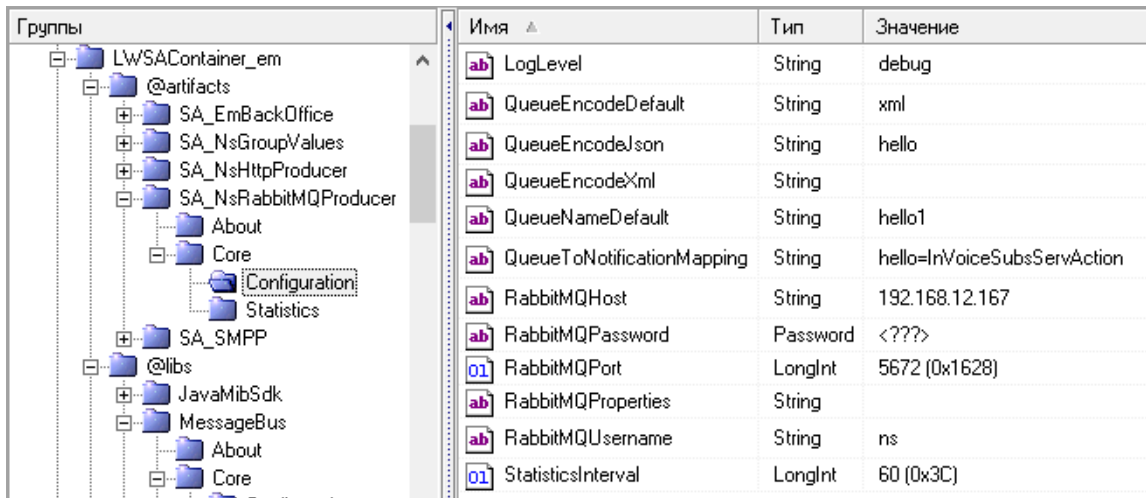


Рис. 22. Пример настройки SA NS RabbitMQProducer

2. Убедитесь, что настроено соединение от NS SLES EM к LSC.

SA NS RabbitMQProducer использует интерфейс `{http://www.bercut.com/wsd/!ns/rabbitmqproducer}NS_RabbitMQProducer`.

SA NS RabbitMQProducer выводит статистику работы:

1. Статистику принятых и отправленных сообщений — в MIB-группе `/LWSAContainer/@artifacts/SA_NsRabbitMQProducer/Core/Statistics/Performance/General`:

- `InRequestsCnt` — количество входящих сообщений.
- `InRequestsMax` — максимальное количество входящих сообщений.
- `InRequestsMin` — минимальное количество входящих сообщений.
- `InRequestsAvg` — среднее количество входящих сообщений.
- `OutSuccessfulCnt` — количество успешных исходящих сообщений.
- `OutSuccessfulMax` — максимальное количество успешных исходящих сообщений.
- `OutSuccessfulMin` — минимальное количество успешных исходящих сообщений.
- `OutSuccessfulAvg` — среднее количество успешных исходящих сообщений.
- `OutFailedCnt` — количество неуспешных исходящих сообщений.
- `OutFailedMax` — максимальное количество неуспешных исходящих сообщений.
- `OutFailedMin` — минимальное количество неуспешных исходящих сообщений.
- `OutFailedAvg` — среднее количество неуспешных исходящих сообщений.
- `CurrentLoadTPS` — текущая нагрузка, транзакций в секунду в течение интервала обновления.

2. Статистику в разрезе очередей — в MIB-группе `/LWSAContainer/@artifacts/SA_NsRabbitMQProducer/Core/Statistics/Performance/PerQueue`:

- `Queue` — Название очереди.
- `InRequestsCnt` — количество входящих сообщений.
- `InRequestsMax` — максимальное количество входящих сообщений.
- `InRequestsMin` — минимальное количество входящих сообщений.
- `InRequestsAvg` — среднее количество входящих сообщений.
- `OutSuccessfulCnt` — количество успешных исходящих сообщений.
- `OutSuccessfulMax` — максимальное количество успешных исходящих сообщений.
- `OutSuccessfulMin` — минимальное количество успешных исходящих сообщений.
- `OutSuccessfulAvg` — среднее количество успешных исходящих сообщений.

- *OutFailedCnt* — количество неуспешных исходящих сообщений.
- *OutFailedMax* — максимальное количество неуспешных исходящих сообщений.
- *OutFailedMin* — минимальное количество неуспешных исходящих сообщений.
- *OutFailedAvg* — среднее количество неуспешных исходящих сообщений.
- *CurrentLoadTPS* — текущая нагрузка, транзакций в секунду в течение интервала обновления.

4.9. Установка и настройка адаптера для БД

Настройка SA NS DbProducer не является обязательным требованием для функционирования системы Notification System.

SA NS DbProducer — универсальный адаптер для обращения к базам данных и получения данных из них через JDBC. Для взаимодействия с базой данных используется любой источник данных, настроенный в LW SA Container.

i Примечание. Компонент поддерживает работу с драйверами БД PostgreSQL и Oracle.

Для каждого набора критериев определяется *файл маппинга*, в котором:

- задается инструкция SQL;
- поддерживаются объекты PreparedStatement для обеспечения безопасности;
- поддерживаются массивы значений — для передачи в значении критерия списка значений;
- при необходимости задается несколько инструкций с оператором SELECT для получения различных критериев;
- поддерживаются микросервисы типа Getter для получения значения критериев и типа Action для вызова процедуры, вставки или обновления значений.

Все операции выполняются в рамках транзакции — автономной или распределенной. Все ошибки при запросе приводят к выполнению операции ROLLBACK для транзакции. Валидация файлов маппинга выполняется вне транзакции.

При вызове операции микросервиса выполняется синхронное обращение к БД.

! Внимание! В запросе должна содержаться одна строка или 0 строк, если отсутствует ответ или все его параметры указаны как обязательные.

По результату выполнения запроса в трейс выводится информация о выполненном запросе и его параметрах и о времени выполнения запроса в миллисекундах.

Ограничения

1. Одному файлу маппинга соответствует одна таблица одной базы данных. При необходимости использования нескольких таблиц в запросе необходимо использовать оператор UNION.
2. Размер набора результатов, если он есть, должен быть равен одной строке. При большем или меньшем количестве строк происходит ошибка.
3. Атрибуты nullable и default в XSD-файле не поддерживаются.
4. Адаптер не может самостоятельно генерировать события — обращаться к БД и вызывать систему Notification System.
5. Объектные типы БД не поддерживаются.
6. В OracleDataBase не поддерживаются массивы.
7. Параметры имеют плоскую структуру.

Установка и настройка адаптера для БД

1. Установите SA NS DbProducer — [Установка SA NS DbProducer](#).
2. Настройте источник данных — [Настройка источника данных](#).
3. Создайте файл маппинга — [Создание файла маппинга](#).
4. Создайте XSD-файл — [Создание XSD-файла](#).
5. Настройте SA NS DbProducer — [Настройка SA NS DbProducer](#).
6. Задайте маршрут от NsSLES к LW SA Container — [Настройка маршрута](#).

Установка SA NS DbProducer

Установите SA NS DbProducer. Для этого скопируйте из дистрибутива системы NotificationSystem/deployable/lwsa_deployable/ файл SA_NS_DbProducer_v1.war в каталог на сервере LWSA/webapps.

Настройка источника данных

1. Создайте MIB-группу `/LWSAContainer_embo/DataSources/ConnectionsList/<jndi-name>/`.
2. Настройте подключение к базе данных. Для этого перейдите в MIB-группу `/LWSAContainer_embo/DataSources/ConnectionsList/<jndi-name>/ProtocolLayers/Jdbc` и задайте значения параметров:
 - `Url`;
 - `UserName`;
 - `Password`;
 - `ConnectionType`.
3. В `/LWSA_Kafka/Core/Configuration/PortTypeXAConnections/` создайте группу для `portType {http://www.bercut.com/wsdl/em/dbproducer}NS_DbProducer`.
4. В MIB-группе `/LWSA_Kafka/Core/Configuration/PortTypeXAConnections/{http://www.bercut.com/wsdl/em/dbproducer}NS_DbProducer` присвойте значение `<jndi-name>` параметру `DataSourceName`.

Примечание. Если источник данных недоступен, адаптер отключает свой `portType`. Для корректной работы этой функции настройте мониторинг `DbConnectionMonitor` [1].

Создание файла маппинга

Создайте файл маппинга и опишите в нем, как должна выполняться подстановка параметров в запросе. Название файла маппинга должно соответствовать шаблону `*.query.xml`.

Пример файла маппинга:

```
<db-mapping>
  <request>
    <rtsib-element>DBAdapterTestRequest</rtsib-element>

    <param>
      <name>name</name>
      <to>1</to>
    </param>

    <param-value>
      <value>24</value>
      <to>2</to>
      <type>int</type>
    </param-value>

    <param-value>
      <value>${timevar}</value>
```



```

        <to>3</to>
        <type>time</type>
    </param-value>
</request>

<query> select name, timevar, datevar, datetimevar, intvar, intarrayvar, durationvar
        from test
        where name like ? and intvar = ? and timevar = ?
</query>

<support-fm>true</support-fm>

<response>
    <rtsib-element>DBAdapterTestResponse</rtsib-element>

    <param>
        <from>1</from>
        <to>name</to>
    </param>
    <param>
        <from>2</from>
        <to>timevar</to>
    </param>
    <param>
        <from>3</from>
        <to>datevar</to>
    </param>
    <param>
        <from>4</from>
        <to>datetimevar</to>
    </param>
    <param>
        <from>5</from>
        <to>intvar</to>
    </param>
    <param>
        <from>6</from>
        <to>arrayintvar</to>
    </param>
    <param>
        <from>7</from>
        <to>durationvar</to>
    </param>
</response>
</db-mapping>

```

Где:

- **request** — маппинг входных параметров. Может быть пустым, но должен быть обязательно;
- **request/rtsib-element** — связь с входным XSD-элементом — обязательно;
- **request/param/name** — имя параметра из XSD-файла;
- **request/param/to** — индекс, в который должен быть записан параметр. Тип определяется на основании XSD-файла;
- **request/param-value/value** — значение параметра. Ссылка на конкретный параметр. Допускается *синтаксис FreeMarker*. Пример: `{name}`;
- **request/param-value/to** — индекс, в который должен быть записан параметр;
- **param/type** — тип параметра — опционально;
- **query** — запрос — обязательно. Может быть в виде объекта PreparedStatement. Должен относиться к таблице в базе данных, к которой настроено подключение.

i Примечание. Пример таблицы *test*:

```
CREATE TABLE test (
  id serial primary key,
  name character varying(2000),
  timevar time NOT NULL,
  datevar date NOT NULL,
  datetimevar timestamp NOT NULL,
  intvar integer NOT NULL,
  intarrayvar integer[],
  durationvar character varying(2000)
)
```

- **response**³ — маппинг выходных параметров — опционально;
- **response/rtsib-element** — связь с выходным XSD-элементом — обязательно;
- **response/param** — маппинг значений из resultSet на элементы XSD-структуры.

i Примечание. *request/param/to* и *request/param-value/to* не должны повторно заполнять один индекс. Индексы в файле должны быть проставлены последовательно и без пропусков.

В файле маппинга допустимо использовать следующие входные параметры:

- **<param>** — маппинг значения происходит строго по имени параметра из XSD-описания;
- **<param-value>** — имеет два подтипа параметров:
 - прямое указание значения в теге **<value>**;
 - FreeMarker-выражение *типа plain*, в которое можно передать ссылку на параметр из XSD-описания.

! Внимание! Чтобы адаптер определил, что FreeMarker-выражение должно распознаваться как массив, передайте в значении **<type>** один из используемых типов с **['']** на конце. Пример:

```
<param-value>
  <value>${intvar}</value>
  <to>3</to>
  <type>int[]</type>
</param-value>
```

Поддерживаются следующие XSD-типы:

- **xsd:string** — любой набор символов;
- **xsd:boolean**⁴ — значения 'True' и 'False';
- **xsd:int** — положительные и отрицательные целочисленные значения;
- **xsd:decimal** — значения с плавающей точкой;
- **xsd:unsigned-int** — положительные целочисленные значения;
- **xsd:negative-int** — отрицательные целочисленные значения;
- **xsd:datetime** — дата и время в формате ISO 8601 с поддержкой TimeZone;
- **xsd:date** — дата в формате ISO 8601 с поддержкой TimeZone;

³ Если **response** — это пустой список, вызывается **executeUpdate**. Если ожидаются выходные параметры, вызывается **executeQuery**.

⁴ В БД Oracle передается как **INTEGER**, так как в ней отсутствует **boolean num**.

- `xsd:time` — время в формате ISO 8601 с поддержкой `TimeZone`;
- `xsd:duration` — временной промежуток⁵.

Примечание. Отдельно выделяется любой из поддерживаемых типов с атрибутом `unbounded`. Тип с этим атрибутом распознается как массив и добавляется в базу как массив значений соответствующего типа.

Скопируйте файл маппинга в каталог `em_data`.

Создание XSD-файла

Создайте XSD-описание со схемой элементов `request` и `response`. Значения `<xsd:complexType name>` должны соответствовать `request/rtsib-element` и `response/rtsib-element` в файле маппинга.

Поместите это описание и все необходимые XSD-схемы и интерфейсы в JAR-файл и скопируйте в каталог `em_data`.

Примечание. Если запрос выполняет операцию `UPDATE`, то `response` не заполняется — `<sequence>` остается пустым.

Примечание. Адаптер не поддерживает типы `nillable`, поэтому отсутствие значения передается через `minOccurs="0"`.

Пример XSD-описания:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://www.tele2.com/events/any-system"
targetNamespace="http://www.tele2.com/events/any-system"
xmlns:commd="http://bercut.com/em/schema/CommonDefinitions"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.0">
<xsd:import schemaLocation="BA_EM_CommonDefinitions.xsd" namespace="http://bercut.com/em/schema/CommonDefinitions"/>

  <xsd:element name="DBAdapterTestRequest" type="tns:t-DBAdapterTestRequest"
  commd:anyType="request"></xsd:element>
  <xsd:element name="DBAdapterTestResponse" type="tns:t-DBAdapterTestResponse"
  commd:anyType="response"></xsd:element>

  <xsd:complexType name="t-DBAdapterTestRequest">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"></xsd:element>
      <xsd:element name="timevar" type="xsd:time"></xsd:element>
      <xsd:element name="datevar" type="xsd:date"></xsd:element>
      <xsd:element name="datetimevar" type="xsd:dateTime"></xsd:element>
      <xsd:element name="intvar" type="xsd:int"></xsd:element>
      <xsd:element name="durationvar" type="xsd:duration"></xsd:element>
      <xsd:element name="arrayintvar" type="xsd:int" minOccurs="0"
maxOccurs="unbounded"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="t-DBAdapterTestResponse">
    <xsd:sequence>
      </xsd:sequence>
    </xsd:complexType>

</xsd:schema>
```

⁵ В БД отобразится как формат `string`.

Настройка SA NS DbProducer

1. Перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NS_DbProducer/Core/Configuration`. Для параметра `ReloadMappingFiles` задайте значение 'True' для перезагрузки файлов маппинга и XSD-схем.
2. Перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NS_DbProducer/Core/Configuration/Elements` и создайте вложенные группы с названием, которое строго совпадает с названием входного элемента `request/rtsib-element` в созданной XSD-схеме.
3. В созданных подгруппах задайте значения для параметра:
 - `MappingFile` — имя файла маппинга. Должно соответствовать шаблону `*.query.xml`.

Настройка маршрута

Настройте соединение NsSLES с LW SA Container. Перейдите в MIB-группу `/SLES_em/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` и создайте маршрут: `{http://www.bercut.com/wsdl/em/dbproducer}NS_DbProducer`. В подгруппе `ScalingTable` задайте параметры соединения с LW SA Container, на котором развернут SA NS DbProducer.

4.10. Установка и настройка SA NS GroupValues

Настройка SA NS GroupValues не является обязательным требованием для функционирования системы Notification System.

Компонент SA NS GroupValues используется для работы с группами в системе Notification System с помощью методов SOAP API.

Установка SA NS GroupValues

Установите SA NS GroupValues. Для этого:

1. Остановите LW SA Container.
2. Скопируйте из дистрибутива системы `NotificationSystem/deployable/lwsa_deployable/` файл `SA_NS_GroupValues_v1.war` в каталог на сервере `LWSA/webapps`.
3. Запустите контейнер LW SA Container.

Настройка SA NS GroupValues

1. Настройте соединение с DB GROUP VALUES. Перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NsGroupValues/GroupValues/Configuration/DbConnection/` и настройте значение параметра:
 - `JndiDataSource` = `jdbc/groupsdb`.
2. Перейдите в MIB-группу `/LWSAContainer/DataSources/ConnectionsList/jdbc/groupsdb/ProtocolLayers/Jdbc` и задайте значения параметров:
 - `Url`
 - `UserName`
 - `Password`
 - `ConnectionType` = `postgre`
3. Перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NsGroupValues/Core/Configuration/` и в значении параметра `Permissions` укажите пользователей, имеющих доступ к API.

4.11. Установка и настройка SA NS ScheduledScenario

Установите SA NS ScheduledScenario, настройте взаимодействие LW SA Container, на котором установлен SA NS ScheduledScenario с NsSLES и системой Scheduler Subsystem.

Адаптер SA NS ScheduledScenario предназначен для взаимодействия с подсистемой Scheduler Subsystem [19] в процессе выполнения отложенных сценариев. Интерфейс взаимодействия Notification System и SA NS ScheduledScenario определен в AOI_NS_ScheduledScenario.wsdl.

! Внимание! Для работы с системой Notification System 2.10 используйте версию системы Scheduler Subsystem не ниже 3.2.

Установка SA NS ScheduledScenario

Установите SA NS ScheduledScenario. Для этого:

1. Остановите LW SA Container.
2. Скопируйте из дистрибутива системы NotificationSystem/deployable/lwsa_deployable/ файл SA_NS_ScheduledScenario_v1.war в каталог на сервере LWSA/webapps.
3. Запустите LW SA Container.

Настройка SA NS ScheduledScenario

1. Настройте MIB-параметры SA NS ScheduledScenario. Перейдите в MIB-группу /LWSAContainer/@artifacts/SA_NsScheduledScenario/Core/Configuration/ и задайте значения параметрам:

- *RtsibErrorBehavior* — описание поведения при ошибках опправки сообщения по RTSIB.

Возможны три типа поведения:

- *IGNORE* — игнорирование ошибки. Сообщение в трейсы на уровне debug.
- *ALARM* — сообщение в трейсы на уровне alarm.
- *SUSPEND* — сообщение в трейсы на уровне alarm. Приостановка работы на определенное время.

Поведение по умолчанию — *ALARM*.

Заранее заданные типы ошибок:

- *TIMEOUT* — тайм-аут при отправке в BRE.
- *ROUTE_UNAVAILABLE* — маршрут не настроен.
- *UNKNOWN_COMPONENT* — компонент, к которому настроен маршрут, не найден.
- *COMPONENT_UNAVAILABLE* — компонент недоступен.
- *DEFAULT* — прочие ошибки.

Вы можете задать другие типы ошибок. Для этого укажите строку, содержащую окончание названия исключения. Например, если нужно обработать ошибку `java.lang.IllegalArgumentException`, можно указать *IllegalArgumentException=IGNORE*.

Разделитель между поведением и типом ошибки — '='. Разделить между записями — перенос строки или ';'.

Изменения применяются без перезагрузки.

- *BreErrorBehavior* — описание поведения при получении ошибок выполнения от BRE.

Возможны три типа поведения:

- *IGNORE* — игнорирование ошибки. Сообщение в трейсы на уровне debug.
- *ALARM* — сообщение в трейсы на уровне alarm.
- *SUSPEND* — сообщение в трейсы на уровне alarm. Приостановка работы на определенное время.

Поведение по умолчанию — *ALARM*.

Заранее заданные типы ошибок:

- *NOT_SUPPORTED_MESSAGE* — ошибка может быть получена от BRE, если туда отправлено сообщение, для обработки которого нет ни одного активного сценария.
- *DEFAULT* — прочие ошибки.

Вы можете задать другие типы ошибок. Для этого укажите строку, содержащую окончание названия исключения. Например, если нужно обработать ошибку `java.lang.IllegalArgumentException`, можно указать *IllegalArgumentException=IGNORE*.

Разделитель между поведением и типом ошибки — '='. Разделить между записями — перенос строки или ';'.

- *SuspenseRetryTimeoutInMillis* — время ожидания перед возобновлением работы в миллисекундах, если сработало поведение *SUSPEND*.
- *TaskTimeoutInSeconds* — тайм-аут в секундах, передаваемый в SA Scheduler при создании задания.
- *MaxPollTimeoutInMillis* — максимальное время ожидания в миллисекундах перед следующим запросом в SA Scheduler, если задания не получены.
- *PollThreadCount* — количество потоков для обращения к SA Scheduler.
- *TaskCountPerPoll* — количество заданий, получаемых за один запрос к SA Scheduler.
- *MaxExecutionTime* — максимальное время в секундах, отводимое для выполнения отложенного сценария.
- *MaxLifeTime* — время в секундах, в течение которого отложенный сценарий должен быть выполнен, если он по каким-то причинам не был выполнен в указанное время. Не может быть меньше *MaxExecutionTime*.

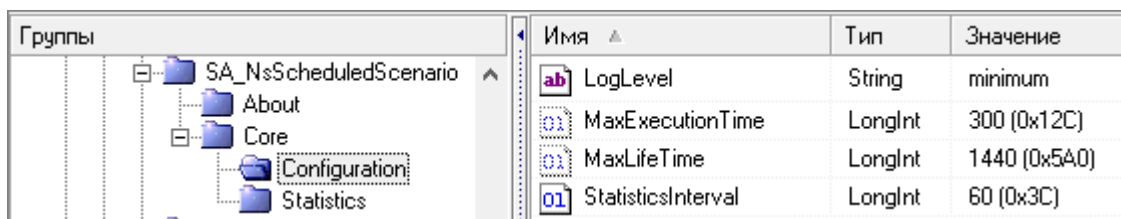


Рис. 23. Настройка MIB-параметров SA NS ScheduledScenario

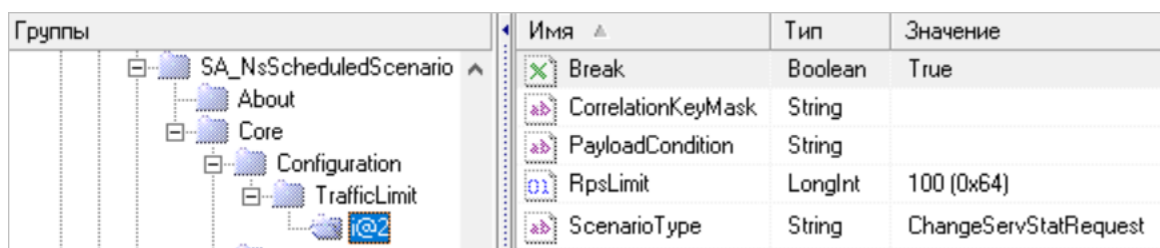
- Для управления нагрузкой, создаваемой вызовом отложенных сценариев в MIB-группе `/LWSAContainer/@artifacts/SA_NsScheduledScenario/Core/Configuration/` перейдите в подгруппу `TrafficLimit`. `TrafficLimit` содержит список условий, по которым устанавливаются ограничения нагрузки. Внутри `TrafficLimit` создайте подгруппы `Condition1`, `Condition2`, ... `ConditionN` по необходимому количеству условий.

Каждое условие накапливает статистику запросов. Если количество запросов превышает максимальное, запросы перестают обрабатываться и выполняются

тогда, когда количество запросов менее максимального. Условия перебираются по порядку. Сначала проверяются частные ограничения, затем — общие.

В подгруппах *ConditionN* задайте значения параметров:

- *RpsLimit* — максимальное количество запросов в секунду. Значение по умолчанию — '100'.
- *ScenarioType* — тип отложенного сценария. Заполнение необязательно.
- *CorrelationKeyMask* — регулярное выражение, которое применяется к полю *CorrelationKey*. Заполнение необязательно.
- *PayloadCondition* — XPath-выражение, с помощью которого можно обратиться к любому параметру отложенного сценария. Заполнение необязательно.
- *Break* — переменная типа Boolean. Если *Break* имеет значение 'True', перебор условий прекращается при достижении максимального количества запросов.



Имя	Тип	Значение
Break	Boolean	True
CorrelationKeyMask	String	
PayloadCondition	String	
RpsLimit	LongInt	100 (0x64)
ScenarioType	String	ChangeServStatRequest

Рис. 24. Пример настройки TrafficLimit

Для каждого условия доступны статистические переменные:

- *FitTasksCnt* — общее количество выполненных запросов, подходящих под условие;
- *FitTasksMax* — максимальное количество выполненных запросов, подходящих под условие, за секунду;
- *FitTasksMin* — минимальное количество выполненных запросов, подходящих под условие, за секунду;
- *FitTasksAvg* — среднее количество выполненных запросов, подходящих под условие, за секунду;
- *UnfitTasksCnt* — общее количество отклоненных запросов, подходящих под условие, за секунду;
- *UnfitTasksMax* — максимальное количество отклоненных запросов, подходящих под условие, за секунду;
- *UnfitTasksMin* — минимальное количество отклоненных запросов, подходящих под условие, за секунду;
- *UnfitTasksAvg* — общее количество отклоненных запросов, подходящих под условие, за секунду;
- *FitTasksAvgSpeed* — средняя скорость выполнения запросов;
- *CurrentRps* — текущее количество выполняемых запросов за секунду;
- *MaxRps* — максимальное количество выполненных запросов за секунду.

3. Настройте соединение NsSLES с LWSA Container. Перейдите в MIB-группу `/SLES_em/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` и создайте маршрут:

- `{http://www.bercut.com/wsdl/ns/scheduledscenario}NS_ScheduledScenarioManager`

В подгруппе *ScalingTable* задайте параметры соединения с LWSA Container, на котором развернут SA NS ScheduledScenario. По данному маршруту будут отправляться запросы управления отложенным исполнением сценария — действия (ACTION) *createScheduledScenario*, *replaceScheduledScenario*, *deleteScheduledScenario*, а также GETTER *checkScheduledScenario*.

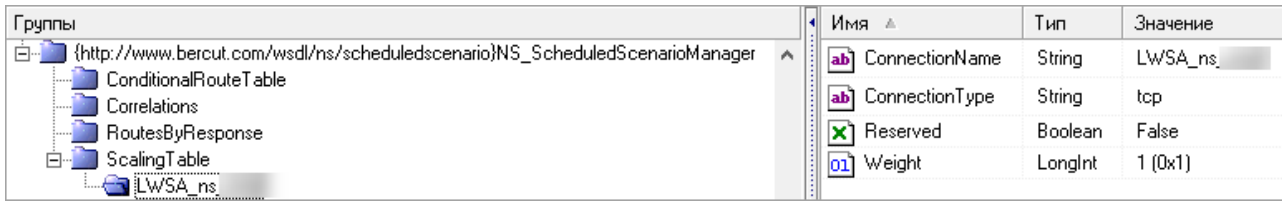


Рис. 25. Настройка маршрута от NsSLES к LW SA Container

Убедитесь, что LWSA Container, на котором развернут SA NS ScheduledScenario, присутствует среди заданных в группе `/SLES_em/@libs/MessageBus/ExternalConnections/`.

4. Настройте соединение LW SA Container с NsSLES. Перейдите в MIB-группу `/LWSA/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` и создайте маршрут:

- `{http://www.bercut.com/wsd/ns/scheduledscenario}NS_ScheduledScenarioExecuter`

В подгруппе `ScalingTable` задайте параметры соединения с NsSLES. По данному маршруту будет отправляться запрос на выполнение отложенного исполнения сценария — событие `scheduledScenarioExecute`.

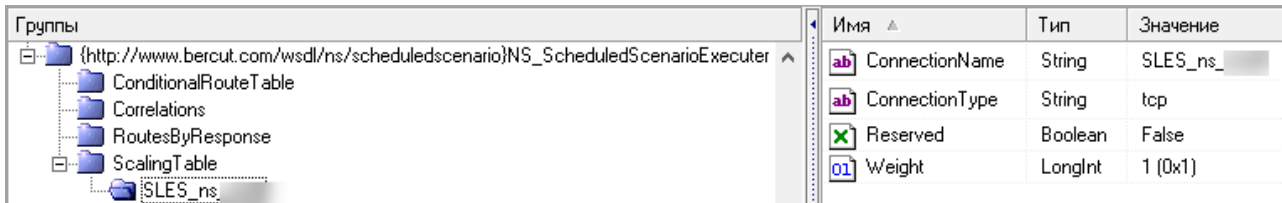


Рис. 26. Настройка маршрута от LW SA Container к NsSLES

5. Настройте соединение LW SA Container с LWSA_Scheduler. LWSA_Scheduler — LWSA Container, на котором установлен SA Scheduler системы Scheduler Subsystem.

Примечание. Подробнее об установке и настройке системы Scheduler Subsystem — [19].

Интерфейс взаимодействия систем NS и Scheduler Subsystem определен в `AOI_Scheduler_v1.wsdl`.

Настройка зависит от места развертывания SA Scheduler:

- Если SA Scheduler и SA NS ScheduledScenario развернуты на одном и том же LW SA Container, то для их взаимодействия достаточно настроить локальное соединение в одной из MIB-групп:
 - `/LWSA/@libs/MessageBus/Core/Configuration/Routing/DefaultScaleTable/local/` и указать `ConnectionType = local`.
 - `/LWSA/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` создайте маршрут `{http://www.bercut.com/spec/aoi/scheduler}SchedulerManagementPortType/` и в подгруппе `ScalingTable` укажите, что маршрут локальный:



Рис. 27. Пример настройки локального маршрута между SA Scheduler и SA NS ScheduledScenario

- Если SA Scheduler и SA NS ScheduledScenario развернуты на разных LW SA Container, то настройте следующие маршруты:
 1. От LW SA Container с SA NS ScheduledScenario к LWSA_Scheduler. На LW SA Container, где развернут SA NS ScheduledScenario, перейдите в MIB-группу `/LWSA/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` и создайте маршрут `{http://www.bercut.com/spec/aoi/scheduler}SchedulerManagementPortType/` в сторону LW SA Container, где развернут SA Scheduler. По данному маршруту будут отправляться запросы на управление задачами — *replaceJob*, *deleteJobs*, *getJobs*.
 2. От LW SA Container к LWSA_Scheduler. На LW SA Container, где развернут SA NS ScheduledScenario, перейдите в MIB-группу `/LWSA/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` и создайте маршрут `{http://www.bercut.com/spec/aoi/scheduler}SchedulerWorkerPortType` в сторону LW SA Container, где развернут SA Scheduler. По данному маршруту будут отправляться запросы на завершение задачи — *releaseTask*.
 3. Убедитесь, что на LW SA Container, где развернут SA Scheduler, настроен маршрут к LW SA Container: `{http://www.bercut.com/spec/aoi/scheduler}SchedulerEventPortType`. По данному маршруту в систему Notification System будут приходить запросы на запуск задачи — *onStartTask*.

4.12. Установка и настройка компонентов системы BRE MicroServices

Настройка компонентов системы BRE MicroServices не является обязательным требованием для функционирования системы Notification System.

При необходимости запуска микросервисов системы выполните следующие настройки:

1. На NsSLES установите все бизнес-процессы из каталогов `\nssles_deployable` и `\sles_common` установочного архива системы BRE MicroServices.

Убедитесь, что в MIB Explorer в `@artifacts` появились каталоги с названием микросервисов.

2. Проверьте, что на NsSLES:

- 2.1. Настроены внешние маршруты в MIB-группе `/NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/`:

- `{http://www.bercut.com/wsd/GetServiceName}GetServiceNamePortType`

В подгруппе `ScalingTable` добавьте маршрут до SLES_Notifications: `/NsSLES/@libs/MessageBus/Core/Configuration/Routing/`

- PortTypeRouting/{http://www.bercut.com/wsd/GetServiceName}GetServiceNamePortType/ScalingTable/SLES_Notifications/*
- *{http://www.bercut.com/wsd/GetPersonalRate}GetPersonalRatePortType*
В подгруппе *ScalingTable* добавьте маршрут до SLES_SOAPAPI: */NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/{http://www.bercut.com/wsd/GetPersonalRate}GetPersonalRatePortType/ScalingTable/SLES_soapAPI/*
- *{http://www.bercut.com/wsd/Tele2SoapAPI/DelayManagementInternal}DelayManagementInternalPortType*
В подгруппе *ScalingTable* добавьте маршрут до SLES_SOAPAPI: */NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/{http://www.bercut.com/wsd/Tele2SoapAPI/DelayManagementInternal}DelayManagementInternalPortType/ScalingTable/SLES_soapAPI/*
- *{http://j2ee.netbeans.org/wsd/CPASubscriptionWSDL}CPASubscriptionWSDLPortType/*
В подгруппе *ScalingTable* добавьте маршрут до SLES_CPA: */NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/{http://j2ee.netbeans.org/wsd/CPASubscriptionWSDL}CPASubscriptionWSDLPortType/ScalingTable/SLES_CPA/*
- *{http://docs.oasis-open.org/ws-tx/wsat/2006/06}ParticipantPortType/*
В подгруппе *ScalingTable* добавьте маршрут до:
 - LWSA_InVoice;
 - LWSA_Custom;
 - SLES_SelfCare;
 - SLES_soapAPI.
- *{http://www.bercut.com/wsd/PromisePayInternal}PromisePayAcceptOffer/*
В подгруппе *ScalingTable* добавьте маршрут до SLES_SelfCare: */NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/{http://www.bercut.com/wsd/PromisePayInternal}PromisePayAcceptOffer/ScalingTable/SLES_SelfCare/*
- *{http://www.bercut.com/wsd/PromisePayInternal}PromisePayAcceptOfferTransactional/*
В подгруппе *ScalingTable* добавьте маршрут до SLES_SelfCare: */NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/{http://www.bercut.com/wsd/PromisePayInternal}PromisePayAcceptOfferTransactional/ScalingTable/SLES_SelfCare/*
- *{http://www.bercut.com/wsd/ChangeSubsRateInternalTransactional}ChangeSubsRateInternal/*
В подгруппе *ScalingTable* добавьте маршрут до SLES_soapAPI: */NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/{http://www.bercut.com/wsd/ChangeSubsRateInternalTransactional}ChangeSubsRateInternal/ScalingTable/SLES_soapAPI/*
- *{http://www.bercut.com/wsd/PCServicesTransactional}PCServices/*

В подгруппе *ScalingTable* добавьте маршрут до SLES_soapAPI:

```
/NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/  
{http://www.bercut.com/wsd/PCServicesTransactional}PCServices/  
ScalingTable/SLES_soapAPI/
```

- {http://www.bercut.com/wsd/RateReconfigurationInternalTransactional}RateReconfigurationInternalPortType/

В подгруппе *ScalingTable* добавьте маршрут до SLES_soapAPI:

```
/NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/  
{http://www.bercut.com/wsd/RateReconfigurationInternalTransactional}  
RateReconfigurationInternalPortType/ScalingTable/SLES_soapAPI/
```

2.2. В группе */NsSLES/@libs/MessageBus/ExternalConnections/ConnectionsList/* указаны все внешние компоненты.

2.3. Убедитесь, что на LWSA_Invoice, LWSA_Custom, SLES_SelfCare и SLES_soapAPI добавлены следующие маршруты к NsSLES в группах <имя LWSA>/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/:

- {http://docs.oasis-open.org/ws-tx/wsat/2006/06}CoordinatorPortType/
- {http://docs.oasis-open.org/ws-tx/wscoor/2006/06}RegistrationPortType/

В подгруппах *ScalingTable* добавьте маршрут к NsSLES.

2.4. Убедитесь, что на LWSA_sp добавлены следующие маршруты к NsSLES в MIB-группе *LWSA_sp/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/*:

- {http://docs.oasis-open.org/ws-tx/wsat/2006/06}CoordinatorPortType/
- {http://docs.oasis-open.org/ws-tx/wscoor/2006/06}RegistrationPortType/

В подгруппах *ScalingTable* добавьте маршрут к NsSLES.

3. Загрузите микросервисы. Вы можете использовать один из следующих методов:

3.1. На сервере, где установлен LWSA_embo, в каталог *em_data* добавьте JAR-файлы из каталогов *\nsbo_deployable* установочного архива системы BRE MicroServices. Затем перезагрузите LWSA_embo.

3.2. В веб-интерфейсе Notification System [5] перейдите на вкладку **Управление > Микросервисы** и установите микросервисы из каталога *\nsbo_deployable* установочного архива системы BRE MicroServices.

Убедитесь, что в MIB Explorer в */LWSA_embo/@artifacts* появились каталоги с названием микросервисов.

4. Проверьте, что на LWSA_embo для микросервисов настроено взаимодействие с NsSLES в MIB-группе */LWSA_embo/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/*:



Рис. 28. Вариант настройки маршрутов

5. Проверьте, что на LW SA Container в каталог em_data добавлен файл T2 Schema NS Events.jar из каталога \nslwsa_schema установочного архива системы BRE MicroServices.
6. Проверьте, что на LW SA Container в MIB-группе LWSA/Core/Configuration/ установлены следующие значения параметров:
 - `SystemPrincipalSource` = `VarSystemPrincipal`.
 - `UserPrincipalSource` = `VarUserPrincipal`.
 - `VarSystemPrincipal` = `WSMS`.
 - `VarUserPrincipal` = `WSMS`.
7. Убедитесь, что для бизнес-процесса BP SpaceSubscription на NsSLES в MIB-группе /NsSLES/@artifacts/BP_SpaceSubscription/Configuration/ установлены значения переменных:
 - `Login` = `t2_selfcare`.
 - `Password` = `t2_selfcare`.

4.13. Установка и настройка SA NS AuditAggregator

Вы можете настроить отображение агрегированных данных в NS Back Office на странице **Отчеты**.

Для настройки отчетов выполните следующие настройки:

1. Установите DB AGGREGATION, если она не была установлена ранее — 4.1.4.4.
2. На LW SA Container установите SA NS AuditAggregator. Вы можете использовать любой контейнер, например LW SA Container EM BO. Скопируйте в каталог LWSA_EMBO/webapps файл SA_NS_AuditAggregator_v1.war из каталога NotificationSystem/deployable/lwsa_deployable.

3. Настройте подключение к DB AGGREGATION и AUDIT в группе `SA_NsAuditAggregator/Core/Configuration/`:
 - `AggrDataSource` — значение DataSource для DB AGGREGATION;
 - `AuditDataSource` — значение DataSource для DB AUDIT;
 - `ScanPeriodInDays` — укажите количество дней, за которое осуществляется сканирование, в днях.
4. Убедитесь, что параметр `PREV_DAYS_FOR_CREATE` в таблице `NS_AGGR_SETTINGS` базы данных AGGREGATION равен или больше значения MIB-параметра `ScanPeriodInDays` — возможное время максимального простоя `SA_NS_AuditAggregator`.
5. Проверьте значение параметра `MAX_STORAGE_PERIOD` в таблице `NS_AGGR_SETTINGS` базы данных AGGREGATION — время хранения агрегированных данных.
6. Убедитесь, что в трейсах нет ошибок.

Через час после запуска в NS Back Office на странице **Отчеты** появятся данные.

i Примечание. В течение первого часа после запуска, пока данные не загрузились, на странице **Отчеты** система может выводить ошибки получения данных.

4.14. Установка и настройка SA KeyValuesStorage

Установите и настройте SA KeyValues Storage.

Предварительные требования

Для работы системы KeyValuesStorage необходимо, чтобы были установлены LW SA Container версии 3.5.25 и выше и СУБД Tarantool версии 2.8.3.

Установка SA KeyValues Storage

Установите SA KeyValuesStorage. Для этого:

1. Остановите LW SA Container.
2. Скопируйте из дистрибутива системы `keyvalues-storage-1.1-multiplatform-release.zip` файл `sa-kv.war` в каталог на сервере `LWSA/webapps`.
3. Запустите контейнер LW SA Container.

Настройка SA KeyValuesStorage

1. Перейдите в MIB-группу `LWSA/DataSources/ConnectionsList` и создайте два DataSource для подключения к Tarantool. Имя пользователя и пароль должны совпадать с именем пользователя и паролем в настройках Tarantool.

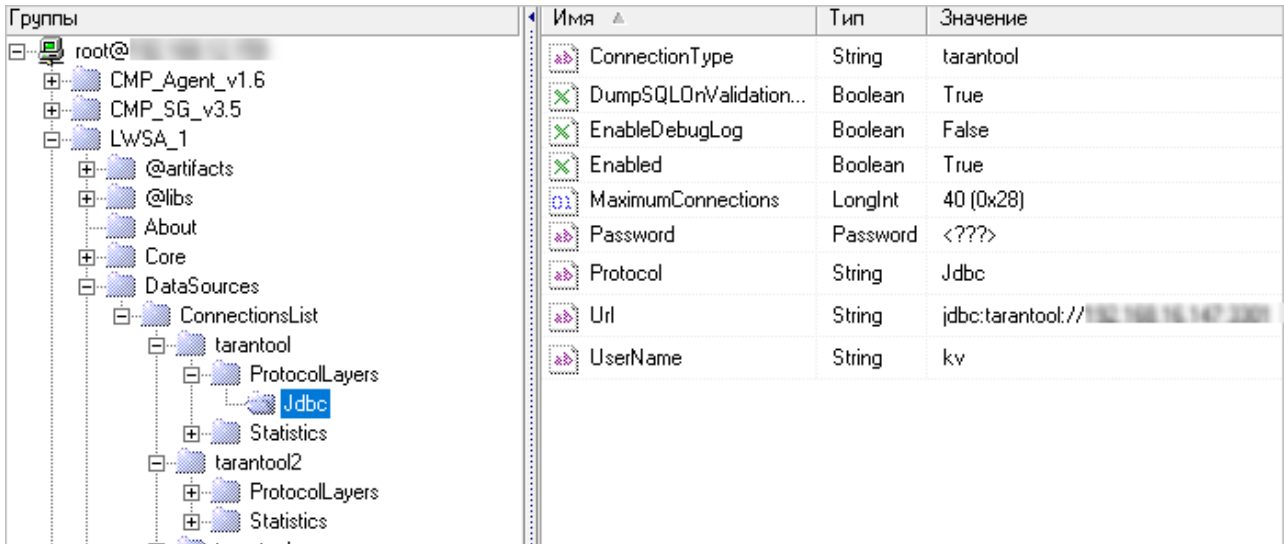


Рис. 29. Вариант настройки подключения к Tarantool

2. Перейдите в MIB-группу LWSA/Core/Configuration/DBConnectionMonitoring/DataSources и настройте мониторинг для каждого источника данных.

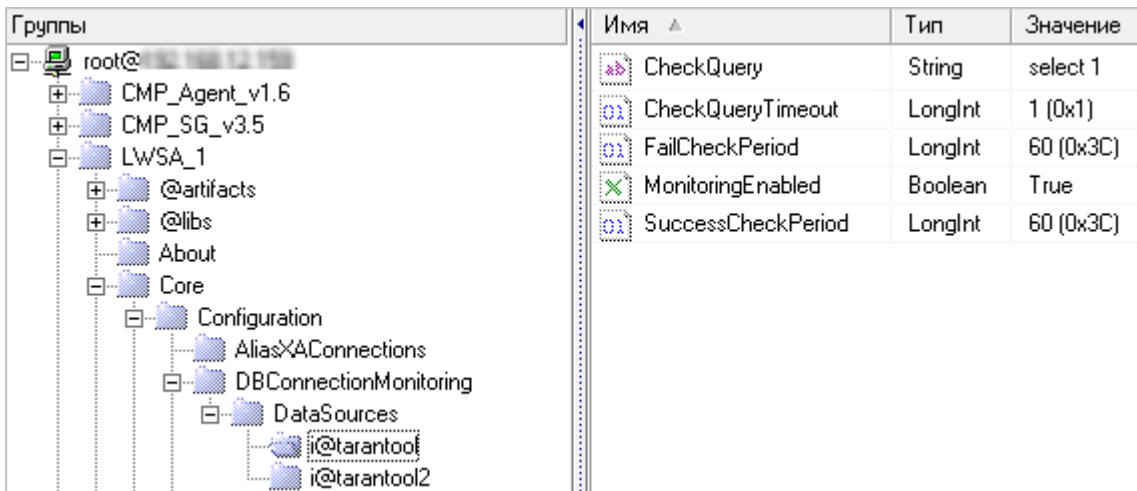


Рис. 30. Вариант настройки мониторинга

3. Перейдите в MIB-группу LWSA/@artifacts/SA_KeysValuesStorage/Core/Configuration и для параметра DataSource через запятую без пробела укажите названия созданных источников данных.

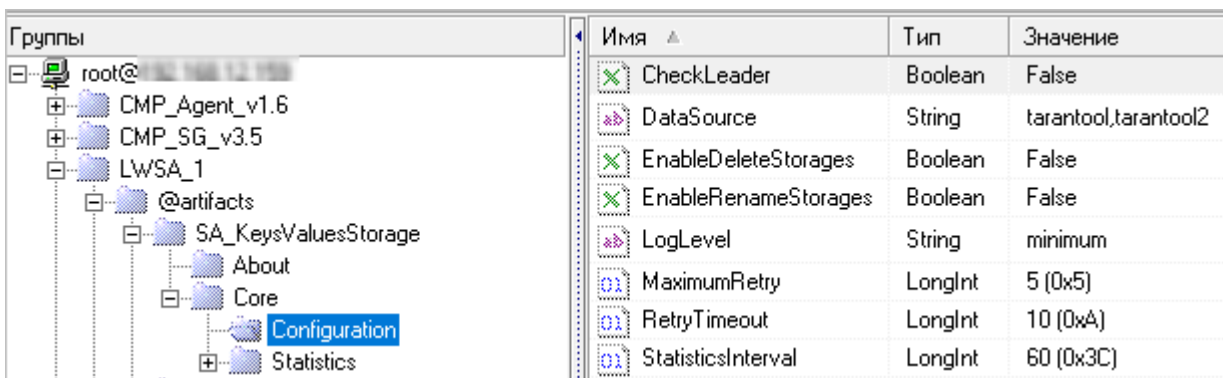


Рис. 31. Вариант настройки параметра DataSource

4. Перейдите в MIB-группу `/LWSA/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` и настройте следующие маршруты:

- `{http://www.bercut.com/spec/dsi/KeysValuesStorage}KeysValuesManagePortType;`
- `{http://www.bercut.com/spec/dsi/KeysValuesStorage}KeysValuesStoragePortType.`

4.15. Установка и настройка SA NS KVStorageManagement

Установите SA NS KVStorageManagement.

SA NS KVStorageManagement используется для взаимодействия с системой KeysValues Storage. Он позволяет управлять данными, привязанными к определенным ключам в KeysValues DB на платформе Tarantool.

Компонент позволяет в качестве критерия сценария выполнить проверку наличия значения по заданному ключу или в качестве операции сценария создать или удалить значение по заданному ключу.

i Примечание. При проверке наличия значения или при удалении значения задайте наименование хранилища, так как ключ уникален в одном хранилище. При создании значения, если наименование хранилища не указано, создается новое хранилище.

Установка SA NS KVStorageManagement

Установите SA NS KVStorageManagement. Для этого:

1. Остановите LW SA Container.
2. Скопируйте из дистрибутива системы `NotificationSystem/deployable/lwsa_deployable/` файл `SA_NS_KVStorageManagement_v1.war` в каталог на сервере `LWSA/webapps`.
3. Запустите контейнер LW SA Container.

Настройка SA NS KVStorageManagement

1. Перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NsKVStorageManagement/Core/Configuration/` и присвойте параметру `ReloadFiles` значение 'True' для включения перезагрузки микросервисов из каталога `em_data`. Для работы с системой KeysValuesStorage необходимо, чтобы был загружен микросервис `AOI_NS_KeyValueUniversalOperation.jar`.
2. Настройте маршрут от NsSLES к LW SA Container. Для этого перейдите в MIB-группу `/NsSLES/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` и создайте маршрут `{http://www.bercut.com/wsd/ns/kvstoragemanagement}NS_KVStorageManagement`. В подгруппе `ScalingTable` задайте параметры соединения с LW SA Container, на котором развернут SA NS KVStorageManagement.
3. Настройте маршруты от LW SA Container, на котором развернут SA NS KVStorageManagement до LW SA Container, на котором развернут SA KeysValuesStorage. Для этого в MIB-группе `/LWSA/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/` создайте маршруты `{http://www.bercut.com/spec/dsi/KeysValuesStorage}KeysValuesManagePortType` и `{http://www.bercut.com/spec/dsi/KeysValuesStorage}KeysValuesStoragePortType`. В подгруппе `ScalingTable` задайте параметры соединения с LW SA Container, на котором развернут SA KeysValuesStorage.

4.16. Настройка кеширования ответов на внешние вызовы

Настройте кеширование на NS SLES EM.

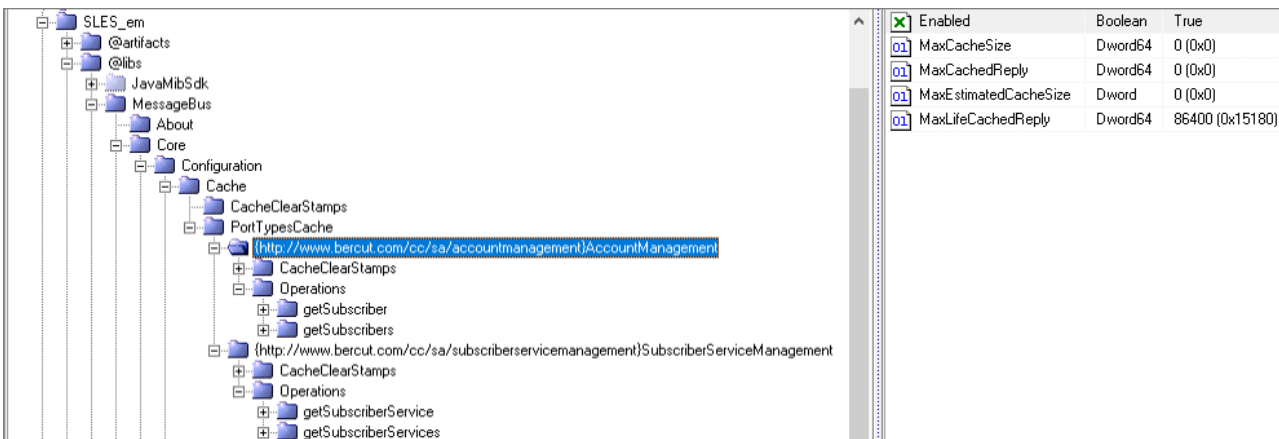
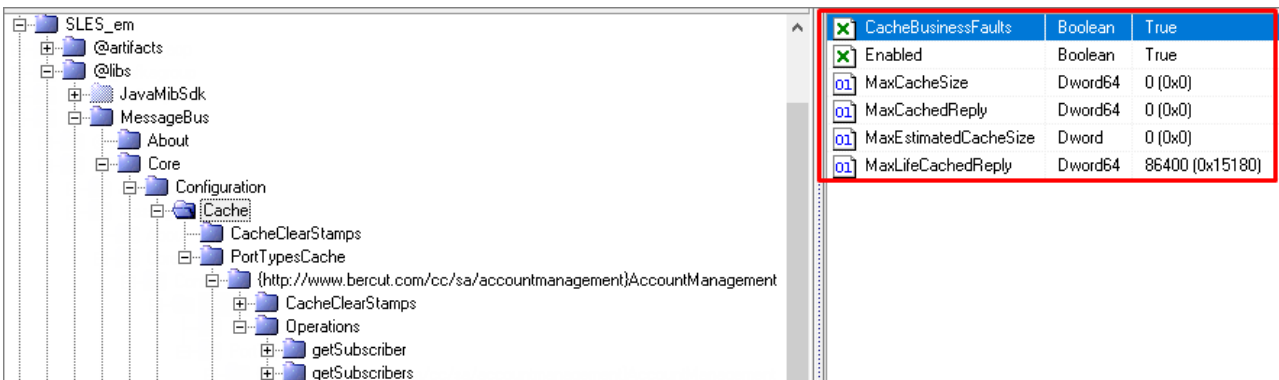
Кеширование поможет уменьшить нагрузку для вызовов во внешние системы. Настройка кеширования ответов на внешние вызовы выполняется в группах: [1]:

- @libs/MessageBus/Core/Configuration/Cache — для всех portType;
- @libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType> — для portType;
- @libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType>/Operations/<имя операции> — для операции portType.

Для уменьшения нагрузки на IN@Voice DB выполните следующие действия:

1. Настройте кеширование для следующих portType:

- {http://www.bercut.com/cc/sa/accountmanagement}AccountManagement. Для операций:
 - getSubscribers
 - getSubscriber.
- {http://www.bercut.com/cc/sa/subscriberservicemanagement}SubscriberServiceManagement. Для операций:
 - getSubscriberServices
 - getSubscriberService.



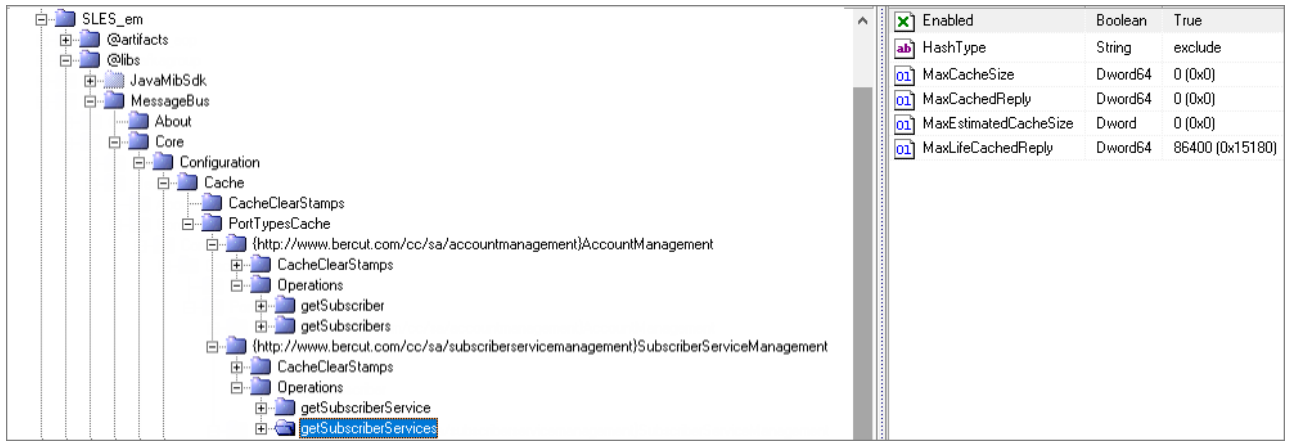


Рис. 32. Вариант настроек кеширования

2. Задайте XPath-пути для подсчета хеша в переменных XPath групп @libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType>/Operations/<имя операции>/XPaths/targetDate.

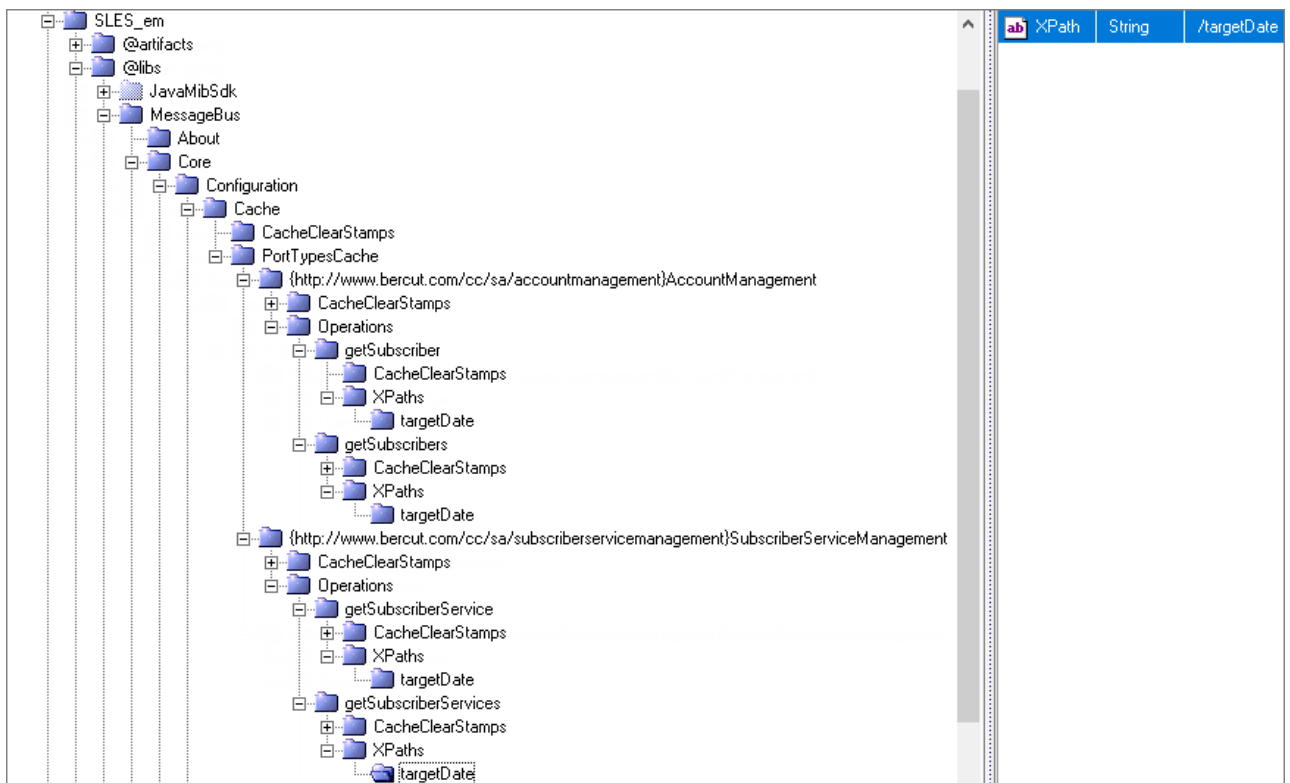


Рис. 33. Вариант настройки XPath-пути

5. Обновление системы

Если у вас уже *установлены и настроены компоненты системы*, вы можете обновить систему.

Обновление системы с версии 2.15 до 2.16

Порядок обновления:

1. Обновите NS SLES:

- 1.1. Остановите SLES с помощью SSM [1].
- 1.2. Удалите из NS SLES все каталоги, кроме `bin` и `deployed`.
- 1.3. Распакуйте архив `notification-system-2.16-multiplatform-release.zip/ns-sles-3.5-multiplatform-release.zip`.
- 1.4. Из каталога `notification-system-2.16-multiplatform-release/ns-sles-3.5-multiplatform-release/` скопируйте в NS SLES все каталоги, кроме `options.sh`, и установите SLES [1].
- 1.5. Запустите NS SLES.
- 1.6. Убедитесь в ATLAS MIB Explorer, что версия NS SLES стала актуальной. Вы можете увидеть это в MIB-группах `/NsSLES/About/` и `/NsSLES/@libs/NsInterpreter/About/`.
- 1.7. Убедитесь, что в момент запуска NS SLES в трейсах не было ошибок.

2. Обновите LW SA Container:

- 2.1. Остановите контейнер LWSA с помощью SSM [1].
- 2.2. Установите LWSA из каталога установки `lw-sa-container-3.5-multiplatform-release.zip`.
- 2.3. Запустите контейнер LWSA.
- 2.4. Проверьте в ATLAS MIB Explorer, что версия LWSA стала актуальной. Вы можете увидеть это в MIB-группе `/LWSAContainer/About/`.
- 2.5. Убедитесь, что в момент запуска контейнера LWSA в трейсах не было ошибок.

3. Обновите все специализированные адаптеры SA.

4. Обновите BO Notification System:

- 4.1. Остановите контейнер LWSA_EMBO с помощью SSM [1].
- 4.2. Удалите каталог `ROOT` и файл `ROOT.war` из каталога `LWSA_EMBO/webapps`.
- 4.3. Очистите каталог `LWSA_EMBO/work/Catalina/localhost/_/` и `LWSA_EMBO/BERCLwsa/temp/`.
- 4.4. Установите LWSA_EMBO из каталога установки `lw-sa-container-3.5-multiplatform-release.zip`.
- 4.5. Скопируйте файл `ROOT.war` из каталога `notification-system-2.16-multiplatform-release.zip/backoffice` и поместите его в каталог `LWSA_EMBO/webapps`.
- 4.6. Запустите контейнер LWSA_EMBO.
- 4.7. Проверьте версию LWSA_EMBO в MIB-группе `/LWSAContainer_bo/About/`. Версия должна быть актуальной.
- 4.8. Проверьте версию Back Office в MIB-группе `/LWSAContainer_bo/@artifacts/SA_EmBackOffice/About/`. Версия должна быть актуальной.
- 4.9. Убедитесь, что в момент запуска LWSA_EMBO в трейсах не было ошибок.
- 4.10. Проверьте, можно ли авторизоваться в BO Notification System с правами пользователя `root`.

6. Логирование

Настройте параметры хранения файлов в системе.

Выполните следующие настройки:

1. Добавьте в файл `/export/home/bercut/.profile` настройку максимального количества открытых файлов:

```
ulimit -n 16384
```

2. Настройте ограничение для хранения трейсовых сообщений от компонентов в MIB-группе `/CMP_Agent_v1.6/TraceLogger/Configuration`:
 - Установите значение переменной `Strategy=SizeLimit`. Это соответствует стратегии ограничения журнала событий по количеству файлов и их размеру.
 - Увеличьте объем хранилища. Для этого установите значение переменной `MaxSizeInKb=102400`.
 - Перезапустите компонент ATLAS Agent.
 - Убедитесь, что в системе достаточно места для хранения файлов журнала событий.
3. Увеличьте значение переменной `MaxFiles` с 10 до 50 в группе `/CMP_Agent_v1.6/TraceLogger/Configuration` для того, чтобы увеличить место для трейсовых сообщений стандартного каталога `/opt/BERCatlas/data/` до 50 Гб.

7. Мониторинг системы

Рекомендации по мониторингу состояния системы.

Для мониторинга системы вы можете использовать:

- основные показатели — [7.1](#),
- приложения Unica — [7.2](#),
- время отклика — [7.3](#).

7.1. Основные показатели

1. Мониторинг виртуального сервера по основным показателям: CPU, память, место на дисках.
2. Мониторинг активности всех запущенных приложений на хосте: nsSLES, LWSA, SG, StartStopManager, CMP_JvmMonitor_v1.0, Agent и тд.
3. Мониторинг активности всех адаптеров SA, развернутых на LWSAem, LWSAembo, LWSA__EM_kafka, LWSA_scheduler.
4. Мониторинг активности всех бизнес-процессов BP, развернутых на nsSLES.
5. Мониторинг состояния подключения ко всем базам данных на LWSAem, LWSAembo, LWSA__EM_kafka, LWSA_scheduler:

```
/CMP_LWSAContaine<префикс>/DataSources/ConnectionsList/<имя бд>/  
Statistics/StateInfo/Status
```

6. Мониторинг состояния доступности всех настроенных маршрутов в LWSAem, LWSAembo, LWSA__EM_kafka, LWSA_scheduler:

```
/LWSA<префикс>/@libs/MessageBus/Core/Statistics/StateInfo/RouteStatus/  
<имя PT>/RouteStatus
```

7. Мониторинг состояния всех исходящих адаптивных очередей на LWSAem, LWSAembo, LWSA__EM_kafka, LWSA_scheduler:

```
/LWSA<префикс>/@libs/MessageBus/Core/Statistics/StateInfo/RouteStatus/  
<имя PT>/QueueStatus
```

8. Мониторинг доступности всех входящих PT на LWSAem, LWSAembo, LWSA__EM_kafka, LWSA_scheduler:

```
/LWSA<префикс>/@libs/MessageBus/Core/Statistics/StateInfo/  
IncomingPortTypes/<имя PT>/State/Status/Status
```

9. Мониторинг состояния всех входящих адаптивных очередей на LWSAem, LWSAembo, LWSA__EM_kafka, LWSA_scheduler:

```
/LWSA<префикс>/@libs/MessageBus/Core/Statistics/StateInfo/  
IncomingPortTypes/<имя PT>/State/AdaptiveQueue/QueueStatus
```

10. Мониторинг состояния доступности всех настроенных маршрутов в nsSLES:

```
/NsSLES_<префикс>/@libs/MessageBus/Core/Statistics/StateInfo/  
RouteStatus/<имя PT>/RouteStatus
```

11. Мониторинг состояния доступности всех настроенных маршрутов в SG:

```
/SG_<префикс>/@libs/LIB_MessageBus_v3.3/Core/Statistics/StateInfo/  
RouteStatus/<имя PT>RouteStatus
```

12. Мониторинг состояния всех исходящих адаптивных очередей в nsSLES:

```
/NsSLES<префикс>/@libs/MessageBus/Core/Statistics/StateInfo/RouteStatus/<имя PT>/QueueStatus
```

13. Мониторинг доступности всех входящих PT в nsSLES:

```
/NsSLES<префикс>/@libs/MessageBus/Core/Statistics/StateInfo/IncomingPortTypes/<имя PT>/State/Status/Status
```

14. Мониторинг доступности портов всех приложений Platform v3 (все SG, LWSA и nsSLES):

- <имя приложения>/@libs/MessageBus/DataServer/Statistics/Status
- <имя приложения>/@libs/MessageBus/HttpServer/Statistics/Status
- <имя приложения>/@libs/MessageBus/LocalServer/Statistics/Status
- <имя приложения>/@libs/MessageBus/ServiceServer/Statistics/Status

15. Мониторинг потребления памяти и CPU всех Java-приложений, запущенных на сервере (через JvmMonitor):

- /CMP_JvmMonitor_v1.0/JVM/Instances/<имя приложения>/Statistics/StateInfo/CPU/CPUUsage на значение > 70%
- /CMP_JvmMonitor_v1.0/JVM/Instances/<имя приложения>/Statistics/StateInfo/Memory/HeapMemoryUsage > 80%

7.2. Мониторинг приложений Unica

1. Мониторинг подключения к DB:

- /UNICA_ML<префикс>/Security/Providers/Oracle/Status/In-Connected
- /UNICA_ML<префикс>/Security/Providers/Oracle/Status/Out1-Connected
- /UNICA_ML<префикс>/Security/Providers/Oracle/Status/Out2-Connected

2. Мониторинг подключений к SMPP серверам:

```
/UNICA_ML<префикс>/Security/Providers/Smpp/<номер сервера>/Status/Connected
```

3. Мониторинг ошибок доставки SMS:

- /UNICA_ML<префикс>/Security/Providers/Smpp/<номер сервера>/Statistics/Faults & Errors/Cancel-SM-Requests-Rejected > 0
- /UNICA_ML<префикс>/Security/Providers/Smpp/<номер сервера>/Statistics/Faults & Errors/Submit-SM-Requests-Rejected > 0
- /UNICA_ML<префикс>/Security/Providers/Smpp/<номер сервера>/Statistics/Faults & Errors/Unanswered-Requests-Count > 0

4. Мониторинг среднего времени отклика SMPP серверов:

i Примечание. Порог срабатывания аларма выбирается администратором системы на основании среднего времени отклика нормальной работы под нагрузкой (анализ графиков в Nagios или другой системы).

- /UNICA<префикс>/Security/Providers/Smpp/<номер сервера>/Statistics/Performance/Speed-Submit-Average
- /UNICA<префикс>/Security/Providers/Smpp/<номер сервера>/Statistics/Performance/Speed-Cancel-Average
- /UNICA<префикс>/Security/Providers/Smpp/<номер сервера>/Statistics/Performance/Speed-Total-Average

7.3. Время отклика

Мониторинг среднего времени отклика всех входящих PT на SG, LWSAem, LWSAembo, LWSA__EM_kafka, LWSA_scheduler и nsSLES:

```
<имя приложения>/@libs/MessageBus/Core/Statistics/StateInfo/  
IncomingPortTypes/<имя PT>RequestResponseTimeAvg
```

Порог срабатывания аларма выбирается администратором системы на основании среднего времени отклика нормальной работы под нагрузкой (анализ графиков в Nagios или другой системы).

Аларм должен срабатывать при превышении порога значения 2 раза в течении 10 минут. Значение может быть изменено администратором.

Приложение 1. Работа с отложенными сценариями

Создание и настройка отложенного исполнения сценариев.

Для работы с отложенным исполнением сценариев вы должны:

1. Создать микросервис для отложенного исполнения сценариев и загрузить его в NS Back Office — [П 1.1.](#)
2. Настроить сценарии — [П 1.2.](#)

i Примечание. При добавлении новых параметров проверьте, что обновлен XSD-файл и в нем присутствуют эти параметры.

Вариант настройки отложенного исполнения сценариев на примере события `updateLocation` — [П 1.3.](#)

П 1.1. Создание микросервиса

Создайте микросервис для отложенного исполнения сценариев, загрузите его в NS Back Office.

Для создания отложенного исполнения сценария:

1. Создайте XSD-файл с описанием входных параметров для отложенного исполнения сценария. При создании проверьте, что в XSD-файле:
 - присутствует импорт данных из файла `NS_ScheduledScenarioDefinition.xsd`;
 - заданы обязательные параметры;
 - задан атрибут **`cd:anyType="request"`**. В атрибуте `anyType` укажите параметры запроса, который должен получить сценарий при отложенном исполнении. Обычно описывает действие, которое сценарий должен выполнить при отложенном исполнении.

Пример содержимого XSD-файла:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.bercut.com/wsd1/ms/ScheduledCodeScenario"
  xmlns:tns="http://www.bercut.com/wsd1/ms/ScheduledCodeScenario"
  xmlns:cd="http://bercut.com/em/schema/CommonDefinitions"
  xmlns:ssdef="http://www.bercut.com/schema/ns/ScheduledScenarioDefinition"
  version="1.0">

  <xsd:import schemaLocation="BA_EM_CommonDefinitions.xsd"
    namespace="http://bercut.com/em/schema/CommonDefinitions"/>
  <xsd:import schemaLocation="NS_ScheduledScenarioDefinition.xsd"
    namespace="http://www.bercut.com/schema/ns/ScheduledScenarioDefinition"/>

  <!-- ключевой element, таких element (с anyType) может быть несколько в одной
xsd/jar -->
  <xsd:element name="promoCodeRequest" cd:anyType="request">
    <xsd:complexType>
      <xsd:sequence>
        <!-- обязательные параметры -->
        <xsd:element ref="ssdef:targetDate" minOccurs="1"/>
        <xsd:element ref="ssdef:correlationKey" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

<!-- пользовательские параметры -->
    <xsd:element name="activatePromoCodeMsisdn" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>{MSISDN активатора промокода}
Номера абонента - активатора промо кода</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
    
```

2. Создайте JAR-файл, который содержит XSD-файл, описывающий набор входных параметров. Минимальный набор файлов (без требуемого XSD-файла) содержится в файле AOI NS SchedulerScenario. Вы можете создавать новые файлы на основе AOI NS SchedulerScenario, добавив туда XSD-файл.

i Примечание. Создаваемый JAR-файл должен зависеть от:

```

<component name="AOI NS ScheduledScenario" mainstream="1.0" />
<component name="NS ScheduledScenarioDefinition" mainstream="1.0" use-as="wsdl" />
    
```

Пример содержимого JAR-файла:

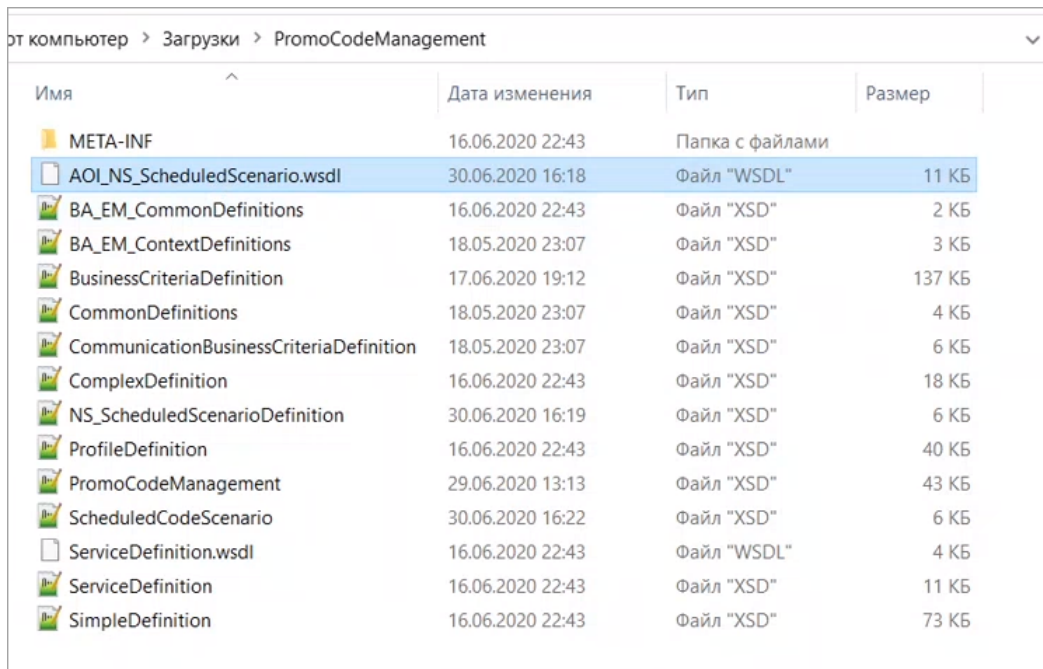


Рис. 34. Пример содержимого JAR-файла

3. Созданный JAR-файл загрузите в NS Back Office на вкладке **Микросервисы**.

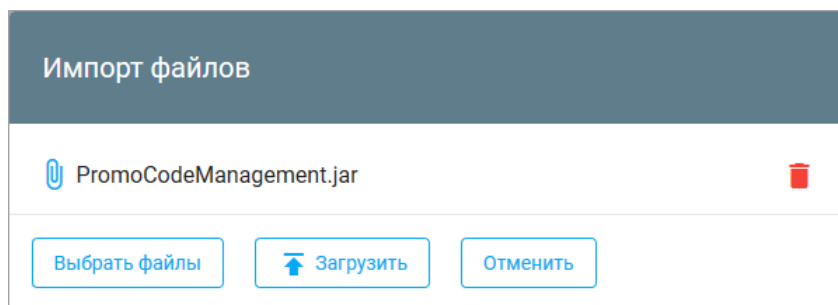


Рис. 35. Импорт файла

Вариант создания микросервиса на примере события *updateLocation* — П 1.3.1.

П 1.2. Настройка отложенного исполнения сценариев

Настройте сценарий-инициатор запуска отложенного исполнения сценария и сценарий, который запускается отложено.

Для работы механизма отложенного исполнения сценариев вы должны настроить два сценария:

- Сценария А — инициатора запуска отложенного сценария. Этот сценарий будет передавать параметры запуска отложенного сценария в Scheduler Subsystem.
- Сценария Б — отложенного сценария, который будет запущен со стороны Scheduler Subsystem в заданное время. При этом сценарий Б может выполнять любые действия, в соответствии с заданными параметрами.

Особенности настройки сценария А

1. Перейдите к сценарию, который будет инициировать запуск отложенного исполнения сценария. На вкладке **Схема сценария** создайте действие формирования отложенного исполнения сценария, либо его замены. Создайте новую операцию и в качестве запускаемого микросервиса выберите один из следующих:

- Создание отложенного исполнения сценария — выберите, чтобы создать новое действие по отложенному исполнению сценария:

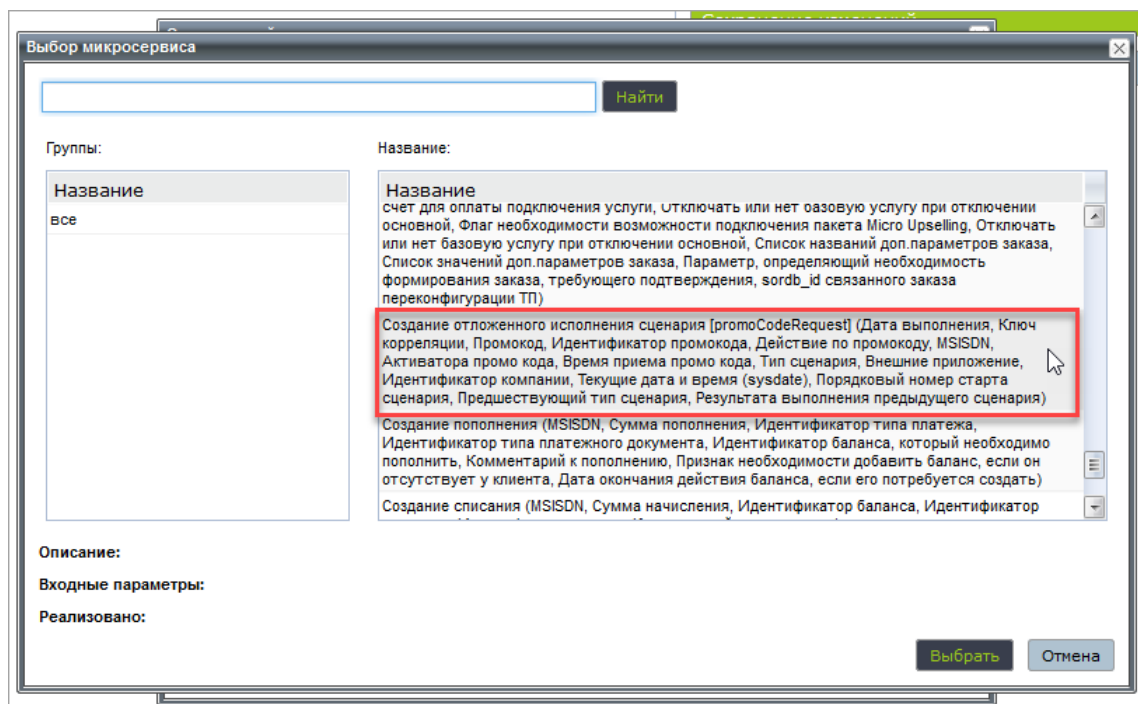


Рис. 36. Создание отложенного исполнения сценария

- Замена отложенного исполнения сценария — выберите, чтобы изменить параметры ранее созданного исполнения сценария:

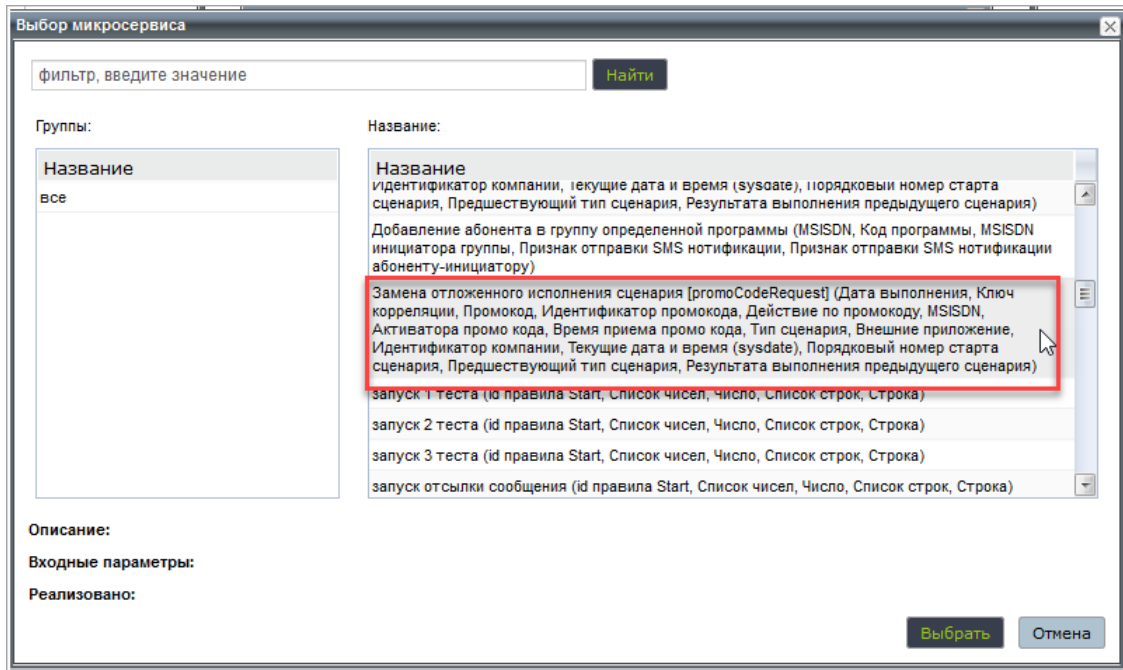


Рис. 37. Замена отложенного исполнения сценария

- Удаление отложенного исполнения сценария — выберите, чтобы удалить ранее созданное отложенное исполнение сценария:

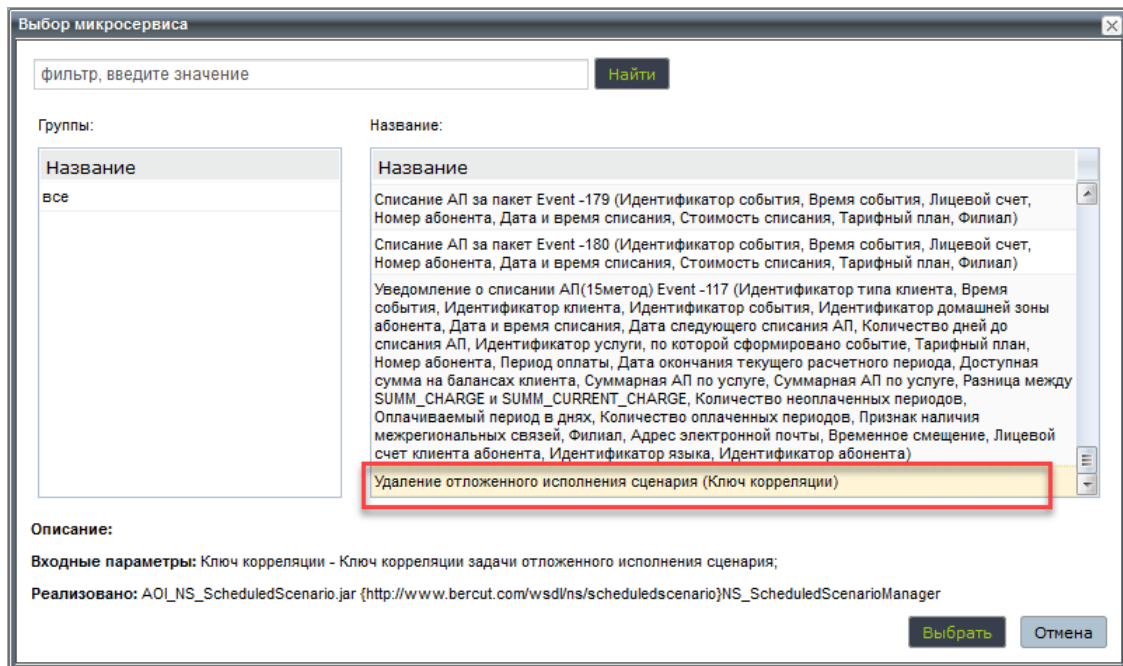


Рис. 38. Удаление отложенного исполнения сценария

Вы также можете проверить наличие уже созданного ранее отложенного исполнения сценария. Для этого вы можете использовать два способа:

- 1.1. С использованием критериев. На вкладке **Схема сценария** нажмите **Доступные критерии** и добавьте новый критерий. На странице **Выбор критерия** выберите один из критериев:

- Дата выполнения — дата выполнения отложенного исполнения сценария.
- Признак наличия — признак существования задачи отложенного исполнения сценария.

- Идентификатор задачи — идентификатор задачи в системе Scheduler Subsystem.

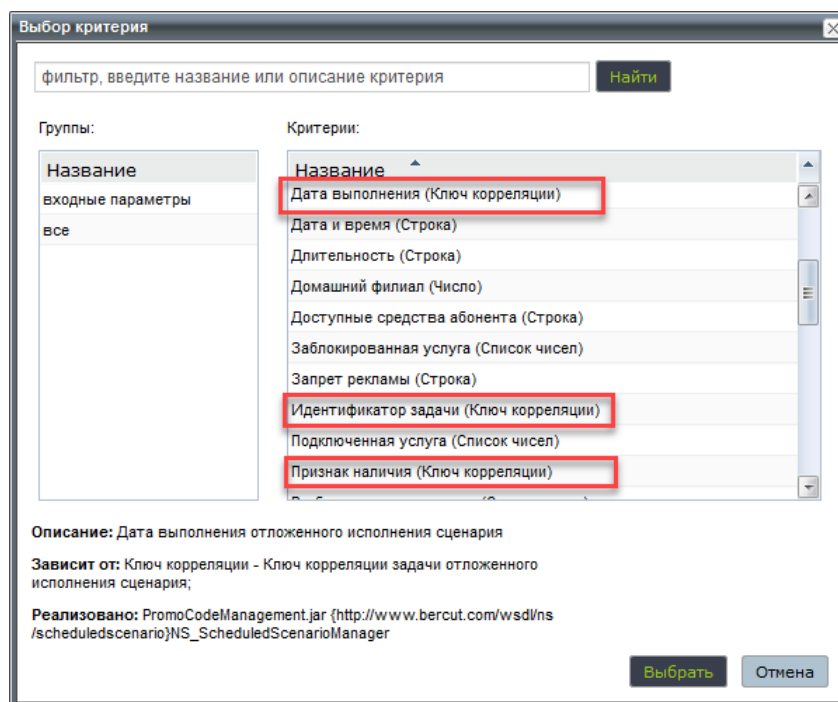


Рис. 39. Выбор критерия для проверки наличия отложенного исполнения сценария

- 1.2. С использованием макроподстановок. На вкладке **Схема сценария** нажмите **Доступные макроподстановки**, выберите критерий и добавьте его в макроподстановку:

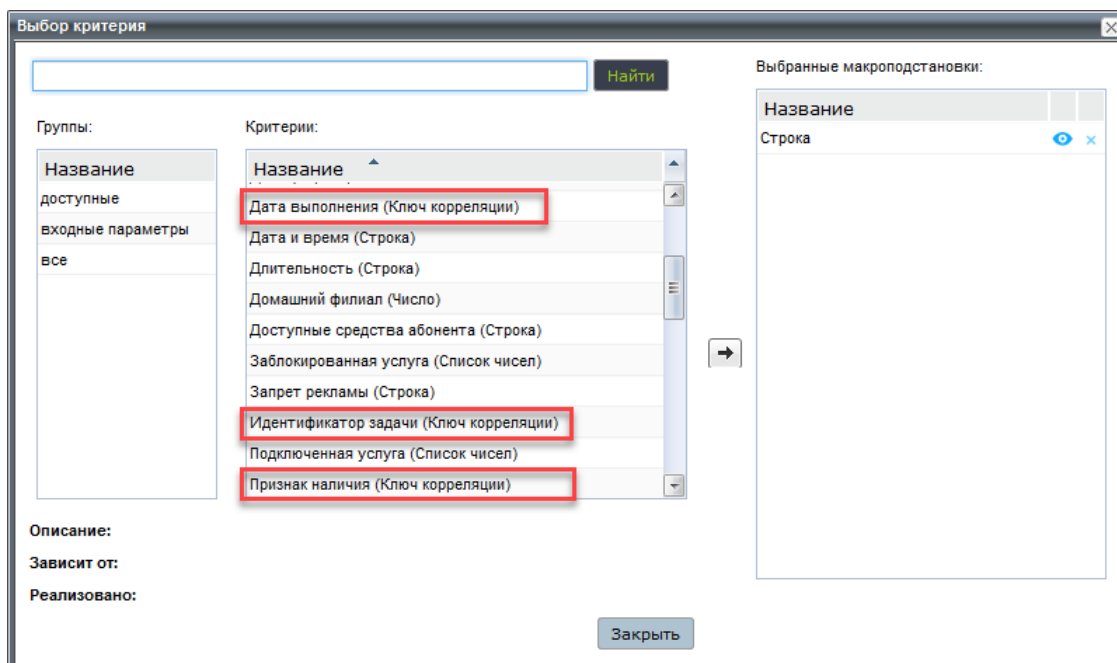


Рис. 40. Выбор макроподстановки для проверки наличия отложенного исполнения сценария

2. Добавьте необходимые критерии, создайте правила.

В таблице правил для каждого правила в действии заполните параметр **ключ корреляции**, который должен быть уникальным. При заполнении ключа корреляции вы можете использовать макроподстановки или задать значение вручную.

Особенности настройки сценария Б

В NS Back Office для продукта создайте сценарий, который будет запущен в определенное время. В качестве источника события укажите «Выполнение отложенного исполнения сценария [xXXRequest]», где xXXRequest — имя, заданное в XSD-файле. Например `<xsd:element name="xXXRequest" cd:anyType="request">`.

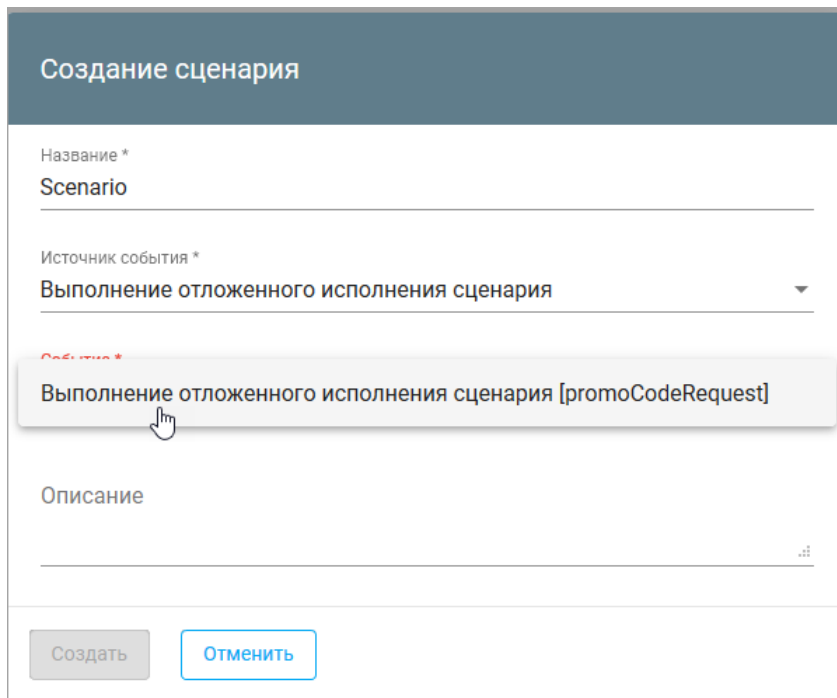


Рис. 41. Создание сценария

Подробнее о работе в NS BackOffice — [5].

Вариант настройки сценариев на примере события *updateLocation*:

- сценария А — [П 1.3.2](#);
- сценария Б — [П 1.3.3](#).

П 1.3. Вариант настройки отложенного исполнения сценариев

Вариант настройки отложенного исполнения сценария на примере события updateLocation.

Для работы механизма отложенного исполнения сценариев вы должны настроить два сценария:

- Сценария А — инициатора запуска отложенного сценария. Этот сценарий будет передавать параметры запуска отложенного сценария в Scheduler Subsystem.
- Сценария Б — отложенного сценария, который будет запущен со стороны Scheduler Subsystem в заданное время. При этом сценарий Б может выполнять любые действия, в соответствии с заданными параметрами.

Механизм работы с отложенным сценарием описан на примере события *updateLocation*. В зависимости от MSISDN через указанное время будет выполняться свой отложенный сценарий для каждого MSISDN.

Для создания отложенного исполнения сценария:

1. Создайте файл микросервиса — П 1.3.1.
2. Настройте сценарий А — П 1.3.2.
3. Настройте сценарий Б — П 1.3.3.

П 1.3.1. Создание микросервиса

Создайте микросервис для отложенного исполнения сценариев, загрузите его в NS Back Office.

Порядок создания файла микросервиса:

1. Создайте XSD-файл `ScheduledCodeScenario.xsd`.

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.bercut.com/wsd1/ms/ScheduledCodeScenario"
  xmlns:tns="http://www.bercut.com/wsd1/ms/ScheduledCodeScenario"
  xmlns:bcddef="http://www.bercut.com/schema/BusinessCriteriaDefinition"
  xmlns:srvdef="http://www.bercut.com/spec/schema/ServiceDefinition"
  xmlns:cd="http://bercut.com/em/schema/CommonDefinitions"
  xmlns:ssdef="http://www.bercut.com/schema/ns/ScheduledScenarioDefinition"
  version="1.0">
  <xsd:import schemaLocation="BA_EM_CommonDefinitions.xsd" namespace="http://bercut.com/em/schema/CommonDefinitions"/>
  <xsd:import namespace="http://www.bercut.com/spec/schema/ServiceDefinition" schemaLocation="ServiceDefinition.xsd"/>
  <xsd:import namespace="http://www.bercut.com/schema/BusinessCriteriaDefinition" schemaLocation="BusinessCriteriaDefinition.xsd"/>
  <xsd:import namespace="http://www.bercut.com/specs/schemas/common-definitions" schemaLocation="CommonDefinitions.xsd"/>
  <xsd:import schemaLocation="NS_ScheduledScenarioDefinition.xsd" namespace="http://www.bercut.com/schema/ns/ScheduledScenarioDefinition"/>
  <xsd:element name="uLRequest" cd:anyType="request">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ssdef:targetDate" minOccurs="1"/>
        <xsd:element ref="ssdef:correlationKey" minOccurs="0"/>
        <xsd:element ref="bcddef:msisdn"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

Рис. 42. Вариант создания файла `ScheduledCodeScenario.xsd`

Проверьте, что файл соответствует условиям:

- 1.1. В нем присутствует обязательный импорт

```
<xsd:import schemaLocation="NS_ScheduledScenarioDefinition.xsd"
namespace="http://www.bercut.com/schema/ns/ScheduledScenarioDefinition"/>
```

- 1.2. Задан обязательный атрибут:

```
<xsd:element name="uLRequest" cd:anyType="request">
```

! **Примечание.** Элемент `name="uLRequest"` задает имя, которое будет добавлено к названию события в NS BackOffice.

- 1.3. Заданы обязательные параметры:

```
<xsd:element ref="ssdef:targetDate" minOccurs="1"/>
<xsd:element ref="ssdef:correlationKey" minOccurs="0"/>
```

! **Внимание!** Эти параметры должны присутствовать всегда вне зависимости от события для которого создается.

Так как в рассматриваемом примере достаточно только параметра MSISDN указан только он: `<xsd:element ref="bcddef:msisdn"/>`. Можно задать любые пользовательские параметры.

При каждом дальнейшем изменении этого файла необходимо менять версию. Не допускается менять порядок следования, имена и значения параметров.

При добавлении каждый новый параметр добавляется в конец списка пользовательских параметров.

2. Добавьте созданный файл в `AOI_NS_ScheduledScenario.jar`. При первом создании имя `AOI_NS_ScheduledScenario` можно поменять на другое. В данном случае это `SchedWSMS.jar`.
3. Загрузите JAR-файл в NS Back Office на вкладке **Микросервисы**.

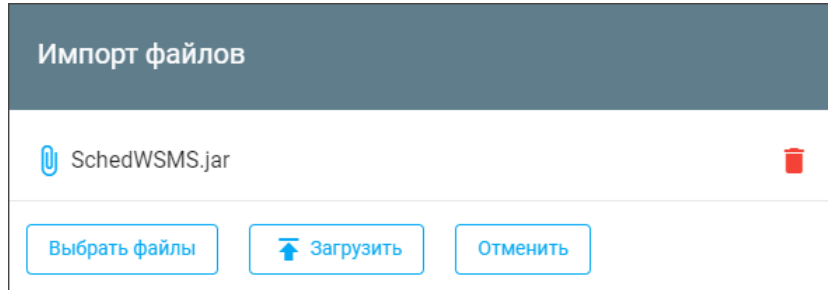


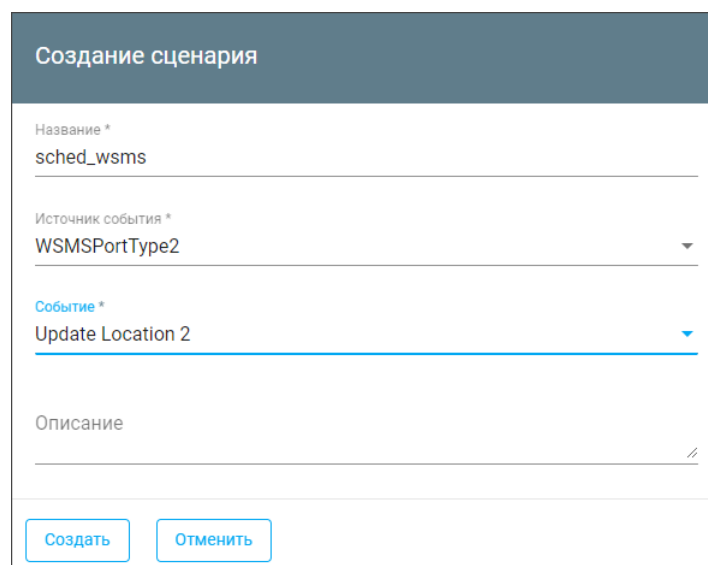
Рис. 43. Вариант настройки импорта файлов

П 1.3.2. Настройка сценария А

Создайте сценарий, настройте схему сценария и таблицу правил.

Создайте и настройте сценарий-инициатор запуска отложенного сценария (сценарий А):

1. Создайте сценарий на *странице сценариев продукта* вкладки **Продукты**.
2. В открывшемся окне **Создание сценария** задайте:
 - **Название** — название нового сценария.
 - **Источник события** — выберите из раскрывающегося списка интерфейс внешней системы, которая является источником события.
 - **Событие** — выберите из раскрывающегося списка операцию, которая является событием-инициатором выполнения сценария. Выберите входящее событие `updateLocation`.
 - **Описание** — введите описание сценария.



3. На странице меню **Схема сценария** настройте сценарий — П 1.3.2.1.
4. На странице меню **Таблица правил** настройте действия таблицы правил — П 1.3.2.2.

П 1.3.2.1. Настройка схемы сценария

Задайте действия, которые будут определять реакцию системы Notification System на событие. При необходимости определите набор критериев для работы с правилами.

Страница меню **Схема сценария** вкладки **Продукт** состоит из двух областей: *Действия* и *Правила*.

1. В области **Действие** нажмите **Добавить действие**. В открывшемся окне **Создание действия** создайте действие формирования отложенного исполнения сценария, либо его замены. Создайте операцию.

Редактирование операции

Название: create

Микросервис: Создание отложенного исполнения сценария [uLRequest] [Выбрать](#)

Описание:

Входные параметры:

Название	Обязательный	Доступен в Excel	Значение
Дата выполнения	да	<input checked="" type="checkbox"/>	~S{datetime_chang... datetime,"+1m"}}
Ключ корреляции	нет	<input checked="" type="checkbox"/>	--
MSISDN	да	<input type="checkbox"/>	БК: MSISDN

Сохранить Отмена

Рис. 45. Вариант создания операции

2. Нажмите **Выбрать** справа от поля **Микросервис** и в окне **Выбор микросервиса** со списком доступных микросервисов выберите микросервис для создания отложенного исполнения сценария (сценария Б).

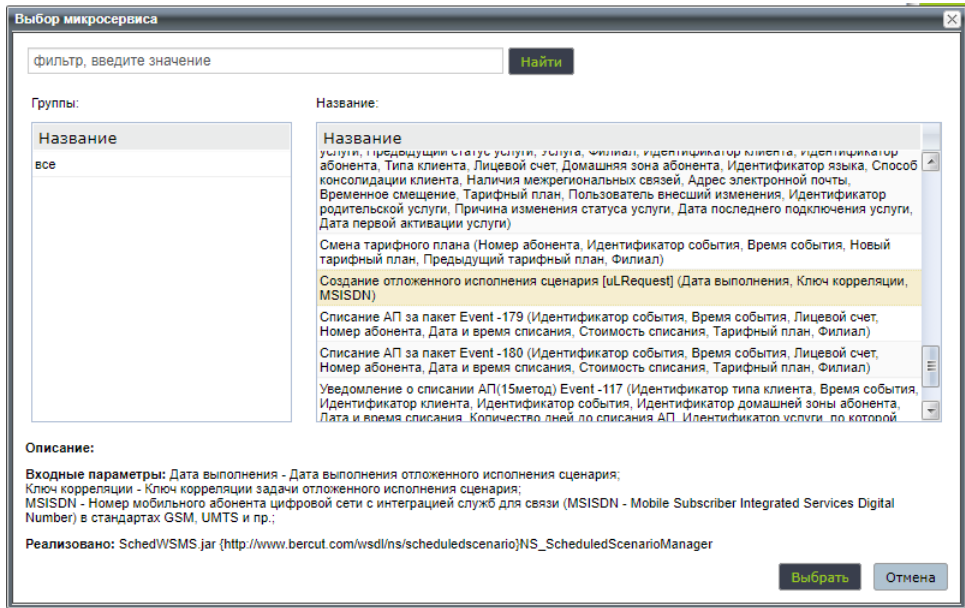


Рис. 46. Вариант создания действия

3. Проверьте, что в операции содержатся заданные в XSD-файле входные параметры:

- **Дата выполнения** – дата, в которую должен будет выполнен отложенный сценарий. В данном случае согласно `~${datetime_change(.now? datetime, "+1m")}` отложенный сценарий будет выполнен через минуту. Указанный именно здесь этот параметр будет автоматически присваиваться всем новым создаваемым далее правилам. В таблице правил его можно поменять для любого правила. Порядок работы с датой и временем описан в [5].
- **Ключ корреляции** – уникальный идентификатор.
- **MSISDN**.

4. Настройте критерии. В области **Правила** нажмите **Доступные критерии**.

- В открывшемся окне **Настройка критериев** нажмите **Добавить**. Откроется окно **Добавление критерия**.
- В поле **Критерий** нажмите **Выбрать** справа от поля ввода. Из списка доступных критериев выберите критерии, которые будут соответствовать бизнес-процессам оператора связи. В данном случае MSISDN:

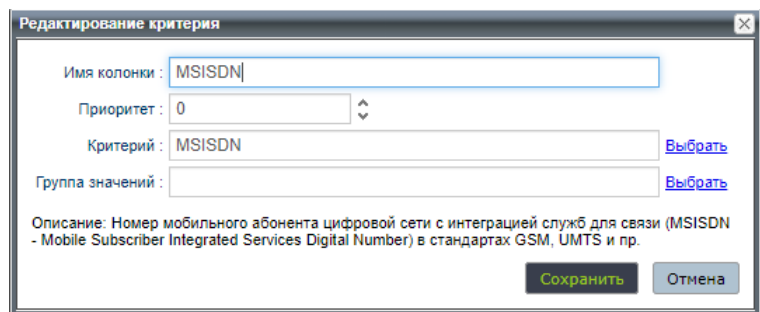


Рис. 47. Вариант добавления критерия

5. Сохраните изменения.

П 1.3.2.2. Настройка таблицы правил

Заполните таблицу правил.

Перейдите на вкладку **Продукт**, меню **Таблица правил** и заполните таблицу правил.

! Внимание! Пользователь настраивает работу объектов системы в соответствии с бизнес-процессами оператора связи под его задачи. Набор критериев, для которых выбираются операнды и значения, полностью зависит от настроек схемы сценария, который может отличаться для каждой отдельно взятой задачи.

В данном случае создайте правила, в которых для одного MSISDN будет выполняться одно действие отложенного сценария Б, для другого MSISDN — другое правило отложенного сценария Б.

1. Нажмите **Редактировать таблицу**.
2. Создайте правило, вызвав контекстное меню левой кнопкой мыши в области первого столбца.

		MSISDN		
№	Описание	Оператор	Значение	
1		равен	79014566663	...
2		равен	79019965208	...

Рис. 48. Вариант настройки таблицы правил

3. Укажите:

- 3.1. Описание.
- 3.2. Операторы и значения для критериев, созданные в схеме сценария — [П 1.3.2.1](#).
- 3.3. Действие правила. Параметры действия — действия, созданные в схеме сценария.

В данном случае создайте действие «Создание отложенного выполнения сценария» для каждого MSISDN.

Например:

- Для **79014566663**:

Действия

create

Дата выполнения

Ключ корреляции

Result

Рис. 49. Вариант создания действия для MSISDN 79014566663

Для действия настройте параметры:

- дата выполнения — через минуту;

- уникальный идентификатор — ключ корреляции. По ключу корреляции будет происходить связь с отложенным сценарием Б. В зависимости от обрабатываемого события и бизнес требований предпочтительнее указывать ключ корреляции как связку уникальных параметров. При заполнении значения вы также можете использовать макроподстановки. Например для WSMS можно использовать связку MSISDN и VLR.
- Для 79019965208:

Параметр	Значение	Состояние
Дата выполнения	<code>\${datetime_change(.now?datetime,\"+1m\")}</code>	Включено
Ключ корреляции	22222_22222	Выключено
Result		Выключено

Рис. 50. Вариант создания действия для MSISDN 79019965208

Для действия настройте параметры: своя дата выполнения и ключ корреляции.

4. Сохраните изменения и примените сценарий на тест.

П 1.3.3. Настройка сценария Б

Создайте сценарий, настройте схему сценария и таблицу правил.

Создайте и настройте отложенный сценарий (сценарий Б):

1. Создайте сценарий на *странице сценариев продукта* вкладки **Продукты**.
2. В открывшемся окне **Создание сценария** задайте:
 - **Название** — название нового сценария.
 - **Источник события** — выберите из раскрывающегося списка *Выполнение отложенного исполнения сценария*.
 - **Событие** — выберите из раскрывающегося списка операцию, которая является событием-инициатором выполнения сценария. Выберите событие *Выполнение отложенного исполнения сценария [xXXRequest]*, где xXXRequest — имя, заданное в XSD-файле. Например, `<xsd:element name="xXXRequest" cd:anyType="request">`.
 - **Описание** — введите описание сценария.

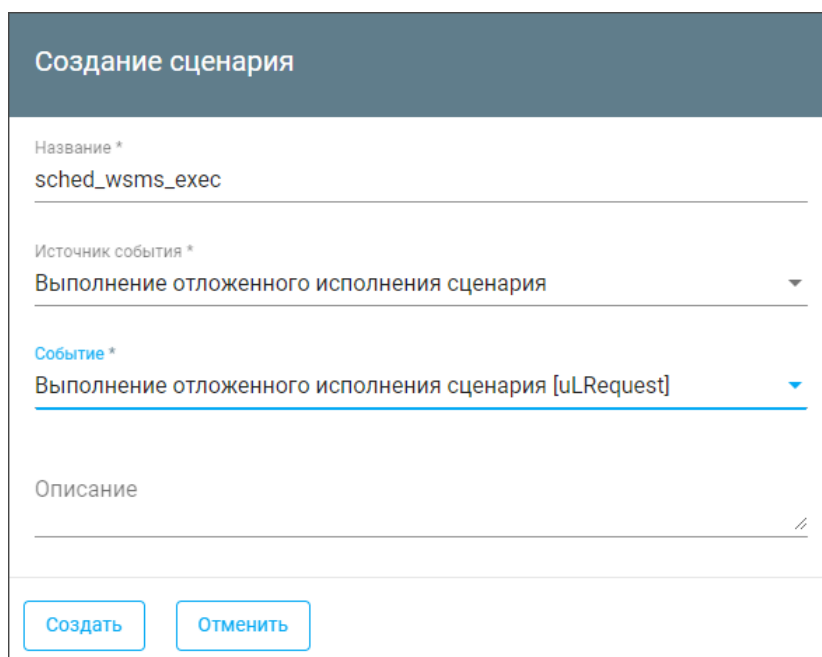


Рис. 51. Вариант настройки сценария Б

3. На странице меню **Схема сценария** настройте сценарий — [П 1.3.3.1](#).
4. На странице меню **Таблица правил** настройте действия таблицы правил — [П 1.3.3.2](#).

П 1.3.3.1. Настройка схемы сценария

Задайте действия, которые будут определять реакцию системы Notification System на событие. При необходимости определите набор критериев для работы с правилами.

Страница меню **Схема сценария** вкладки **Продукт** состоит из двух областей: *Действия* и *Правила*.

1. В области **Действие** нажмите **Добавить действие**. В открывшемся окне **Создание действия**:
 - Задайте название действия в поле «Имя».
 - При необходимости убедитесь в наличии выходных параметров в нижней части окна.

Поставьте флаг «Доступен в Excel», если необходимо, чтобы эти параметры отображались в файле Excel.

Для сценария Б вы можете создавать любые действия и операции которые реализованы установленными в системе микросервисами. В данном примере описано создание двух действий «Отправка SMS», каждое с разным текстом.

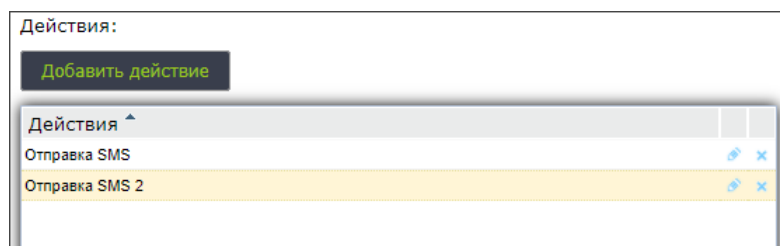


Рис. 52. Вариант создания действий

2. Настройте критерии. В области **Правила** нажмите **Доступные критерии**.

- В открывшемся окне **Настройка критериев** нажмите **Добавить**. Откроется окно **Добавление критерия**.
- В поле **Критерий** нажмите **Выбрать** справа от поля ввода. Из списка доступных критериев выберите критерии, которые будут соответствовать бизнес-процессам оператора связи. В данном случае **Ключ корреляции**:

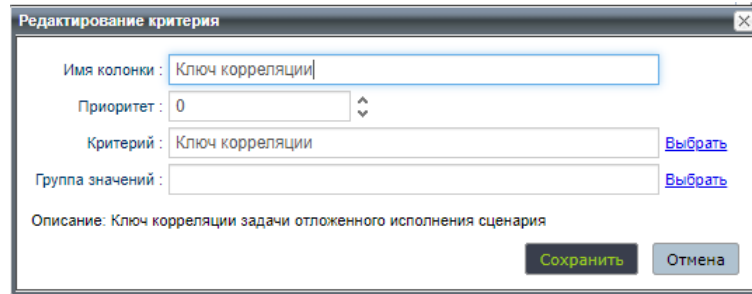


Рис. 53. Вариант добавления критерия

3. Сохраните изменения.

П 1.3.3.2. Настройка таблицы правил

Заполните таблицу правил.

Перейдите на вкладку **Продукт**, меню **Таблица правил** и заполните таблицу правил.

! Внимание! Пользователь настраивает работу объектов системы в соответствии с бизнес-процессами оператора связи под его задачи. Набор критериев, для которых выбираются операнды и значения, полностью зависит от настроек схемы сценария, который может отличаться для каждой отдельно взятой задачи.

В данном случае добавьте значения ключа корреляции, создайте действия с текстом отправляемых SMS-сообщений:

1. Нажмите **Редактировать таблицу**.
2. Создайте правило, вызвав контекстное меню левой кнопкой мыши в области первого столбца.

		Ключ корреляции		
№	Описание	Оператор	Значение	
☰ 1		равен	11111_11111	...
☰ 2		равен	22222_22222	...

Рис. 54. Вариант настройки таблицы правил

3. Укажите:

- 3.1. Описание.
- 3.2. Операторы и значения для критериев, созданные в схеме сценария — П 1.3.3.1.
- 3.3. Действие правила. Параметры действия — действия, созданные в схеме сценария.

В данном случае создайте действие «Отправка SMS» в зависимости от ключа корреляции:

Например:

The screenshot shows a configuration window titled "Действия" (Actions). At the top, there is a dropdown menu labeled "Выбор действия" (Action selection) with "Отправка SMS" (SMS Sending) selected. Below this, the configuration for "Отправка SMS" is shown with several fields and toggle switches:

- Адрес отправителя** (Sender address): "Tele2" with a "Vx" icon and a toggle switch.
- Текст сообщения** (Message text): "Привет111111111111111111" with a "Vx" icon and a toggle switch.
- Начало интервала доставки** (Start of delivery interval): empty field with a "Vx" icon and a toggle switch.
- Окончание интервала доставки** (End of delivery interval): empty field with a "Vx" icon and a toggle switch.
- Смещение временной зоны** (Time zone offset): empty field with a "Vx" icon and a toggle switch.
- Сохранить информацию** (Save information): checked checkbox with a "Vx" icon and a toggle switch.

At the bottom left of the window is an "OK" button.

Рис. 55. Вариант настройки действия для отправителя с MSISDN 79014566663

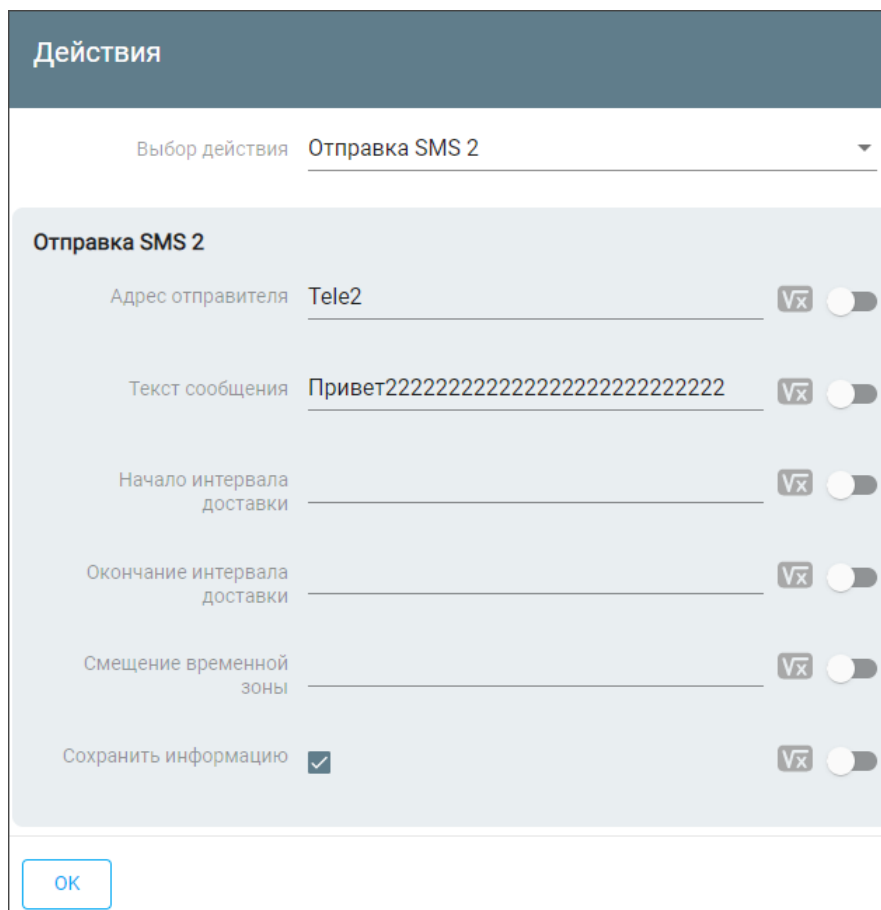


Рис. 56. Вариант настройки действия для отправителя с MSISDN 79019965208

4. Сохраните изменения и примените сценарий на тест.

В описанном примере в случае поступления события **updateLocation** для абонента с MSISDN 79014566663 через 1 минуту ему придет SMS-сообщение с текстом Привет111111111111111111.

В случае поступления события **updateLocation** для абонента с MSISDN 79019965208 через 1 минуту ему придет SMS-сообщение с текстом: Привет222222222222222222222222222222.

Приложение 2. Создание событий

Вы можете создавать события нескольких типов для Notification System:

1. *Пользовательские события* — события, которые будут обрабатываться сценариями, настроенными в Notification System.
2. *События* для отложенных сценариев — события, которые используются для настройки запуска отложенного исполнения сценариев.
3. *События с поддержкой транзакционности*, если вам необходимо выполнять несколько операций в рамках транзакции
4. *События* для интеграции с Apache Kafka:
 - для *SA NS Kafka Consumer* — события для обработки сообщений, получаемых из Apache Kafka.
 - для *SA NS Kafka Producer* — события для обработки сообщений перед отправкой в Apache Kafka.
5. *События* для передачи сообщений по HTTP.

П 2.1. Пользовательское событие

Вы можете создавать и редактировать события, которые будут обрабатываться сценариями, настроенными в Notification System.

После создания файла микросервиса вы можете загрузить его в NS BackOffice и настроить сценарий для его использования.

Общий порядок работы с микросервисами и справочниками:

1. *Создайте* микросервис.
2. *Загрузите* созданный микросервис в систему.
3. Создайте продукт.
4. Настройте *справочники* и *сценарии*.
5. Чтобы проверить действие сформированных правил, используйте пункт меню **Применить на тест**. Черновик сценария загрузится на тестовое окружение.

i Примечание. Подробнее об объектах системы Notification System и компонентах ее архитектуры — [6].

Пользователь настраивает работу объектов системы в соответствии с бизнес-процессами оператора связи под его задачи. Набор критериев, для которых выбираются операнды и значения, полностью зависит от настроек схемы сценария. Набор критериев может отличаться для каждой отдельно взятой задачи.

П 2.1.1. Создание микросервиса

Создайте WSDL-файл интерфейса и сформируйте JAR-файл микросервиса.

Микросервис — бизнес-процесс, предназначенный для получения значений критериев или выполнения бизнес-задачи, или интерфейс. Микросервис представляет из себя JAR-архив, который содержит интерфейсы и библиотеки.

Вы можете сами создавать микросервисы. Для создания микросервиса выполните следующие действия:

1. Создайте интерфейс и укажите в нем события — П 2.1.1.1.

2. Добавьте файл в один каталог с интерфейсом общие WSDL- и XSD-файлы — [П 2.1.1.2.](#)
3. Задайте критерии для входных и выходных параметров — [П 2.1.1.3.](#)
4. При возможности выполните валидацию WSDL-файла интерфейса — [П 2.1.1.4.](#)
5. Сформируйте JAR-файл микросервиса — [П 2.1.1.5.](#)

Вы также можете обновить микросервис — [П 2.1.5.1.](#)

После создания файла микросервиса вы можете *загрузить* его в NS BackOffice и *настроить* сценарий для его использования.

П 2.1.1.1. Создание интерфейса

Интерфейс — это WSDL-файл, в котором описан portType. Используемый portType в качестве операций содержит события.

Вы можете создавать свои собственные интерфейсы, а затем с помощью их формировать микросервисы. Интерфейс создается для каждого типа событий и реализует portType. Может содержать одно или несколько событий (операций) одного типа. Одно событие соответствует одной операции этого porttype.

Входные и выходные параметры операции — это критерии. Каждая операция может содержать неограниченное число критериев.

При создании интерфейсов вы можете использовать критерии, заранее заданные в файле BusinessCriteriaDefinition.xsd, или создавать свои критерии.

Для удобства при создании WSDL-файла интерфейса вы можете использовать готовый [шаблон](#).

П 2.1.1.1.1. Шаблон WSDL-файла

Готовый шаблон для создания интерфейса.

Для создания WSDL-файла пользовательского интерфейса вы можете использовать готовый шаблон:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="AOI_<<NAME>>"
  targetNamespace="http://www.bercut.com/wsdl/<<NAME>>"
  xmlns:tns="http://www.bercut.com/wsdl/<<NAME>>"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
  xmlns:servd="http://www.bercut.com/spec/schema/ServiceDefinition"
  xmlns:bcd="http://www.bercut.com/schema/BusinessCriteriaDefinition"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <types>
    <xsd:schema targetNamespace="http://www.bercut.com/wsdl/<<NAME>>" version="X.Y">
      <xsd:import schemaLocation="ServiceDefinition.xsd"
        namespace="http://www.bercut.com/spec/schema/ServiceDefinition"/>
      <xsd:import schemaLocation="SimpleDefinition.xsd"
        namespace="http://www.bercut.com/spec/schema/SimpleDefinition"/>
      <xsd:import schemaLocation="BusinessCriteriaDefinition.xsd"
        namespace="http://www.bercut.com/schema/BusinessCriteriaDefinition"/>

      <xsd:element name="get<<NAME>>Request">
        <xsd:complexType>
          <xsd:sequence>
            <!-- параметры запроса -->
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="get<<NAME>>Response">
        <xsd:complexType>
          <xsd:sequence>
```

```

        <!-- параметры ответа -->
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
</xsd:schema>
</types>

<message name="BusinessFault">
    <part name="fault" element="servd:businessFault"/>
</message>
<message name="get<<NAME>>Request">
    <part name="parameters" element="tns:get<<NAME>>Request"/>
</message>
<message name="get<<NAME>>Response">
    <part name="parameters" element="tns:get<<NAME>>Response"/>
</message>

<portType name="<<NAME>>PortType">
    <documentation>{<<ИМЯ ИСТОЧНИКА>>}</documentation>
    <operation name="get<<NAME>>">
        <documentation>{<<ИМЯ СОБЫТИЯ>>[ACTION,GETTER]}<<ПОЛНОЕ ОПИСАНИЕ СОБЫТИЯ>></
documentation>
        <input name="get<<NAME>>Request" message="tns:get<<NAME>>Request"/>
        <output name="get<<NAME>>Response" message="tns:get<<NAME>>Response"/>
        <fault name="BusinessFault" message="tns:BusinessFault"/>
    </operation>
</portType>
<binding name="<<NAME>>PortTypeBinding" type="tns:<<NAME>>PortType">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="get<<NAME>>">
        <input name="get<<NAME>>Request">
            <soap:body use="literal" parts="parameters"/>
        </input>
        <output name="get<<NAME>>Response">
            <soap:body use="literal"/>
        </output>
        <fault name="BusinessFault">
            <soap:fault use="literal" name="BusinessFault"/>
        </fault>
    </operation>
</binding>

<service name="<<NAME>>Service">
    <port name="<<NAME>>ServicePort"
        binding="tns:<<NAME>>PortTypeBinding">
        <soap:address location="http://localhost:${HttpDefaultPort}/<<NAME>>Service/
<<NAME>>Port"/>
    </port>
</service>
<plnk:partnerLinkType name="WSMSPlt">
    <plnk:role name="<<NAME>>PortTypeRole" portType="tns:<<NAME>>PortType"/>
</plnk:partnerLinkType>
</definitions>

```

П 2.1.1.1.2. Создание WSDL-файла по шаблону

Вы можете создать WSDL-файл интерфейса по шаблону.

Для создания WSDL-файла интерфейса по шаблону:

1. Создайте каталог, в котором будут храниться файлы интерфейса.
2. Создайте в каталоге новый файл и скопируйте туда *шаблон*.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <definitions name="AOI <<NAME>>"
3    targetNamespace="http://www.bercut.com/wsd/ <<NAME>>"
4    xmlns:tns="http://www.bercut.com/wsd/ <<NAME>>"
5    xmlns="http://schemas.xmlsoap.org/wsd/"
6    xmlns:wsdl="http://schemas.xmlsoap.org/wsd/"
7    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
8    xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
9    xmlns:servd="http://www.bercut.com/spec/schema/ServiceDefinition"
10   xmlns:bcd="http://www.bercut.com/schema/BusinessCriteriaDefinition"
11   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
12 <types>
13   <xsd:schema targetNamespace="http://www.bercut.com/wsd/ <<NAME>>" version="1.1">
14     <xsd:import schemaLocation="ServiceDefinition.xsd" namespace="http://www.bercut.com/spec/schema/ServiceDefinition"/>
15     <xsd:import schemaLocation="SimpleDefinition.xsd" namespace="http://www.bercut.com/spec/schema/SimpleDefinition"/>
16     <xsd:import schemaLocation="BusinessCriteriaDefinition.xsd" namespace="http://www.bercut.com/schema/BusinessCriteriaDefinition"/>
17
18     <xsd:element name="get<<NAME>>Request">
19       <xsd:complexType>
20         <xsd:sequence>
21           <!-- параметры запроса -->
22         </xsd:sequence>
23       </xsd:complexType>
24     </xsd:element>

```

Рис. 57. Создание WSDL-файла

3. Придумайте для интерфейса название <<NAME>>, которое будет идентифицировать событие интерфейса.

Примечание. Допустимо использовать буквы английского алфавита, цифры и знак нижнего подчеркивания.

Пример: WsmsTrplGroup

4. Замените в содержимом файла имя <<NAME>> на придуманное.

5. Укажите номер версии интерфейса в блоке:

```
<xsd:schema targetNamespace="http://www.bercut.com/wsd/ <<NAME>>" version="X.Y">
```

X.Y — номер версии.

6. Вы также можете настроить *критерии* на этом этапе.

7. Задайте имя источника и имя события в блоке:

```

<portType name="<<NAME>>PortType">
  <documentation>{<<ИМЯ ИСТОЧНИКА>>}</documentation>
  <operation name="get<<NAME>>">
    <documentation>{<<ИМЯ СОБЫТИЯ>> [ACTION, GETTER]}<<ПОЛНОЕ ОПИСАНИЕ СОБЫТИЯ>></documentation>

```

- Задайте описание portType. В теге <documentation> для portType задайте имя источника — назначение portType.

Примечание. Содержимое этого тега будет отображаться в NS BackOffice [5] в окне создания сценария в параметре *Источник события*.

- Задайте описание события. В блоке <operation/> теге <documentation> для каждого события (операции) задайте:
 - имя события — краткое описание события;

Примечание. Содержимое этого тега будет отображаться в NS BackOffice [5] в окне создания сценария в параметре *Событие*.

- полное описание события — подробное описание события.

Пример:

```

<portType name="WsmsTrplGroupPortType">
  <documentation>{Справочная информация для Welcome SMS}</documentation>
  <operation name="getWsmsTrplGroup">
    <documentation>{Получить информацию по группе ТП[ACTION, GETTER]}

```

Получить информацию по группе тарифных планов</documentation>

Пример отображения имени источника и имени события в NS BackOffice:

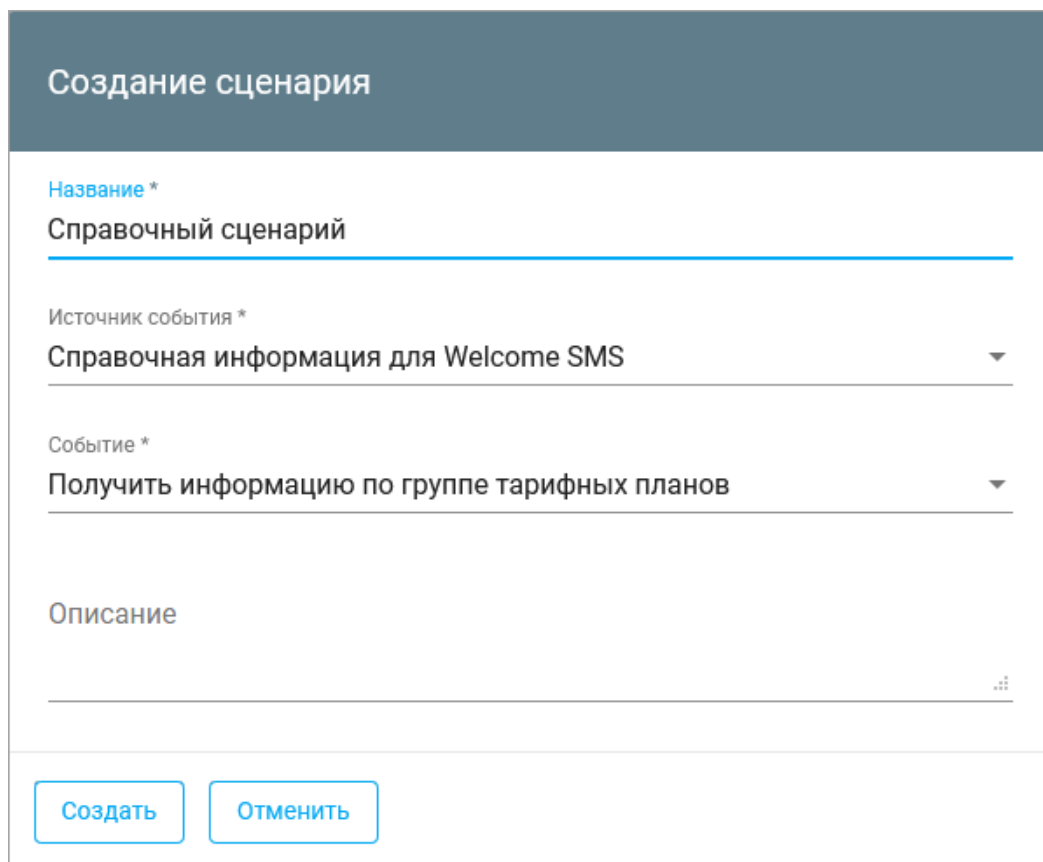


Рис. 58. Пример отображения в NS BackOffice:

8. Сохраните файл интерфейса AOI_<<NAME>>.wsdl. Например, AOI_WsmsTrplGroup.wsdl.

П 2.1.1.2. Добавление WSDL- и XSD-файлов

Добавьте в каталог с WSDL-файлом файлы.

Порядок настройки:

1. В каталог с WSDL-файлом сохраните файлы:

- SimpleDefinition.xsd
- ServiceDefinition.xsd
- ServiceDefinition.wsdl
- ProfileDefinition.xsd
- ComplexDefinition.xsd
- BusinessCriteriaDefinition.xsd
- \META-INF\MANIFEST.MF

Файлы можно взять из любого существующего микросервиса.

Эти файлы содержат список заданных ранее критериев, которые вы сможете использовать при настройке входных и выходных параметров ([П 2.1.1.3](#)).

Пример содержимого каталога:

Имя	Дата изменения	Тип	Размер
META-INF	07.08.2019 23:07	Папка с файлами	
AOI_WsmtsTrplGroup.wsdl	22.06.2020 13:03	Файл "WSDL"	5 КБ
BusinessCriteriaDefinition.xsd	07.08.2019 23:07	Файл "XSD"	86 КБ
ComplexDefinition.xsd	07.08.2019 23:07	Файл "XSD"	18 КБ
ProfileDefinition.xsd	07.08.2019 23:07	Файл "XSD"	37 КБ
ServiceDefinition.wsdl	07.08.2019 23:07	Файл "WSDL"	4 КБ
ServiceDefinition.xsd	07.08.2019 23:07	Файл "XSD"	11 КБ
SimpleDefinition.xsd	07.08.2019 23:07	Файл "XSD"	73 КБ

Рис. 59. Пример содержимого каталога

2. Для файла MANIFEST.MF проследите, чтобы номер версии, указанной в блоке Implementation-Version: X.Y, был больше или равен версии существующего сервиса.

Пример содержимого файла MANIFEST.MF:

```
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.8.3
Created-By: 1.6.0_45-b06 (Sun Microsystems Inc.)
Implementation-Title: AOI_WsmtsTrplGroup
Specification-Title: AOI_WsmtsTrplGroup
Specification-Version: 1.0
Specification-Vendor: Bercut Ltd.
Implementation-Version: 1.0.0.25
Implementation-Vendor: Bercut Ltd.
```

П 2.1.1.3. Добавление входных и выходных параметров

Откройте созданный файл и добавьте входные и выходные параметры.

Откройте созданный WSDL-файл и добавьте входные и выходные параметры. В качестве параметров используются критерии.

Вы можете добавлять:

- ссылки на существующие критерии, заданные в файле BusinessCriteriaDefinition.xsd (П 2.1.1.3.1);

і Примечание. Предпочтительно использовать этот способ.

- собственные критерии (П 2.1.1.3.2).

Количество критериев, которые вы можете задать, неограничено.

Критерии задаются в блоках:

- для входных параметров — в блоке:

```
<xsd:element name="get<<NAME>>Request">
  <xsd:complexType>
    <xsd:sequence>
      <!-- параметры запроса -->
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

- для выходных параметров — в блоке:

```
<xsd:element name="get<<NAME>>Response">
  <xsd:complexType>
    <xsd:sequence>
```

```
        <!-- параметры ответа -->
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Пример настроенных критериев:

```
<xsd:element name="getWsmsTrplGroupRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="bcd:enabledRateId"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="getWsmsTrplGroupResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="TrplGroupRussia" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation>{Тарифная группа Россия}Тарифная группа Россия</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

П 2.1.1.3.1. Добавление ссылки на критерий

В WSDL-файле добавьте ссылки на критерии, заданные в файле *BusinessCriteriaDefinition.xsd*.

Для добавления ссылки на критерий:

1. Откройте для редактирования *созданный* WSDL-файл.
2. Найдите *блоки* для входных и выходных параметров.
3. После тега `<xsd:sequence>` добавьте ссылку на критерий из файла *BusinessCriteriaDefinition.xsd* в формате:

```
<xsd:element ref="<пространство имен>:<имя критерия>"/>
```

! Внимание! Запрещено добавлять или изменять критерии в файле *BusinessCriteriaDefinition.xsd*. Это может привести к ошибкам применения сценариев.

4. Добавьте ссылки на критерии для входных и выходных параметров.
5. Все критерии, которые добавляются при создании первой версии интерфейса, являются обязательными. У этих критериев отсутствует параметр `minOccurs="0"`.

! Внимание! При создании *последующих* версий справочника с каждым новым параметром обязательно должен быть соотнесен параметр `minOccurs="0"`.

При создании следующих версий

Пример:

```
<xsd:element name="getWsmsTrplGroupRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="bcd:enabledRateId"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

П 2.1.1.3.2. Добавление критерия

Добавьте требуемые критерии в WSDL-файл.

Если в файле BusinessCriteriaDefinition.xsd не нашлось подходящих критериев, вы можете задать собственный критерий. Для этого:

1. Откройте для редактирования *созданный* WSDL-файл.
2. Найдите *блоки* для входных и выходных параметров.
3. Для добавления критерия после тега <xsd:sequence> добавьте блок вида:

```
<xsd:element name="<<ID КРИТЕРИЯ>>" type="<<ТИП>>">
  <xsd:annotation>
    <xsd:documentation>{<<НАЗВАНИЕ КРИТЕРИЯ>><<ОПИСАНИЕ КРИТЕРИЯ>>}</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

i Примечание. Вы можете добавлять только линейные критерии, а не комплексные.

! Внимание! При создании *последующих* версий справочника с каждым новым параметром обязательно должен быть соотнесен параметр `minOccurs="0"`.

4. Задайте параметры:

- <<ID КРИТЕРИЯ>> — идентификатор критерия. Задается один раз и не может быть изменен впоследствии.
- <<ТИП>> — тип параметра в схеме W3C [23].
- <<НАЗВАНИЕ КРИТЕРИЯ>> — краткое название критерия, которое видит пользователь в NS BackOffice [5]. Это описание можно поменять в NS BackOffice. Пример: *Тарифная группа РФ*.
- <<ОПИСАНИЕ КРИТЕРИЯ>> — развернутое описание критерия, которое видит пользователь в NS BackOffice [5]. Это описание можно поменять в NS BackOffice. Пример: *Тарифная группа Россия*.

Пример:

```
<xsd:element name="getWsmsTrplGroupResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="TrplGroupRussia" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation>{Тарифная группа Россия}Тарифная группа Россия</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

П 2.1.1.4. Валидация XML-файла

По возможности выполните валидацию XML-файла для созданного WSDL-файла в сторонних системах. Это поможет избежать ошибок синтаксиса в XML-файле.

П 2.1.1.5. Формирование JAR-файла

Сформируйте JAR-файл микросервиса.

Микросервис представляет из себя JAR-архив. Для формирования архива:

1. Перейдите в каталог, в котором находится созданный WSDL-файл и *остальные WSDL и XSD-файлы*.
2. Создайте ZIP-архив.

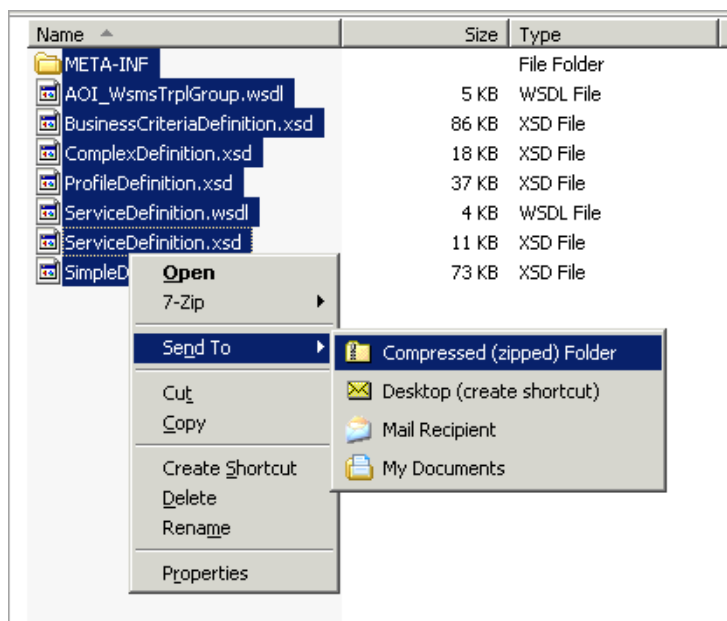


Рис. 60. Пример создания архива

3. В сформированном архиве замените расширение `.zip` на `.jar`.

П 2.1.2. Загрузка микросервисов в систему

Загрузите микросервис в NS BackOffice.

Микросервисы используются в сценариях для получения значения критериев или выполнения действия.

Для загрузки микросервисов:

1. Войдите в BackOffice под учетной записью пользователя, который обладает правом добавления микросервисов.

Примечание. Описание соответствующей ролевой модели — [5].

2. Перейдите на страницу **Управление > Микросервисы** и нажмите [+ Добавить](#).
3. В открывшемся окне **Импорт файлов** нажмите [Выбрать файлы](#) и выберите созданный JAR-файл на вашем компьютере.

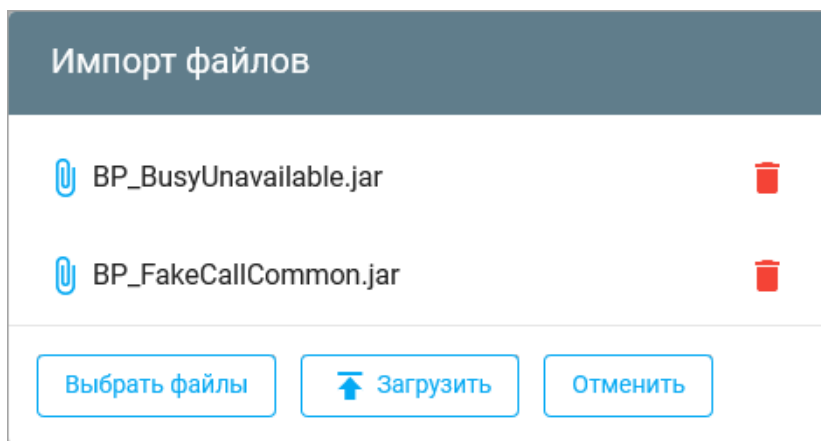


Рис. 61. Вариант окна импорта для загрузки микросервиса

4. Нажмите **Загрузить**. При успешной загрузке JAR-файлов информация о микросервисе отобразится на странице «Микросервисы» — в списке появятся новые критерии или операции.
5. Проверьте, что вы загрузили верную версию справочника. Версия, указанная для микросервиса соответствует указанной в файле `MANIFEST.MF` ([П 2.1.1.2](#)).

! Внимание! Если в файле `MANIFEST.MF` вы не проконтролировали номер версии, то это приведет к ошибке применения сценария и ошибкам при работе со сценарием. Такой сценарий можно будет только удалить.

Подробное описание загрузки приведено в руководстве пользователя — [\[5\]](#).

П 2.1.3. Просмотр и заполнение справочников

При необходимости настройте справочник для каждого добавленного микросервиса в `NS BackOffice`.

Перед настройкой справочников ознакомьтесь с определениями следующих объектов системы:

- **Критерий** — параметр события или результат вызова микросервиса. Критерии используются при определении реакции системы на то или иное действие. Для каждого критерия система отображает источник справочника значений.
- **Справочник значений** — источник значений для критерия. В системе можно выделить несколько типов справочников:
 - Пользовательский справочник;
 - Внешняя система;
 - База данных;
 - Псевдоним;
 - Не задан.

Авторизуйтесь под учетной записью пользователя с ролью, которая дает право только на просмотр критериев. Страница **Критерии** будет иметь для вас следующий вид — [П 2.1.3.1](#). Пользователям с этой ролью список критериев доступен для просмотра, но недоступен для редактирования.

Авторизуйтесь под учетной записью с ролью, которая дает право только на просмотр и редактирование критериев. В этом случае вы сможете заполнять справочники критериев — [П 2.1.3.2](#).

i Примечание. Описание соответствующей ролевой модели — [\[5\]](#).

П 2.1.3.1. Просмотр справочников

Для просмотра справочников:

1. Авторизуйтесь в `Back Office` под именем под учетной записью пользователя с ролью, имеющей право только на просмотр критериев.
2. Перейдите на страницу **Управление > Критерии** и выберите критерий. Вам будут доступны для просмотра свойства критерия.

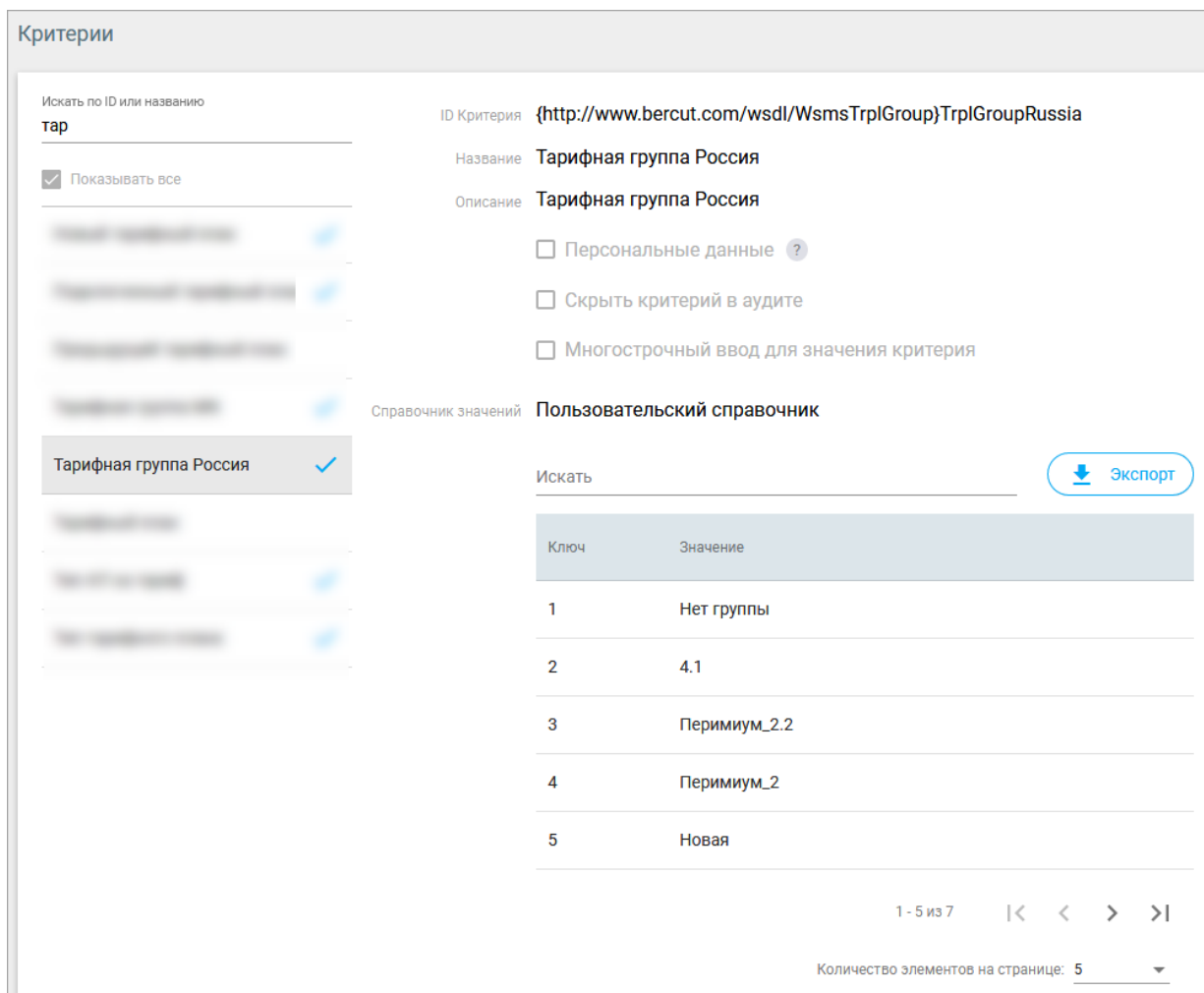


Рис. 62. Вариант отображения справочника для пользователя с ролью User

П 2.1.3.2. Заполнение справочников

При необходимости в настройте справочник для каждого добавленного микросервиса в NS BackOffice.

Для заполнения справочников:

1. Авторизуйтесь в Back Office под учетной записью пользователя с ролью, которая дает право на просмотр и редактирование критериев.
2. Перейдите на страницу **Управление > Критерии** и выберите критерий.
3. Нажмите **Редактировать** настройте параметры для критерия. В поле **Справочник значений** выберите тип справочника и настройте параметры.
 - **Описание** — описание критерия, которое система будет отображать при наведении на критерий.
 - **Скрыть значения критерия в аудите** — поставьте этот флаг, если значения критерия должны быть доступны только пользователям с ролью 'Администратор системы' при просмотре в аудите.
 - **Скрыть критерий в аудите** — поставьте этот флаг, чтобы критерий не отображался в аудите;
 - **Многострочный ввод для значения критерия** — поставьте этот флаг, чтобы при вводе значений параметров в окне **Действия** на вкладке **Таблица правил** страницы «Продукт» был доступен многострочный ввод.
 - **Справочник значений** — выберите тип справочника:

- Пользовательский справочник;
- Внешняя система;
- База данных;
- Псевдоним;
- Не задан.

Тип справочника определяет, что будет служить источником данных для критерия.

Описание настройки для каждого из типов справочников — [5].

Ниже приведен вариант настройки пользовательских справочников.

Примечание. Для критериев доступен поиск по ключу и по значению критерия.

Примечание. Для таблицы критериев справочника доступна сортировка по ключу и значению.

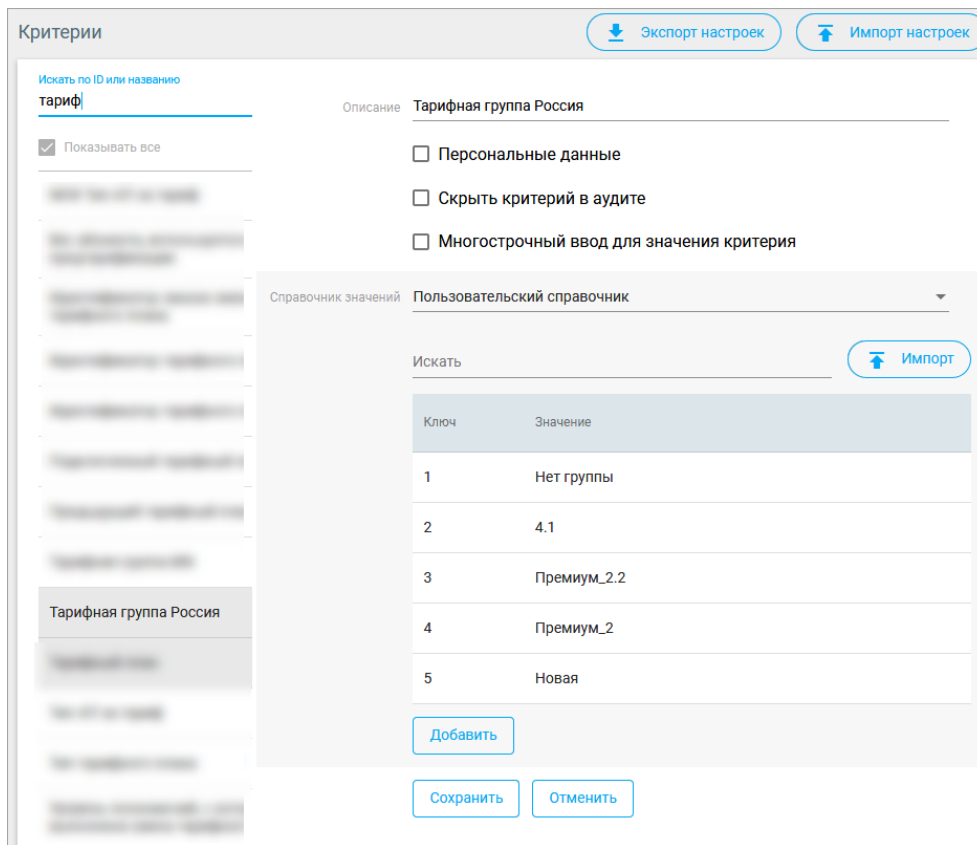


Рис. 63. Вариант настройки справочника типа «Пользовательский справочник»

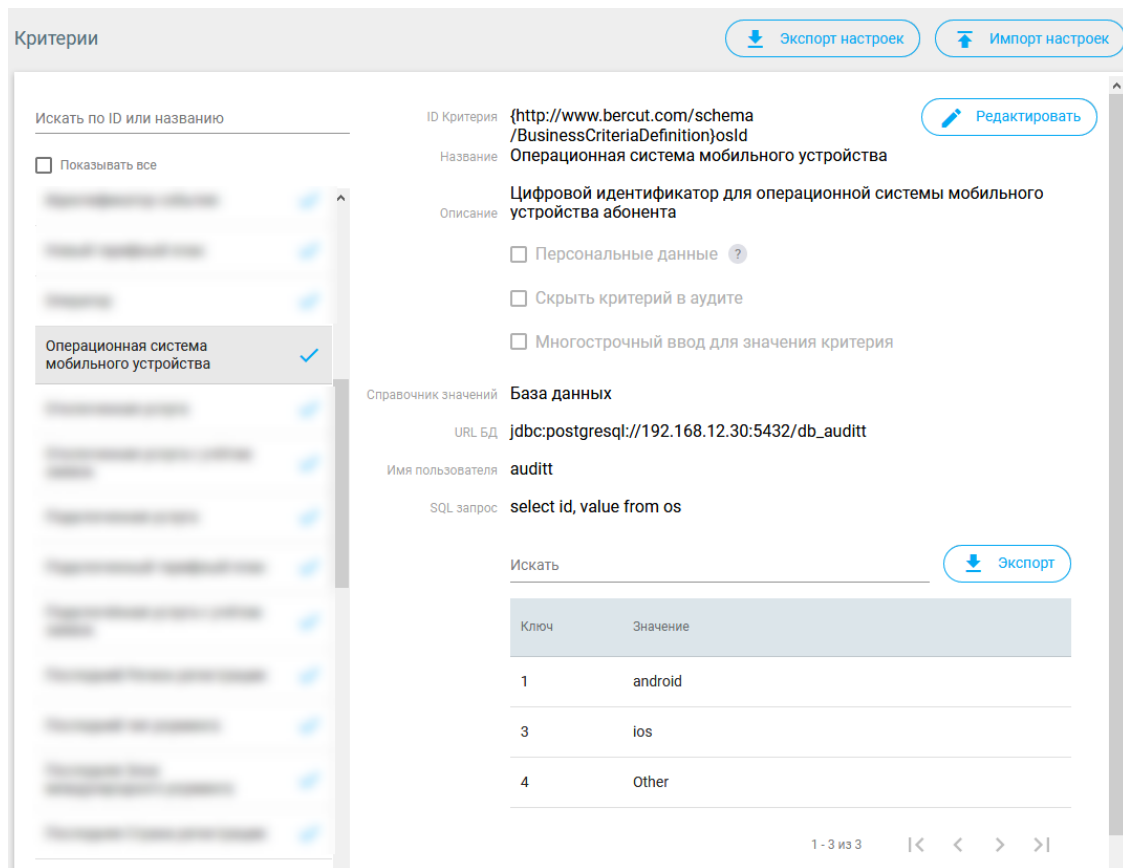


Рис. 64. Вариант настройки справочника типа «База данных»

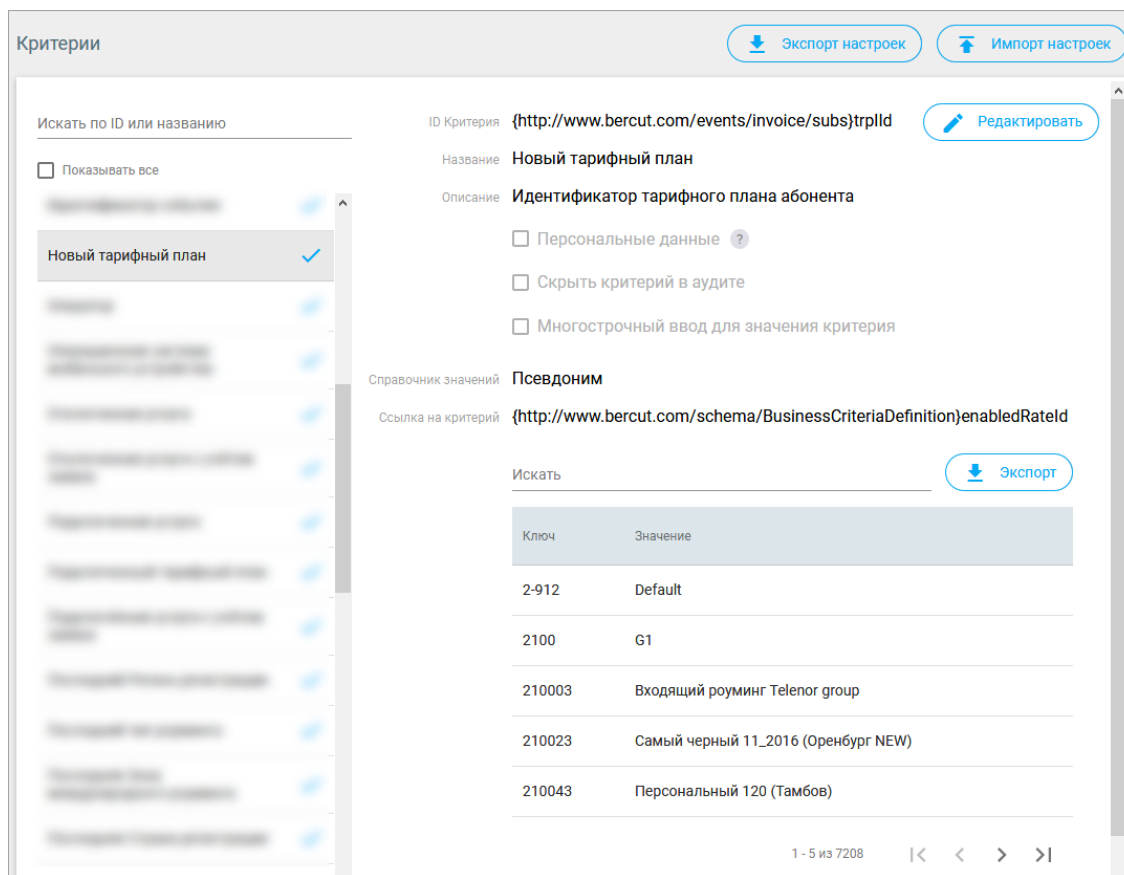


Рис. 65. Вариант настройки справочника типа «Псевдоним»

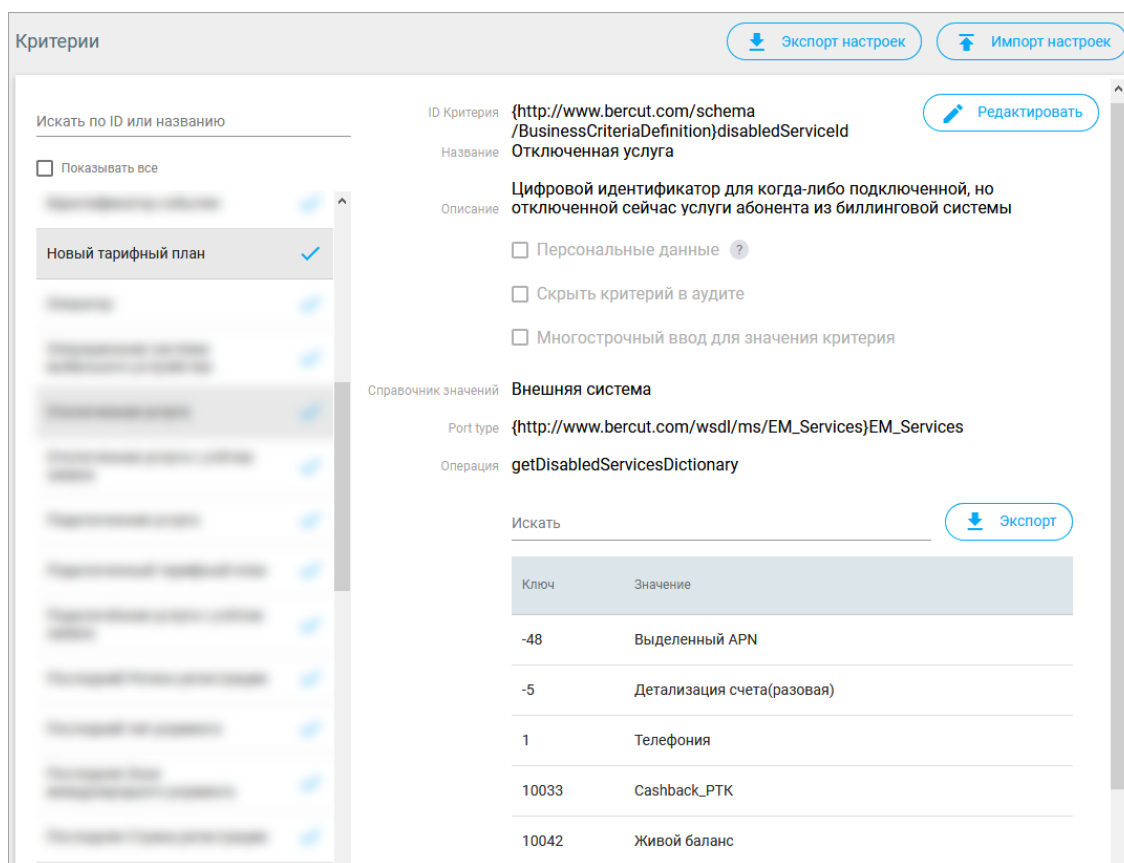


Рис. 66. Вариант настройки справочника типа «Внешняя система»

П 2.1.4. Создание и настройка сценария

В NS BackOffice создайте сценарий с использованием загруженного микросервиса.

Примечание. В разделе представлены настройки тестового контура по проверке конкретного решения. Эти данные носят ознакомительный характер и могут содержать не все настройки, необходимые для продуктивного контура.

Предварительные действия

После загрузки микросервиса в BackOffice, но перед созданием сценария, убедитесь, что для критериев, включенных в микросервис, верно настроены справочники. При необходимости [настройте](#) их.

Добавление и настройка сценария

Порядок настройки:

1. При необходимости создайте продукт.
2. Добавьте сценарий — [П 2.1.4.1.](#)
3. Настройте схему сценария — [П 2.1.4.2.](#)
4. Настройте таблицу правил — [П 2.1.4.3.](#)

П 2.1.4.1. Добавление сценария

В NS BackOffice добавьте сценарий для созданного микросервиса.

Перед добавлением сценария убедитесь, что создан продукт, в котором вы будете добавлять сценарии. При необходимости создайте его.

Описание страниц веб-интерфейса BackOffice и работы со страницей сценариев — [5].

Для добавления сценария:

1. Перейдите на страницу **Продукт** и выберите продукт.
2. На странице сценариев продукта нажмите **Добавить сценарий**.
3. В окне **Создание сценария** выберите параметры *созданного* интерфейса.

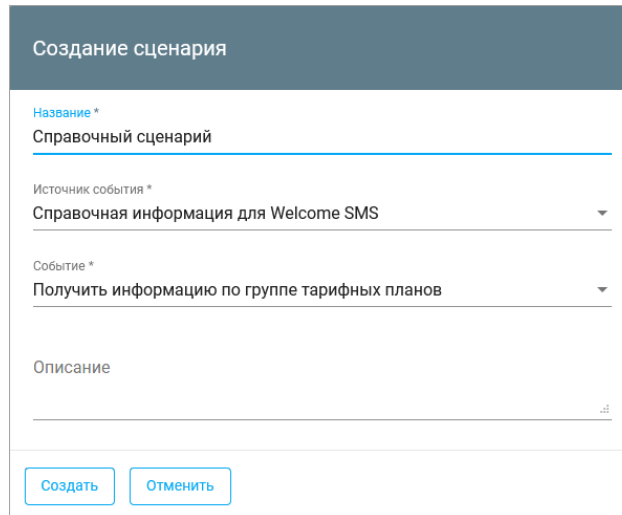


Рис. 67. Пример окна «Создание сценария»

Система Notification System создаст сценарий. Далее настройте схему сценария — [П 2.1.4.2.](#)

П 2.1.4.2. Настройка схемы сценария

В NS BackOffice настройте действия для сценария.

Описание страниц веб-интерфейса BackOffice и работы со страницей сценариев — [5].

На странице созданного сценария перейдите на вкладку **Схема сценария** и настройте схему сценария. Вкладка меню **Схема сценария** страницы **Продукт** состоит из двух областей: *Действия* и *Правила*:

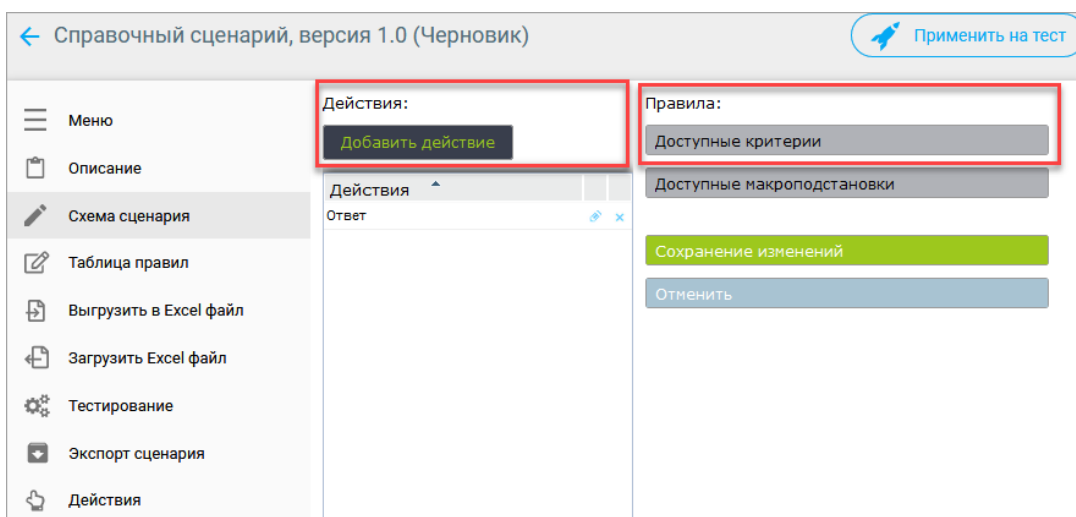


Рис. 68. Области «Действия» и «Правила»

1. Добавьте действие — в области **Действие** нажмите **Добавить действие**. В открывшемся окне **Создание действия**:
 - Задайте название действия в поле «Имя».
 - Убедитесь в наличии выходных параметров в нижней части окна.

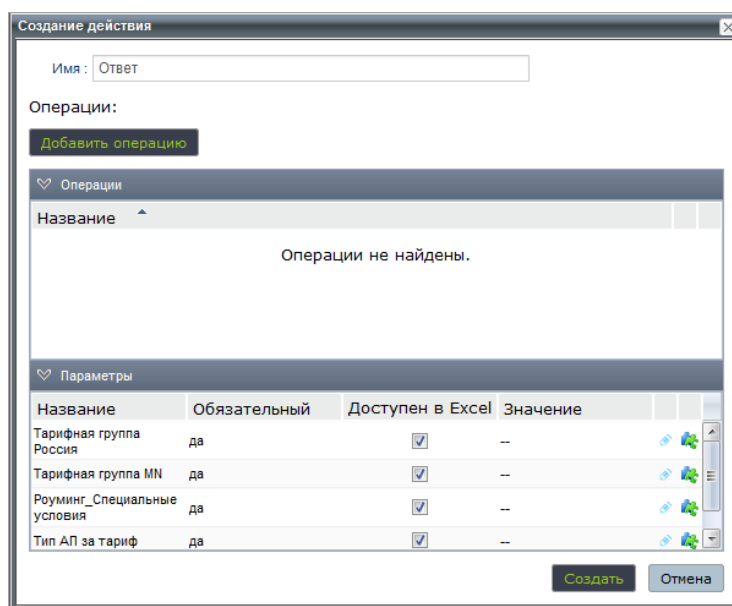


Рис. 69. Вариант добавления действия

2. Если необходимо, добавьте операцию. В окне **Создание действия** нажмите **Добавить операцию**. В окне **Новая операция** справа от поля микросервис нажмите **Выбрать**.
Выберите микросервис и добавьте его в операцию.
3. Настройте критерии. В области **Правила** нажмите **Доступные критерии**. В открывшемся окне **Настройка критериев** нажмите **Добавить**. Откроется окно **Добавление критерия**.

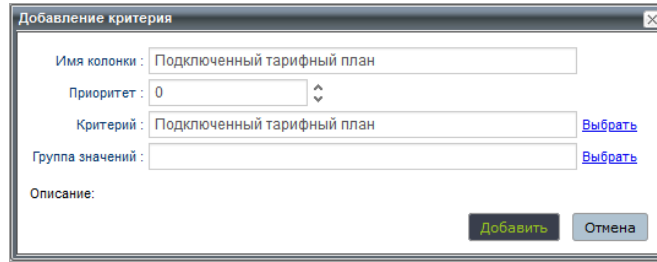


Рис. 70. Вариант настройки критерия в окне «Добавление критерия»

В поле **Критерий** нажмите **Выбрать** справа от поля ввода. Из списка доступных критериев выберите критерии, которые будут соответствовать бизнес-процессам оператора связи.

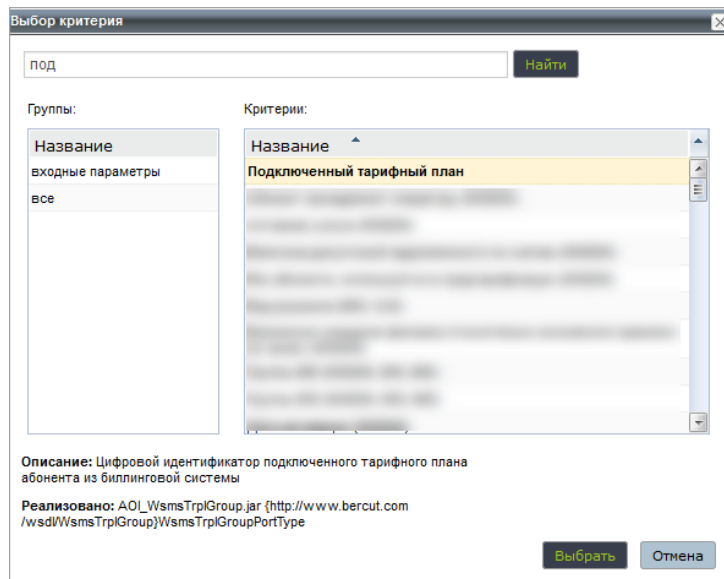


Рис. 71. Вариант настройки критерия «Подключенный тарифный план» в окне «Выбор критерия»

4. Нажмите **Сохранить** для сохранения критерия.

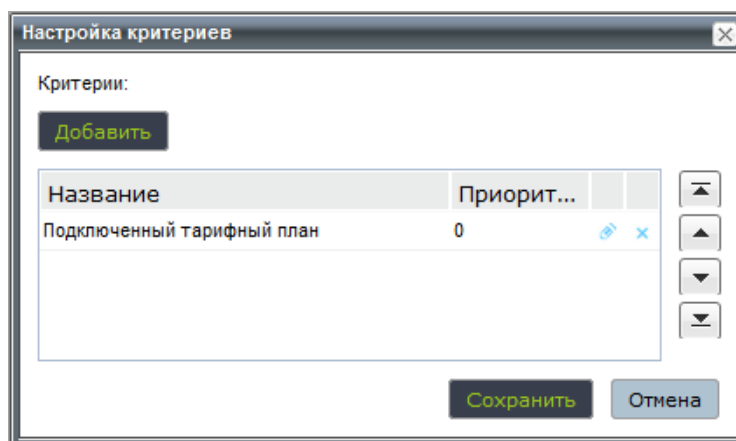


Рис. 72. Вариант настройки критерия

При необходимости задайте выражение для вычисления значения параметра. Для этого нажмите  и в окне **Значение параметра** выберите формат представления результата.

5. Сохраните схему сценария.

Далее настройте таблицу правил — [П 2.1.4.3](#).

П 2.1.4.3. Настройка таблицы правил

Заполните таблицу правил.

Для задания правил вы можете использовать:

- веб-интерфейс системы Notification System: страница **Продукт**, меню **Таблица правил**;
- файл Excel: вкладка **Продукт**, меню **Выгрузить в Excel файл**.

Перед настройкой таблицы правил убедитесь в том, что понимаете работу основных объектов системы:

- *Правило* — набор логических выражений (условий), при соблюдении которых Notification System выполняет действие, соответствующее правилу. В результате исполнения сценария система применяет только одно — первое подходящее правило. При расчете правила система последовательно вычисляет значения входящих в него логических выражений. Расчет ведется в порядке приоритета критериев.
- *Действие* — часть сценария, которая определяет реакцию системы на событие. В сценарий входит одно или несколько действий.
- *Критерий* — параметр события или результат вызова микросервиса.
- *Микросервис* — бизнес-процесс, предназначенный для получения значений критериев или выполнения бизнес-задачи.
- *Макроподстановка* используется при настройке правил сценария. В результате макроподстановки система заменит наименование макроса результатом вызова микросервиса, который ему соответствует.
- *Группа значений* – набор значений, которые могут использоваться в выражениях. Группа содержит списки значений или другие подгруппы, которые в свою очередь включают в себя списки значений. Группы значений используются в правилах при определении реакции системы на то или иное действие.

Настройте работу объектов системы в соответствии с бизнес-процессами оператора связи.

Порядок настройки

Каждая строка таблицы — отдельное правило. Порядок настройки в веб-интерфейсе системы Notification System:

1. Нажмите **Редактировать таблицу**.
2. Создайте правила, соответствующие бизнес-процессам оператора связи, вызвав контекстное меню левой кнопкой мыши в области столбца **№**. Укажите:
 - Столбец **Описание** — название программы.
 - Операторы и значения для критериев, созданных в схеме сценария ([П 2.1.4.2](#)).

Примечание. Убедитесь, что у вас настроены справочники (П 2.1.3.2). Тогда при добавлении значений в таблицу правил вы сможете выбирать значения из настроенных справочников:

Описание	Подключенный тарифный план		Тарифная группа Россия		Тарифная группа MN	
	Операт...	Значен...	Операт...	Значен...	Операт...	Значен...
☰						
☰						
☰			равен	4.1 (2)	равен	Премиум_2
☰			равен			

Искать _____

- 4.1 (2)
- qqqqqq (6)
- Нет группы (1)
- Новая (5)
- Новая Новая (7)
- Перимиум_2 (4)
- Перимиум_2.2 (3)

Рис. 73. Вариант выбора значения из справочника при добавлении правил

- Столбец **Действие** — действия правила. Параметры действия — это действия из схемы сценария (П 2.1.4.2).

3. Нажмите **Сохранить**, чтобы сохранить настроенные правила.

№	Описание	Подключенный тарифный план		...
		Оператор	Значение	
1		равен	Мой разговор 12_2019 (Ростов-на-Дону) (412...	...
2		равен	Мой онлайн (Ростов-на-Дону) (213405)	...
3		равен	Мой онлайн + new (Ростов-на-Дону) (313604)	...
4		равен	Мой онлайн+ 08_2019 (Тамбов) (212698)	...
5		равен	Мой онлайн+ 09_2019(Тамбов) (313243)	...
6		равен	Мой онлайн+ 11_2018 (Ростов-на-Дону) (2120...	...
7		равен	Мой онлайн+ 11_2018 (Тамбов) (212089)	...
8		равен	Мой онлайн 12_2019 (Ростов-на-Дону) (21268...	...

Рис. 74. Вариант настройки таблицы правил

П 2.1.5. Обновление микросервиса и сценария

П 2.1.5.1. Обновление микросервиса

Обновите WSDL-файл интерфейса, сформируйте JAR-файл микросервиса.

Для обновления *созданного* ранее WSDL-интерфейса:

1. Откройте на редактирование WSDL-файл и добавьте в него новые критерии. Критерии добавляйте в конец списка параметров с атрибутом `minOccurs="0"`.

! Внимание! Ни в коем случае не изменяйте идентификатор критериев, их порядок в файле и не удаляйте их. Это приведет к ошибкам применения сценария и во время работы со сценарием. Такой сценарий можно будет только удалить.

2. Обновите версию справочника в блоке `<xsd:schema targetNamespace="http://www.bercut.com/wsd/ <<NAME>>" version="X.Y">`
X.Y — номер версии.
3. Для файла MANIFEST.MF убедитесь, что номер версии, указанной в блоке `Implementation-Version: X.Y` был больше или равен версии существующего сервиса.
4. По возможности выполните валидацию файла.
5. Сформируйте JAR-архив ([П 2.1.1.5](#)).

Пример обновления WSDL-интерфейса:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="AOI_WsmtsTrplGroup"
targetNamespace="http://www.bercut.com/wsd/WsmtsTrplGroup"
xmlns="http://schemas.xmlsoap.org/wsd/"
xmlns:wsd="http://schemas.xmlsoap.org/wsd/"
xmlns:nsns="http://com.bercut/schema/NsModel"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.bercut.com/wsd/WsmtsTrplGroup"
xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
xmlns:servd="http://www.bercut.com/spec/schema/ServiceDefinition"
xmlns:bcd="http://www.bercut.com/schema/BusinessCriteriaDefinition"
xmlns:soap="http://schemas.xmlsoap.org/wsd/soap/">
<types>
<xsd:schema targetNamespace="http://www.bercut.com/wsd/WsmtsTrplGroup"
version="1.1">
<xsd:import schemaLocation="ServiceDefinition.xsd"
namespace="http://www.bercut.com/spec/schema/ServiceDefinition"/>
<xsd:import schemaLocation="BusinessCriteriaDefinition.xsd"
namespace="http://www.bercut.com/schema/BusinessCriteriaDefinition"/>
<xsd:import schemaLocation="SimpleDefinition.xsd"
namespace="http://www.bercut.com/spec/schema/SimpleDefinition"/>

<xsd:element name="getWsmtsTrplGroupRequest">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="bcd:enabledRateId"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name="getWsmtsTrplGroupResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="TrplGroupRussia" type="xsd:string">
<xsd:annotation>
<xsd:documentation>{Тарифная группа Россия}Тарифная группа Россия/<
xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TrplGroupMN" type="xsd:string">
<xsd:annotation>
```

```

    <xsd:documentation>{Тарифная группа MN}Тарифная группа MN</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="TrplGroupSpec" type="xsd:string">
  <xsd:annotation>

  <xsd:documentation>{Роуминг_Специальные условия}Тарифная группа Специальные условия</
xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="TrplApType" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>{Тип АП за тариф}Тип АП за тариф</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="osId" type="xsd:string" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>{Оперционная система}ОС абонента</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
  </xsd:complexType>
</xsd:schema>
</types>
<message name="BusinessFault">
  <part name="fault" element="servd:businessFault"/>
</message>
<message name="getWsmsTrplGroupRequest">
  <part name="parameters" element="tns:getWsmsTrplGroupRequest"/>
</message>
<message name="getWsmsTrplGroupResponse">
  <part name="parameters" element="tns:getWsmsTrplGroupResponse"/>
</message>

<portType name="WsmsTrplGroupPortType">
  <documentation>{Справочная информация для Welcome SMS}</documentation>
  <operation name="getWsmsTrplGroup">
    <documentation>{Получить информацию по группе тарифных планов[ACTION,GETTER]}
    Получить информацию по группе тарифных планов</documentation>
    <input name="getWsmsTrplGroupRequest"
      message="tns:getWsmsTrplGroupRequest"/>
    <output name="getWsmsTrplGroupResponse"
      message="tns:getWsmsTrplGroupResponse"/>
    <fault name="BusinessFault"
      message="tns:BusinessFault"/>
  </operation>
</portType>
<binding name="WsmsTrplGroupPortTypeBinding" type="tns:WsmsTrplGroupPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getWsmsTrplGroup">
    <input name="getWsmsTrplGroupRequest">
      <soap:body use="literal" parts="parameters"/>
    </input>
    <output name="getWsmsTrplGroupResponse">
      <soap:body use="literal"/>
    </output>
    <fault name="BusinessFault">
      <soap:fault use="literal" name="BusinessFault"/>
    </fault>
  </operation>
</binding>

<service name="WsmsTrplGroupService">
  <port name="WsmsTrplGroupServicePort"
    binding="tns:WsmsTrplGroupPortTypeBinding">
    <soap:address
      location="http://localhost:${HttpDefaultPort}/WsmsTrplGroupService/
WsmsTrplGroupPort"/>
  </port>
</service>
<plnk:partnerLinkType name="WSMSPlt">
  <plnk:role name="WsmsTrplGroupPortTypeRole"

```

```
portType="tns:WsmsTrplGroupPortType"/>
</plnk:partnerLinkType>
</definitions>
```

П 2.1.5.2. Обновление сценария

В NS BackOffice вы можете изменить схему сценария и настроить новые параметры таблицы правил для загруженного микросервиса.

Примечание. В разделе представлены настройки тестового контура по проверке конкретного решения. Эти данные носят ознакомительный характер и могут содержать не все настройки, необходимые для продуктивного контура.

Обновление сценария выполняется для новой версии микросервиса.

Обновление сценария

Порядок обновления:

1. **Загрузите** новую версию микросервиса в систему.
2. Откройте схему сценария, которую вы создали. На вкладке меню **Схема сценария** страницы **Продукт** откройте созданное действие и убедитесь, что новый параметр появился в списке параметров в окне **Редактирование действия**:

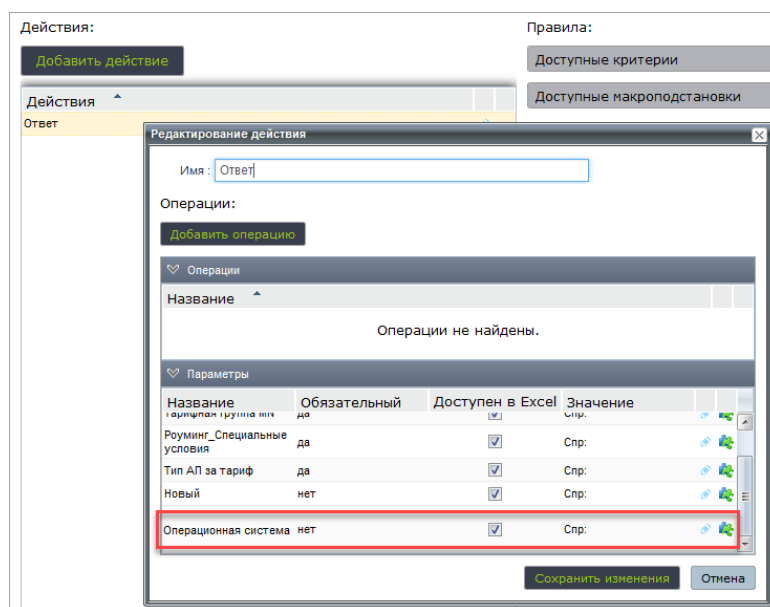


Рис. 75. Вариант списка параметров для новой версии микросервиса

Убедитесь, что для нового параметра активен флаг **Доступен в Excel-файле**.

Примечание. Если флаг не активирован, параметр не будет отображаться и в поде **Действие** таблицы правил, и в Excel-файле.

3. Перейдите в меню **Таблица правил** и убедитесь, что новый параметр появился в окне **Действия**. При необходимости задайте значение параметра и сохраните изменения.

Убедитесь, что у вас настроены справочники (П 2.1.3.2) для всех параметров. Тогда при добавлении значений в таблицу правил вы сможете выбирать значения из настроенных справочников:

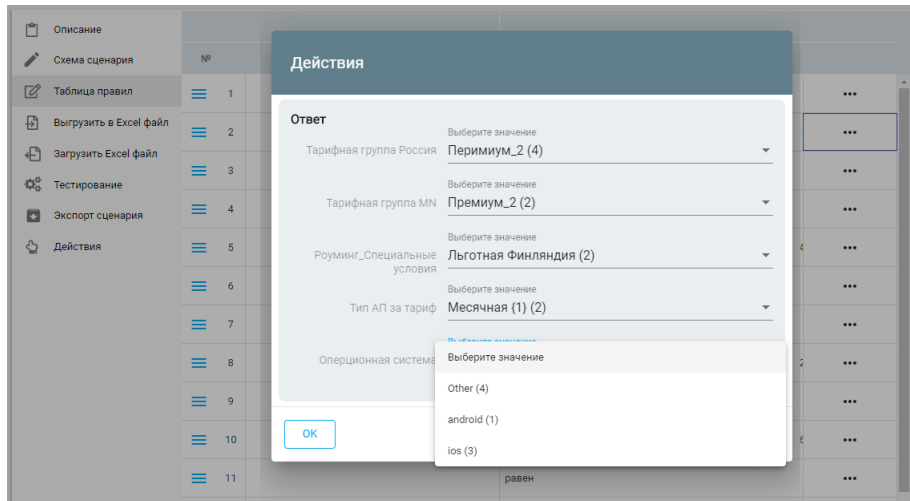


Рис. 76. Вариант отображения в таблице правила с настроенными справочниками

4. На вкладке меню **Схема сценария** страницы **Продукт** для сценариев, для которых вы хотите использовать добавленный критерий, его необходимо добавить в схему сценария.

Критерии созданного и настроенного [сценария](#) на основе [загруженного](#) микросервиса доступны для использования в других сценариях.

П 2.2. Событие для отложенного сценария

Создание микросервиса, который содержит событие для запуска отложенного сценария, описано в [П 1.1](#).

П 2.3. Событие с поддержкой транзакционности

Начиная с версии Notification System 2.10 вы можете выполнять несколько операций в рамках транзакции. Для этого создайте микросервис с поддержкой транзакционности.

Особенности создания микросервиса с поддержкой транзакционности:

1. В WSDL-файл добавить атрибут **transaction=true** для соответствующего porttype. Например:

```
<plnk:partnerLinkType name="EndInterfaceTx" xmlns:ext="http://bercut.com/wsdl/extensions" ext:transaction="true">
```

Примечание. Пространство имен для транзакционности — `xmlns:ext="http://bercut.com/wsdl/extensions"`.

Внимание! Добавление атрибута **transaction="true"** ведет к нарушению обратной совместимости интерфейса.

2. В конкретной операции задайте параметры — режим выполнения операции в рамках транзакционности. Параметры задаются в квадратных скобках через запятую после ACTION.

Допустимые значения параметров:

- **NON_TX** — данная операция не должна выполняться в рамках транзакции, хотя весь интерфейс, который вы указали в шаге 1, помечен как транзакционный.

Пример:

```
<operation name="OnEndMandatoryTx">
  <documentation>{вызов операции End(транзакционный с
  NON_TX) с обязательными параметрами[ACTION, NON_TX]}</documentation>
```

- **WITH_COMP** — пользователь в Back Office должен обязательно настроить компенсацию для данной операции, хотя весь интерфейс, который вы указали в шаге 1, помечен как транзакционный.

Пример:

```
<operation name="OnEndTxС">
  <documentation>{вызов операции End(транзакционный+компенс) [ACTION, WITH_COMP]}</documentation>
```

Если параметры не заданы, операция выполняется в рамках транзакции. Операция ROLLBACK для транзакции в случае возникновения ошибки выполняется через шину RTSIB.

П 2.4. События для интеграции с Apache Kafka

П 2.4.1. Событие для обработки SA NS Kafka Consumer

Вы можете создавать новые события для обработки сообщения из Apache Kafka.

При получении сообщений из Apache Kafka компонент SA NS Kafka Consumer сравнивает их с указанными в XSD-файлах. Notification System может обрабатывать только те сообщения, формат которых совпадает с указанным в XSD-файлах.

Чтобы Notification System могла обрабатывать новое сообщение из Apache Kafka, вы должны создать для него событие в новом XSD-файле или добавить событие в один из существующих файлов.

Вы можете создать событие двумя способами:

1. С помощью JAR-файла микросервиса.
2. С помощью файла маппинга.

Создание события с помощью JAR-файла микросервиса

Для создания события в новом файле:

1. Создайте XSD-файл — [П 2.4.1.1.](#)
2. Сформируйте JAR-файл микросервиса — [П 2.4.1.2.](#)
3. Итоговый JAR-файл файл:
 - 3.1. Загрузите в NS Back Office на вкладке **Управление > Микросервисы**. Процесс загрузки аналогичен описанному в разд. [П 2.1.2.](#)
 - 3.2. Добавьте в каталог `LWSA\em_data` на сервере, где установлен LWSA Container с SA NS Kafka Consumer.
4. Чтобы Notification System загрузила файл, перешелкните значение MIB-параметра `/SA_NsKafkaConsumer/Core/Configuration/ReloadFiles (4.6)` в значение True.

5. Создайте файл маппинга для сообщения. Например `file.consumer.xml`.

Примечание. Подробнее про преобразование с помощью файла маппинга — П 3.1.

Пример содержимого файла:

```
<consumer-mapping>
  <rtsib-element>notify-10014</rtsib-element>
  <detection>
    <type>VALUE</type>
    <expression>//*:msgTypeId</expression>
    <value>-10014</value>
  </detection>
  <body-encoding>utf-8</body-encoding>
  <body-format>xml</body-format>
  <mapping>
    <param>
      <from>//*:msgTypeId</from>
      <to>eventId</to>
    </param>
    <param>
      <from>//*:subsId</from>
      <to>subsId</to>
    </param>
    <param>
      <from>//*:eventContext/*:param[*:key='USI_ID']/*:value</from>
      <to>newSubsUsiId</to>
    </param>
    <param>
      <from>//*:eventContext/*:param[*:key='USI_ID_OLD']/*:value</from>
      <to>oldSubsUsiId</to>
    </param>
    <param>
      <from>//*:eventContext/*:param[*:key='IMSI']/*:value</from>
      <to>newSubsImsi</to>
    </param>
    <param>
      <from>//*:eventContext/*:param[*:key='IMSI_OLD']/*:value</from>
      <to>oldSubsImsi</to>
    </param>
  </mapping>
</consumer-mapping>
```

6. Загрузите созданные файлы:

- Созданный JAR-файл микросервиса:

1. На LWSA Container EM BO — в каталог `$LWSA/em_data` на сервере или в веб-интерфейс NS Back Office на вкладке **Микросервисы**.
2. На LWSA Container, на котором установлен SA Ns Kafka Consumer,— в каталог `$LWSA/em_data` на сервере.

- Созданные XML-файлы маппинга — на LWSA Container, на котором установлен SA Ns Kafka Consumer, в каталог `$LWSA/em_data` на сервере.

7. Настройте MIB-параметры SA Ns Kafka Consumer. Перейдите в MIB-группу `/LWSAContainer/@artifacts/SA_NsKafkaConsumer/Core/Configuration/Topics/` и создайте новую подгруппу. Имя подгруппы — это имя топика.

В подгруппе укажите имена созданных XML-файлов в значении MIB-параметров:

- `ConversionMappingFiles` — название файла маппинга для преобразования, должно оканчиваться на `.consumer.xml`. Если не заполнено, то будет срабатывать маппинг по умолчанию.

Примечание. Компонент должен наследоваться от AOI NS KafkaConsumer.

В разделе приведен пример создания XSD-файла для получения из Apache Kafka сообщения следующего формата:

```
{
  "other-event": {
    "ErrorName" : "Ошибка создания УЗ. Есть ОТТ учетка",
    "spaceid" : "",
    "ErrorType" : "ErrCreateОТТ",
    "response.trace.global_code" : "winkproxy_10.12.75.121:7090_towink_end",
    "response.result.error.message" : "Телефон привязан к действующей учётной записи",
    "msisdnOld" : "",
    "action" : "",
    "msisdn" : "79920303502",
    "response.status" : "412",
    "msisdnNew" : "",
    "timestamp" : "2020-04-29T09:58:35.445+0300",
    "response.path" : ""
  }
}
```

Создание события с помощью файла маппинга

Для создания события вы также можете использовать файл маппинга — XML-файл, соответствующий схеме `consumer-mapping.xsd`. Это позволяет преобразовывать сообщения, получаемые из Apache Kafka, из любого формата.

Примечание. Правила преобразования с помощью файла маппинга — общие для всех адаптеров `consumer` и описаны в разделе [П 3.1](#).

Обработка блока `trace`

Начиная с версии SA NS Kafka Consumer 1.4 добавлена обработка блока `trace` в JSON-формате.

Примечание. Логика чтения блока `trace` реализована только для JSON-формата и маппинга по умолчанию. Для маппинга с помощью файлов маппинга используйте в файле маппинга тег `<to-event-context>`.

По умолчанию обработка блока `trace` выключена — значение `TraceBlockEnabled=False`.

Пример:

1. SA NS Kafka Consumer получает сообщение вида:

```
{
  "StartRequestAnyParams":{
    "ruleIdStart":"test_rule",
    "dataDigit":[1,2,3]
  },
  "trace":{
    "requestTime":"2021-03-12T15:37:29.828+0300",
    "requestId":"b596e9c0-832f-11eb-8b28-005056b0dde0",
    "notifyId":"notify-88"
  }
}
```

2. После маппинга по умолчанию весь блок `trace` попадает в `EventContext` сообщения:

```
<ber-ns0:kafkaRequest>
  <ber-ns0:Header>
    <ber-ns0:kafkaTopicName></ber-ns0:kafkaTopicName>
  </ber-ns0:Header>
  <ber-ns0:Body>
    <ber-ns2:StartRequestAnyParams>
      <ber-ns1:ruleIdStart>test_rule</ber-ns1:ruleIdStart>
      <ber-ns1:dataDigit>1</ber-ns1:dataDigit>
      <ber-ns1:dataDigit>2</ber-ns1:dataDigit>
      <ber-ns1:dataDigit>3</ber-ns1:dataDigit>
    </ber-ns2:StartRequestAnyParams>
  </ber-ns0:Body>
</ber-ns0:kafkaRequest>
```

```

        </ber-ns2:StartRequestAnyParams>
    </ber-ns0:Body>
    <ber-ns0:EventContext>
        <ber-ns3:eventContextParameter>
            <ber-ns3:name>requestTime</ber-ns3:name>
            <ber-ns3:value>2021-03-12T15:37:29.828+0300</ber-ns3:value>
        </ber-ns3:eventContextParameter>
        <ber-ns3:eventContextParameter>
            <ber-ns3:name>requestId</ber-ns3:name>
            <ber-ns3:value>b596e9c0-832f-11eb-8b28-005056b0dde0</ber-ns3:value>
        </ber-ns3:eventContextParameter>
        <ber-ns3:eventContextParameter>
            <ber-ns3:name>notifyId</ber-ns3:name>
            <ber-ns3:value>notify-88</ber-ns3:value>
        </ber-ns3:eventContextParameter>
    </ber-ns0:EventContext>
</ber-ns0:kafkaRequest>

```

П 2.4.1.1. Создание XSD-файла

Для удобства при создании XSD-файла интерфейса вы можете использовать готовый [шаблон](#).

П 2.4.1.1.1. Создание шаблона XSD-файла

При создании XSD-файла вы можете использовать готовый шаблон:

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.tele2.com/events/any-system"
  targetNamespace="http://www.tele2.com/events/any-system"
  xmlns:commnd="http://bercut.com/em/schema/CommonDefinitions"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">

  <xsd:import schemaLocation="BA_EM_CommonDefinitions.xsd"
    namespace="http://bercut.com/em/schema/CommonDefinitions"/>

  <xsd:element name="other-event" type="tns:t-other-event" commnd:anyType="request">
    <xsd:annotation>
      <xsd:documentation xml:lang="ru">{Какое-то событие}описание</xsd:documentation>
    </xsd:annotation>
  </xsd:element>

  <xsd:complexType name="t-other-event">
    <xsd:annotation>
      <xsd:documentation xml:lang="ru">Название события </xsd:documentation>
    </xsd:annotation>
    <xsd:all>
      <xsd:element name="ErrorName" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation xml:lang="ru">{ErrorName}ErrorName Описание ошибки
</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="spaceid" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation xml:lang="ru">{spaceid}spaceid</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="ErrorType" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation xml:lang="ru">{ErrorType}ErrorType описание типа ошибки
</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="response.trace.global_code" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation xml:lang="ru">
{response.trace.global_code}response.trace.global_code описание</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:all>
  </xsd:complexType>

```

```

    <xsd:element name="response.result.error.message" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{response.result.error.message} Текст ошибки
      </xsd:documentation>
    </xsd:element>
    <xsd:element name="msisdnOld" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{msisdnOld}msisdnOld описание параметра
      </xsd:documentation>
    </xsd:element>
    <xsd:element name="action" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{action}action описание параметра
      </xsd:documentation>
    </xsd:element>
    <xsd:element name="msisdn" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{msisdn}msisdn описание параметра
      </xsd:documentation>
    </xsd:element>
    <xsd:element name="response.status" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">
{response.status}response.status описание параметра</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="msisdnNew" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{msisdnNew} описание параметра
      </xsd:documentation>
    </xsd:element>
    <xsd:element name="timestamp" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{timestamp}timestamp описание параметра
      </xsd:documentation>
    </xsd:element>
    <xsd:element name="response.path" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">
{response.path}response.path описание параметра</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
</xsd:schema>

```

i Примечание. Шаблон содержит параметры события, которые соответствуют сообщению, получаемому из Apache Kafka (П 2.4.1). Вы должны задать параметры, соответствующие вашему событию.

В общем виде шаблон выглядит следующим образом:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- В блоке задается описание используемых пространств имен — namespaces -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.tele2.com/events/<<NAME>>"
  targetNamespace="http://www.tele2.com/events/<<NAME>>"
  xmlns:commd="http://bercut.com/em/schema/CommonDefinitions"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">

  <xsd:import schemaLocation="BA_EM_CommonDefinitions.xsd"
  namespace="http://bercut.com/em/schema/CommonDefinitions"/>

  <xsd:element name="<<ИМЯ СОБЫТИЯ>>"

```

```

type="tns:<<ТИП СОБЫТИЯ, обычно t-<<ИМЯ СОБЫТИЯ>>>>" commd:anyType="request">
  <xsd:annotation>
    <xsd:documentation xml:lang="ru">{Описание события для отображения в NS Back
Office}Описание события</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="<<ТИП СОБЫТИЯ>>">
  <xsd:annotation>
    <xsd:documentation xml:lang="ru">Название события </xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="<параметр события>" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">
{<параметр события>}<параметр события> Описание параметра</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="<параметр события 2>" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{<параметр события 2>}<параметр события
2> Описание параметра 2</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
</xsd:schema>

```

П 2.4.1.1.2. Создание XSD-файла по шаблону

Вы можете создать XSD-файл интерфейса по шаблону.

Для создания XSD-файла по шаблону:

1. Создайте каталог, в котором будут храниться файлы.
2. Создайте в каталоге новый файл и скопируйте туда [шаблон](#).
3. Задайте пространство имен для данного события в двух местах в блоке:

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.tele2.com/events/имя namespace"
  targetNamespace="http://www.tele2.com/events/имя namespace"
  xmlns:commd="http://bercut.com/em/schema/CommonDefinitions"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">

```

i Примечание. Название пространства имен должно отличаться для разных групп событий.

4. Задайте основные параметры события в блоке:

```

<xsd:element name="<<ИМЯ СОБЫТИЯ>>"
type="tns:<<ТИП СОБЫТИЯ, обычно t-<<ИМЯ СОБЫТИЯ>>>>" commd:anyType="request">
  <xsd:annotation>
    <xsd:documentation xml:lang="ru">{Описание события для отображения в NS
Back Office}Описание события</xsd:documentation>
  </xsd:annotation>
</xsd:element>

```

- 4.1. Укажите имя события. Имя события соответствует имени корневого элемента в JSON.
- 4.2. Укажите тип события. Тип события обычно задается как имя события с префиксом *t-*.
- 4.3. Убедитесь, что для события задан атрибут anyType.
- 4.4. В блоке `<xsd:documentation>` задайте имя события: в фигурных скобках укажите имя события, после — описание.

5. Укажите тип события и описание в блоке:

```
<xsd:complexType name="<<ТИП СОБЫТИЯ>>">
  <xsd:annotation>
    <xsd:documentation xml:lang="ru">Название события </xsd:documentation>
  </xsd:annotation>
```

6. Добавьте в событие параметры. Для каждого параметра задайте имя, тип и описание.

- Имя параметра — соответствует параметру сообщения, получаемому из Apache Kafka (П 2.4.1).
- Тип параметра — укажите тип параметра, который равен типу параметра получаемого сообщения.

i Примечание. Если формат даты соответствует XSD-формату — например, 2020-04-29T09:58:35.445+0300 — замените тип на `xsd:dateTime`.

- Описание параметра — укажите описание параметра. Оно должно быть понятно для пользователя веб-интерфейса.

```
<xsd:element name="<параметр события>" type="<тип параметра>">
  <xsd:annotation>
    <xsd:documentation xml:lang="ru">
      {<параметр события>}<параметр события> Описание параметра</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
```

7. Сохраните получившийся файл.

П 2.4.1.2. Создание JAR-файла

Добавьте в каталог с XSD-файлом другие файлы, после чего создайте JAR-файл.

1. Добавьте в каталог с *созданным* XSD-файлом другие файлы:

- AOI_NS_KafkaConsumer.wsdl
- BA_EM_CommonDefinitions.xsd
- ComplexDefinition.xsd
- NS_KafkaDefinition.xsd
- ProfileDefinition.xsd
- ServiceDefinition.wsdl
- ServiceDefinition.xsd
- SimpleDefinition.xsd
- \META-INF\MANIFEST.MF

Файлы можно взять из любого существующего микросервиса, предназначенного для работы с Apache Kafka.

i Примечание. Убедитесь, что JAR-файл содержит файл `AOI_NS_KafkaConsumer.wsdl`. В противном случае система Notification System не сможет его установить.

Эти файлы содержат список заданных ранее критериев, которые вы сможете использовать при настройке параметров события.

Имя	Дата изменения	Тип	Размер
META-INF	18.12.2019 13:57	Папка с файлами	
AOI_NS_KafkaConsumer.wsdl	18.12.2019 13:57	Файл "WSDL"	4 КБ
BA_EM_CommonDefinitions.xsd	18.12.2019 13:57	Файл "XSD"	2 КБ
ComplexDefinition.xsd	18.12.2019 13:57	Файл "XSD"	18 КБ
NS_KafkaDefinition.xsd	18.12.2019 13:57	Файл "XSD"	2 КБ
other-event.xsd	30.06.2020 13:17	Файл "XSD"	5 КБ
ProfileDefinition.xsd	18.12.2019 13:57	Файл "XSD"	37 КБ
ServiceDefinition.wsdl	18.12.2019 13:57	Файл "WSDL"	4 КБ
ServiceDefinition.xsd	18.12.2019 13:57	Файл "XSD"	11 КБ
SimpleDefinition.xsd	18.12.2019 13:57	Файл "XSD"	73 КБ

Рис. 77. Пример содержимого каталога

2. Для создания JAR-файла:

- 2.1. Перейдите в каталог, в котором находится созданный XSD-файл, а также остальные WSDL- и XSD-файлы.
- 2.2. Создайте ZIP-архив.
- 2.3. В сформированном архиве замените расширение .zip на .jar.

П 2.4.2. Событие для обработки SA NS Kafka Producer

Вы можете создавать новые события для обработки сообщения перед отправкой в Apache Kafka.

При получении сообщений от Notification System компонент SA NS Kafka Producer преобразует сообщение в JSON или XML, отправляет сообщение в Apache Kafka.

Для создания события в новом файле:

1. Создайте XSD-файл — [П 2.4.2.1.](#)
2. Сформируйте JAR-файл микросервиса — [П 2.4.2.2.](#)
3. Получившийся JAR-файл файл:
 - 3.1. Загрузите в NS Back Office на вкладке **Управление > Микросервисы**. Процесс загрузки аналогичен описанному в разд. [П 2.1.2.](#)
 - 3.2. Добавьте в каталог LWSA\em_data на сервере, где установлен LWSA Container с SA NS Kafka Producer.
4. Чтобы Notification System загрузила файл, перещелкните значение MIB-параметра /SA_NsKafkaProducer/Core/Configuration/ReloadFiles (4.6) в значение True.

В дальнейшем вы сможете выбрать созданное действие на странице **Выбор микросервиса**:

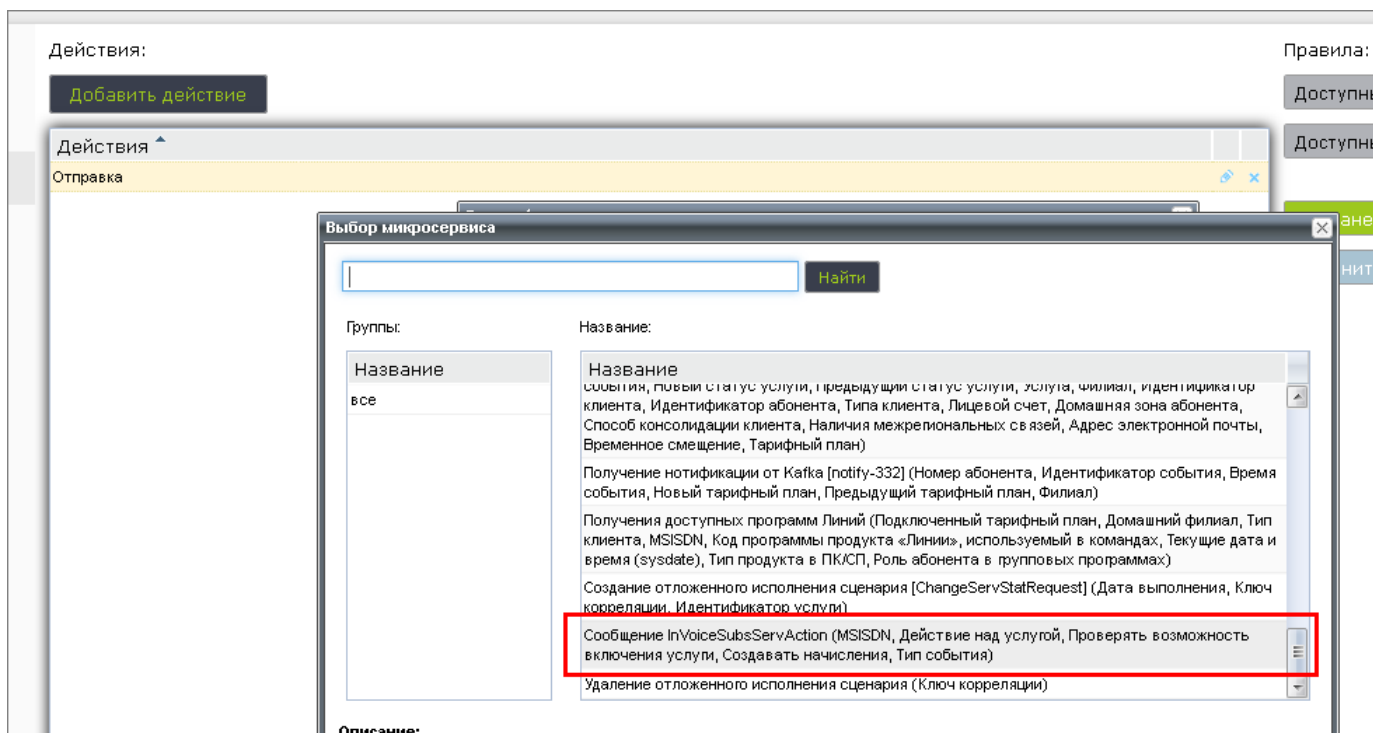


Рис. 78. Вариант выбора созданного действия

П 2.4.2.1. Создание XSD-файла

Для удобства при создании XSD-файла интерфейса вы можете использовать готовый [шаблон](#).

П 2.4.2.1.1. Создание шаблона XSD-файла

При создании XSD-файла вы можете использовать готовый шаблон:

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.bercut.com/action/invoice/subs_serv"
  xmlns:tns="http://www.bercut.com/action/invoice/subs_serv"
  xmlns:cd="http://bercut.com/em/schema/CommonDefinitions"
  xmlns:bcd="http://www.bercut.com/schema/BusinessCriteriaDefinition"
  xmlns:cxdf="http://www.bercut.com/spec/schema/ComplexDefinition"
  version="1.0">

  <xsd:import schemaLocation="BusinessCriteriaDefinition.xsd" namespace="http://www.bercut.com/schema/BusinessCriteriaDefinition"/>
  <xsd:import schemaLocation="BA_EM_CommonDefinitions.xsd" namespace="http://bercut.com/em/schema/CommonDefinitions"/>
  <xsd:import schemaLocation="ComplexDefinition.xsd" namespace="http://www.bercut.com/spec/schema/ComplexDefinition"/>

  <xsd:element name="InVoiceSubsServAction" cd:anyType="request">
    <xsd:annotation>
      <xsd:documentation>{Какое-то событие}описание</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="bcd:msisdn"/>
        <xsd:element ref="bcd:serviceAction"/>
        <xsd:element ref="bcd:checkSwitchOn"/>
        <xsd:element ref="bcd:chargeOrder"/>
        <xsd:element name="eventId" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>{Тип события}Тип события</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```



```

        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

і Примечание. Шаблон содержит параметры события, соответствующие сообщению, получаемому из бизнес-процесса. Вы должны задать параметры, соответствующие вашему событию.

При настройке соблюдайте следующие правила:

1. Основной тег `element` должен содержать атрибут `cd:anyType="request"` — это операция (действие), которую пользователь выбирает в веб-интерфейсе Notification System.
2. Основной тег `element` должен иметь дочерний тег `<xsd:annotation>`.
3. Внутри тега `<xsd:sequence>` для параметров рекомендуется использовать теги `element` через `ref`: `<xsd:element ref="<Элемент>" />`. Вы также можете определить свои теги `element` внутри, хотя это делать не рекомендуется.

В общем виде шаблон выглядит следующим образом:

```

<?xml version="1.0" encoding="UTF-8"?>
<!--В блоке задается описание используемых пространств имен – namespaces -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.bercut.com/action/invoice/subs_serv"
    xmlns:tns="http://www.bercut.com/action/invoice/subs_serv"
    xmlns:cd="http://bercut.com/em/schema/CommonDefinitions"
    xmlns:bcd="http://www.bercut.com/schema/BusinessCriteriaDefinition"
    xmlns:cxdf="http://www.bercut.com/spec/schema/ComplexDefinition"
    version="1.0">
    <xsd:import schemaLocation="BusinessCriteriaDefinition.xsd" namespace="http://
www.bercut.com/schema/BusinessCriteriaDefinition"/>
    <xsd:import schemaLocation="BA_EM_CommonDefinitions.xsd" namespace="http://bercut.com/
em/schema/CommonDefinitions"/>
    <xsd:import schemaLocation="ComplexDefinition.xsd" namespace="http://www.bercut.com/
spec/schema/ComplexDefinition"/>
    <xsd:element name="<<ИМЯ СОБЫТИЯ>>" cd:anyType="request">
        <xsd:annotation>
            <xsd:documentation>{Описание события для отображения в NS Back
Office}Описание события</xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="bcd:msisdn"/>
                <xsd:element ref="bcd:serviceAction"/>
                <xsd:element ref="bcd:checkSwitchOn"/>
                <xsd:element ref="bcd:chargeOrder"/>
                <xsd:element name="<параметр события>" type="xsd:string">
                    <xsd:annotation>
                        <xsd:documentation>{Тип события}Тип события</xsd:documentation>
                    </xsd:annotation>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>

```

П 2.4.2.1.2. Создание XSD-файла по шаблону

Вы можете создать XSD-файл интерфейса по шаблону.

Для создания XSD-файла по шаблону:

1. Создайте каталог, в котором будут храниться файлы.
2. Создайте в каталоге новый файл и скопируйте туда [шаблон](#).

3. Задайте пространство имен для данного события в двух местах в блоке:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.bercut.com/action/invoice/subs_serv"
  xmlns:tns="http://www.bercut.com/action/invoice/subs_serv"
  xmlns:cd="http://bercut.com/em/schema/CommonDefinitions"
  xmlns:bcd="http://www.bercut.com/schema/BusinessCriteriaDefinition"
  xmlns:cxdf="http://www.bercut.com/spec/schema/ComplexDefinition"

  version="1.0">
```

Примечание. Название пространства имен должно отличаться для разных групп событий.

4. Задайте основные параметры события в блоке:

```
<xsd:element name="InVoiceSubsServAction" cd:anyType="request">
  <xsd:annotation>
    <xsd:documentation>{Какое-то событие}описание</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="bcd:msisdn"/>
      <xsd:element ref="bcd:serviceAction"/>
      <xsd:element ref="bcd:checkSwitchOn"/>
      <xsd:element ref="bcd:chargeOrder"/>
      <xsd:element name="eventId" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation>{Тип события}Тип события</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

4.1. Укажите имя основного события. Имя события соответствует имени корневого элемента в JSON.

4.2. Убедитесь, что для события задан атрибут `cd:anyType="request"` — это операция (действие), которую пользователь выбирает в веб-интерфейсе Notification System.

4.3. В блоке `<xsd:documentation>` задайте имя события: в фигурных скобках укажите имя события, после — описание.

5. Добавьте в событие параметры. Рекомендуется использовать параметры `element` через `ref`, например: `<xsd:element ref="bcd:msisdn"/>`. Вы также можете определить свои параметры `element` внутри, хотя это делать не рекомендуется.

5.1. Чтобы определить параметр через `ref`, укажите параметр в формате:

```
<xsd:element ref="<параметр события>" />
```

Примечание. Формат параметра события должен быть описан в одном из пространств имен.

5.2. Чтобы определить свой параметр, задайте имя, тип и описание.

- Имя параметра — соответствует параметру сообщения, получаемому из бизнес-процесса.
- Тип параметра — укажите тип параметра, который типу параметра получаемого сообщения.

Примечание. Если формат даты соответствует XSD-формату — например, `2020-04-29T09:58:35.445+0300` — замените тип на `xsd:dateTime`.

- Описание параметра — укажите описание параметра. Оно должно быть понятно для пользователя веб-интерфейса.

```
<xsd:element name="<параметр события>" type="<тип параметра>"
  <xsd:annotation
    <xsd:documentation xml:lang="ru">
      {<параметр события>}<параметр события> Описание параметра</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
```

6. Сохраните итоговый файл.

П 2.4.2.2. Создание JAR-файла

Добавьте в каталог с XSD-файлом другие файлы, после чего создайте JAR-файл.

1. Добавьте в каталог с *созданным* XSD-файлом другие файлы:

- AOI_NS_KafkaProducer.wsdl
- BA_EM_CommonDefinitions.xsd
- BusinessCriteriaDefinition.xsd
- ComplexDefinition.xsd
- NS_KafkaDefinition.xsd
- ProfileDefinition.xsd
- ServiceDefinition.wsdl
- ServiceDefinition.xsd
- SimpleDefinition.xsd
- \META-INF\MANIFEST.MF

Файлы можно взять из любого существующего микросервиса, предназначенного для работы с Apache Kafka.

Эти файлы содержат список заданных ранее критериев, которые вы сможете использовать при настройке параметров события.

Примечание. AOI_NS_KafkaProducer.wsdl должен иметь версию не ниже следующей:

```
<component name="AOI NS KafkaProducer" mainstream="1.2" use-as="wsdl" />
```

Имя	Дата изменения	Тип	Размер
META-INF	19.10.2019 16:33	Папка с файлами	
AOI_NS_KafkaProducer.wsdl	19.10.2019 16:33	Файл "WSDL"	4 КБ
BA_EM_CommonDefinitions.xsd	19.10.2019 16:33	Файл "XSD"	2 КБ
BusinessCriteriaDefinition.xsd	19.10.2019 16:33	Файл "XSD"	87 КБ
ComplexDefinition.xsd	19.10.2019 16:33	Файл "XSD"	18 КБ
InVoiceSubsServAction.xsd	21.04.2021 17:26	Файл "XSD"	2 КБ
NS_KafkaDefinition.xsd	19.10.2019 16:33	Файл "XSD"	2 КБ
ProfileDefinition.xsd	19.10.2019 16:33	Файл "XSD"	37 КБ
ServiceDefinition.wsdl	19.10.2019 16:33	Файл "WSDL"	4 КБ
ServiceDefinition.xsd	19.10.2019 16:33	Файл "XSD"	11 КБ
SimpleDefinition.xsd	19.10.2019 16:33	Файл "XSD"	73 КБ

Рис. 79. Пример содержимого каталога

2. Для создания JAR-файла:

- 2.1. Перейдите в каталог, в котором находится созданный XSD-файл, а также остальные WSDL- и XSD-файлы.
- 2.2. Создайте ZIP-архив.
- 2.3. В сформированном архиве замените расширение .zip на .jar.

П 2.5. События для SA NS HttpProducer

Для создания события:

1. Создайте XSD-файл, описывающий структуру response и request:

```
<xsd:element name="TestRtcmProducerRequest"
type="tns:TestRtcmProducerRequestType" cd:anyType="request">
  <xsd:annotation>
    <xsd:documentation xml:lang="ru">{новости request}описание</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:element name="TestRtcmProducerResponse" type="tns:TestRtcmProducerResponseType"
cd:anyType="response">
  <xsd:annotation>
    <xsd:documentation xml:lang="ru">{новости response}описание</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="TestRtcmProducerRequestType">
  <xsd:sequence>
    <xsd:element name="msisdn" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{msisdn}msisdn</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="trace_test" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{trace_test}trace_test</
xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <!-- xsd:element ref="ctxd:keyValueContext" minOccurs="0"/-->
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TestRtcmProducerResponseType">
  <xsd:sequence>
    <xsd:element name="isOk" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation xml:lang="ru">{isOk}isOk</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

2. Сформируйте JAR-файл микросервиса, содержащий в себе файлы AOI_NS_HttpProducer.wsdl и NS_HttpDefinition.xsd.
3. Создайте файл маппинга для преобразования запроса, отправляемого в RabbitMQ. Например news.producer.xml. Пример содержимого файла:

```
<producer-mapping>
  <rtsib-element>newsRequest</rtsib-element>
  <body-encoding>utf-8</body-encoding>
  <body-format>json</body-format>
  <empty-line>nothing</empty-line>
  <template><![CDATA[
  {
    "idea_id":${idea_id.rawValue},
    "session_store":${session_store},
```

```

    "after":${after},
    "limit":${limit}
  }
]]></template>
</producer-mapping>

```

4. Создайте файл маппинга для преобразования ответа, получаемого от RabbitMQ. Например news.producer-resp.xml. Пример содержимого файла:

```

<producer-mapping>
  <rtsib-element>newsResponse</rtsib-element>
  <detection>
    <type>EXIST</type>
    <expression>$.result</expression>
  </detection>
  <body-encoding>windows-1251</body-encoding>
  <body-format>json</body-format>
  <mapping>
    <param>
      <from>$.mes_title</from>
      <to>mes_title</to>
    </param>
    <param>
      <from>$.mes_body</from>
      <to>mes_body</to>
    </param>
    <param>
      <from>$.mes_date</from>
      <to>mes_date</to>
    </param>
  </mapping>
</producer-mapping>

```

5. Загрузите созданные файлы:

- Созданный JAR-файл микросервиса:

1. На LWSA Container EM BO — в каталог \$LWSA/em_data на сервере или в веб-интерфейс NS Back Office на вкладке **Микросервисы**.
2. На LWSA Container, на котором установлен SA NS Http Producer,— в каталог \$LWSA/em_data на сервере.

- Созданные XML-файлы маппинга — на LWSA Container, на котором установлен SA NsHttpProducer,— в каталог \$LWSA/em_data на сервере.

6. Настройте MIB-параметры SA NsHttpProducer. Перейдите в MIB-группу /LWSAContainer/@artifacts/SA_NsHttpProducer/Core/Configuration/Elements/ и создайте новую подгруппу. Имя подгруппы — имя элемента в теге <rtsib-element> в файле маппинга.

В подгруппе укажите имена созданных XML-файлов в значении MIB-параметров:

- *RequestConversionMappingFile* — название файла маппинга для запроса request для преобразования, должно оканчиваться на .producer.xml. Если не заполнено, то будет срабатывать маппинг по умолчанию.
- *ResponseConversionMappingFile* — название файла маппинга для ответа response для преобразования, должно оканчиваться на .producer-resp.xml. Если не заполнено, то будет срабатывать маппинг по умолчанию.

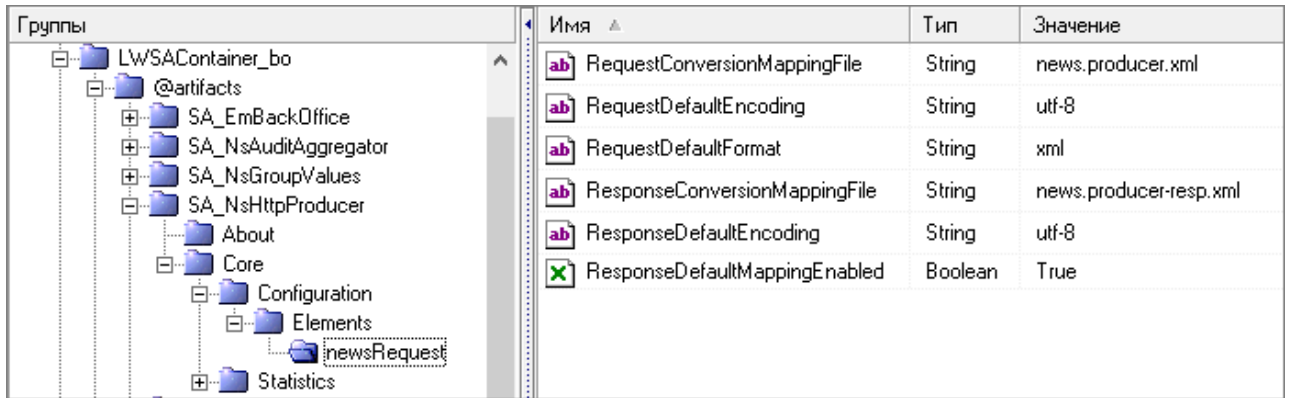


Рис. 80. Вариант настройки SA NsHttpProducer

Приложение 3. Правила преобразования сообщений адаптерами Consumer

Правила преобразования сообщений и обработки ошибок универсальными адаптерами Consumer.

Преобразование используется для компонента SA NS KafkaConsumer (4.6).

Преобразование сообщений делится на два типа:

- преобразование с помощью файла маппинга (П 3.1);
- преобразование плоской структуры (П 3.2).

Результат преобразования — это сообщение, которое имеет структуру, описанную в специальном XSD-файле. Это сообщение является входным сообщением для anyType-микросервиса и помечено атрибутом `anyType="request"`.

Алгоритм преобразования

1. Consumer получает сообщение из определенного топика или очереди.
2. Проверяется значение MIB-параметра `ConversionMappingFiles` в MIB-группе `SA_NsKafkaConsumer/Core/Configuration/Topics/`. Если есть, берет файл маппинга оттуда.
3. Если MIB-параметра `ConversionMappingFiles` нет, берем файл маппинга из MIB-параметра `ConversionMappingFiles` MIB-группы `SA_NsKafkaConsumer/Core/Configuration/`.
4. Если файлы маппинга есть, пытается применить их по очереди в порядке перечисления в значении MIB-параметра.
5. Если найден подходящий файл маппинга, то преобразует сообщение с помощью файла маппинга (П 3.1).
6. Если подходящий файл маппинга не найден или список файлов маппинга был пустой, пытается сделать преобразование плоской структуры (П 3.2).

П 3.1. Преобразование с помощью файла маппинга

Файл маппинга — это XML-файл, соответствующий схеме `consumer-mapping.xsd`. Кодировка файла маппинга UTF-8.

Файл автоматически загрузится при запуске приложения, если название файла оканчивается на `.consumer.xml` и он лежит в каталоге на сервере `../LWSAC/em_data`.

Файлы загружаются в алфавитном порядке. Если какой-то из файлов не удалось загрузить, он будет пропущен, а в трейс сообщений система выведет с соответствующую ошибку.

Чтобы настроить преобразование, создайте файл маппинга и опишите в нем преобразование. Содержимое файла маппинга зависит от входного формата сообщения.

Пример преобразований для форматов:

- XML — П 3.1.1;
- JSON — П 3.1.2.

Структура файла маппинга

- **body-encoding** — кодировка тела сообщения;

- **body-format** — формат тела сообщения xml или json;
- **rtsib-element** — название корневого элемента в XSD-структуре, на которую будет происходить маппинг.
- **detection** — настройка, позволяющая определить, подходит ли входящее сообщение для данного файла маппинга. Если настройки нет, по умолчанию считается, что подходит.
 - **expression** — выражение в формате XPath или jsonPath, в зависимости от body-format, которое указывает на элемент-маркер.
 - **type** — принимает одно из значений:
 - **EXIST** — достаточно наличия элемента-маркера;
 - **VALUE** — система сравнит значение элемента-маркера с ожидаемым.
 - **value** — ожидаемое значение, с которым будет сравниваться значение из элемента-маркера, если в **type** указано VALUE.
- **mapping** — маппинг значений из сообщения с параметрами из XSD-структуры.
 - **param** — маппинг для каждого параметра. Если для параметра нет маппинга, то возможно два варианта:
 1. Параметр не появится в результирующем сообщении;
 2. Параметр появится со значением по-умолчанию, если он является обязательным в описании XSD-структуры.
 - **from** — выражение в формате XPath или jsonPath, в зависимости от body-format. Указывает, откуда нужно выбрать значения из входящего сообщения. Значений может быть несколько. Если параметр является массивом, будут переданы все значения, если нет — только первое значение.
 - **to** — название параметра в XSD-структуре.
 - **to-event-context** — название параметра в блоке EventContext, который передается вместе с основным сообщением в сторону Notification System. Поле не может быть пустым.

i **Примечание.** Должен быть задан один из параметров:

- **to**
- **to-event-context.**

Пример схемы consumer-mapping.xsd до SA NS Kafka Consumer версии 1.4

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="consumer-mapping">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="rtsib-element"/>
        <xs:element name="detection">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="type">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="EXIST"/>
                    <xs:enumeration value="VALUE"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="expression" type="xs:string"/>
              <xs:element name="value" type="xs:string" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element type="xs:string" name="body-encoding"/>
<xs:element type="xs:string" name="body-format"/>
<xs:element name="mapping">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="param" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="from"/>
            <xs:element type="xs:string" name="to"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Пример схемы consumer-mapping.xsd для SA NS Kafka Consumer версии 1.4 и выше

i Примечание. Начиная с версии SA NS Kafka Consumer 1.4 появилась возможность указывать мапинг параметров в параметре `EventContext`.

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="consumer-mapping">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="rtsib-element"/>
        <xs:element name="detection">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="type">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="exist"/>
                    <xs:enumeration value="value"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="expression" type="xs:string"/>
              <xs:element name="value" type="xs:string" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element type="xs:string" name="body-encoding"/>
        <xs:element name="body-format">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="xml"/>
              <xs:enumeration value="json"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="mapping">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="param" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="from"/>
                    <xs:element type="xs:string" name="to"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

```

    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

Пример содержимого файла маппинга до SA NS Kafka Consumer версии 1.4

```

<?xml version="1.0" encoding="UTF-8"?>
<consumer-mapping>
  <rtsib-element>notify-pcrf</rtsib-element>
  <detection>
    <type>VALUE</type>
    <expression>$.consumptionPCRF.eventID</expression>
    <value>9</value>
  </detection>
  <body-encoding>utf-8</body-encoding>
  <body-format>json</body-format>
  <mapping>
    <param>
      <from>$.consumption.eventID</from>
      <to>notifyId</to>
    </param>
  </mapping>
</consumer-mapping>

```

Пример содержимого файла маппинга до SA NS Kafka Consumer версии 1.4 и выше

```

<?xml version="1.0" encoding="UTF-8"?>
<consumer-mapping>
  <rtsib-element>StartRequestAnyParams</rtsib-element>
  <detection>
    <type>EXIST</type>
    <expression>//StartRequestAnyParams</expression>
  </detection>
  <body-encoding>utf-8</body-encoding>
  <body-format>xml</body-format>
  <mapping>
    <param>
      <from>//StartRequestAnyParams/ruleIdStart</from>
      <to>ruleIdStart</to>
    </param>
    <param>
      <from>//trace/requestTime</from>
      <to-event-context>dataTextSingle</to-event-context>
    </param>
  </mapping>
</consumer-mapping>

```

П 3.1.1. XML-преобразование

Пример создания XML-преобразования.

Для преобразования из формата xml используется xPath 1.0 с учетом пространства имен (namespace).

Например, если входящее сообщение выглядит так:

```

<notify-331 xmlns="http://www.bercut.com/events/invoice/subs"
  xmlns:test="http://www.bercut.com/events/invoice/test">
  <msisdn>111</msisdn>
  <test:msisdn>222</test:msisdn>
</notify-331>

```

то:

- xPath вида **//msisdn** не вернет результата.

- XPath вида `//*["msisdn"=local-name()]` вернет результат: `111,222`.

Для проверки XPath вы можете использовать сторонние тестировщики. Например — <https://www.freeformatter.com/xpath-tester.html>.

XML Input

Option 1: Copy-paste your XML document here

```
<notify-331 xmlns="http://www.bercut.com/events/invoice/subs" xmlns:test="http://www.bercut.com/events/invoice/test">
  <msisdn>111</msisdn>
  <test:msisdn>222</test:msisdn>
</notify-331>
```

Option 2: Or upload your XML document

Выберите файл

XPath expression

`//*["msisdn"=local-name()]`

Include the XML item type in output:

Yes

XPath result:

```
Element='<msisdn xmlns="http://www.bercut.com/events/invoice/subs">111</msisdn>'
Element='<test:msisdn xmlns:test="http://www.bercut.com/events/invoice/test">222</test:msisdn>'
```

Рис. 81. Пример проверки XPath

Пример отправляемого сообщения:

Примечание. В отправляемом сообщении нет знаков переноса строки. Они добавлены для наглядности отображения сообщения.

```
<message>
  <StartRequestAnyParams>
    <ruleIdStart>111</ruleIdStart>
    <dataDigit>1</dataDigit>
    <dataDigit>2</dataDigit>
    <dataDigit>3</dataDigit>
  </StartRequestAnyParams>
  <trace>
    <requestTime>2021-03-12T15:37:29.828+0300</requestTime>
    <someId>1</someId>
    <someId>2</someId>
  </trace>
</message>
```

Пример файла маппинга:

```
<consumer-mapping>
  <rtsib-element>StartRequestAnyParams</rtsib-element>
  <detection>
    <type>EXIST</type>
    <expression>//StartRequestAnyParams</expression>
  </detection>
  <body-encoding>utf-8</body-encoding>
  <body-format>xml</body-format>
  <mapping>
    <param>
      <from>//StartRequestAnyParams/ruleIdStart</from>
      <to>ruleIdStart</to>
```

```

    </param>
    <param>
      <from>//trace/requestTime</from>
      <to>dataTextSingle</to>
    </param>
    <param>
      <from>//trace/requestTime</from>
      <to-event-context>dataTextSingle</to-event-context>
    </param>
    <param>
      <from>//StartRequestAnyParams/dataDigit</from>
      <to>dataDigit</to>
    </param>
    <param>
      <from>//StartRequestAnyParams/dataDigit</from>
      <to-event-context>dataDigit</to-event-context>
    </param>
  </mapping>
</consumer-mapping>

```

Пример получаемого сообщения:

```

<ber-ns0:kafkaRequest>
<ber-ns0:Header>
  <ber-ns0:kafkaTopicName></ber-ns0:kafkaTopicName>
</ber-ns0:Header>
<ber-ns0:Body>
  <ber-ns2:StartRequestAnyParams>
    <ber-ns1:ruleIdStart>111</ber-ns1:ruleIdStart>
    <ber-ns1:dataDigit>1</ber-ns1:dataDigit>
    <ber-ns1:dataDigit>2</ber-ns1:dataDigit>
    <ber-ns1:dataDigit>3</ber-ns1:dataDigit>
    <ber-ns1:dataTextSingle>2021-03-12T15:37:29.828+0300</ber-ns1:dataTextSingle>
  </ber-ns2:StartRequestAnyParams>
</ber-ns0:Body>
<ber-ns0:EventContext>
  <ber-ns3:eventContextParameter>
    <ber-ns3:name>dataTextSingle</ber-ns3:name>
    <ber-ns3:value>2021-03-12T15:37:29.828+0300</ber-ns3:value>
  </ber-ns3:eventContextParameter>
  <ber-ns3:eventContextParameter>
    <ber-ns3:name>dataDigit</ber-ns3:name>
    <ber-ns3:value>1</ber-ns3:value>
    <ber-ns3:value>2</ber-ns3:value>
    <ber-ns3:value>3</ber-ns3:value>
  </ber-ns3:eventContextParameter>
</ber-ns0:EventContext>
</ber-ns0:kafkaRequest>

```

П 3.1.2. JSON-преобразование

Пример создания JSON-преобразования.

Для json используется jsonPath реализация от JayWay с включенной опцией **Always return result list** — всегда возвращать список. Для проверки jsonpath вы можете использовать, например, стороннюю утилиту — <https://jsonpath.herokuapp.com/>.

Пример отправляемого сообщения:

i Примечание. В отправляемом сообщении нет знаков переноса строки. Они добавлены для наглядности отображения сообщения.

```

{
  "StartRequestAnyParams": {
    "ruleIdStart": "test_rule",
    "dataDigit": [
      1,
      2,
      3
    ]
  }
}

```

```

    },
    "trace":{
      "requestTime":"2021-03-12T15:37:29.828+0300",
      "someId":[
        1,
        2,
        3
      ],
      "requestId":"b596e9c0-832f-11eb-8b28-005056b0dde0",
      "notifyId":"notify-88"
    }
  }
}

```

Пример файла маппинга:

```

<consumer-mapping>
  <rtsib-element>StartRequestAnyParams</rtsib-element>
  <detection>
    <type>EXIST</type>
    <expression>$.StartRequestAnyParams</expression>
  </detection>
  <body-encoding>utf-8</body-encoding>
  <body-format>json</body-format>
  <mapping>
    <param>
      <from>$.StartRequestAnyParams.ruleIdStart</from>
      <to>ruleIdStart</to>
    </param>
    <param>
      <from>$.trace.requestTime</from>
      <to>dataTextSingle</to>
    </param>
    <param>
      <from>$.trace.requestTime</from>
      <to-event-context>dataTextSingle</to-event-context>
    </param>
    <param>
      <from>$.StartRequestAnyParams.dataDigit.*</from>
      <to>dataDigit</to>
    </param>
    <param>
      <from>$.StartRequestAnyParams.dataDigit.*</from>
      <to-event-context>dataDigit</to-event-context>
    </param>
  </mapping>
</consumer-mapping>

```

Для обработки параметра-массива укажите:

```

<mapping>
  <param>
    <from>$.msisdn.*</from>
    <to>msisdn</to>
  </param>
</mapping>

```

Пример получаемого сообщения:

```

<ber-ns0:kafkaRequest>
  <ber-ns0:Header>
    <ber-ns0:kafkaTopicName></ber-ns0:kafkaTopicName>
  </ber-ns0:Header>
  <ber-ns0:Body>
    <ber-ns2:StartRequestAnyParams>
      <ber-ns1:ruleIdStart>test_rule</ber-ns1:ruleIdStart>
      <ber-ns1:dataDigit>1</ber-ns1:dataDigit>
      <ber-ns1:dataDigit>2</ber-ns1:dataDigit>
      <ber-ns1:dataDigit>3</ber-ns1:dataDigit>
      <ber-ns1:dataTextSingle>2021-03-12T15:37:29.828+0300</ber-ns1:dataTextSingle>
    </ber-ns2:StartRequestAnyParams>
  </ber-ns0:Body>
  <ber-ns0:EventContext>
    <ber-ns3:eventContextParameter>

```

```
<ber-ns3:name>dataTextSingle</ber-ns3:name>
  <ber-ns3:value>2021-03-12T15:37:29.828+0300</ber-ns3:value>
</ber-ns3:eventContextParameter>
<ber-ns3:eventContextParameter>
  <ber-ns3:name>dataDigit</ber-ns3:name>
  <ber-ns3:value>1</ber-ns3:value>
  <ber-ns3:value>2</ber-ns3:value>
  <ber-ns3:value>3</ber-ns3:value>
</ber-ns3:eventContextParameter>
</ber-ns0:EventContext>
</ber-ns0:kafkaRequest>
```

П 3.2. Преобразование с помощью плоской структуры

Преобразование используется по умолчанию, если не найдено файлов маппинга.

Алгоритм преобразования

1. Система декодирует полученное сообщение, используя значение MIB-параметра *DefaultEncoding*.
2. Ищет в сообщении первый символ { или <.
3. Если символ не найден, выводит сообщение в трейс о невозможности преобразования.
4. Если символ найден, соответственно дальше выполняется XML- или JSON-парсинг, в зависимости от найденного символа.
5. Если каких-то параметров в сообщении не хватает, чтобы заполнить XSD-структуру, возможны два варианта:
 - если параметры не являются обязательными, они пропускаются;
 - если параметры являются обязательными, они заполняются значениями по-умолчанию.

Система ожидает, что после преобразования сообщение будет иметь плоскую структуру, совпадающую с xsd-структурой результирующего сообщения.

Приложение 4. Правила преобразования сообщений адаптерами Producer

Правила преобразования сообщений и обработки ошибок универсальными адаптерами Producer.

Преобразование используется для компонентов SA NS KafkaProducer (4.6), SA NS HttpProducer (4.7), SA NS RabbitMQProducer (4.8).

Преобразование сообщений делится на два типа:

- преобразование с помощью файла маппинга (П 4.1);
- преобразование плоской структуры (П 4.2).

П 4.1. Преобразование с помощью файла маппинга

Файл маппинга — это XML-файл, соответствующий схеме `producer-mapping.xsd`. Кодировка файла маппинга — UTF-8.

Файл автоматически загрузится при запуске приложения, если название файла оканчивается на `.producer.xml` и он лежит в каталоге на сервере `../LWSAC/em_data`.

Файлы загружаются в алфавитном порядке. Если какой-то из файлов не удалось загрузить, он будет пропущен, а в трейсе сообщений система выведет с соответствующую ошибку.

Чтобы настроить преобразование, создайте файл маппинга и опишите в нем преобразование. Содержимое файла маппинга зависит от входного формата сообщения.

Структура файла маппинга

- **body-encoding** — кодировка тела сообщения;
- **body-format** — формат тела сообщения xml или json;
- **rtsib-element** — название корневого элемента в XSD-структуре, который будет использоваться в маппинге.
- **template** — шаблон Freemarker со специальными функциями для удобного маппинга.

Описание синтаксиса шаблона Freemarker для форматов:

- XML — П 4.1.1.

Пример содержимого файла маппинга для формата XML:

```
<producer-mapping>
  <rtsib-element>notify-331</rtsib-element >
  <body-encoding>utf-8</body-encoding>
  <body-format>xml</body-format>
  <template><![CDATA[
<notify-331 xmlns="http://www.bercut.com/events/invoice/subs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <msisdn>${msisdn}</msisdn>
  <eventID>331</eventID>
  <timeStamp>${timeStamp.rawValue?iso_utc}</timeStamp>
  <statusId>${statusId.default("1")}</statusId>
  ${statusIdOld.tag("statusIdOld")}
  ${serviceId.array("serviceId")}
  <branchId>${branchId}_${clntId}</branchId>
</notify-331>
  ]]></template>
```

```
</producer-mapping>
```

▪ JSON — П 4.1.2.

Пример содержимого файла маппинга для формата JSON:

```
<producer-mapping>
  <rtsib-element>InVoiceSubsServAction</rtsib-element >
  <body-encoding>utf-8</body-encoding>
  <body-format>json</body-format>
  <template><![CDATA[
  {
    "type": "simple-event",
    "service_id": ${servId},
    "service_id": ${servId.asNumber},
    "service_action": ${serviceAction},
    ${serviceAction.key("serviceAction")}
    ${clntLink.key("clntLink")}
    ${clntId.key("clntList").withoutComma}
  }
  ]]></template>
</producer-mapping>
```

▪ plain text — П 4.1.3.

Пример схемы producer-mapping.xsd

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="producer-mapping">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="rtsib-element"/>
        <xs:element type="xs:string" name="body-encoding"/>
        <xs:element name="body-format">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="xml"/>
              <xs:enumeration value="json"/>
              <xs:enumeration value="plain"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element type="xs:string" name="template"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

П 4.1.1. Синтаксис XML-преобразования

Синтаксис шаблонов для FreeMarker преобразования.

#	Синтаксис	Описание
1.1	<code>\${param}</code>	<p>Возвращает значение обязательного параметра. Осуществляется экранирование согласно стандарту XML.</p> <p>Пример:</p> <pre><value>\${param1}</value></pre> <p>Результат:</p> <pre><value>VALUE</value></pre>

#	Синтаксис	Описание
		<p>Если <i>param</i> — массив, возвращает значения через запятую. Пример: 1,2,3</p> <p>Если <i>param</i> отсутствует — например, если является необязательным параметров — возвращает пустое значение. В этом случае это ошибкой не является.</p>
1.2	<code>#{param.rawValue}</code>	<p>Возвращает значение в типе FreeMarker (string, number, date, array). К данному значению могут быть применены стандартные модификаторы "?".</p> <p>Значение в объекте FreeMarker будет экранированным по спецсимволам в соответствии с форматом — XML или JSON.</p>
1.3	<code>#{param.attributeValue}</code>	<p>Возвращает значение обязательного параметра как атрибут. Осуществляется экранирование согласно стандарту XML.</p> <p>Пример:</p> <pre><value color=#{param1.attributeValue} /> <value color="red" /></pre> <p>Если <i>param</i> — массив, возвращает значения через запятую. Пример: "1,2,3".</p> <p>Если <i>param</i> отсутствует — например, если является необязательным параметров — две двойные кавычки, внутри которых пустое значение: "".</p>
1.4	<code>#{param.tag(...)}</code>	<p>Возвращает значение и сам тег.</p> <p>Если <i>param</i> — массив, возвращает значения через запятую. Пример: "1,2,3".</p> <p>Если критерия нет, то не возвращает ничего. Пустая строка — отсутствие критерия.</p> <p>Пример:</p> <pre>#{param1.tag("value")} <value>VALUE</value></pre>
1.5	<code>#{param.attribute(...)}</code>	<p>Возвращает как атрибут.</p> <p>Если <i>param1</i> — массив, возвращает значения через запятую. Пример: "1,2,3".</p> <p>Если критерия нет, то не возвращает ничего. Пустая строка — отсутствие критерия.</p> <p>Пример:</p> <pre><value #{param1.attribute("color")} /> <value color="RED" /></pre>
1.6	<code>#{param.array(...)}</code>	<p>Возвращает как список.</p> <p>Если список пустой или критерия нет — не возвращает ничего.</p> <p>Пример:</p>

#	Синтаксис	Описание
		<pre> <code>\${param1.array("color")}</code> <color>RED</color> <color>BLUE</color> <color>BLACK</color> </pre>
1.7	<code>\${param.default(...)}</code>	<p>Если критерий — массив и он пустой — возвращает значение по умолчанию.</p> <p>Если критерия нет или пустая строка — возвращает значение по умолчанию.</p> <p>Пример:</p> <pre> <value>\${param1.default("NOTHING")}</value> <value color=\${param1.attributeValue.default("BLACK")} /> <value>NOTHING</value> <value color="BLACK" /> </pre>

⊘ Ограничение. Если для *array*, *tag*, *attribute* нужна более сложная логика по модификации значения, пишется логика на *Freemarker*

П 4.1.2. Синтаксис JSON-преобразования

Синтаксис шаблонов для *FreeMarker* преобразования.

#	Синтаксис	Описание
2.1	<code>\${param}</code>	<p>Возвращает значение параметра.</p> <p>Если не строковой параметр и он отсутствует — возвращает NULL.</p> <p>Если строковый параметр:</p> <ul style="list-style-type: none"> если строковый параметр обязательный, то параметр не может быть NULL. Возвращается значение, либо пустая строка. если строковый параметр опциональный и не указан, то возвращается NULL. если строковый параметр опциональный и указан пустым, то возвращается пустая строка. <p>Если параметр имеет числовой тип — возвращает значение без "".</p> <p>Если параметр имеет тип <code>boolean</code> — возвращает значение <code>True</code> или <code>False</code>.</p> <p>Если параметр имеет тип массив, с учетом вышеуказанного — возвращает как массив.</p> <p>Для строк выполняется экранирование согласно стандарту JSON.</p> <p>Пример:</p> <pre> { </pre>

#	Синтаксис	Описание
		<pre>"id": \${id}, "author": \${author} }</pre>
2.2	<code>\${param.rawValue}</code>	Возвращает значение в типе FreeMarker — string, number, date или array. К данному значению могут быть применены стандартные модификаторы "?".
2.3	<code>\${param.asString}</code>	<p>Всегда возвращает значение как строку, в двойных кавычках. Если параметр отсутствует или пустой — возвращает "".</p> <p>Если параметр — массив, то соединит значения через запятую в одну строку и добавит квадратные скобки. Пример: ["a","b","c"].</p>
2.4	<code>\${param.asNumber}</code>	<p>Возвращает значение как число, без двойных кавычек. Если параметр отсутствует — возвращает NULL.</p> <p>Если значение содержит символы отличные от 0-9 — происходит ошибка.</p> <p>Если параметр — массив, то соединит значения через запятую в одну строку и добавит квадратные скобки. Пример: [1,2,3].</p>
2.5	<code>\${param.asBoolean}</code>	<p>Возвращает значение как True или False. Если параметр отсутствует — возвращает NULL.</p> <p>Если строка yes, число не равно 0, то возвращает значение True.</p> <p>Если строка no или число равно 0, то возвращает значение False.</p> <p>Если параметр — массив, соединит значения через запятую в одну строку и добавит квадратные скобки. Пример: [true,false,true].</p>
2.6	<code>\${param.key(...)}</code>	<p>Вывод key/value с проверкой наличия параметра.</p> <p>Если параметр отсутствует — равен NULL — ничего не выводит. Если параметр присутствует, то выводит key/value + ','</p> <p>Пример:</p> <pre>{ "id": \${id}, \${authorName.key("author")} "desc": \${desc} }</pre> <p>Результат:</p> <pre>{ "id": 123, "author": "Name", "desc": "...." }</pre> <pre>{ "id": 123,</pre>

#	Синтаксис	Описание
		<pre>"desc": "...."</pre> <p>}</p> <p>Может использоваться с <code>asString</code>, <code>asNumber</code>, <code>asBoolean</code>.</p> <p>Пример:</p> <pre><code>\${msisdn.asNumber.key("contact")}</code></pre> <p>Результат, при <code>\${msisdn} = "123"</code>:</p> <pre>contact: 123</pre> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>! Внимание! Нельзя использовать в другом порядке <code><code>\${msisdn.key("contact").asNumber}</code></code></p> </div>
2.7	<code><code>\${param.key(...).withoutComma}</code></code>	<p>Модификатор для <code>data</code>. Указывает, что запятую в конце добавлять не надо.</p> <p>Пример:</p> <pre>{ \${authorName.key("author").withoutComma} }</pre> <p>Результат:</p> <pre>{ "author": "Name" }</pre>
2.8	<code><code>\${param.key(...).beforeComma}</code></code>	<p>Модификатор для <code>data</code>. Указывает, что запятую надо добавлять вначале.</p> <p>Пример:</p> <pre>{ "id": \${id} \${authorName.data("author").beforeComma} }</pre> <p>Результат:</p> <pre>{ "id": 123 ,"author": "Name" }</pre>
2.9	<code><code>\${param.default(...)}</code></code>	<p>Если параметр отсутствует является пустой строкой. Возвращает значение по умолчанию.</p> <p>Пример:</p> <pre>{ "id": \${id.default("777")} }</pre> <p>Результат:</p> <pre>{ "id": 777 }</pre>
2.10	<code><code>\${param.emptyAsNull}</code></code>	<p>Если значение — пустая строка "" или NULL — возвращает его как NULL.</p>

#	Синтаксис	Описание
		Пример: <pre>{ "id": \${id.emptyAsNull}" }</pre> Результат: <pre>{ "id": null }</pre>
2.11	<code>\${param.withoutQuote}</code>	Возвращает значение без обрамляющих двойных кавычек. Пример: <pre>{ "id": "\${id.withoutQuote}- \${name.withoutQuote}" }</pre> Результат: <pre>{ "id": "34-MEGA" }</pre>

П 4.1.3. Синтаксис преобразования в формат plain text

Синтаксис шаблонов для FreeMarker преобразования.

#	Синтаксис	Описание
3.1	<code>\${param}</code>	Возвращает значение параметра, без какого либо экранирования.
3.2	<code>\${param.rawValue}</code>	Возвращает значение в типе FreeMarker — string, number, date или array. К данному значению могут быть применены стандартные модификаторы "?".

П 4.2. Преобразование с помощью плоской структуры

Преобразование используется по умолчанию, если не найдено файлов маппинга.

Алгоритм преобразования

- Система выбирает формат для преобразования: XML или JSON, в зависимости от значения MIB-параметра *DefaultFormat*.
- Система выполняет преобразование в соответствии с выбранным форматом:
 - Для XML-преобразования: убирает пространства имен.
 - Для JSON-преобразования: преобразует параметры в соответствии с типами, описанными в XSD-файле.
- Если каких-то параметров в сообщении не хватает, чтобы заполнить XSD-структуру, возможны два варианта:
 - если параметры не являются обязательными, они пропускаются;

- если параметры являются обязательными, они заполняются значениями по-умолчанию.

4. Система кодирует сообщение в соответствии со значением MIB-параметра *DefaultEncoding*.

Система ожидает, что после преобразования сообщение будет иметь плоскую структуру, совпадающую с XSD-структурой результирующего сообщения.

Пример файлов для преобразования:

Входящее сообщение от NsSLES

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:ber="http://bercut.com/"
xmlns:ber-addr="http://bercut.com/addressing"
xmlns:ber-pc="http://bercut.com/principalcontext">
    <ber-addr:To>rtsib://<IP-address>:<port></ber-addr:To>
    <ber-addr:From>
      <ber-addr:Address>rtsib://<IP-address>:<port>/<Session ID></ber-addr:Address>
    </ber-addr:From>
    <ber-pc:PrincipalContext>
      <ber-pc:SessionPrincipal>5c02ba1c-b028-4a3c-9d48-1dfd07bec1bb</ber-
pc:SessionPrincipal>
    </ber-pc:PrincipalContext>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:ber-ns0="http://www.bercut.com/schema/
BusinessCriteriaDefinitionTest"
xmlns:ber-ns1="http://www.bercut.com/schema/ns/KafkaDefinition"
xmlns:ber-ns2="http://com.bercut/anytype/ANY_StartInterface">
    <ber-ns1:kafkaRequest>
      <ber-ns1:Body>
        <ber-ns2:StartRequestAnyParams>
          <ber-ns0:ruleIdStart>str</ber-ns0:ruleIdStart>
          <ber-ns0:dataDigit>1</ber-ns0:dataDigit>
          <ber-ns0:dataDigit>2</ber-ns0:dataDigit>
          <ber-ns0:dataDigit>3</ber-ns0:dataDigit>
          <ber-ns0:dataDigitSingle>100</ber-ns0:dataDigitSingle>
          <ber-ns0:dataText>a</ber-ns0:dataText>
          <ber-ns0:dataText>b</ber-ns0:dataText>
          <ber-ns0:dataText>c</ber-ns0:dataText>
          <ber-ns0:dataTextSingle>abc</ber-ns0:dataTextSingle>
        </ber-ns2:StartRequestAnyParams>
      </ber-ns1:Body>
    </ber-ns1:kafkaRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

XML-преобразование

Результат XML-преобразования:

```
<StartRequestAnyParams>
  <dataDigit>1</dataDigit>
  <dataDigit>2</dataDigit>
  <dataDigit>3</dataDigit>
  <dataText>a</dataText>
  <dataText>b</dataText>
  <dataText>c</dataText>
  <ruleIdStart>str</ruleIdStart>
  <dataDigitSingle>100</dataDigitSingle>
  <dataTextSingle>abc</dataTextSingle>
</StartRequestAnyParams>
```

JSON-преобразование

При JSON-преобразовании дополнительно отправляются поля:

- -xmlns — пространство имен входящего корневого элемента;

- trace — служебная информация:
 - requestTime — время запроса;
 - notifyId — источник нотификации;
 - requestId — идентификатор сообщения.

Результат JSON-преобразования:

```
{
  "StartRequestAnyParams": {
    "-xmlns": "http://com.bercut/anytype/ANY_StartInterface",
    "dataDigit": [
      1,
      2,
      3
    ],
    "dataText": [
      "a",
      "b",
      "c"
    ],
    "ruleIdStart": "str",
    "dataDigitSingle": 100,
    "dataTextSingle": "abc"
  },
  "trace": {
    "requestTime": "2020-04-29T09:58:35.445+0300",
    "requestId": "2fc05271-895a-43c2-8380-87a6476c5fdd",
    "notifyId": "StartRequestAnyParams"
  }
}
```

Приложение 5. Реализация обработки событий

Основные артефакты, которые необходимо разработать для реализации обработки событий:

- критерии (П 5.1);
- интерфейс, через который осуществляется получение событий (П 5.2);
- микросервисы, возвращающие значения критериев (П 5.3), и справочники значений для них (П 5.4);
- микросервисы, осуществляющие выполнение действий (П 5.5).

П 5.1. Критерии

Критерии — артефакты, которые используются в интерфейсах.

Критерий характеризуется наименованием, отражающим предметную область, и описанием. Может быть получен с помощью вызова микросервиса — сущности, подключенной по RTSIB и реализующей интерфейс.

Критерии используются в интерфейсах обработки событий и в интерфейсах микросервисов. Один критерий может возвращать разные микросервисы, как и вызов одного микросервиса может вернуть разные критерии.

Значение критерия зависит от аргументов или других критериев.

Пример.

PortType SimplePortType содержит метод getClientType, который на вход получает значение MSISDN, а на выходе возвращает значение ClientType. В данном примере критерий ClientType зависит от критерия MSISDN.

Таблица 7. Примеры критериев

Название БК	Зависимость	Описание
Номер	IMSI	Возвращает MSISDN по IMSI.
Регион абонента	Номер телефона	Возвращает домашний регион абонента по MSISDN.
Регион абонента	IMSI	Возвращает домашний регион абонента по IMSI.
Регион нахождения абонента	VLR	Возвращает регион абонента, в котором он зарегистрировался.
Тип клиента	Номер телефона	Возвращает тип (B2B, B2C) абонента по MSISDN.

Здесь значение критерия «Регион абонента» можно получить по номеру телефона или по IMSI, исходя из того, что доступно в данном сценарии.

Один микросервис может вернуть несколько критериев. Пример MS:

Таблица 8. Пример возвращения микросервисом нескольких критериев

Метод MS для получения	Возвращает бизнес-критерий
getByImsi(IMSI)	MSISDN Регион абонента
getByMsisdn(MSISDN)	IMSI
getRegion(MSISDN)	Регион абонента
InVoiceGetSubscriberInfo(Номер)	Тип клиента Модель расчета абонента
WelcomeSMS(Номер, VLR)	Регион нахождения абонента

Критерий может быть как простым типом (число, строка, дата/время, boolean), так и массивом простых типов, но не может быть сложным типом данных.

Критерий может возвращать значения из справочника. Все критерии находятся в файле BusinessCriteriaDefinition.xsd. Добавление новых бизнес-критериев осуществляется с помощью увеличения версии в атрибуте version.

П 5.1.1. Требования к выделению критериев

Основные принципы выделения критериев:

1. Критерий должен точно и однозначно отражать бизнес-потребность конечного пользователя и предметную область.

Правильный критерий: Тип клиента. Неправильный критерий: CLNT_ID.

Например, для определения льготной зоны роуминга не следует использовать маску VLR. Необходим критерий, который будет содержать в названии именованные зоны роуминга. Таким образом, вместо «VLR — '7953128*'» используйте «Зона роуминга — Сибирь».

2. Наименование критерия должно быть лаконично и понятно.
3. Описание критерия должно уточнять наименование и содержать примеры возможных значений критериев.
4. Один критерий не должен использоваться для разных целей и иметь разную смысловую нагрузку. Пример:

- «Баланс (Номер)» — баланс абонента;
- «Баланс (Название компании)» — остаток средств.

В общем случае значение критерия — это идентификатор. Например:

- критерий «Тип абонента» — '2';
- критерий «Филиал» — '78';
- критерий «Подключенные услуги» — '125', '425', '523'.

Соответствия критериев и идентификаторов приведены в справочниках, которые позволяют выбрать значение из выпадающих списков.

Критерии могут быть сгруппированы для упрощения поиска. При этом один критерий может входить в разные группы: например, АБОНЕНТ, УСЛУГИ, РОУМИНГ.

П 5.1.2. Требования к описанию критериев

Критерии добавляются в файл `BusinessCriteriaDefinition.xsd` или могут быть определены в отдельных файлах XSD или WSDL.

Тип критерия наследуется из файла `SimpleDefinition.xsd`. Например:

type — 'sd:DigitsId'.

Добавление нового критерия увеличивает версию файла `BusinessCriteriaDefinition.xsd` (атрибут `version`).

Описание критерия задается в файле `BusinessCriteriaDefinition.xsd`, описание не должно уточняться в конечном интерфейсе.

Указывается короткое наименование критерия, которое задается в фигурных скобках '{' и '}' в теге `annotation`. Это временный синтаксис, впоследствии короткое имя становится атрибутом.

Короткое наименование критерия начинается с заглавной буквы. Названия критерия должны быть на русском языке. По возможности минимизируйте использование аббревиатур.

Название в теге `element` должно начинаться со строчной буквы с использованием стиля `CamelCase`. Пример: `branchId`, `externalMsisdn`.

Пример описания критерия в файле `BusinessCriteriaDefinition.xsd`:

```
<xsd:element name="branchId" type="sd:DigitsId">
  <xsd:annotation>
    <xsd:documentation xml:lang="ru">{Филиал}Цифровой идентификатор для филиала оператора</
xsd:documentation>
    <xsd:documentation xml:lang="en">{Branch}This a digits identifier of operators branch</
xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

Пример использования критерия в интерфейсе:

```
<xsd:element ref="bcd:msisdn"/>
```

Для критериев обязателен следующий формат: {<краткое описание>}<полное описание>.

Рекомендуется делать описание на русском и английском языках.

П 5.2. Обработка событий

Для каждый тип событий реализуется отдельный интерфейс — `portType`.

Общего принципа выделения интерфейсов нет.

События рекомендуется группировать в интерфейсы, максимально приближенные к бизнес-потребностям. Например: события по списанию абонентской платы, по регистрации в роуминге.

Интерфейс представляет из себя WSDL-описание `portType`, где одно событие соответствует одной операции этого `portType`.

Входные и выходные параметры являются критериями и описаны в файле `BusinessCriteriaDefinition.xsd`. Операция может генерировать два вида исключений ([П 5.5.1](#)).

П 5.3. Получение значений критериев (Getter)

Микросервисы получения значения критериев возвращают один или несколько критериев в зависимости от переданных параметров.

Основная функция микросервисов — возвращать критерии в зависимости от переданных параметров. Один микросервис может вернуть несколько критериев: это зависит от реализации. Например, если микросервис получает профиль абонента, то возвращаются все критерии этого профиля. При использовании в правиле нескольких критериев система вызывает микросервис один раз.

В контексте одного правила не должно быть двух идентичных обращений к внешней системе: одна операция получает всю информацию. При вызове операций микросервисов изменения в исходную систему не вносятся. Множественные вызовы одних и тех же операций для одних и тех же данных должны каждый раз возвращать идентичные результаты. Любому критерию числового типа может соответствовать справочник значений, который используется для визуального отображения значения критериев и для макроподстановок. Пример критерия: тип абонента со значениями:

- 1 — бесконтрактный;
- 2 — средний бизнес;
- 3 — ...

Пример WSDL:

```
<definitions name="AOI_OsTypes" ... >
  <types>
    <xsd:schema ... >
      <xsd:complexType name="GetOsTypeRequestParams">
        <xsd:sequence>
          <xsd:element ref="bcdef:msisdn"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="GetOsTypeResponseParams">
        <xsd:sequence>
          <xsd:element ref="bcdef:osId"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:element name="getOsTypeRequestParams" type="tns:GetOsTypeRequestParams"></xsd:element>
      <xsd:element name="getOsTypeResponseParams" type="tns:GetOsTypeResponseParams"></xsd:element>
    </xsd:schema>
  </types>
  <message name="GetOsTypeRequest">
    <part name="params" element="tns:getOsTypeRequestParams"/>
  </message>
  <message name="GetOsTypeResponse">
    <part name="params" element="tns:getOsTypeResponseParams"/>
  </message>
  <message name="SystemFault">
    <part name="fault" element="nsl:systemFault"/>
  </message>
  <portType name="EM_OsTypes">
    <operation name="getOsType">
      <documentation>Получение модели устройства[GETTER]</documentation>
      <input name="getOsTypeRequest" message="tns:GetOsTypeRequest"/>
      <output name="getOsTypeResponse" message="tns:GetOsTypeResponse"/>
      <fault name="systemFault" message="tns:SystemFault"/>
    </operation>
  </portType>
</definitions>
```

Для операции (тег *operation*) должно быть задано описание через тег *documentation*: {<краткое описание>[GETTER]}<полное описание>, где:

- краткое описание — описание конкретной операции, будет отображено в интерфейсе.

- полное описание — развернутое описание операции.

Теги *message* и *element*, используемые в качестве входных и выходных критериев, должны быть описаны соответствующим образом. Входные и выходные критерии могут являться обязательными и опциональными полями. Описание интерфейса должно отражать реальное поведение системы.

П 5.4. Справочники

Для критерия создается справочник значений, если микросервис возвращает идентификатор в качестве значения критерия.

Если микросервис возвращает в качестве значения критерия идентификатор, то для этого критерия реализуется справочник значений: операция в микросервисе.

В одном микросервисе могут быть реализованы несколько методов, которые возвращают справочник. Метод получает на вход и возвращает типы **GetDictionaryRequestParams** и **GetDictionaryResponseParams**, которые описаны в файле **BA_EM_CommonDictionary.xsd** компонента **BA EM CommonDictionary**.

Пример WSDL:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="AOI_OsTypes" targetNamespace="http://www.bercut.com/wsdl/ms/EM_OsTypes"
...>
  <types>
    <xsd:schema ...>
      <xsd:import schemaLocation="BA_EM_CommonDictionary.xsd" namespace="http://bercut.com/em/schema/dto/commonDictionary"/>
      <xsd:import namespace="http://www.bercut.com/spec/schema/ServiceDefinition"
schemaLocation="ServiceDefinition.xsd"/>
      <xsd:element name="getOsTypesDictionaryRequestParams"
type="dictdefs:GetDictionaryRequestParams"></xsd:element>
      <xsd:element name="getOsTypesDictionaryResponseParams"
type="dictdefs:GetDictionaryResponseParams"></xsd:element>
    </xsd:schema>
  </types>
  <message name="GetOsTypesDictionaryRequest">
    <part name="params" element="tns:getOsTypesDictionaryRequestParams"/>
  </message>
  <message name="GetOsTypesDictionaryResponse">
    <part name="params" element="tns:getOsTypesDictionaryResponseParams"/>
  </message>
  <message name="DictionaryNotFound">
    <part name="fault" element="ns0:getDictionaryFault"/>
  </message>
  <portType name="EM_OsTypes">
    <operation name="getOsTypesDictionary">
      <input name="getOsTypesDictionaryRequest" message="tns:GetOsTypesDictionaryRequest"/>
      <output name="getOsTypesDictionaryResponse"
message="tns:GetOsTypesDictionaryResponse"/>
      <fault name="fault" message="tns:DictionaryNotFound"/>
    </operation>
  </portType>
```

Операция, которая реализует справочник, возвращает имя критерия с помощью заполнения поля **parameterName** типа **GetDictionaryResponseParams** и список его возможных значений. Вызов метода выполняется периодически в зависимости от настроек: значение по умолчанию (мин.) — '15'. Полученные значения кешируются и используются в дальнейшем.

Если по какой-то причине значения справочника не были получены, микросервис возвращает исключение *DictionaryNotFound* с указанием причины.

П 5.5. Действия (Action)

Микросервисы действия вызываются системой при выполнении операции, которые входят в состав действия.

В результате выполнения правил система принимает решение о необходимости совершить действие и вызывает для выполнения его операций микросервис. Микросервисы действия используют те же принципы, что и микросервисы определения значения критериев. Входные и выходные параметры являются критериями. Возможен возврат микросервисом двух видов исключений ([П 5.5.1](#)).

В отличие от микросервисов определения значения критериев, микросервисы действия оказывают влияние на внешнюю систему. Примеры действий: отправка SMS-сообщения, подключение услуги.

Пример описания WSDL для действий:

```
<portType name="EM_SendSms">
  <operation name="sendSMS">
    <documentation>{Отправка SMS[ACTION]} Метод позволяет выполнить отпавку SMS с учетом
    логики antispam в Multilink </documentation>
    <input name="sendSMSRequest" message="tns:SendSMSRequest"/>
    <output name="sendSMSResponse" message="tns:SendSMSResponse"/>
    <fault name="businessFault" message="tns:BusinessFault"/>
    <fault name="systemFault" message="tns:SystemFault"/>
  </operation>
</portType>
```

Для операции (тег *operation*) должно быть задано описание через тег *documentation*: {<краткое описание>[ACTION]}<полное описание>, где:

- краткое описание — описание конкретной операции, будет отображено в интерфейсе.
- полное описание — развернутое описание операции.

Теги *message* и *element*, используемые в качестве входных и выходных критериев, должны быть описаны соответствующим образом.

П 5.5.1. Исключения

В результате сбоя при выполнении действия в процессе вызова микросервиса могут быть сгенерированы два вида исключений.

В процессе выполнения действия при вызове микросервиса возможна ситуация сбоя в работе. Операция микросервиса генерирует два вида исключений:

- *businessFault* — произошла ошибка получения данного критерия. Все правила, которые его используют, не могут быть применены.
- *systemFault* — произошла системная ошибка. Необходимо прекратить выполнение всего сценария.

П 5.6. Макроподстановки

В аргументах вызовов микросервисов могут быть использованы макроподстановки. Макроподстановками являются любые критерии. Если для критерия существует справочник, то в качестве значения используется значение критерия из справочника.

П 5.7. Реализация микросервисов

Микросервис может быть реализован любым доступным способом в рамках Platform v3. В большинстве случаев для этого достаточно бизнес-процесса. Один микросервис содержит один или несколько методов получения критериев, а также несколько методов получения справочников.

Пример:

```
<portType name="EM_Accounts">
  <operation name="getClientInfoByMsisdn">
    <documentation>{Получение клиентской информации[GETTER]}</documentation>
    <input name="getClientInfoByMsisdnRequest"
message="tns:getClientInfoByMsisdnRequest"/>
    <output name="getClientInfoByMsisdnResponse"
message="tns:getClientInfoByMsisdnResponse"/>
    <fault name="BusinessFault" message="tns:BusinessFault"/>
  </operation>
  <operation name="getImsiByMsisdn">
    <documentation>{Получение ismi по msisdn[GETTER]}</documentation>
    <input name="getImsiByMsisdnRequest" message="tns:getImsiByMsisdnRequest"/>
    <output name="getImsiByMsisdnResponse" message="tns:getImsiByMsisdnResponse"/>
    <fault name="BusinessFault" message="tns:BusinessFault"/>
  </operation>

  <operation name="getClientTypeDictionary">
    <input name="getClientTypeDictionaryRequest"
message="tns:getClientTypeDictionaryRequest"/>
    <output name="getClientTypeDictionaryResponse"
message="tns:getClientTypeDictionaryResponse"/>
    <fault name="dictionaryNotFoundFault" message="tns:dictionaryNotFoundFault"/>
  </operation>
  <operation name="getSegmentTypeDictionary">
    <input name="getSegmentTypeDictionaryRequest"
message="tns:getSegmentTypeDictionaryRequest"/>
    <output name="getSegmentTypeDictionaryResponse"
message="tns:getSegmentTypeDictionaryResponse"/>
    <fault name="dictionaryNotFoundFault" message="tns:dictionaryNotFoundFault"/>
  </operation>
  <operation name="getServiceTypeDictionary">
    <input name="getServiceTypeDictionaryRequest"
message="tns:getServiceTypeDictionaryRequest"/>
    <output name="getServiceTypeDictionaryResponse"
message="tns:getServiceTypeDictionaryResponse"/>
    <fault name="dictionaryNotFoundFault" message="tns:dictionaryNotFoundFault"/>
  </operation>
</portType>
```

Физические методы для получения критерия могут быть объединены в один бизнес-процесс или в одну SA. Основное требование — каждая технологическая сущность должна соответствовать законченному объему функциональности.

П 5.8. Требования к наименованию артефактов и расширению интерфейсов

Требования к наименованию артефактов

Формат наименования артефактов:

- Название микросервиса — имя в CamelStyle.
- PortType для интерфейса — <http://www.bercut.com/wsd/ms/><Название микросервиса>.
- Имя файла WSDL — AOI_EM_<Название микросервиса>.wsdl.
- Название BP — EM_<Название микросервиса>.

- PortType для BP — <http://www.bercut.com/bpel/em/><Название микросервиса>.
- Имя BPEL-файла — <Название микросервиса>.bpel.
- Имя ATLAS-описания — BP_EM_<Название микросервиса>_v1.atlas.xml.
- Название JAR-файла — BP_EM_<Название микросервиса>_v1.jar.

Пример:

- Имя MSa — OsTypes.
- PortType для интерфейса — <http://www.bercut.com/wsd/ms/OsTypes>.
- Имя файла WSDL — AOI_EM_OsTypes.wsdl.
- Название BP — EM_OsTypes.
- PortType для BP — <http://www.bercut.com/bpel/em/OsTypes>.
- Имя BPEL-файла — OsTypes.bpel.
- Имя ATLAS-описания — BP_EM_OsTypes_v1.atlas.xml.
- Название JAR-файла — BP_EM_OsTypes_v1.jar.

Требования к расширению интерфейсов

При корректировке версии используется *version* для XSD.

Новые параметры добавляются в конец списка параметров.

Источники информации

1. Platform v3. Установка и настройка компонентов. Руководство администратора.
2. Product Catalog. Описание веб-интерфейса. Руководство пользователя.
3. Unica Multilink. Unica Multilink SMPP. Руководство администратора.
4. Notification System. Схемы данных BLACKLIST, AUDIT, MODEL, GROUP VALUES и AGGREGATION. Справочник.
5. Notification System. Описание веб-интерфейса. Руководство пользователя.
6. Notification System. Установка и настройка компонентов. Обновление системы. Руководство администратора.
7. Notification System. Работа с событиями. Руководство по настройке.
8. Notification System. API-команды. Справочник.
9. ATLAS. Руководство администратора.
10. Welcome SMS. Установка и настройка компонентов. Руководство администратора.
11. Welcome SMS. Настройка продуктового решения. Руководство по настройке.
12. Kafka System. Реализация и настройка компонентов. Руководство по настройке.
13. [Apache Kafka](#).
14. Продукт «Линии». Групповые скидки. Руководство по настройке.
15. T2Group. Установка, настройка и обновление системы. Руководство администратора.
16. Запросы 229780 и 237886. Продукт «Промокоды». Руководство по настройке.
17. PromoCode. Установка и настройка системы. Руководство администратора.
18. Communication Layer. Установка и настройка системы. Руководство администратора.
19. Scheduler Subsystem. Установка и настройка системы. Руководство администратора.
20. <https://wiki.tele2.ru/x/5vSJB>. BRE MicroServices. Описание микросервисов. Справочник. BRE MicroServices. Описание микросервисов. Справочник.
21. SCS. Работа с графическим интерфейсом. Руководство пользователя.
22. [RabbitMQ](#)
23. [W3C Recommendation](#)
24. <https://tomcat.apache.org/tomcat-7.0-doc/realm-howto.html>
25. <https://www.w3.org/TR/xmlschema-2/#duration>
26. <https://kafka.apache.org/documentation/#consumerconfigs>

История изменений

В разделе представлена история изменения документа.

Версия 1.0

Версия документа 1:

Создание документа.

Версия документа 2:

1. Добавлено описание установки и настройки обновления системы до версии 1.0.9.
2. Добавлено описание настроек Multilink DB и Unica SMPP Multilink для MVNO-оператора.

Версия документа 3:

1. Обновлено описание настройки для отправки SMS-сообщений Unica_SMPP_Multilink: добавлено описание переменной *DispatcherGroupIdByOwnerId*, удалено описание переменной *DispatcherGroupId* (не используется).
2. Выделен раздел с настройками MVNO-оператора, в котором описано добавление MVNO-оператора.

Версия документа 4:

В разделах **Настройка SA DWH**, **Настройка SA Roaming Steering** и **Настройка SA Multilink** изменено описание настроек файлов `server.xml` и `context.xml`.

Версия документа 5:

1. Расширено описание архитектуры системы (2) — добавлены разделы:
 - Компонентный состав;
 - Описание PortType.
2. Добавлено приложение с описанием взаимодействия компонентов.

Изменения с версии 1.0 до 2.1

Версия документа 1:

1. Обновлен раздел **Функциональные возможности пользователей**: администратор работает в веб-приложении Back Office Notification System, а не в Product Catalog.
2. Изменена архитектура системы — 2.
3. Обновлен раздел **Установка** (4):
 - Удалены компоненты версии 1.0: SE_EM_Formatter_v1 и SE_EM_Utils_v1.jar.
 - Удалено упоминание о SA_EM_Dictionary.
4. Удален неактуальный раздел **Добавление MVNO-оператора**.
5. Раздел **Обновление до версии 1.0.9** заменен на **Обновление до версии 2.1**.
6. Информация из следующих разделов перенесена в документ «Welcome SMS. Руководство администратора»:
 - Установка Unica SMPP Multilink
 - Настройка взаимодействия компонентов
 - Настройка SLES
 - Настройка SA
 - Настройка LW SA Container

- Настройка Unica SMPP Multilink
- Настройка схемы данных Multilink DB
- Настройки SCP системы Welcome SMS
- Настройка бизнес-процесса BP_EmSendSms
- Настройка отправки роуминговых сообщений для MVNO-оператора
- Приложение 1. Используемые микросервисы
- Приложение 2. Взаимодействие компонентов

7. Добавлены разделы:

- Установка NS BO
- Установка SA NS BlackList — [4.3.1](#)
- Установка схемы данных DB AUDIT — [4.1.4.1](#)
- Установка схемы данных DB BLACKLIST — [4.3.2](#)
- Установка Service Gateway — [4.3.3](#).
- Настройка CMP_NsSLES_v3.3_em
- Настройка CMP_LWSAContainer_v3.3_em.
- Миграция данных в DB BLACKLIST

Изменения с версии 2.1 до 2.2

Версия документа 1:

1. Удален ввиду неактуальности раздел **Функциональные возможности пользователей**. Пользователи работают в веб-интерфейсе системы Notification System согласно ролевой модели [5].
2. В раздел **Установка и настройка компонентов** (4) добавлены шаги 3 и 4.
3. Расширен раздел **Установка и настройка схемы данных DB AUDIT** (4.1.4.1).
4. Схема данных DATAMODEL добавлена в архитектуру системы (2) и в схему развертывания.
5. Добавлены разделы:
 - Установка схемы данных DB DATAMODEL — [4.1.4.2](#).
 - Обновление системы с версии 2.1 до 2.2.
 - Секционирование таблиц DB AUDIT — [4.1.4.1.1](#).
 - Настройка поиска по критериям в DB AUDIT — [4.1.4.1.2](#).
 - Настройка получения аудита через API — [4.4](#)
 - Миграция данных в DB DATAMODEL

Изменения с версии 2.2 до 2.3

Изменения не вносились.

Изменения с версии 2.3 до 2.4

Версия документа 1:

1. Полностью изменена архитектура системы — 2. Добавлены новые компоненты, удалены неиспользуемые.
2. Реализована интеграция с Apache Kafka. Добавлены разделы:
 - **Интеграция с Apache Kafka** — [3.1](#);
 - **Настройка интеграции с Apache Kafka** — [4.6](#).
3. Обновлены разделы с информацией по установке и настройке системы в соответствии с версией 2.4. Добавлен раздел **Установка AuditDeliveryReport** (4.2).
4. Название схемы данных DB DATAMODEL изменено на DB MODEL.

5. Удалена информация про миграцию данных.

Версия документа 2:

Добавлен раздел **Мониторинг системы** — 7.

Изменения с версии 2.4 до 2.5

Версия документа 1:

1. В раздел **Микросервисы** добавлено уточнение: справочники могут быть вызваны микросервисами — 1.6.
2. Обновлен раздел **Архитектура системы**, в архитектуру добавлены компоненты SA AuditDeliveryReport и SA EM BackOffice, обновлена схема взаимодействия системы Notification System с другими системами Bercut — 2.
3. В разделе **Интеграция с Apache Kafka** изменен рисунок **Отправка событий в Apache Kafka** — 3.1.
4. В раздел **Настройка LW SA Container EM BO** добавлена информация о загрузке справочников бизнес-критериев — 4.1.2.
5. Обновлен раздел **Установка и настройка схемы данных DB DATAMODEL** — 4.1.4.2.
6. Раздел **Настройка Apache Kafka** переименован в **Настройка интеграции с Apache Kafka**, добавлено примечание — 4.6.
7. Удалены разделы:
 - **Обновление системы с версии 1.0 до 2.1.**
 - **Обновление системы с версии 2.1 до 2.2.**

Изменения с версии 2.5 до 2.6

Версия документа 1:

1. В раздел **Общие сведения** добавлено упоминание группы значений — 1.
2. В раздел **Архитектура системы** добавлена информация о компонентах SA NS GroupValues, SA NS HttpProducer, SA NS RabbitMQProducer, GROUP VALUES DB — 2.
3. Добавлены разделы:
 - **Группа значений** — 1.8.
 - **Интеграция с внешними системами**, куда перенесен раздел **Интеграция с Apache Kafka** — 3.
 - **Интеграция с RabbitMQ** — 3.2.
 - **Отправка HTTP-сообщений** — 3.3.
 - **Установка и настройка схемы данных DB GROUP VALUES** — 4.1.4.3.
 - **Установка и настройка HTTP-адаптера** — 4.7.
 - **Установка и настройка RabbitMQ-адаптера** — 4.8.
 - **Установка и настройка SA NS GroupValues** — 4.10.
4. В раздел **Настройка поиска по критериям в DB AUDIT** добавлено описание параметров запроса и пример отображения параметра MODEL_EVENT_ID в Back Office — 4.1.4.1.2.
5. Обновлен раздел **Обновление системы** — 5.

Изменения с версии 2.6 до 2.7

Версия документа 1:

1. В раздел **Сценарий** добавлено определение отложенного сценария — 1.2.
2. В разделе **Архитектура системы** в архитектуру добавлен компонент SA NS SchedulerScenario — 2.
3. Внесены изменения в раздел **Обновление системы** — 5.

4. Добавлены разделы:

- 4.1. **Интеграция с Scheduler Subsystem** — [3.4](#);
- 4.2. **Установка и настройка SA NS ScheduledScenario** — [4.11](#).
- 4.3. **Работа с отложенными сценариями** — [Приложение 1](#).
- 4.4. **Создание микросервиса** — [П 1.1](#);
- 4.5. **Настройка отложенного исполнения сценариев** — [П 1.2](#).

Версия документа 2:

В раздел «Настройка интеграции с Apache Kafka» добавлены MIB-параметры `TopicToPartitionKeyFieldValue` и `TopicToPartitionKeyDefaultValue` — [4.6](#).

Версия документа 3:

1. В разделы добавлена информация о варианте настройки отложенного исполнения сценариев:
 - 1.1. **Работа с отложенными сценариями** — [Приложение 1](#);
 - 1.2. **Создание микросервиса** — [П 1.1](#);
 - 1.3. **Настройка отложенного исполнения сценариев** — [П 1.2](#).
2. Добавлены разделы:
 - 2.1. **Вариант настройки отложенного исполнения сценариев** — [П 1.3](#).
 - 2.2. **Создание микросервиса** — [П 1.3.1](#).
 - 2.3. **Настройка сценария А** — [П 1.3.2](#);
 - 2.4. **Настройка схемы сценария** для сценария А — [П 1.3.2.1](#);
 - 2.5. **Настройка таблицы правил** для сценария А — [П 1.3.2.2](#);
 - 2.6. **Настройка сценария Б** — [П 1.3.3](#);
 - 2.7. **Настройка схемы сценария** для сценария Б — [П 1.3.3.1](#);
 - 2.8. **Настройка таблицы правил** для сценария Б — [П 1.3.3.2](#).

Изменения с версии 2.7 до 2.8

Версия документа 1:

1. В раздел **Общие сведения** добавлена информация о привилегиях — [1](#).
2. Добавлен раздел **Ролевая модель доступа** — [1.9](#).
3. Добавлен раздел **Настройка System Principal** — [4.5](#).
4. В разделе **Настройка LW SA Container EM BO** обновлено описание настройки авторизации пользователей в Back Office — [4.1.2](#).
5. Добавлены разделы:
 - 5.1. **Настройка аутентификации через Active Directory** — [4.1.2.1](#);
 - 5.2. **Настройка аутентификации через Tomcat** — [4.1.2.2](#).
6. Обновлен раздел **Обновление системы** — [5](#).

Версия документа 2:

1. В раздел **Настройка NS SLES EM** добавлен пункт об установке микросервисов системы BRE Microservices — [4.1.1](#).
2. Добавлен раздел **Установка и настройка компонентов системы BRE Microservices** — [4.12](#).

Версия документа 3:

1. В разделе **Интеграция с Apache Kafka** обновлен алгоритм работы SA NS Kafka Consumer — [3.1](#).
2. В разделе **Настройка интеграции с Apache Kafka** изменен список MIB-параметров для SA NS Kafka Consumer — [4.6](#);

2.1. Добавлены параметры:

- *BreErrorBehavior*;
- *ConversionErrorBehavior*;
- *ConversionMappingFiles*;
- *DefaultEncoding*;
- *KafkaMaxPartitionFetchBytes*;
- *KafkaProperties*;
- *PollerThreadTimeout*;
- *ReloadFiles*;
- *RtsibErrorBehavior*;
- *Slf4jAtlasLoggerEnabled*;
- *SuspenseRetryTimeout*;
- *TopicThreadCount*;
- *UseWSN*.

2.2. Удален параметр *KafkaPollerThreadCount*.

3. Добавлен раздел **Создание событий** и его подразделы — [Приложение 2](#).
4. Раздел **Пользовательское событие** перенесен из документа «Notification System. Работа с микросервисами и справочниками. Руководство по настройке» — [П 2.1](#).
5. Добавлен раздел **Правила преобразования сообщений адаптерами consumer** и его подразделы — [Приложение 3](#).

Версия документа 4:

1. Из раздела **Настройка аутентификации с помощью Active Directory** удалена неверная настройка аутентификации с помощью Active Directory — [4.1.2.1](#).

Изменения с версии 2.8 до 2.9

Версия документа 1:

1. В разделе **Архитектура системы** добавлен компонент SA NS AuditAggregator, обновлена архитектура системы — [2](#).
2. Изменена схема развертывания в разделе **Установка и настройка компонентов** — [4](#).
3. Изменена версия PostgreSQL на 12 в разделах:
 - 3.1. **Установка и настройка схемы данных DB AUDIT** — [4.1.4.1](#).
 - 3.2. **Установка и настройка схемы данных DB MODEL** — [4.1.4.2](#).
 - 3.3. **Установка и настройка схемы данных DB GROUP VALUES** — [4.1.4.3](#).
4. Добавлен раздел **Установка и настройка схемы данных DB AGGREGATION** — [4.1.4.4](#).
5. В разделе **Настройка интеграции с Apache Kafka** для SA NS Kafka Consumer добавлен MIB-параметр *DefaultMappingEnabled* — [4.6](#).
6. Добавлен раздел **Настройка отчетов** — [4.13](#).
7. Обновлен текст раздела **Обновление системы** — [5](#).

Версия документа 2:

1. Добавлен раздел **Правила преобразования сообщений адаптерами producer** и его подразделы — [Приложение 4](#).
2. Добавлен раздел **Событие для обработки SA NS Kafka Producer** и его подразделы — [П 2.4.2](#).
3. В разделе **Настройка интеграции с Apache Kafka** обновлены настройки SA NS KafkaProducer — [4.6](#):
 - 3.1. удалены MIB-параметры:

- *TopicToPartitionKeyDefaultValue*;
- *TopicToPartitionKeyFieldValue*.

3.2. добавлены MIB-параметры:

- *ConversionMappingFiles*;
- *DefaultEncoding*;
- *DefaultMethod*;
- *KafkaProperties*;
- *ReloadFiles*;
- *Slf4jAtlasLoggerEnabled*.

3.3. параметр *TopicNameDefault* переименован в *DefaultTopic*.

3.4. параметр *TopicEncodeDefault* переименован в *DefaultFormat*.

3.5. добавлена группа *SA_NsKafkaProducer/Core/Configuration/Elements/* с параметрами:

- *ConversionMappingFiles*;
- *DefaultEncoding*;
- *DefaultFormat*;
- *DefaultPartitionKey*;
- *FieldPartitionKey*;
- *Topic*.

Изменения с версии 2.9 до 2.10

Версия документа 1:

1. В разделе **Установка и настройка компонентов** обновлена версия Java на 1.13 — [4](#).
2. В разделе **Настройка NS SLES EM** версия SLES изменена на 3.5, добавлен маршрут {<http://www.bercut.com/wsd1/ns/interpreter/status>}NS_InterpreterStatus — [4.1.1](#).
3. Раздел **Настройка LW SA Container EM** переименован в **Настройка LW SA Container** — [4.1.3](#).
4. В разделе **Установка и настройка схемы данных DB AGGREGATION** добавлена настройка новых отчетов — [4.1.4.4](#).
5. Версия компонентов LW SA Container и SLES изменена на 3.5 в разделах:
 - **Настройка LW SA Container EM BO** — [4.1.2](#).
 - **Настройка интеграции с Apache Kafka** — [4.6](#).
6. Имя LWSA_em изменено на LW SA Container в разделах:
 - 6.1. **Настройка AuditDeliveryReport** — [4.2](#);
 - 6.2. **Установка и настройка схемы данных DB BLACKLIST** — [4.3.2](#);
 - 6.3. **Установка и настройка HTTP-адаптера** — [4.7](#);
 - 6.4. **Установка и настройка RabbitMQ-адаптера** — [4.8](#);
 - 6.5. **Установка и настройка SA NS GroupValues** — [4.10](#);
 - 6.6. **Установка и настройка SA NS ScheduledScenario** — [4.11](#).
7. В раздел **Установка и настройка компонентов системы BRE MicroServices** добавлены новые маршруты — [4.12](#).
8. Обновлен раздел **Обновление системы** — [5](#).
9. Добавлен раздел **Событие с поддержкой транзакционности** — [П 2.3](#).

Версия документа 2:

1. В разделе **Интеграция с Scheduler Subsystem** указана используемая версия Scheduler Subsystem (не ниже 3.2), добавлено примечание про обработку отложенных сценариев — [3.4](#).
2. В раздел **Установка и настройка SA NS ScheduledScenario** добавлены MIB-параметры *MaxExecutionTime* и *MaxLifeTime* — [4.11](#).
3. В разделе **Преобразование с помощью файла маппинга** добавлено описание параметра **to-event-context** — [П 3.1](#).
4. В разделе **XML-преобразование** обновлены примеры преобразований — [П 3.1.1](#).
5. В разделе **JSON-преобразование** обновлены примеры преобразований, добавлен пример обработки параметра-массива — [П 3.1.2](#).

Версия документа 3:

1. В раздел «Установка и настройка схемы данных DB AGGREGATION» добавлена настройка параметра *InitSql* — [4.1.4.4](#).
2. В разделе **Установка и настройка HTTP-адаптера** — [4.7](#):
 - 2.1. удалены MIB-параметры:
 - *MessageToNotificationMapping*;
 - *MessageEncodeDefault*;
 - *MessageEncodeJson*;
 - *MessageEncodeXml*.
 - 2.2. добавлены MIB-параметры:
 - *ConnectTimeout*;
 - *ConnectionRequestTimeout*;
 - *ReloadFiles*;
 - *RequestConversionMappingFiles*;
 - *RequestDefaultEncoding*;
 - *RequestDefaultFormat*;
 - *ResponseConversionMappingFiles*;
 - *ResponseDefaultEncoding*;
 - *ResponseDefaultMappingEnabled*;
 - *SocketTimeout*.
 - 2.3. добавлены MIB-параметры для группы *SA_NsHttpProducer/Core/Configuration/Elements/*.
3. В раздел «Настройка интеграции с Apache Kafka» — [4.6](#):
 - 3.1. Добавлен MIB-параметр *TraceBlockEnabled* для SA NsKafkaConsumer.
 - 3.2. Добавлены примечания о логике обработки для MIB-группы / *SA_NsKafkaConsumer/Core/Configuration/Topics/* и параметра *KafkaTopics*.
 - 3.3. В MIB-группу /*SA_NsKafkaConsumer/Core/Configuration/Topics/*<Имя топика> добавлены MIB-параметры *CustomDateTimeFormats* и *DefaultMappingEnabled*.
4. Раздел «Настройка отчетов» переименован в «Установка и настройка SA NS AuditAggregator» — [4.13](#).
5. В разделе «Создание событий» расширен список событий — [Приложение 2](#).
6. В раздел «Событие для обработки SA NS Kafka Consumer» добавлено описание преобразования с помощью файла маппинга и обработки блока trace — [П 2.4.1](#).
7. Добавлен раздел «События для SA NS HttpProducer» — [П 2.5](#).
8. В раздел «Правила преобразования сообщений адаптерами Consumer» добавлены адаптеры, для которых доступны преобразования по правилам. Текст раздела «Правила преобразования сообщений» перенесен в этот раздел — [Приложение 3](#).
9. В раздел «Правила преобразования сообщений адаптерами Producer» добавлены адаптеры, для которых доступны преобразования по правилам — [Приложение 4](#).
10. В раздел «Преобразование с помощью файла маппинга» добавлены примеры описания маппинга для форматов XML и JSON — [П 4.1](#).

Изменения с версии 2.10 до 2.11

Версия документа 1:

Изменен раздел **Обновление системы** — 5.

Версия документа 2:

В разделе **Обновление системы** обновлена версия устанавливаемого патча схемы данных MODEL на 1.1.5 — 5.

Версия документа 3:

Добавлено [Приложение 5](#) «Реализация обработки событий».

Изменения с версии 2.11 до 2.12

Версия документа 1:

1. Изменен раздел **Обновление системы** — 5.
2. В разделе **Настройка аутентификации через Active Directory** — 4.1.2.1 добавлен MIB-параметр: *EmailAttribute*.
3. В разделе **Настройка LW SA Container EM BO** добавлена настройка URL-префикса — 4.1.2.
4. Добавлен раздел **Настройка расширенного аудита** — 4.1.4.1.3.

Версия документа 2:

1. В раздел **Настройка расширенного аудита** добавлена настройка маршрутизации — 4.1.4.1.3.
2. В раздел **Обновление системы** добавлены настройка маршрутизации, патч DB Audit и настройка новых привилегий — 5.

Изменения с версии 2.12 до 2.13

Версия документа 1:

1. Изменен раздел **Обновление системы** — 5.
2. В разделе **Установка и настройка SA NS ScheduledScenario** — 4.11:
 - добавлена настройка соединения с базой данных;
 - добавлены MIB-параметры:
 - *RtsibErrorBehavior*;
 - *BreErrorBehavior*;
 - *SuspenseRetryTimeoutInMillis*;
 - *TaskTimeoutInSeconds*;
 - *MaxPollTimeoutInMillis*;
 - *PollThreadCount*;
 - *TaskCountPerPoll*;
 - добавлено описание настройки и статистики таблицы *TrafficLimit* для управления выполнением отложенных сценариев.
3. В раздел **Установка и настройка схемы данных DB AGGREGATION** добавлена установка инструкций *scheduled_event.xml* и *scheduled_events_error-list.xml* — 4.1.4.4.
4. Обновлена архитектура системы — 2.

Версия документа 2:

1. Из раздела **Установка и настройка схемы данных DB AGGREGATION** удалена установка инструкции *scheduled_events_error-list.xml* — инструкция для установки этого отчета устанавливается при обновлении NS BackOffice — 4.1.4.4.

2. Инструкция по настройке соединения с базой данных Scheduler дополнена и перенесена из раздела **Установка и настройка SA NS ScheduledScenario** в раздел **Настройка LW SA Container EM VO** — [4.1.2](#).
3. Изменен раздел **Обновление системы** — [5](#).

Версия документа 3:

1. Добавлен раздел **Установка и настройка адаптера для БД** — [4.9](#).
2. Добавлен раздел **Установка и настройка SA NS KVStorageManagement** — [4.15](#).
3. Обновлена архитектура системы — [2](#).
4. Изменен раздел **Обновление системы** — [5](#).

Версия документа 4:

1. В раздел **Обновление системы** добавлены пункт об обновлении системы BRE Microservices, ссылка на описание установки и настройки SA KeyValues Storage и примечание об обновлении NS SLES — [5](#).
2. Добавлен раздел **Установка и настройка SA KeyValuesStorage** — [4.14](#).
3. Дополнен раздел **Установка и настройка SA NS KVStorageManagement** — [4.15](#).

Изменения с версии 2.13 до 2.14

Версия документа 1:

1. Изменен раздел **Обновление системы** — [5](#).

Версия документа 2:

1. Добавлен пункт о настройке кеширования ответов на внешние вызовы — [4.1.1](#).
2. Добавлен раздел «Настройке кеширования ответов на внешние вызовы» — [4.16](#).

Изменения с версии 2.14 до 2.15

Версия документа 1:

1. Изменен раздел **Обновление системы** — [5](#).

Изменения с версии 2.15 до 2.16

Версия документа 1:

1. Изменен раздел **Обновление системы** — [5](#).
2. В раздел **Установка и настройка схемы данных DB AUDIT** добавлен пункт о настройке трассировки стека — [4.1.4.1](#).
3. В раздел **Настройка интеграции с Apache Kafka** для SA NS KafkaProducer добавлен MIB-параметр *DefaultJsonMethod* — [4.6](#).
4. В раздел **Настройка аутентификации через Active Directory** добавлен MIB-параметр *Mode* — [4.1.2.1](#).

Версия документа 2:

1. В раздел **Настройка аутентификации через Active Directory** внесены уточнения — [4.1.2.1](#).

Изменения с версии 2.16 до 2.17

Версия документа 1:

Изменения не вносились.

Версия документа 2:

1. Добавлен раздел [Установка и настройка схем базы данных](#):

- В данный раздел вынесено описание общих действий по установке и настройке БД.
- Добавлены настройки резервирования.
- В подразделах для каждой БД оставлены кастомные настройки.