

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ИНПОЛЮС»
СИСТЕМА МОНИТОРИНГА И КОНФИГУРИРОВАНИЯ КОРПОРАТИВНОЙ
ИНТЕГРАЦИОННОЙ ШИНЫ POLUS ESB

Руководство пользователя

На 128 листах

СОДЕРЖАНИЕ

СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ	5
ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ	6
Назначение Системы	6
Основные категории задач Системы	6
Требования к архитектуре системы.....	8
УПРАВЛЕНИЕ REST API	14
ПРОКСИ-СЕРВИСЫ	18
Простой сервис-посредник	19
Безопасный прокси-сервис.....	20
Прокси-сервис на основе WSDL	21
Логирующий прокси-сервис	22
Трансформирующий прокси-сервис	23
ТОЧКИ ВХОДА (INBOUND ENDPOINTS)	25
Прослушивающие точки входа.....	26
Опрашивающие точки входа.....	35
Точки входа, основанные на событиях.....	43
Пример создания прослушивающей точки входа HTTP	43
Пример создания опрашивающей точки входа — JMS	45
ТОЧКИ ВЫЗОВА	46
Добавление точки вызова по URL	46
Добавление Точки Вызова По Умолчанию	48
Добавление Точки Вызова HTTP	48
Добавление точки вызова WSDL.....	50
Добавление отказоустойчивой группы	50
Добавление группы балансировки нагрузки	52
Добавление точки вызова по шаблону	52
Добавление списка получателей.....	54
Добавление динамических точек вызова	54
Параметры, общие для точек вызова по URL, HTTP, WSDL и по умолчанию	55
ПОСЛЕДОВАТЕЛЬНОСТИ МЕДИАЦИИ	57
Добавление последовательности	58
Добавление динамической последовательности.....	58

Редактирование последовательности	60
Типы медиаторов.....	62
СЕРВИСЫ ДАННЫХ	67
Установка сервера сервисов данных Polus ESB и доступ к консоли управления.....	67
Создание экземпляра (instance) БД и заполнение базы данными.....	68
Создание сервиса данных.....	69
Вызов сервиса данных.....	73
Генератор автоматизированных сервисов данных	75
Редактирование сервиса данных	81
Загрузка сервиса данных	85
Удаление сервиса данных.....	86
ПРИЛОЖЕНИЯ	88
НАСТРОЙКА ПОЛЬЗОВАТЕЛЕЙ	92
Добавление нового пользователя и назначение ролей	92
Импорт пользователей	93
НАСТРОЙКА РОЛЕЙ.....	98
Добавление роли пользователя	98
Редактирование или удаление роли.....	99
Обновление имен ролей	99
ЗАПЛАНИРОВАННЫЕ ЗАДАЧИ.....	101
ШАБЛОНЫ.....	104
Управление шаблонами последовательностей	104
Управление шаблонами точек вызова	107
УПРАВЛЕНИЕ ХРАНИЛИЩАМИ СООБЩЕНИЙ	110
Добавление хранилища сообщений в памяти	110
Добавление хранилища сообщений JMS	111
Добавление пользовательского хранилища сообщений.....	112
Возможные действия в хранилище сообщений	112
УПРАВЛЕНИЕ ПРОЦЕССОРАМИ СООБЩЕНИЙ.....	114
Добавление процессора пересылки запланированных сообщений	114
Добавление Процессора Выборки Сообщений.....	115
Добавление Пользовательского Процессора Сообщений	116
Действия с процессорами сообщений.....	117
КОННЕКТОРЫ.....	118

ТОПИКИ И ПОДПИСКИ.....	120
РАЗВЕРНУТЫЕ СЕРВИСЫ	124

СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ

API	Applied Programming Interface – интерфейс прикладного программирования, вызов методов которого, как правило, доступен внешним системам
Axis2	Каркас (framework) разработки Web-сервисов с открытым кодом (open source)
ESB	Enterprise Service Bus - связующее ПО (middleware) класса «Сервисная шина предприятия»
HTTP	HyperText Transfer Protocol - протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.
HTTPS	HyperText Transfer Protocol Secure - расширение протокола HTTP для поддержки шифрования в целях повышения безопасности
JMS	Java Messaging Service - стандарт связующего ПО (middleware) для обмена сообщениями
REST	Representational State Transfer - архитектурный стиль взаимодействия компонентов распределённого приложения в сети.
SOAP	Simple Object Access Protocol - протокол обмена структурированными сообщениями в распределённой вычислительной среде, широко используемый для вызова Web-сервисов
URL	Unified Resource Locator – унифицированный указатель ресурса (файла, сервиса и т.д.), определяющий адрес ресурса в глобальной сети
WebSocket	Протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером, используя постоянное соединение
WSDL	Web Service Definition Language - язык описания веб-сервисов и доступа к ним, основанный на языке XML.
XML	Extensible Markup Language – расширяемый язык разметки документов, широко используется как формат сообщений в распределенной среде
XSL	Extensible Stylesheet Language – набор стандартов, описывающих языки преобразования и визуализации XML-документов

ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ

Назначение Системы

Система предназначена для мониторинга и конфигурирования корпоративной интеграционной шины Polus ESB и должна предоставлять возможность создавать и конфигурировать интеграционные решения на основании событийной модели и передачи сообщений между информационными системами (источниками и получателями) с использованием графического интерфейса.

Основные категории задач Системы

Система обеспечивает надежный, унифицированный и гибко настраиваемый механизм обмена данными между информационными системами в рамках предприятия и решает следующие задачи:

- обеспечивает настраиваемую маршрутизацию и гарантированную доставку сообщений от источников к получателям;
- реализует трансформацию данных между форматами, требуемыми различными приложениями, а также дополнение их необходимой информацией (обогащение данными) из внешних источников информации (баз данных, файловых хранилищ, корпоративных информационных систем, сторонних сервисов);
- предоставляет визуальный интерфейс для создания и конфигурирования параметров сообщений и сценариев обмена данными;
- обеспечивает возможность разработки и конфигурирования адаптеров к внешним информационным системам (как готовых, так и реализованных самостоятельно);
- обеспечивает механизмы аутентификации, авторизации и разграничения прав доступа к функциям и данным системы;
- реализует механизм мониторинга и аудита исполняемых процессов и операций;
- обеспечивает безопасность передачи данных при помощи шифрования.

С точки зрения единого информационного пространства предприятия и оптимизации бизнес-

процессов предприятия внедрение системы позволяет решать следующие задачи:

- формирование единого информационного пространства в гетерогенных средах;
- получение масштабируемой архитектуры управления уровня компании;
- обеспечение синхронизации данных в различных информационных системах;
- делегирование задачи построения информационных потоков аналитическим отделам;
- уменьшение общей сложности интеграционной схемы и снижение требования к пропускной способности каналов;
- снижение транзакционных издержек при обмене данными между различными подразделениями;
- снижение общих затрат на обслуживание и сопровождение информационных систем.

Требования к архитектуре системы

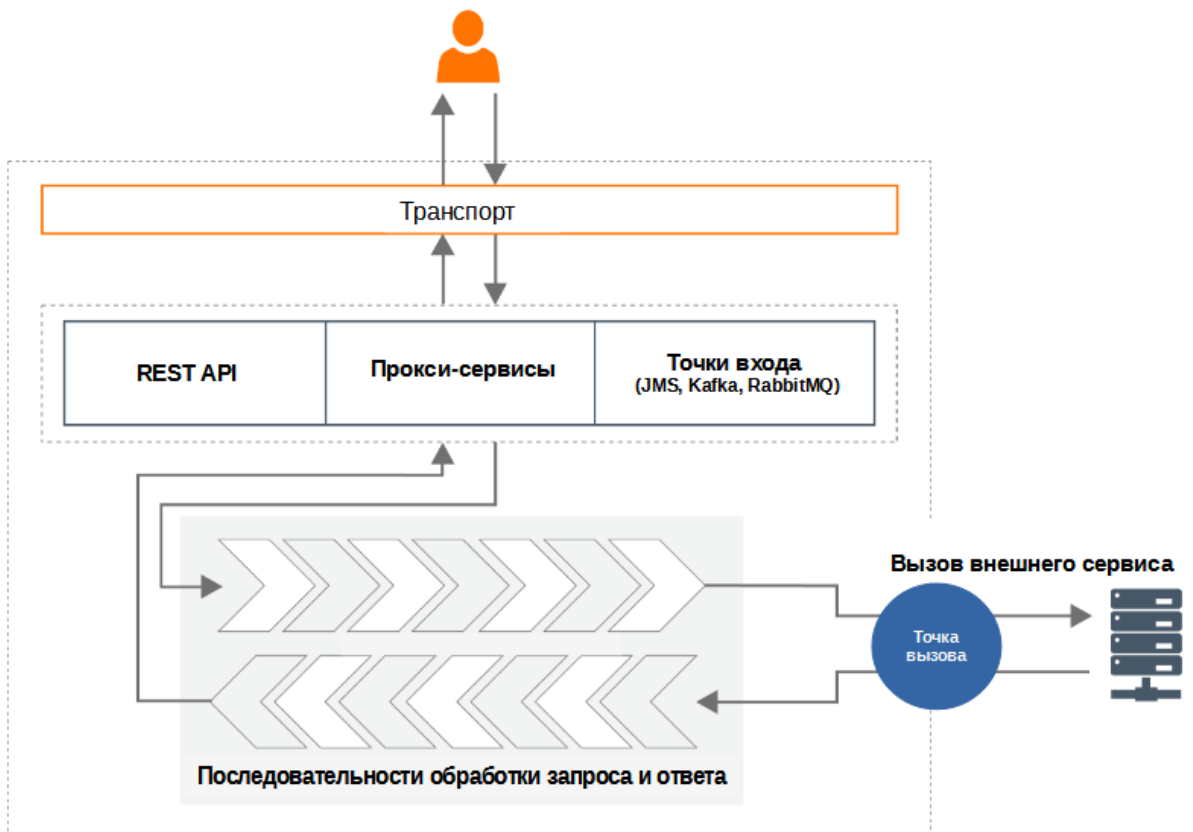


Рис.1. Общая архитектура системы

Общая схема работы Polus ESB реализована следующим образом:

- внешнее сообщение/запрос поступает от клиента по определенному протоколу уровня приложения и отправляется первичному обработчику запроса определенного типа (REST API, прокси-сервис, точка входа - inbound endpoint);
- далее Polus ESB выполняет последовательность обработки сообщений;
- обработанное сообщение передается точке выхода (endpoint), являющейся интерфейсом доступа к системе-получателю.

Первичные обработчики входных сообщений

Первичные обработчики входных сообщений — это объекты Polus ESB, которые запускают выполнение последовательности дальнейшей обработки сообщения.

Система предоставляет 3 типа первичных обработчиков:

- REST API;
- прокси-сервисы (SOAP);
- точки входа (inbound endpoints).

REST API обрабатывает только те входные сообщения, которые ссылаются на конкретный *URL контекст*, определяемый пользователем при конфигурировании REST API (по аналогии с сервлетом, обрабатывающим только запросы, относящиеся к конкретному Web-приложению, контекст которого определяется соответствующим URL в web.xml).

Далее входящее сообщение передается *последовательности обработки запросов* (in-sequence), а по окончании обработки — системе-получателю. Ответы от систем-получателей направляются *последовательности обработки ответов* (out-sequence). Также существует возможность определить *последовательность обработки ошибок* (fault-sequence), которая выполняется в случае возникновения исключения.

REST API определяет один или более ресурсов, являющихся логическими компонентами API, доступ к каждому из которых предоставляется посредством соответствующего HTTP-вызова (GET, POST, PUT, DELETE).

Прокси-сервис обрабатывает входное сообщение перед отправкой его заданной системе-получателю (endpoint). Существует возможность настройки подключения дополнительной функциональности обработки сообщения при вызове прокси-сервиса до передачи сообщения сервису-получателю (endpoint). Вызов прокси-сервиса доступен не только по HTTP/S, но и с использованием других протоколов уровня приложения (JMS, FTP). Сам прокси-сервис создается в соответствии с Axis2 архитектурой и доступен и как SOAP, и как REST сервис.

Точки входа (inbound endpoints) позволяют не только непосредственно получать входные сообщения (т.е. функционировать как listeners), но и периодически опрашивать различные источники данных (например — JMS-очереди) на предмет появления новых сообщений.

Обработка сообщений

Медиаторы — это отдельные обработчики сообщений, вызываемые после приема входного запроса/сообщения и выполнения первичной обработки при помощи REST API, прокси-сервисов или inbound endpoints. В рамках Polus ESB реализована библиотека медиаторов, обеспечивающих функциональные возможности для реализации стандартных действий по обработке сообщений (фильтрация, трансформация, обогащение дополнительными данными, отправка получателю). Также есть возможность подключить медиатор, реализованный пользователем.

Последовательность — это объект Polus ESB, который реализует последовательный вызов определенных медиаторов и механизм каналов (pipes) между ними, т. е. передачу результатов работы медиатора на вход следующему. Перечень медиаторов и их порядок вызова в рамках последовательности описан в настройках.

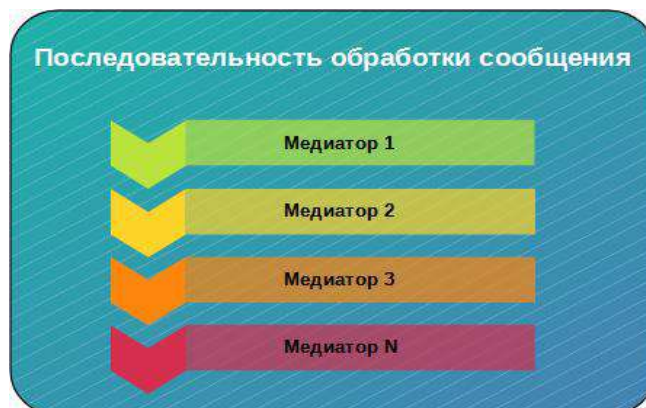


Рис. 2. Последовательность и медиаторы

Система предоставляет возможность сконфигурировать последовательности для:

- обработки полученного сообщения (in-sequence);
- обработки ответа от системы-получателя (out-sequence);
- обработки ошибок (fault-sequence).

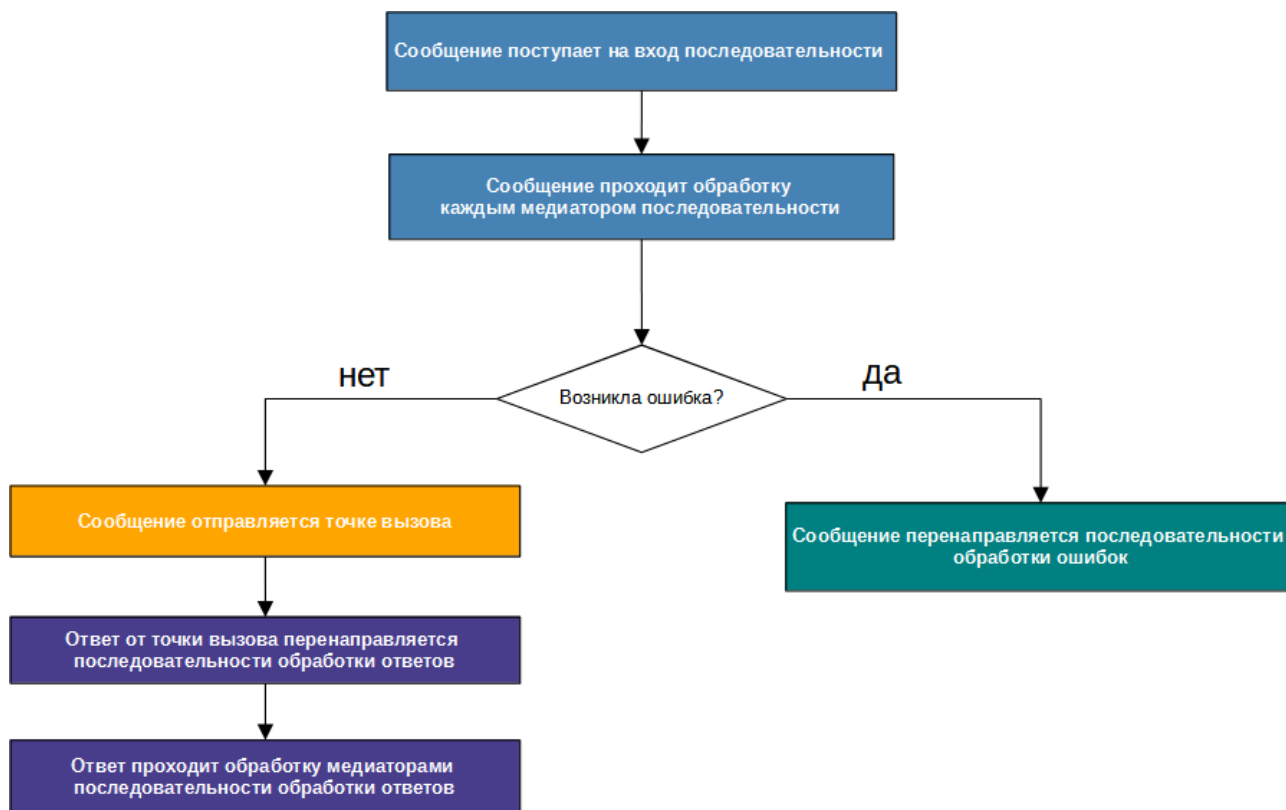


Рис.3. Обработка сообщения: запрос, ответ и последовательность ошибок

Система реализует механизм хранения сообщений между вызовами в рамках последовательности.

Этот подход позволяет реализовать следующие сценарии:

- обслуживание трафика для внешних сервисов-получателей, которые могут принимать сообщения только с заданной пропускной способностью, тогда как входящий трафик поступает с разной скоростью;
- обеспечение гарантированной доставки сообщений в случае сбоев - если обработчики сообщений в определенный момент недоступны, сообщение остается записанным во внутреннее хранилище до тех пор, пока Polus ESB не сможет их вновь обработать.

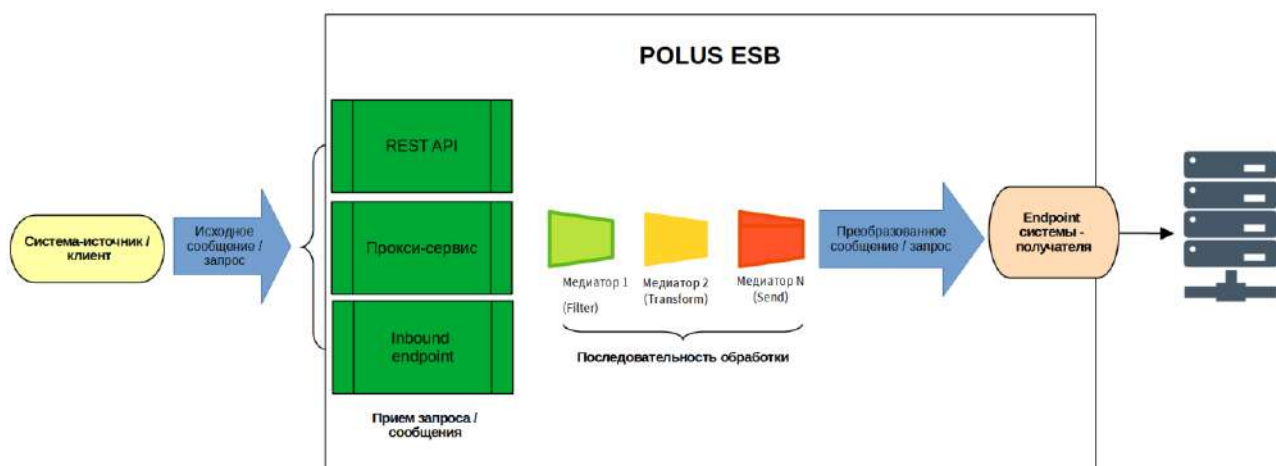


Рис.4. Прием, обработка и передача сообщения внешнему сервису (получателю)

Система реализует также механизм настраиваемых **коннекторов** к широко используемым внешним системам (таким, как JIRA и Google Docs) и позволяет подключать коннекторы, разрабатываемые пользователем. Обработчики в рамках последовательности имеют возможность вызова коннекторов по стандартному интерфейсу.

Взаимодействие с системами-получателями сообщений

Endpoints (точки вызова)

Polus ESB реализует механизм endpoints (точек вызова) – интерфейсов вызова внешних систем, которым в конечном итоге передается обработанное сообщение/запрос. Параметры точки вызова являются настраиваемыми и представляют собой конкретную строку соединения (URL, почтовый ящик, очередь JMS, WebSocket и т. д.), а также прочие параметры, необходимые для подключения и вызова системы-получателя.

Механизм шаблонов и реестр конфигураций

Polus ESB предоставляет механизм шаблонов для настроек/конфигурации всех объектов системы. Пользователь имеет возможность создавать конфигурации последовательностей, точек выхода (endpoints) и других объектов системы в соответствии с определенным прототипом.

Все настройки (конфигурации) объектов системы, включая и дополнительные ресурсы типа WSDL или XSL-преобразований, хранятся в реестре. К каждому элементу настройки обеспечивается доступ по определенному ключу (аналог строки XPath или пути к файлу в файловой системе).

УПРАВЛЕНИЕ REST API

Используйте страницу *Управление API* для добавления новых API или изменения/удаления существующих API. Как показано на Рисунке 1, страница Развёрнутые API показывает имена существующих API. Действие Edit перенаправляет пользователя на страницу Редактировать API, в то время как действие Удалить запрашивает подтверждение и удаляет API из конфигурации. Действие Добавить API перенаправляет пользователя на страницу Добавить API.

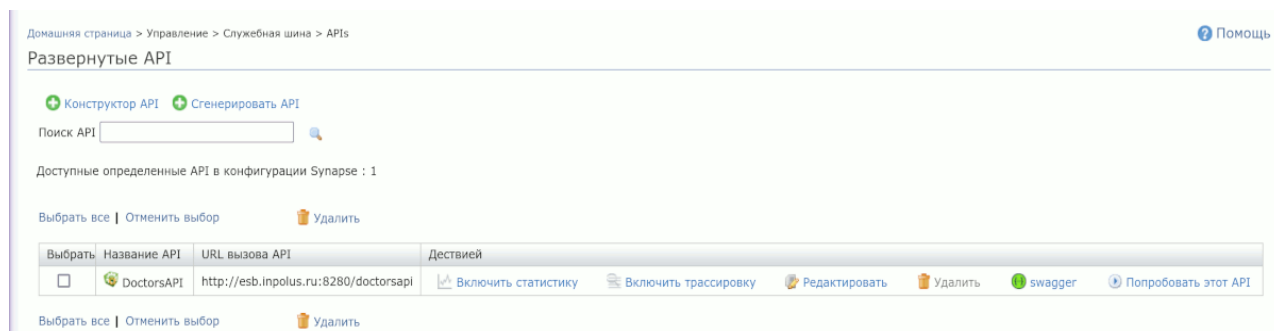


Рис. 5. Просмотр Существующих API

Добавление API

Щёлкните по ссылке **Добавить API**. Появится страница *Добавить API*, как показано ниже.

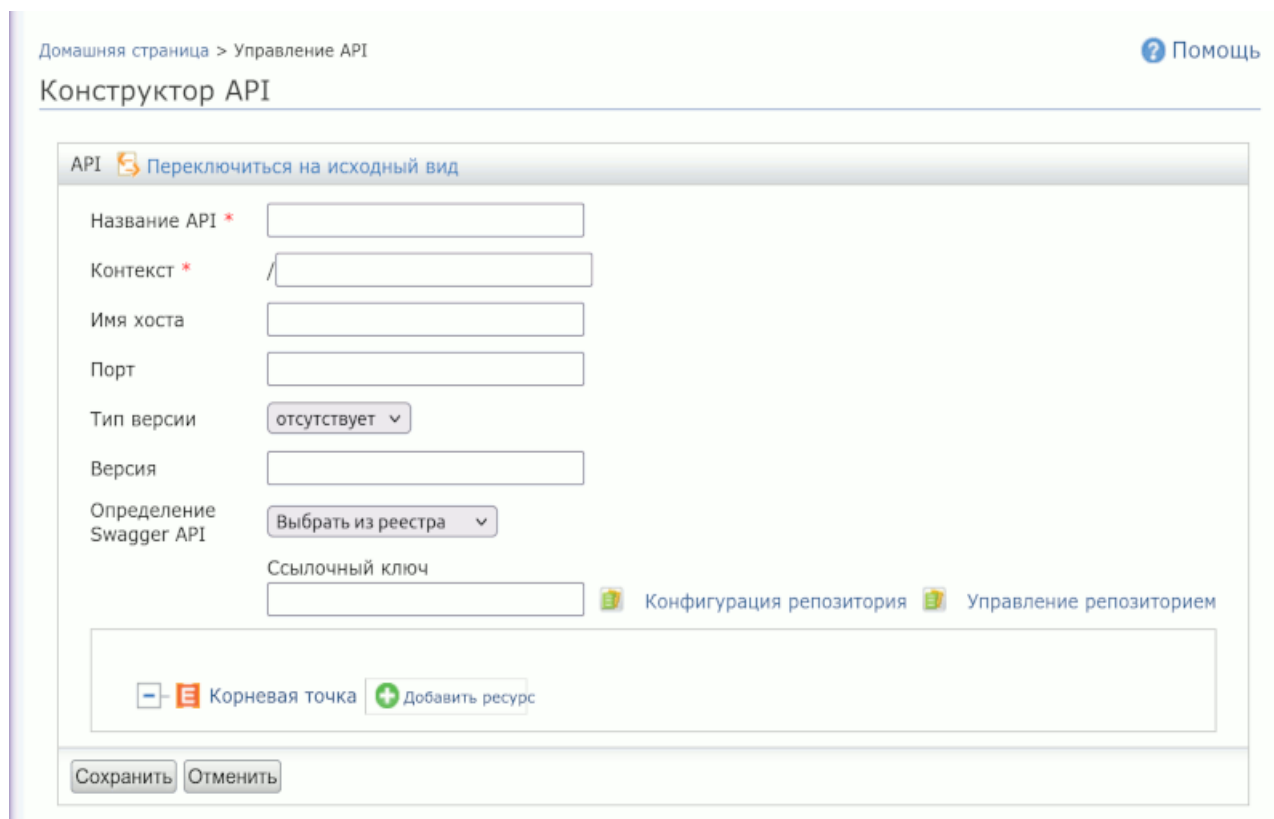


Рис. 6. Добавление API

API требует Имя, Контекст и не меньше одного Ресурса. Контекст всегда должен содержать предшествующий символ '/'. В окне проектирования этот символ добавлен по умолчанию, поэтому пользователю не требуется вводить его повторно. (Обратите внимание на символ / перед полем ввода Контекста). Имя узла может использоваться для фильтрации (входящие сообщения содержат 'имя узла' в своих заголовках) полученных ресурсов REST API с входящими запросами от клиента.

Щёлкните по ссылке **Добавить Ресурс** для добавления Ресурса к API. Появится форма для ввода информации Ресурса под секцией управления API, как показано на Рисунке 7.

Домашняя страница > Управление API ? Помощь

Конструктор API

API [Перекл​ючить​ся на исходный вид](#)

Название API *

Контекст *

Имя хоста

Порт

Тип версии отсутствует ▾

Версия

Определение Swagger API Выбрать из реестра ▾

Ссылочный ключ [Конфигурация репозитория](#) [Управление репозиторием](#)

[-](#) [+](#) Корневая точка [+](#) Добавить ресурс

Ресурсы

Метды GET POST PUT DELETE OPTIONS HEAD PATCH

Стиль URL Нет ▾

Входящая последовательность

Нет

Определить в соответствии

Выбрать из репозитория

Использовать существующую последовательность

Исходящая последовательность

Нет

Определить в соответствии

Выбрать из репозитория

Использовать существующую последовательность

Последовательность ошибок

Нет

Определить в соответствии

Выбрать из репозитория

Использовать существующую последовательность

Рис. 7. Информация о Ресурсе

Требования к Ресурсу:

- Ресурс должен содержать не меньше одного метода. Методы назначаются Ресурсу включением соответствующих флажков.
- API может содержать не больше одного Ресурса по умолчанию (Ресурса без шаблона uri и

без маппинга url).

– Ресурс может иметь Входную Последовательность, Выходную Последовательность и Последовательность обработки ошибок, конфигурация может быть выполнена на месте или выбором конфигурации из реестра или может использовать существующую последовательность, заданную в локальном реестре.

Щёлкните кнопку **Update** для обновления Ресурса. Секция информации о Ресурсе исчезнет и дочерний элемент Ресурса будет добавлен к дереву API, как показано на Рисунке 4.

Щелчок по иконке Ресурса в дереве загрузит секцию Информации о Ресурсе обратно вместе с детальным описанием указанного Ресурса.

The screenshot shows the 'API Constructor' (Конструктор API) interface. At the top, there is a breadcrumb 'Домашняя страница > Управление API' and a 'Помощь' (Help) link. The main title is 'Конструктор API'. Below the title, there is a section for 'API' with a link to 'Переключиться на исходный вид' (Switch to original view). The configuration form includes the following fields and controls:

- Название API * (API Name) - text input
- Контекст * (Context) - text input with a slash character
- Имя хоста (Host name) - text input
- Порт (Port) - text input
- Тип версии (Version type) - dropdown menu with 'отсутствует' (none) selected
- Версия (Version) - text input
- Определение Swagger API (Swagger API definition) - dropdown menu with 'Выбрать из реестра' (Select from registry) selected
- Ссылочный ключ (Reference key) - text input

Below the form, there are two links: 'Конфигурация репозитория' (Repository configuration) and 'Управление репозиторием' (Repository management). The main area displays an API tree structure:

- [-] Корневая точка (+) Добавить ресурс (Root node (+) Add resource)
- [-] Ресурс (Удалить) (Resource (Delete))
- [-] Ресурс (Удалить) (Resource (Delete))

At the bottom of the form, there are 'Сохранить' (Save) and 'Отменить' (Cancel) buttons.

Рис. 8. Дерево API

Когда все данные API и Ресурса будут введены, щёлкните кнопку «**Сохранить**», что приведёт к сохранению API в конфигурации ESB. Также будет показано подтверждающее сообщение, как изображено на рисунке ниже.

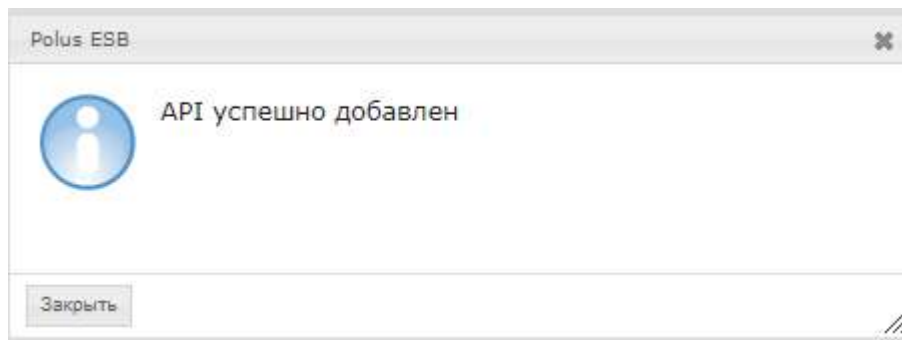


Рис. 9. Сообщение о добавлении API

Изменение API

Изменение сохранённых API выполняется щелчком по ссылке Edit на странице Развёрнутые API. Это перенаправит пользователя на страницу, на которой будет показана информация об API. В режиме редактирования изменение имени API невозможно.

Возможно Добавление новых Ресурсов, Изменение/Удаление существующих Ресурсов.

Когда изменение данных API будет закончено, щёлкните кнопку **Сохранить**, что приведёт к сохранению деталей API в конфигурации ESB. Будет показано подтверждающее сообщение, как изображено на рисунке ниже.

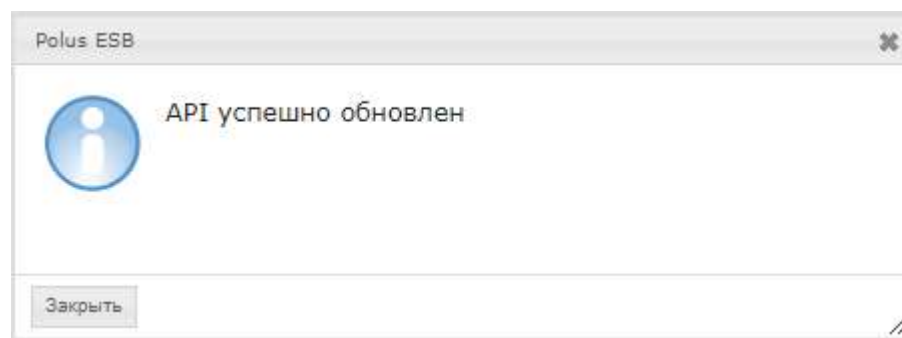


Рис.10. Сообщение об изменении API

Просмотр исходного кода

Щелчок по ссылке **Переключиться на исходный код** показывает исходный код API в формате xml. Виды исходного кода и режима конструктора синхронизированы. Изменения, сделанные в окне режима конструктора, отражаются в окне исходного кода и наоборот, как показано на рисунке ниже.

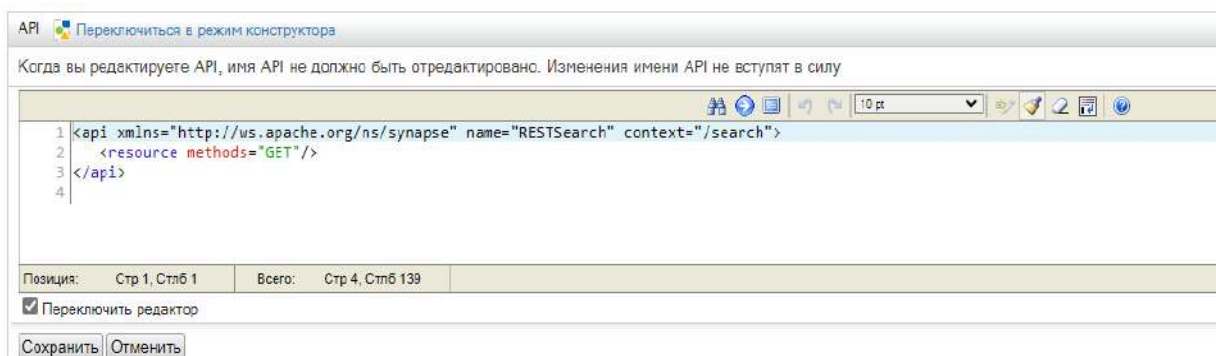


Рис.11. Просмотр исходного кода.

ПРОКСИ-СЕРВИСЫ

Прокси-сервис действует как размещённый (deployed) на Polus ESB сервис и обычно находится перед точкой вызова другого (внешнего) сервиса. Прокси-сервис может быть создан и сконфигурирован с использованием транспортного протокола, схемы, WSDL, или параметров Quality of Service (таких как WS-Security, WS-Reliable Messaging), отличных от них же у исходного сервиса. Он может обрабатывать сообщения перед тем, как они достигнут точки вызова, и ответы перед тем, как они достигнут клиента.

Клиенты могут отправлять запросы на прокси-сервисы непосредственно в ESB, поскольку клиент видит его как сервис, размещенный на Polus ESB. Например, они могут выполнять функцию `?wsdl` и показывать WSDL виртуального прокси-сервиса. В конфигурации ESB такие запросы можно обрабатывать как угодно. Вы можете обработать сообщение и отправить его в реальный сервис, возможно, работающий на другом компьютере (но в целом отправка сообщения — не обязательна). Вы можете указать набор задач, которые должны быть выполнены с сообщениями, полученными для прокси-сервиса, и завершить поток или отправить сообщение обратно клиенту, даже не отправляя его настоящему сервису.

Добавление Прокси-сервиса

В левом навигационном меню выберите **Прокси-сервис**. Это приведёт вас на страницу **Создать прокси-сервис из Шаблона**.

Выбрать шаблон	
Трансформирующий прокси	Создайте прокси-сервис, который преобразует все входящие запросы с помощью XSLT, а затем перенаправляет их на заданные точки вызова. При необходимости ответы, возвращаемые сервисом, также могут быть преобразованы.
Безопасный прокси	Создайте прокси с задействованным WS-Security. Сервис будет обрабатывать WS-Security входящие запросы и перенаправлять их в незащищенный сервис.
Прокси на основе WSDL	Создайте прокси-сервис из существующего WSDL веб-сервиса. Информация о точке вызова извлекается из удаленно размещенного WSDL сервиса.
Логирующий прокси	Создайте прокси-сервис, который регистрирует все входящие запросы и перенаправляет их на указанную точку вызова. При необходимости ответы, возвращаемые сервису, могут быть зарегистрированы до их направления клиенту.
Простой сервис-посредник	Создайте простой прокси-сервис для указанной точки вызова. Прокси-сервис не выполняет никакой дополнительной обработки входящих сообщений, и передает управление точке вызова
Пользовательский прокси	Запустите мастер создания прокси-сервиса и создайте новый прокси-сервис с нуля. Настройте каждый аспект прокси-сервера, включая очереди, точки вызова, транспорты и другие параметры QoS.

Рис. 12. Выбор шаблона прокси-сервиса

Выберите соответствующий вашим требованиям шаблон. Если вы хотите создать расширенный прокси-сервис, или если ни один из доступных шаблонов не соответствует вашим требованиям, выберите параметр **Пользовательский прокси-сервис** в конце списка шаблонов. Это запустит мастер создания прокси-сервиса, который позволит вам настроить все параметры

прокси-сервиса, включая последовательности, точки вызова, транспортные протоколы, параметры WSDL и параметры Quality of Service.

Страница **Создать Прокси-сервис из Шаблона** предоставляет вам следующий набор встроенных шаблонов для простого и быстрого создания и настройки прокси-сервисов.

- Простой сервис-посредник
- Безопасный прокси
- Прокси-сервис на основе WSDL
- Логирующий прокси-сервис
- Трансформирующий прокси-сервис
- Пользовательский прокси-сервис

Простой сервис-посредник

Этот шаблон можно использовать для создания простого прокси-сервиса для указанной точки вызова. Здесь нет никаких доступных настроек для определения последовательностей.

Этот прокси-сервис не выполняет никакой обработки проходящих через него сообщений.

The screenshot shows a web-based configuration page titled "Простой сервис-посредник" (Simple Proxy Service). At the top right, there is a "Помощь" (Help) link. Below the title, a descriptive text states: "Создайте простой прокси-сервис для указанной точки вызова. Прокси-сервис не выполняет никакой дополнительной обработки входящих сообщений, и передает управление точке вызова" (Create a simple proxy service for the specified endpoint. The proxy service does not perform any additional processing of incoming messages and passes control to the endpoint). The main configuration area is divided into three sections: 1. "Настройки прокси-сервиса" (Proxy Service Settings) with fields for "Имя прокси-сервиса*" (Proxy Service Name*), "Целевая точка вызова*" (Target Endpoint*) with radio buttons for "Введите URL" (Enter URL) and "Выбрать из реестра" (Select from registry), and "Целевой URL*" (Target URL*). 2. "Опубликовать параметры WSDL" (Publish WSDL Parameters) with a dropdown menu for "Публикация WSDL" (WSDL Publication) set to "Нет" (No). 3. "Транспорт" (Transport) with checkboxes for "vfs", "http" (checked), "https" (checked), and "local". At the bottom, there are "Создать" (Create) and "Отмена" (Cancel) buttons.

Рис.13. Простой сервис-посредник

Параметры

- *Настройки прокси-сервиса:* пользователь должен указать имя прокси-сервиса и URL точки вызова. Для точки вызова вводится непосредственно URL или выбирается запись из реестра.
- *Параметры WSDL:* пользователь может предоставить WSDL в виде записи реестра,

URL или встроенного (явного) определения WSDL. Если пользователь не укажет параметр WSDL, будет опубликован WSDL по умолчанию.

– *Настройки транспортного протокола*: здесь можно указать параметры транспортного протокола прокси-сервиса.

Примечание: Если пользователь настраивает ESB с Реестром управления и загружает все необходимые файлы в формате *.gar, он может предоставить только базовый WSDL в разделе «параметры WSDL». Нет необходимости предоставлять все необходимые импортированные схемы в качестве ресурсов WSDL, поскольку ESB может получить импортированные ресурсы, используя только базовый WSDL. Обратите внимание, что для вышеуказанной функции пользователю необходимо раскомментировать следующие строки файла @synapse.properties, который находится в папке ESB_HOME/lib/core/WEB_INF/classes.

```
synapse.wSDL.resolver=org.wso2.carbon.mediation.initializer.RegistryWSDLLocator
synapse.schema.resolver=org.wso2.carbon.mediation.initializer.RegistryXmlSchemaURIResolver
```

Безопасный прокси-сервис

Данный тип прокси-сервиса можно использовать в случае необходимости настройки WS-Security.

Безопасный прокси

Создайте прокси с задействованным WS-Security. Сервис будет обрабатывать WS-Security входящие запросы и перенаправлять их в незащищенный сервис.

Настройки прокси-сервиса

Имя прокси-сервиса*

Целевая точка вызова* Введите URL Выбрать из реестра

Целевой URL*

Политика безопасности* Реестр конфигурации Реестр

Опубликовать параметры WSDL

Публикация WSDL

Транспорт

vfs

http

https

local

Рис.14. Безопасный прокси-сервис

В дополнение к стандартным параметрам прокси-сервиса (настройки прокси-сервиса, настройки WSDL, настройки транспортного протокола - см. раздел «Простой сервис-посредник») пользователь должен указать политику безопасности.

Примечание: Если пользователь настраивает ESB с Реестром управления и загружает все необходимые файлы в формате *.gar, он может предоставить только базовый WSDL в разделе «параметры WSDL». Нет необходимости предоставлять все необходимые импортированные схемы в качестве ресурсов WSDL, поскольку ESB может получить импортированные ресурсы, используя только базовый WSDL. Обратите внимание, что для вышеуказанной функции пользователю необходимо раскомментировать следующие строки файла @synapse.properties, который находится в папке ESB_HOME/lib/core/WEB_INF/classes.

```
synapse.wSDL.resolver=org.wso2.carbon.mediation.initializer.RegistryWSDLLocator
synapse.schema.resolver=org.wso2.carbon.mediation.initializer.RegistryXmlSchemaURIResolver
```

Прокси-сервис на основе WSDL

Этот шаблон можно использовать для создания прокси-сервиса из WSDL существующего Web-сервиса. Информация о точке вызова извлекается из удаленно размещенного WSDL реального сервиса.

Прокси на основе WSDL ? Помощь

Создайте прокси-сервис из существующего WSDL веб-сервиса. Информация о точке вызова извлекается из удаленно размещенного WSDL сервиса.

Настройки прокси-сервиса

Имя прокси-сервиса*

WSDL-URI*

WSDL-сервис*

WSDL-порт*

Опубликовать тот же сервисный контракт

Варианты публикации сервисного контракта

Публикация WSDL

Транспорт

vfs

http

https

local

Рис.15. Прокси-сервис на основе WSDL.

В дополнение к стандартным параметрам для данного типа прокси-сервиса в настройках

нужно дополнительно указать URI WSDL, сервис (значение <wsdl:service>) и номер порта (<wsdl:port>).

Примечание: Если пользователь настраивает ESB с Реестром управления и загружает все необходимые файлы в формате *.gar, он может предоставить только базовый WSDL в разделе «параметры WSDL». Нет необходимости предоставлять все необходимые импортированные схемы в качестве ресурсов WSDL, поскольку ESB может получить импортированные ресурсы, используя только базовый WSDL. Обратите внимание, что для вышеуказанной функции пользователю необходимо раскомментировать следующие строки файла @synapse.properties, который находится в папке ESB_HOME/lib/core/WEB_INF/classes.

synapse.wsdl.resolver=org.wso2.carbon.mediation.initializer.RegistryWSDLLocator

synapse.schema.resolver=org.wso2.carbon.mediation.initializer.RegistryXmlSchemaURIResolver

Логирующий прокси-сервис

Этот шаблон можно использовать для создания прокси-сервиса, который логирует все входящие запросы и перенаправляет их заданной точке вызова. При необходимости ответы, возвращаемые конечной службой, могут быть залогированы до их направления клиенту.

Логирующий прокси

Создайте прокси-сервис, который регистрирует все входящие запросы и перенаправляет их на указанную точку вызова. При необходимости ответы, возвращаемые сервису, могут быть зарегистрированы до их направления клиенту.

Настройки прокси-сервиса

Имя прокси-сервиса*

Целевая точка вызова* Введите URL Выбрать из реестра

Целевой URL*

Уровень журнала запроса

Уровень журнала ответов

Опубликовать параметры WSDL

Публикация WSDL

Транспорт

vfs

http

https

local

Создать Отмена

Рис.16. Логирующий прокси-сервис

В дополнение к стандартным параметрам для данного типа прокси-сервиса нужно указать уровень логирования при обработке запроса и ответа.

Примечание: Если пользователь настраивает ESB с Реестром управления и загружает все необходимые файлы в формате *.gar, он может предоставить только базовый WSDL в разделе «параметры WSDL». Нет необходимости предоставлять все необходимые импортированные схемы в качестве ресурсов WSDL, поскольку ESB может получить импортированные ресурсы, используя только базовый WSDL. Обратите внимание, что для вышеуказанной функции пользователю необходимо раскомментировать следующие строки файла @synapse.properties, который находится в папке ESB_HOME/lib/core/WEB_INF/classes.

synapse.wSDL.resolver=org.wso2.carbon.mediation.initializer.RegistryWSDLLocator

synapse.schema.resolver=org.wso2.carbon.mediation.initializer.RegistryXmlSchemaURIResolver

Трансформирующий прокси-сервис

Этот шаблон можно использовать для создания прокси-сервиса, когда пользователь пытается реализовать сценарий, в котором происходит трансформация (изменение формата) сообщения. Данный тип прокси может преобразовывать запросы и ответы на основе указанных конфигураций XSLT.

Помощь

Трансформирующий прокси

Создайте прокси-сервис, который преобразует все входящие запросы с помощью XSLT, а затем перенаправляет их на заданные точки вызова. При необходимости ответы, возвращаемые сервисом, также могут быть преобразованы.

Настройки прокси-сервиса

Имя прокси-сервиса*

Целевая точка вызова* Введите URL Выбрать из реестра

Целевой URL*

Запросить XSLT Реестр конфигурации Реестр

Преобразование ответов

Опубликовать параметры WSDL

Публикация WSDL

Транспорт

vfs

http

https

local

Рис. 17. Трансформирующий прокси-сервис

В дополнение к стандартным параметрам необходимо указать XSLT-преобразование, выбрав его из реестра.

Примечание: Если пользователь настраивает ESB с Реестром управления и загружает все необходимые файлы в формате *.gar, он может предоставить только базовый WSDL в разделе «параметры WSDL». Нет необходимости предоставлять все необходимые импортированные схемы в качестве ресурсов WSDL, поскольку ESB может получить импортированные ресурсы, используя только базовый WSDL. Обратите внимание, что для вышеуказанной функции пользователю необходимо раскомментировать следующие строки файла @synapse.properties, который находится в папке ESB_HOME/lib/core/WEB_INF/classes.

synapse.wsdl.resolver=org.wso2.carbon.mediation.initializer.RegistryWSDLLocator
synapse.schema.resolver=org.wso2.carbon.mediation.initializer.RegistryXmlSchemaURIResolver

ТОЧКИ ВХОДА (INBOUND ENDPOINTS)

Точки входа предназначены для первичной обработки входного сообщения/запроса и передачи на уровень обработки (медиаторы, последовательности) без использования Axis2: точка входа, получив сообщение/запрос по одному из протоколов уровня приложения, непосредственно передает его последовательности, минуя Axis2.

В Polus ESB реализованы три категории точек входа, в каждой из них — несколько типов:

- прослушивающие (listening);
- опрашивающие (polling);
- основанные на событиях (event-based).

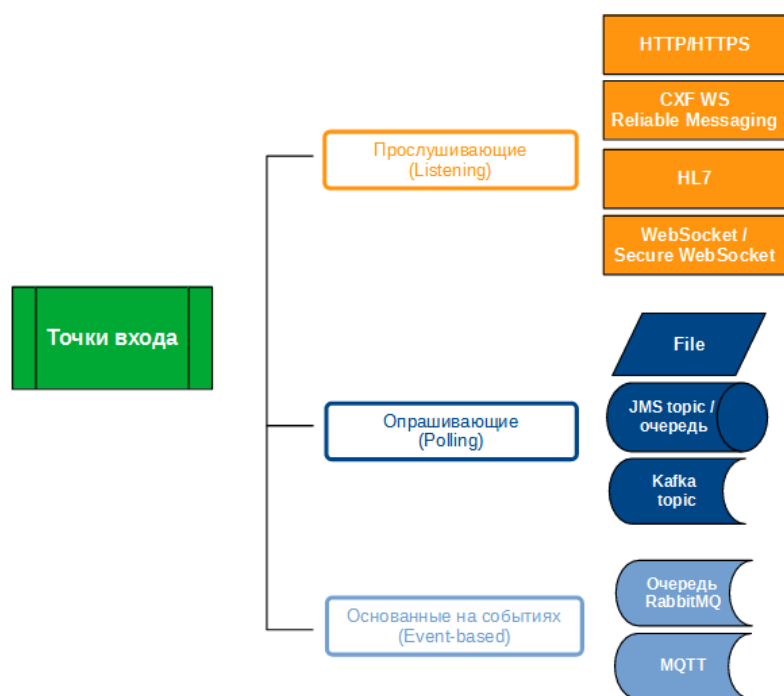


Рис.18. Типы точек входа (inbound endpoints)

Для точек входа всех типов необходимо определять *последовательность обработки сообщения* и *последовательность обработки ошибок*.

Прослушивающие точки входа принимают входящие запросы/сообщения на определенном *порту* и соответствуют следующим условиям:

- прием и последующая обработка запроса происходят синхронно;
- результат передается клиенту, отправившему запрос/сообщение точке входа;
- получив сообщение, прослушивающая точка входа передает его соответствующей *последовательности* для дальнейшей обработки;

- последовательность в результате обработки в итоге передает сообщение *точке вызова* и возвращает результат, полученный от точки вызова, исходной прослушивающей точке входа.

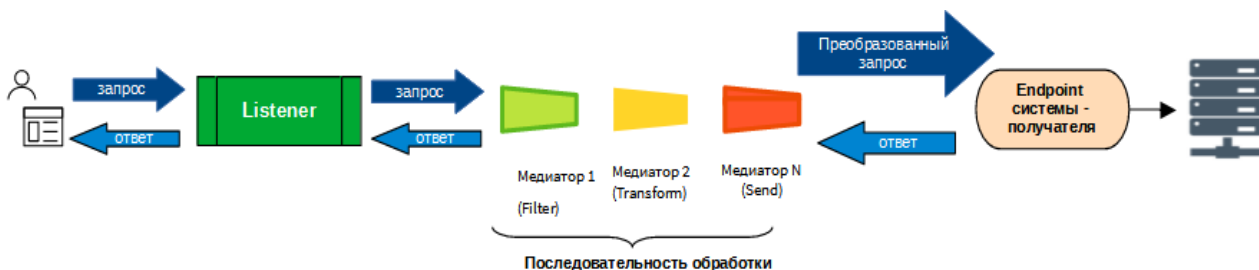


Рис.19. Прослушивающая точка входа (listener)

Опрашивающие точки входа периодически проверяют некий «контейнер данных» (каталог, JMS-очередь или JMS/Kafka топик), и как только новые данные становятся доступны — передают их на обработку соответствующей последовательности. Опрашивающие точки входа поддерживают асинхронную обработку.

Значение временного интервала опроса на наличие новых данных в «контейнере» - настраиваемый параметр.

Точки входа, **основанные на событиях**, однократно подключаются к «серверу событий» и далее «потребляют» (consume) события, т. е. реагируют на каждое вновь возникающее «событие» и передают соответствующее сообщение указанной последовательности.

Для каждого типа точек входа система поддерживает как predefined точки, так и возможность создавать пользовательские.

Прослушивающие точки входа

Система поддерживает следующий набор predefined прослушивающих точек входа:

Таблица 1. Подтипы прослушивающих точек входа

Подтип прослушивающей точки входа	Специфические для подтипа настраиваемые параметры
HTTP/HTTPS	<ul style="list-style-type: none"> – Порт, на котором точка входа будет «прослушивать» запросы; – параметры пула обработки запросов — оптимальное/максимальное количество потоков в пуле, время жизни дополнительно выделяемого потока; – параметры диспетчеризации — если сообщение нужно передать для дальнейшей обработки не последовательности, а REST API или Прокси-сервису;

	– параметры хранилища ключей для HTTPS точки входа.
<i>CXF WebServices Reliable Messaging</i>	Подтип предназначен для гарантированной (в соответствии со спецификацией <i>WS Reliable Messaging</i>) обработки входных SOAP-сообщений и должен конфигурироваться следующими параметрами: <ul style="list-style-type: none"> – порт для «прослушивания» входных SOAP-сообщений; – путь к конфигурационному файлу CXF; – флаг использования SSL.
<i>HL7</i>	Порт, параметры валидации, величина тайм-аута и другие параметры, специфичные для протокола
<i>WebSocket / Secure WebSocket</i>	<ul style="list-style-type: none"> – Порт, на котором точка входа будет «прослушивать» запросы; – параметры настройки оповещения клиента; – параметры пула обработки запросов; – тип обрабатываемого контента; – параметры хранилища ключей и алгоритмов шифрования для точки входа <i>Secure WebSocket</i>.

Параметры типов прослушивающих точек входа

Точки входа HTTP

Параметры конфигурации рабочего пула.

По умолчанию точки входа HTTP совместно используют пул рабочих потоков для обработки входящих запросов. Если вам нужен отдельный пул рабочих потоков для точки входа, вам необходимо настроить следующие параметры:

Таблица 2. Параметры точки входа HTTP

Параметр	Описание	Значение по умолчанию
<code>inbound.worker.pool.size.core</code>	Начальное количество потоков в пуле рабочих потоков. Это значение можно соответствующим образом изменить в зависимости от количества обрабатываемых сообщений. Максимальное значение, которое здесь можно указать, — это значение параметра <code>inbound.worker.pool.size.max</code> .	400
<code>inbound.worker.pool.size.max</code>	Максимальное количество потоков в пуле рабочих потоков. Укажите максимальное ограничение, чтобы избежать снижения производительности, которое может произойти из-за переключения контекста.	500
<code>inbound.worker.thread.keep.alive.seconds</code>	Время поддержания активности дополнительных потоков в рабочем пуле. Это значение должно быть меньше времени ожидания сокета. Когда это время истечет для	60

Параметр	Описание	Значение по умолчанию
	дополнительного потока, он будет уничтожен. Назначение этого параметра — оптимизировать использование ресурсов, избегая потерь, возникающих в результате наличия дополнительных неиспользуемых потоков.	
<code>inbound.worker.pool.queue.length</code>	Длина очереди, используемой для хранения готовых к выполнению задач, которые должны выполняться рабочим пулом. Пул потоков начинает ставить задания в очередь, когда все существующие потоки заняты и в пуле достигнуто максимальное количество потоков. Значение этого параметра должно быть -1, чтобы использовать неограниченную очередь. Если используется ограниченная очередь, и очередь заполняется до предела, и любая дальнейшая попытка отправки заданий завершится неудачей, в результате чего синапс отбросит некоторые сообщения.	-1
<code>inbound.thread.group.id</code>	Уникальный идентификатор группы потоков.	Группа входящих рабочих потоков PassThrough
<code>inbound.thread.id</code>	Уникальный идентификатор потока.	PassThroughInboundWorkerThread
<code>dispatch.filter.pattern</code>	Регулярное выражение, определяющее прокси-сервисы и API для предоставления доступа через точку входа. Укажите выражение <code>.*</code> , чтобы предоставить все прокси-сервисы и API, или укажите выражение, подобное <code>^(/foo /bar /services/MyProxy)\$</code> , чтобы определить набор сервисов для предоставления через точку входа. Если вы не укажете выражение, будет доступна только определенная последовательность точки входа.	blank

Параметр точки входа для прокси-сервисов

Если прокси-сервис должен предоставляться только через точки входа, в конфигурации прокси-сервера необходимо установить следующий параметр сервиса.

Таблица 3. Параметры прокси-сервиса, доступного только через точки входа

Параметр сервиса	Описание	Значение по умолчанию
inbound.only	<p>Должен ли прокси-сервис предоставляться только через точки входа.</p> <p>Если установлено значение true, все запросы, которые прокси-сервис получает через обычный транспорт, будут отклонены. Прокси-сервис будет обрабатывать только те запросы, которые получены через точки входа.</p>	false

Точки входа HTTPS

Таблица 4. Параметры, специфичные для точки входа HTTPS

Параметр	Описание	Обязательный?
inbound.http.port	Порт, на котором должен быть запущен прослушиватель точки вызова.	Да
keystore	Расположение KeyStore, где хранятся ключи.	Да
truststore	Расположение TrustStore, в котором хранятся ключи.	Нет
SSLVerifyClient	Используется при включении взаимной проверки.	Нет
HttpsProtocols	Поддерживаемые протоколы.	Нет
SSLProtocol	Поддерживаемый протокол SSL.	Нет
CertificateRevocationVerifier	<p>Когда для атрибута enable установлено значение true, это подтверждает и проверяет статус отзыва сертификатов хоста с помощью OCSP/CRL при установлении HTTPS-соединений.</p> <p>Если для атрибута enable этого параметра установлено значение true, вам также необходимо указать CacheSize и CacheDelay. CacheSize — максимальный размер кэша. CacheDelay — продолжительность времени между двумя последовательными запланированными задачами управления кэшем, которые выполняют служебную работу для кэша.</p>	Нет

Параметры конфигурации рабочего пула для точки входа HTTPS – те же, что и для точки входа HTTP.

Точки входа WebSocket

Таблица 5. Параметры точки входа WebSocket

Параметр	Описание	Обязательный?
inbound.ws.port	Порт прослушивателя netty, на котором прослушивается входящий трафик WebSocket.	Да
ws.client.side.broadcast.level	Уровень клиентской широковещательной рассылки, который определяет, как кадры WebSocket передаются от точки входа WebSocket клиенту. Широковещательная рассылка происходит на основе абонентского пути клиента, подключенного к точке входа WebSocket. Возможны следующие три уровня: 0 — только уникальный клиент может получить кадр от точки входа WebSocket. 1 — все клиенты, подключенные к одному и тому же абонентскому пути, получают кадр WebSocket. 2 — Все клиенты, подключенные к одному и тому же абонентскому пути, кроме того, кто публикует фрейм во входящие, получают фрейм WebSocket.	Да
ws.outflow.dispatch.sequence	Последовательность для медиации между серверной частью и клиентом.	Да
ws.outflow.dispatch.fault.sequence	Последовательность сбоев для пути медиации между серверной частью и клиентом.	Да
ws.boss.thread.pool.size	Размер netty boss пула	Нет
ws.worker.thread.pool.size	Размер пула рабочих потоков.	Нет
ws.subprotocol.handler.class	Укажите один или несколько необходимых настраиваемых классов обработчиков подпротокола. Разделяйте каждый пользовательский класс обработчика подпротокола точкой с запятой.	Нет
ws.default.content.type	Указывает тип содержимого кадров веб-сокета, получаемых от точки входа.	Нет
ws.shutdown.status.code	Указывает код состояния кадра закрытого веб-сокета, отправляемого при закрытии точки входа.	Нет
ws.shutdown.status.message	Указывает сообщение о состоянии закрытого фрейма веб-сокета, когда точка входа закрыта.	Нет
ws.pipeline.handler.class	Полное имя класса реализованного вами класса обработчика конвейера.	Нет

Точки входа Secure WebSocket

Таблица 6. Параметры точки входа Secure WebSocket

Параметр	Описание	Обязательный?
inbound.ws.port	Порт прослушивателя netty, на котором прослушивается входящий трафик WebSocket.	Да
ws.client.side.broadcast.level	Уровень клиентской широковещательной рассылки, который определяет, как кадры WebSocket передаются от точки входа WebSocket клиенту. Широковещательная рассылка происходит на основе абонентского пути клиента, подключенного к точке входа WebSocket. Возможны следующие три уровня: 0 — только уникальный клиент может получить кадр от точки входа WebSocket. 1 — все клиенты, подключенные к одному и тому же абонентскому пути, получают кадр WebSocket. 2 — Все клиенты, подключенные к одному и тому же абонентскому пути, кроме того, кто публикует фрейм во входящие, получают фрейм WebSocket.	Да
ws.outflow.dispatch.sequence	Последовательность для медиации между серверной частью и клиентом.	Да
ws.outflow.dispatch.fault.sequence	Последовательность сбоев для пути медиации между серверной частью и клиентом.	Да
wss.ssl.key.store.file	Расположение хранилища ключей, в котором хранятся ключи.	Да
wss.ssl.key.store.pass	Пароль для доступа к файлу хранилища ключей.	Да
wss.ssl.trust.store.file	Местоположение хранилища доверенных сертификатов, в котором хранятся ключи.	Да
wss.ssl.trust.store.pass	Пароль для доступа к файлу хранилища доверенных сертификатов.	Да
wss.ssl.cert.pass	Пароль сертификата SSL.	Да
ws.boss.thread.pool.size	Размер netty boss пула	Нет
ws.worker.thread.pool.size	Размер пула рабочих потоков.	Нет
ws.subprotocol.handler.class	Укажите один или несколько необходимых настраиваемых классов обработчиков подпротокола. Разделяйте каждый пользовательский класс обработчика подпротокола точкой с запятой.	Нет
ws.default.content.type	Указывает тип содержимого кадров веб-сокета, получаемых от точки входа.	Нет
ws.shutdown.status.code	Указывает код состояния кадра закрытого веб-сокета, отправляемого при закрытии точки входа.	Нет

Параметр	Описание	Обязательный?
ws.shutdown.status.message	Указывает сообщение о состоянии закрытого фрейма веб-сокета, когда точка входа закрыта.	Нет
ws.pipeline.handler.class	Полное имя класса реализованного вами класса обработчика конвейера.	Нет
wss.ssl.protocols	Включает протокол SSL для конкретной точки входа WebSocket. Значение по умолчанию — «TLS». Вы можете изменить его на версию (версии) TLS, которые включены с протоколом SSL (т. е. TLSv1, TLSv1.1, TLSv1.2). Например, <code><parameter name="wss.ssl.protocols">TLSv1.1,TLSv1.2</parameter></code>	Нет
wss.ssl.cipher.suites	Включает указанные наборы шифров для конкретной точки входа WebSocket. Например, <code><parameter name="wss.ssl.cipher.suites">TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_DHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_128_CBC_SHA</parameter></code>	Нет

Точка входа HL7

Таблица 7. Параметры точки входа HL7

Параметр	Описание	Значение по умолчанию	Возможные значения
inbound.hl7.Port	Порт, на котором нужно запустить прослушиватель MLLP.	N/A	Номер порта
inbound.hl7.AutoAck	Следует ли отправлять автоматическое подтверждение при получении сообщения. Если установлено значение false, вы можете определить тип отправляемого подтверждения HL7. Дополнительные сведения см. в разделе Свойства уровня медиации точки входа HL7.	true	true false
inbound.hl7.ValidateMessage	Это включает проверку сообщений HL7.	true	true false
inbound.hl7.TimeOut	Интервал ожидания в миллисекундах для запуска сообщения NACK.	10000	[0..9]*
inbound.hl7.CharSet	Набор символов, используемый для	UTF-8	ISO-8859-1

Параметр	Описание	Значение по умолчанию	Возможные значения
	кодирования и декодирования сообщений. Некоторые многобайтовые кодировки символов (например, UTF-16, UTF-32) могут привести к тому, что значения байтов будут равны символам кодирования MLLP или значениям байтов меньше 0x1F, что приведет к ошибкам.		UTF-8 US-ASCII
inbound.hl7.BuildInvalidMessages	Если для параметра inbound.hl7.ValidateMessage задано значение false и входящее сообщение является недопустимым, этот параметр указывает, должно ли необработанное сообщение, полученное через транспорт MLLP, передаваться на уровень медиации.	false	true false
inbound.hl7.PassthroughInvalidMessages	Если для параметра inbound.hl7.BuildInvalidMessages задано значение true, этот параметр уведомляет транспортный отправитель Axis2 HL7 о необходимости использования необработанного сообщения.	false	true false
inbound.hl7.MessagePreProcessor	Здесь можно определить реализацию интерфейса org.wso2.carbon.inbound.endpoint.protocol.hl7.HL7MessagePreprocessor. Он предоставляет точку расширения для перехвата входящих сообщений до того, как произойдет какой-либо анализ сообщений.	N/A	Полное имя класса.

Ниже приведены свойства уровня медиации, которые можно задать для точки входа HL7. Область действия этих свойств является областью действия по умолчанию.

Таблица 8. Свойства уровня медиации точки входа HL7

Свойство	Описание
<property name="HL7_RESULT_MODE" value="ACK NACK" scope="default"/>	Это используется для определения типа отправляемого подтверждения HL7. Если для параметра inbound.hl7.AutoAck установлено значение true, это свойство не действует.
<property name="HL7_NACK_MESSAGE" value="<ERROR MESSAGE>" scope="default" />	Это используется для определения пользовательского сообщения об ошибке, которое будет отправлено, если вы установили свойство HL7_RESULT_MODE как NACK.
<property	Если для параметра inbound.hl7.AutoAck установлено

Свойство	Описание
name="HL7_APPLICATION_ACK" value="true" scope="default"/>	значение false и клиенту не отправляется немедленный автоматически сгенерированный ACK, это свойство определяет, должны ли мы автоматически генерировать ACK для запроса после завершения потока передачи. Если для параметра inbound.hl7.AutoAck и этого свойства задано значение false, в качестве ответа необходимо сгенерировать сообщение ACK в правильном формате.

Настройка точки входа HL7

Точку входа HL7 можно настроить с помощью файла <EI_HOME>/conf/hl7.properties.

Поддерживаемый список параметров настройки для этого файла выглядит следующим образом:

Таблица 9. Настройки файла свойств точки входа HL7

Property	Description	Value
hl7_id_generator	По умолчанию библиотека синтаксического анализа NAPI HL7 использует генератор идентификаторов на основе файлов для создания уникальных идентификаторов элементов управления. Чтобы использовать генератор идентификаторов на основе UUID, вы можете изменить его на «uuid».	file uuid (default = file)
worker_threads_core	Определяет размер пула входящих рабочих потоков HL7.	[0..9]* (default = 100)
io_thread_count	Определяет количество потоков ввода-вывода, которые использует IO Reactor. Рекомендуется установить это значение равному числу ядер на машине.	[0..9]* (default = 2)
so_timeout	Определяет время ожидания сокета TCP.	[0..9]* (default = 0)
connect_timeout	Определяет время ожидания соединения TCP.	[0..9]* (default = 0)
so_keep_alive	Определяет сохранение активности сокета TCP.	true false (default = true)
so_rcvbuf	Определяет размер приемного буфера сокета TCP.	[0..9]* (default = 0 использует значение ОС по умолчанию. Максимальное значение зависит от настроек ОС).
so_sndbuf	Определяет размер буфера отправки сокета TCP.	[0..9]* (default = 0 использует значение ОС по умолчанию. Максимальное значение зависит от настроек ОС).

Для настройки пользовательской прослушивающей точки, помимо параметров, общих для всех точек входа (*последовательность обработки сообщений и последовательность обработки ошибок*) достаточно определить только разработанный пользователем класс реализации.

Опрашивающие точки входа

Система поддерживает следующий набор predefined опрашивающих точек входа:

Таблица 10. Подтипы опрашивающих точек входа

Подтип прослушивающей точки входа	Специфические для подтипа настраиваемые параметры
<i>Файл</i>	<i>Подтип предназначен для обработки файлов в определенном каталоге и должен конфигурироваться следующими параметрами:</i> <ul style="list-style-type: none">– каталог для хранения входных файлов («контейнер данных»);– каталоги для хранения обработанных файлов и файлов, при обработке которых возникла ошибка;– шаблон имени обрабатываемых файлов;– тип контента;– интервал опроса;– параметры, специфичные для FTP/SFTP.
<i>JMS: очередь или топик</i>	<i>– URL для JNDI-провайдера;</i> <ul style="list-style-type: none">– интервал опроса;– специфичные для JMS параметры (JNDI-имя очереди или топика, параметры аутентификации, количество параллельных потоков вычитывания сообщений, тип контента, тайм-аут).
<i>Кafka-топик</i>	<i>– Хост и порт для Zookeeper-сервера;</i> <ul style="list-style-type: none">– интервал опроса;– специфичные для Apache Kafka параметры (тип потребителя сообщений (consumer), тип контента, перечень топиков, параметры Kafka consumer groups, тайм-аут, количество потоков).

Таблица 11. Параметры точки входа типа Файл

Имя параметра	Описание	Обязательный ?	Возможные значения	Значение по умолчанию
interval	Продолжительность времени в миллисекундах между двумя сканированиями файлов, которые проверяют наличие обновлений.	Yes		
sequential	Файлы будут обрабатываться последовательно, если для этого параметра установлено значение true.	Yes		true
coordination	Это должно быть «true» для кластерных развертываний, чтобы два узла не могли получить один и тот же файл.	Yes		true
transport.vfs.FileURI	<p>URI расположения ваших файлов. Это должно быть исходное расположение файлов (если вы настраиваете ESB для чтения файлов) или место назначения файлов (если вы настраиваете ESB для отправки файлов). Вы можете указать параметры уровня соединения в URL-адресе (см. Параметры URL-адреса VFS).</p> <p>Когда вам нужно получить доступ к абсолютному пути URL-адреса, вы можете определить URL-адрес с помощью sftpPathFromRoot, как показано ниже. Также обратите внимание, что transport.vfs.AvoidPermissionCheck является обязательным параметром для этого URL-адреса при использовании FTPS.</p> <pre><parameter name="transport.vfs.FileURI"> sftp://[username[: password]@] hostname[: port] [absolute-path]?sftpPathFromRoot=true; transport.vfs.AvoidPermissionCheck=true</parameter></pre>	Yes	Действительный URI файла в следующей форме: file://<путь>	
transport.vfs.ContentType	<p>Тип содержимого файлов, обрабатываемых транспортом. Чтобы указать кодировку при чтении файла, укажите после типа содержимого точку с запятой и набор символов. Например:</p> <pre><parameter name="transport.vfs.ContentType">text/plain;charset=UTF-32</parameter></pre> <p>При записи файла вы можете установить другую кодировку с помощью свойства CHARACTER_SET_ENCODING:</p> <pre><property name="CHARACTER_SET_ENCODING" value="UTF-8" scope="axis2" type="STRING"/></pre>	Yes	Допустимый тип содержимого для файлов (например, text/xml). Вы можете указать кодировку после типа контента, например: text/plain;charset=UTF-32	
transport.vfs.FileNamePattern	Если VFS listener должен обрабатывать только подмножество файлов, доступных в указанном местоположении файла URI, используйте этот параметр, чтобы выбрать эти файлы по имени с помощью регулярного выражения.	No	Регулярное выражение для выбора файлов по имени (например, *.xml)	
transport.vfs.ActionAfterProcess	Следует ли перемещать или удалять файлы после их обработки транспортом.	No	MOVE или DELETE	DELETE
transport.vfs.ActionAfterFailure	Следует ли перемещать или удалять файлы в случае сбоя.	No	MOVE или DELETE	DELETE
transport.vfs.MoveAfterProcess	Куда переместить файлы после обработки, если значение, указанное как transport.vfs.ActionAfterProcess, равно MOVE .	Да, если ActionAfterProcess равно MOVE	Действительный URI файла	
transport.vfs.MoveAfterFailure	Куда переместить файлы после обработки, если значение, указанное как transport.vfs.ActionAfterFailure, равно MOVE .	Да, если ActionAfterFailure равно MOVE	Действительный URI файла	
transport.vfs.ReplyFileURI	Место, куда транспорт должен записывать файлы ответов.	No	Действительный URI	

Имя параметра	Описание	Обязательный ?	Возможные значения	Значение по умолчанию
			файла	
transport.vfs.ReplyFileName	Имя для файлов ответов, записываемых транспортом.	No	Допустимое имя файла	response.xml
transport.vfs.MoveTimestampFormat	Шаблон/формат метки времени, добавляемой к именам файлов в качестве префиксов при перемещении файлов.	No	Действительный шаблон временной метки (например: уууу-ММ-дд"ТНН:mm:ss.SSSZ)	
transport.vfs.Streaming	Должны ли файлы передаваться в потоковом режиме, что полезно при передаче больших файлов.	No	True или false	false
transport.vfs.ReconnectTimeout	Значение времени ожидания повторного подключения в секундах, которое будет использоваться в случае ошибки при передаче файлов.	No	Положительное целое число	30 sec
transport.vfs.MaxRetryCount	Максимальное количество повторных попыток в случае ошибок.	No	Положительное целое число	3
transport.vfs.MoveAfterFailedMove	Место для перемещения сбойного файла.	No	Действительный URI файла	
transport.vfs.FailedRecordsFileName	Имя файла, в котором хранится список файлов с ошибками.	No	Допустимое имя файла	vfs-move-failed-records. характеристики
transport.vfs.FailedRecordsFileDestination	Место для хранения файла ошибочных записей.	No	URI папки	repository/conf/
transport.vfs.MoveFailedRecordTimestampFormat	Формат метки времени для записей в файле ошибочных записей. Файл записей о сбоях содержит имя файла, в котором произошел сбой, и отметку времени его сбоя.	No	Действительный шаблон временной метки (например: уууу-ММ-дд"ТНН:mm:ss.SSSZ)	dd-MM-yyуу НН:mm:ss
transport.vfs.FailedRecordNextRetryDuration	Время ожидания в миллисекундах перед повторной попыткой выполнения задачи перемещения.	No	Положительное целое число	3000 миллисекунд
transport.vfs.Locking	По умолчанию блокировка файлов включена в транспорте VFS. Этот параметр позволяет настроить поведение блокировки для каждого сервиса отдельно. Вы также можете отключить блокировку глобально, указав параметр на уровне получателя и выборочно включив блокировку только для набора сервисов.	No	enable или disable	enable
transport.vfs.FileProcessCount	Этот параметр позволяет регулировать VFS listener, обрабатывая файлы пакетами. Укажите количество файлов, которые вы хотите обработать в каждом пакете.	No	Положительное целое число, например 10	N/A
transport.vfs.FileProcessInterval	Интервал в миллисекундах между пакетами обработки файлов.	No	Положительное целое число, например 1000	N/A
transport.vfs.DistributedLock	Это применимо только в кластерных развертываниях. Установите значение true, если вам нужно избежать одновременной обработки одного и того же файла несколькими серверами.	No	true или false	N/A
transport.vfs.DistributedTimeout	Время ожидания в секундах для распределенной блокировки.	No	Положительное целое число,	N/A

Имя параметра	Описание	Обязательный ?	Возможные значения	Значение по умолчанию
			например 10	
transport.vfs.AutoLockRelease	Установите значение true, если вам нужно снять блокировку, чтобы файлы не обрабатывались из-за ошибочной блокировки. Это работает вместе с параметрами transport.vfs.AutoLockReleaseInterval и transport.vfs.LockReleaseSameNode.	No	true или false	N/A
transport.vfs.AutoLockReleaseInterval	Интервал снятия блокировки в миллисекундах.	No	Положительное целое число, например 1000	N/A
transport.vfs.LockReleaseSameNode	Установите значение true, если вам нужно снять блокировки только на том же рабочем узле. Если для этого параметра установлено значение false, блокировки, накопленные другими узлами, будут сняты в соответствии со значением, указанным в transport.vfs.AutoLockReleaseInterval .	No	true или false	true
transport.vfs.FileSortAttribute	Атрибут, по которому файлы должны сортироваться и обрабатываться.	No	NONE, Name, Size и Lastmodified timestamp	N/A
transport.vfs.FileSortAscending	Порядок сортировки для сортировки и обработки файлов. Если установлено значение true, файлы будут сортироваться в порядке возрастания на основе атрибута, указанного вами в transport.vfs.FileSortAttribute.	No	true или false	true
transport.vfs.CreateFolder	Установите значение true, чтобы создать папку, если папка не существует при перемещении файлов.	No	true или false	false
transport.vfs.SubFolderTimestampFormat	Шаблон/формат временных меток, добавляемых в структуру папок при перемещении файлов. Вам необходимо установить для transport.vfs.CreateFolder значение true, чтобы указать значение для этого параметра.	No	Действительный шаблон временной метки (например: yyyy-MM-dd'THH:mm:ss.SSSZ)	N/A
transport.vfs.Build	Установите значение true, если вам нужно создать содержимое внутри файла перед внедрением файла в механизм медиации. В случае ошибки сборки файл не будет внедрен в механизм медиации.	No	true или false	false

Таблица 12. Параметры точки входа JMS

Имя параметра	Описание	Обязательный?	Возможные значения	Значение по умолчанию
java.naming.factory.initial	Класс фабрики начального контекста JNDI. Класс должен реализовать интерфейс java.naming.spi.InitialContextFactory.	Yes	Допустимое имя класса	-
java.naming.provider.url	URL-адрес поставщика JNDI.	Yes	Действительный URL	-
transport.jms.ConnectionFactoryJNDIName	Имя JNDI фабрики соединений.	Yes	-	-
sequential	Должны ли сообщения опрашиваться и вводиться последовательно или нет.	Yes	true, false	true
transport.jms.ConnectionFactoryType	Тип фабрики соединений.	No	queue, topic	queue
transport.jms.Destination	JNDI-имя пункта назначения.	No	-	Значением по умолчанию является имя

Имя параметра	Описание	Обязательный?	Возможные значения	Значение по умолчанию
				сервиса.
transport.jms.SessionAcknowledgement	Режим подтверждения сеанса JMS.	No	AUTO_ACKNOWLEDGE, CLIENT_ACKNOWLEDGE, DUPS_OK_ACKNOWLEDGE, SESSION_TRANSACTED	AUTO_ACKNOWLEDGE
transport.jms.CacheLevel	Уровень кэша ресурсов JMS.	No	0-нет, 1-подключение, 2-сеанс, 3-потребитель Чтобы подписаться на темы, установите значение transport.jms.CacheLevel равным 3.	0
transport.jms.UserName	Имя пользователя подключения JMS.	No	-	-
transport.jms.Password	Пароль подключения JMS.	No	-	-
transport.jms.JMSSpecVersion	Версия API JMS.	No	1.0.2b, 1.1, 2.0	1.1
transport.jms.SubscriptionDurable	Является ли фабрика соединений устойчивой по подписке или нет.	No	true, false	false
transport.jms.DurableSubscriberClientId	Параметр ClientId при использовании устойчивых подписок.	Требуется, если значение, указанное как transport.jms.SubscriptionDurable, равно true.	-	-
transport.jms.DurableSubscriberName	Имя постоянного подписчика.	Требуется, если значение, указанное как transport.jms.SubscriptionDurable, равно true.	-	-
transport.jms.MessageSelector	Реализация селектора сообщений.	No	-	
transport.jms.ReceiveTimeout	Время ожидания сообщения JMS во время опроса. Установите значение этого параметра в отрицательное целое число, чтобы ждать бесконечно. Установите его на ноль, чтобы предотвратить ожидание.	No	Количество миллисекунд ожидания.	1
transport.jms.ContentType	Как входящий listener должен определять тип содержимого полученных сообщений. Приоритет всегда отдается типу сообщения JMS.	No	Простое строковое значение, и в этом случае listener транспорта предполагает, что полученные сообщения всегда имеют указанный тип содержимого или набор правил. Дополнительные сведения см. на странице http://axis.apache.org/axis2/java/transport/jms.html#Service_configuration .	-
transport.jms.ContentTypeProperty	Получите тип содержимого из свойства сообщения.	No	contentType	-
transport.jms.ReplyDestination	Место назначения, в котором хранится ответ, сгенерированный серверной службой.	No	-	ReplyTo из сообщения

Имя параметра	Описание	Обязательный?	Возможные значения	Значение по умолчанию
transport.jms. PubSubNoLocal	Должны ли сообщения публиковаться через то же соединение, через которое они были получены.	No	true, false	false
transport.jms. SharedSubscription	Если установлено значение true, сообщения будут пересылаться только одному из потребителей, а потребители будут делиться сообщениями, опубликованными в теме.	No	true, false	false
pinnedServers	Список имен серверов синапсов, разделенных запятыми или пробелами, на которых должна быть развернута эта точка входа. Если нет закрепленного списка серверов, конфигурация точки входа будет развернута на всех экземплярах сервера.	No	Список допустимых имен серверов синапсов	-
transport.jms. ConcurrentConsumers	Количество параллельных потоков, которые необходимо запустить для обработки сообщений при опросе.	No	Любое положительное целое число. Для топиков это всегда должно быть 1.	1
transport.jms. retry.duration	Интервал повторных попыток повторного подключения к серверу JMS.	No	Интервал повтора в миллисекундах.	-
transport.jms. RetriesBeforeSuspension	Количество последовательных сбоев медиации, после которых опрос должен быть приостановлен.	No	Укажите любое положительное числовое значение в зависимости от ваших требований. Опрос будет приостановлен, когда число отказов медиации достигнет указанного значения.	-
transport.jms. PollingSuspensionPeriod	Период времени, в течение которого опрос должен быть приостановлен, если задан параметр transport.jms.RetriesBeforeSuspension.	No	Укажите требуемый период времени в миллисекундах.	60000

Таблица 13. Параметры точки входа Kafka

Имя параметра	Описание	Обязательный ?	Возможные значения	Значение по умолчанию
zookeeper.connect	Хост и порт сервера ZooKeeper (hostname:port).	Yes	localhost:2181	
consumer.type	Тип конфигурации потребителя. Это может быть либо простой, либо высокоуровневый.	Yes	highlevel, simple	
interval	Интервал опроса для точки входа для опроса сообщений.	Yes		
coordination	Если установлено значение true в настройке кластера, это будет запускать входящий трафик только на одном рабочем узле.	Yes	true, false	true
sequential	Поведение при выполнении данной последовательности.	Yes	true, false	true
topics	Категория для подачи сообщений. Конфигурация kafka высокого уровня может иметь более одного топика. Вы можете указать несколько названий топиков в виде значений, разделенных запятыми.	Yes		
content.type	Содержание сообщения.	Yes	application/xml, application/json	
group.id	Если все экземпляры потребителей имеют одну и ту же группу потребителей, это работает как традиционная очередь, балансирующая нагрузку на потребителей. Если все экземпляры потребителей имеют разные группы потребителей, это работает как публикация-подписка, и все сообщения передаются всем потребителям.	Yes		
thread.count	Количество потоков.	No		1
consumer.id	Идентификатор потребителя.	No		null
socket.timeout.ms	Время ожидания сокета для сетевых запросов.	No		30 * 1000
socket.receive.buffer.bytes	Буфер приема сокета для сетевых запросов.	No		64 * 1024
fetch.message.max.bytes	Количество байтов сообщений, которые нужно попытаться извлечь для каждого раздела топика в каждом запросе на выборку.	No		1024 * 1024
num.consumer.fetchers	Потоки сборщика чисел, используемые для выборки данных.	No		1
auto.commit.enable	Установленное смещение, используемое в качестве позиции, с которой новый потребитель начнет работу в случае сбоя процесса.	No	true, false	true
auto.commit.interval.ms	Частота в мс, с которой потребительские смещения передаются в zookeeper.	No		60 * 1000
queued.max.message.chunk	Максимальное количество блоков сообщений, буферизованных для использования. Каждый фрагмент может достигать значения, указанного в fetch.message.max.bytes.	No		2
rebalance.max.retries	Максимальное количество повторных попыток.	No		4
fetch.min.bytes	Минимальный объем данных, который сервер должен вернуть для запроса на выборку.	No		1
fetch.wait.max.ms	Максимальное время, в течение которого сервер будет блокироваться перед ответом на запрос выборки, если данных недостаточно для немедленного обслуживания fetch.min.bytes.	No		100
rebalance.backoff.ms	Время задержки между повторными попытками во время ребалансировки.	No		2000
refresh.leader.backoff.ms	Время задержки перед попыткой определить	No		200

Имя параметра	Описание	Обязательный ?	Возможные значения	Значение по умолчанию
s	лидера раздела, который только что потерял своего лидера.			
auto.offset.reset	Установите одно из следующих значений в зависимости от того, что вам нужно делать, когда в ZooKeeper нет начального смещения или если смещение выходит за пределы допустимого диапазона. smallest - автоматически сбросить смещение до наименьшего смещения. largest - Автоматически сбрасывать смещение до наибольшего смещения. anything else - выдать исключение потребителю.	No	smallest, largest, anything else	largest
consumer.timeout.ms	Интервал тайм-аута, по истечении которого потребителю должно быть выдано исключение тайм-аута, если сообщение недоступно для потребления. Рекомендуется установить это значение меньше, чем интервал точки входа Kafka.	No		3000
exclude.internal.topics	Установите значение true, если сообщения из внутренних топиков, таких как смещения, должны быть доступны потребителю.	No	true, false	true
partition.assignment.strategy	Стратегия назначения разделов, которая будет использоваться при назначении разделов потребителем потокам.	No	range, roundrobin	range
client.id	Указанная пользователем строка, отправляемая в каждом запросе для помощи в отслеживании вызовов.	No		value of group id
zookeeper.session.timeout.ms	Значение времени ожидания сеанса ZooKeeper в миллисекундах.	No		6000
zookeeper.connection.timeout.ms	Максимальное время в миллисекундах, которое клиент должен ждать при установлении соединения с ZooKeeper.	No		6000
zookeeper.sync.time.ms	Разница во времени в миллисекундах, в течение которой подписчик ZooKeeper может отставать от лидера ZooKeeper.	No		2000
offsets.storage	Место хранения смещений.	No	zookeeper, kafka	zookeeper
offsets.channel.backoff.ms	Период отсрочки в миллисекундах при повторном подключении к каналу смещений или повторной попытке неудачных запросов на выборку/фиксацию смещения.	No		1000
offsets.channel.socket.timeout.ms	Время ожидания сокета в миллисекундах при чтении ответов на запросы выборки/фиксации смещения.	No		10000
offsets.commit.max.retries	Максимально допустимое количество повторных попыток. Если запрос метаданных потребителя по какой-либо причине завершается сбоем, происходит повторная попытка, но это не влияет на это ограничение.	No		5
dual.commit.enabled	Если для offsets.storage установлено значение kafka, смещения фиксации могут быть двойными для ZooKeeper. Установите для этого параметра значение true, если вам нужно выполнить миграцию из хранилища смещения на основе zookeeper в хранилище смещения на основе kafka.	No	true, false	true

Примечание

В конфигурациях высокого уровня kafka вместо параметров топиков используются следующие параметры:

```
<parameter name="topic.filter">test</parameter>
```

```
<parameter name="filter.from.whitelist">true</parameter>
```

Таблица 14. Параметры для использования фильтров топиков

Parameter	Description	Required
topic.filter	Название фильтра топика.	Да
filter.from.whitelist	Если установлено значение true, сообщения потребляются из белого списка (включаются). Если установлено значение false, сообщения потребляются из черного списка (исключаются).	Да

Таблица 15. Параметры низкоуровневой настройки топиков Kafka

Parameter	Description	Required
simple.topic	Категория для подачи сообщений.	Да
simple.brokers	Конкретное имя брокера Kafka.	Да
simple.port	Конкретный номер порта сервера Kafka.	Да
simple.partition	Раздел топика.	Да
simple.max.messages.to.read	Максимальное количество сообщений для извлечения.	Да

Для настройки пользовательской опрашивающей точки, помимо параметров, общих для всех точек входа (*последовательность обработки сообщений* и *последовательность обработки ошибок*), достаточно определить только разработанный пользователем класс реализации и *интервал опроса*.

Точки входа, основанные на событиях

Система поддерживает предопределенные точки входа для работы с *брокером сообщений RabbitMQ* и обработку сообщений по *протоколу MQTT*.

Для настройки пользовательской точки, основанной на событиях, помимо параметров, общих для всех точек входа (*последовательность обработки сообщений* и *последовательность обработки ошибок*), достаточно определить только разработанный пользователем класс реализации.

Пример создания прослушивающей точки входа HTTP

Добавляем новую точку входа: «Точки входа» / «Добавить точку входа»

Указываем имя («Название») и подтип точки входа (HTTP), нажимаем кнопку «Следующий»:

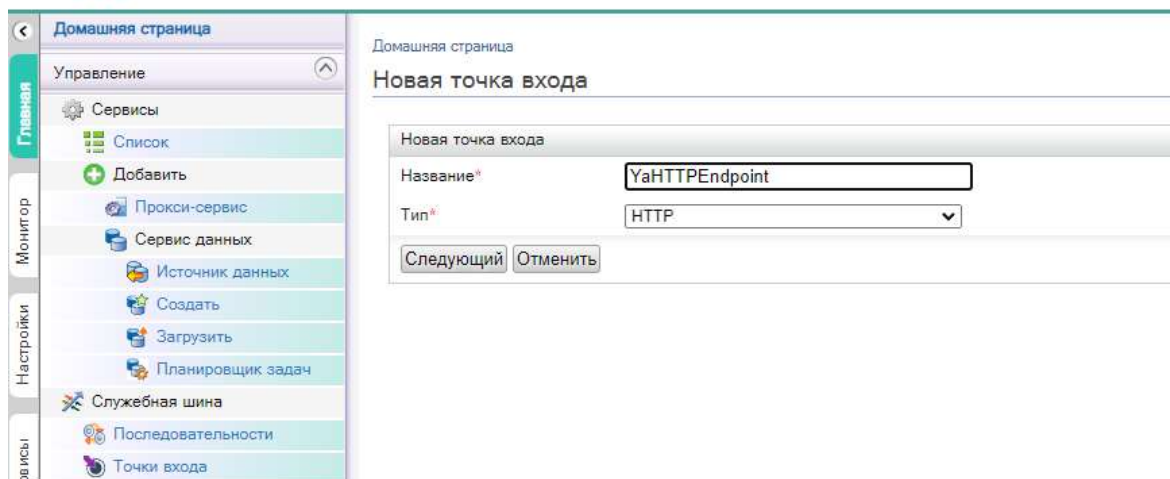


Рис.20. Создание новой точки входа HTTP

Далее указываем значения параметров точки входа; параметры «*Последовательность*» и «*Последовательность ошибок*» являются общими для всех точек входа.

Для точек входа HTTP обязательным параметром является *inbound.port.http* – порт, на котором точка входа будет «прослушивать» входящие запросы. Прочие параметры нужны для конфигурирования пула потоков обработки входящих запросов для точки вызова HTTP

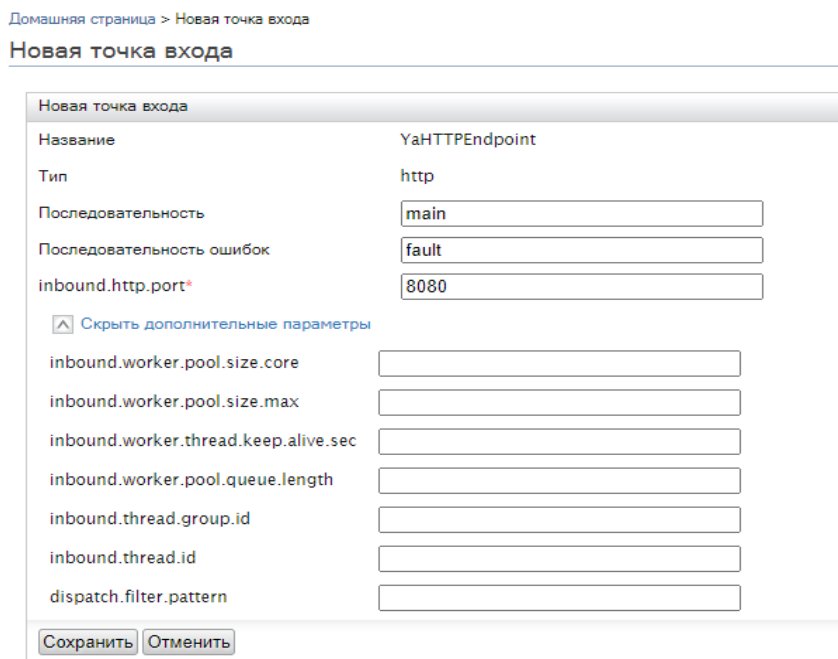


Рис.21. Параметры новой точки входа HTTP

При нажатии на кнопку «Сохранить» созданная точка входа появляется в общем списке:

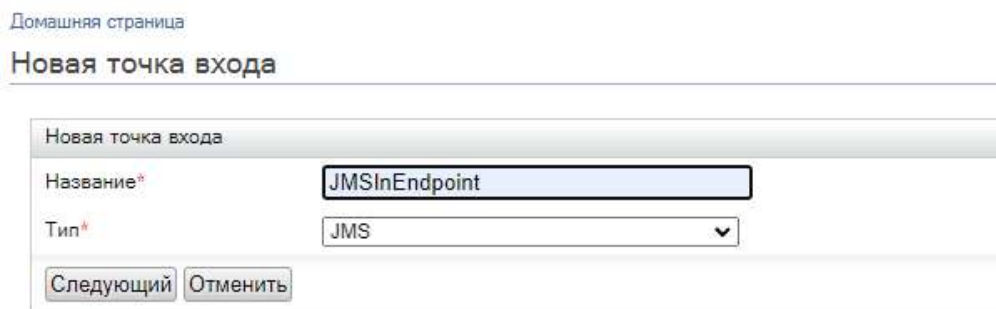


Название	Действие
YaHTTPEndpoint	<input type="button" value="Включить статистику"/> <input type="button" value="Включить трассировку"/> <input type="button" value="Редактировать"/> <input type="button" value="Удалить"/>

Рис.22. Список точек входа

Пример создания опрашивающей точки входа — JMS

Добавляем новую точку входа: «Точки входа» / «Добавить точку входа», указываем название, выбираем подтип (JMS) и нажимаем «Следующий»:



Домашняя страница

Новая точка входа

Новая точка входа

Название* JMSInEndpoint

Тип* JMS

Следующий Отменить

Рис. 23. Новая точка входа JMS

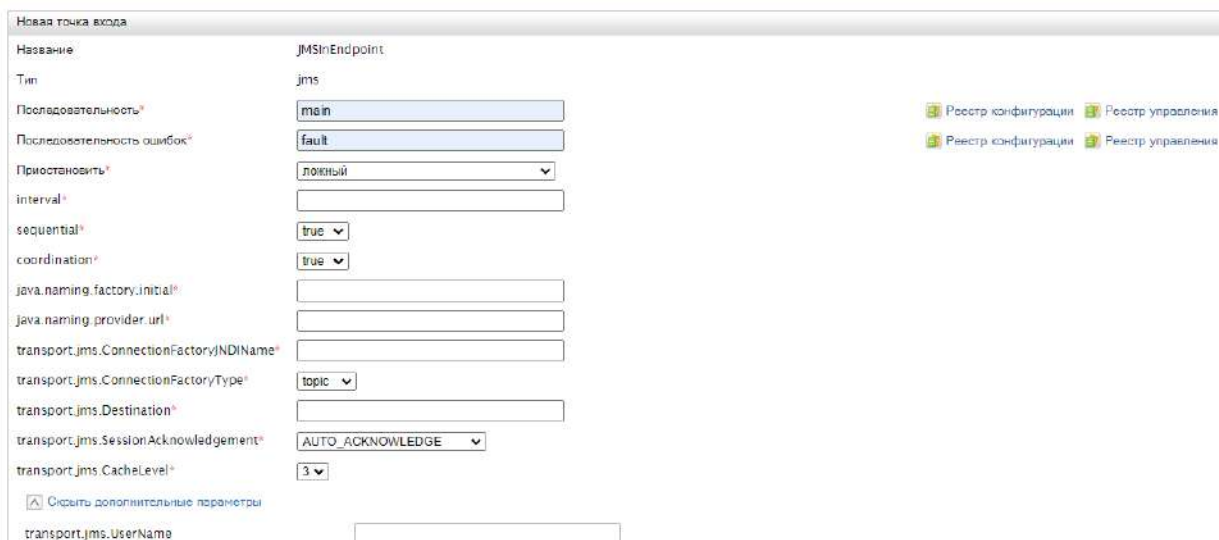
Устанавливаем параметры, общие для всех опрашивающих точек входа:

- **Последовательность** - последовательность, которой точка входа передает сообщение для дальнейшей обработки
- **Последовательность ошибок** — последовательность, которой передается сообщение на дальнейшую обработку при возникновении ошибки
- **sequential** – обработка сообщения последовательностью в том же потоке (true) или независимо (поток выбирается из пула)
- **suspend** – создать точку входа в статусе «приостановлена» (true) или «активна» (false)
- **interval** – интервал (в миллисекундах), с которым точка входа опрашивает источник данных
- **coordination** – актуален для кластерного режима: в случае true опрос выполняет точка входа только на одном узле кластера

Для точки входа JMS нужно также выбрать тип источника данных — топик или очередь JMS, а также установить значения специфичных для протокола JMS параметров:

Домашняя страница > Новая точка входа

Новая точка входа



Новая точка входа

Название JMSInEndpoint

Тип jms

Последовательность* main

Последовательность ошибок* fault

Приостановить* ложный

interval*

sequential* true

coordination* true

java.naming.factory.initial*

java.naming.provider.url*

transport.jms.ConnectionFactoryJNDIName*

transport.jms.ConnectionFactoryType* topic

transport.jms.Destination*

transport.jms.SessionAcknowledgement* AUTO_ACKNOWLEDGE

transport.jms.CacheLevel* 3

Скрыть дополнительные параметры

transport.jms.UserName

Реестр конфигурации Реестр управления

Рис. 24. Параметры точки входа JMS

ТОЧКИ ВЫЗОВА

Используйте функцию управления точками вызова, чтобы добавлять, удалять и изменять точки вызова.

1. В навигаторе в разделе Управление > Службная шина, щелкните **Точки Вызова**. Появится страница **Управление Точками Вызова**.
2. На вкладке **Добавить Точку Вызова**, выберите тип точки вызова, которую вы хотите добавить. Доступные варианты:

Таблица 16. Типы точек вызова

<i>Точка вызова по URL</i>	<i>Определяет прямой URL сервиса</i>
<i>Точка вызова по умолчанию</i>	<i>Определяет дополнительную конфигурацию для целевого объекта по умолчанию</i>
<i>Отказоустойчивая группа</i>	<i>Определяет точки вызова, к которым сервис попытается подключиться в случае сбоя. Это будет происходить в циклическом порядке</i>
<i>Точка вызова HTTP</i>	<i>Определяет точку вызова REST сервиса на основе шаблона URI.</i>
<i>Точка вызова баланса нагрузки</i>	<i>Определяет группы точек вызова для реплицированных сервисов. Входящие запросы будут направляться на эти точки вызова в циклическом режиме. Эти точки вызова также автоматически обрабатывают отказоустойчивые случаи.</i>
<i>Список получателей</i>	<i>Определяет список точек вызова, на которые будет перенаправлено сообщение</i>
<i>Точка вызова по шаблону</i>	<i>Точка вызова, созданная по предопределенному шаблону</i>
<i>Точка вызова WSDL</i>	<i>Определяет WSDL, сервис и порт</i>

Добавление точки вызова по URL

1. На вкладке **Добавить точку вызова**, щёлкните **Точка вызова по URL**

The screenshot shows a web interface for adding a 'Point of Call by URL'. At the top, there is a breadcrumb 'Домашняя страница > Точка вызова по URL' and the title 'Точка вызова по URL'. Below the title, there is a sub-header 'Точка вызова по URL' and a link 'Перейти к исходному виду'. The form contains two input fields: 'Название' (Name) and 'Адрес' (Address), with a 'Тест' button next to the address field. There is a checkbox 'Показать дополнительные параметры' (Show additional parameters) which is checked. Below this is a section titled 'Свойства точки вызова' (Point of Call Properties) with a '+ Добавить свойство' (Add property) button. At the bottom, there are three buttons: 'Сохранить и закрыть' (Save and close), 'Сохранить в реестре' (Save in registry), and 'Отменить' (Cancel).

Рис. 25. Точка вызова по URL


2. Если вы хотите настроить дополнительные параметры, щелкните раскрывающуюся панель *Показать Дополнительные Параметры*. Появится страница со всеми доступными параметрами точки вызова по URL.

Содержание сообщения

Форматировать

Оптимизировать

Состояние приостановки точки вызова


Коды ошибок  Выберите коды ошибок

Начальная продолжительность (миллисекунды)

Максимальная продолжительность (миллисекунды)

Фактор

Состояние тайм-аута точки вызова

Коды ошибок  Выберите коды ошибок

Повторные попытки перед приостановкой

Задержка повторной попытки (миллисекунды)

Время ожидания соединения

Действие

Максимальное время ожидания (миллисекунды)

QoS

WS-Адресация

WS-Безопасность

WS-RM

Рис.26. Настройка точки вызова по URL — Дополнительные параметры

3. Введите необходимые данные в поля. Следующие поля определяют конфигурацию Адресной Точки Вызова.
- Имя: Уникальное имя для точки вызова.
 - Адрес: URL точки вызова. Вы можете проверить доступность данного URL на лету, просто щелкнув *Проверить Адрес*.
 - Формат: Формат сообщения для точки вызова. Доступные значения:
 - Оставить Как Есть: Исходящее сообщение не преобразуется;
 - SOAP 1.1: Преобразование сообщения в формат SOAP 1.1;
 - SOAP 1.2: Преобразование сообщения в формат SOAP 1.2;
 - Обычный Старый формат XML (POX) — Преобразование в обычный старый формат XML;
 - Передача репрезентативного состояния (REST/GET) — Преобразование в запрос HTTP методом GET;
 - Оптимизировать: Оптимизация для сообщения, которое передает двоичные данные. Доступные значения:

- Оставить Как Есть: Без специальной Оптимизации. Сохранить исходное сообщение;
- SwA: Оптимизировано как сообщение SOAP with Attachment (SOAP с Вложением);
- Оптимизировано как MTOM (механизм оптимизации передачи сообщений).

Примечание: остальные поля являются общими для точек вызова типа URL, HTTP, По умолчанию и WSDL.

Добавление Точки Вызова По Умолчанию

1. На вкладке **Добавить точку вызова** щёлкните **Точка вызова по умолчанию**. Появится страница **Точка вызова по умолчанию**. Вы можете переключиться на дополнительные параметры, щёлкнув раскрывающуюся панель *Показать дополнительные параметры*.

[Домашняя страница](#) > [Точка вызова по умолчанию](#)

Точка вызова по умолчанию

Точка вызова [Переключиться на исходный вид](#)

Название *

Показать дополнительные параметры

Свойства точки вызова

[+](#) Добавить свойство

Рис. 27. Точка вызова по умолчанию

2. Введите необходимые данные в поля. Точка Вызова По Умолчанию отличается от Адресной Точки Вызова только наличием свойства URL.

Примечание: остальные поля являются общими для точек вызова типа URL, HTTP, По умолчанию и WSDL.

Добавление Точки Вызова HTTP

1. На вкладке **Добавить Точку Вызова** щёлкните **Точка Вызова HTTP**

Шаблон точки вызова HTTP Переключиться на исходный вид

Название *

Шаблон URI *

OPTIONS

HTTP-метод

Показать дополнительные параметры

Свойства точки вызова

Добавить свойство

Рис. 28. Настройка точки вызова HTTP

2. Если вы хотите изменить дополнительные параметры, щелкните раскрывающуюся панель *Показать дополнительные параметры*. Появится страница со всеми доступными настройками точки вызова HTTP.

Скрыть дополнительные параметры

Состояние приостановки точки вызова

Коды ошибок

Начальная продолжительность (миллисекунды)

Максимальная продолжительность (миллисекунды)

Фактор

Состояние тайм-аута точки вызова

Коды ошибок

Повторные попытки перед приостановкой

Задержка повторной попытки (миллисекунды)

Время ожидания соединения

Действие

Максимальное время ожидания (миллисекунды)

Описание точки вызова

Описание

Свойства точки вызова

Добавить свойство

Рис. 29. Настройки точки вызова HTTP — Дополнительные параметры

3. Введите необходимые данные в поля. Следующие поля определяют конфигурацию Точки Вызова HTTP.
- Имя: Уникальное имя для точки вызова.
 - Шаблон URI: Шаблон URL точки вызова, в котором переменные шаблона помещены в фигурные скобки с префиксом 'uri.var'. Пример:
`http://wso2.com/{uri.var.pageId}/{uri.var.docId}`.
 - Оставить Как Есть: Используется входящий метод HTTP.
 - GET: HTTP GET Метод.
 - POST: HTTP POST Метод.
 - PUT: HTTP PUT Метод.
 - DELETE: HTTP DELETE Метод.
 - HEAD: HTTP HEAD Метод.
 - OPTIONS: HTTP OPTIONS Метод.
 - PATCH: HTTP PATCH Метод.

Примечание: остальные поля являются общими для точек вызова типа URL, HTTP, По умолчанию и WSDL.

Добавление точки вызова WSDL

1. В списке **Добавить точку вызова** щёлкните **Точка вызова WSDL**. Появится страница **Точка Вызова WSDL**.

Домашняя страница > Точка вызова WSDL

Точка вызова WSDL

Точка вызова WSDL [Перейти к исходному виду](#)

Название *

WSDL URI *

Сервис *

Порт *

Показать дополнительные параметры

Свойства точки вызова

[+ Добавить свойство](#)

Рис.30. Настройка Точки вызова WSDL

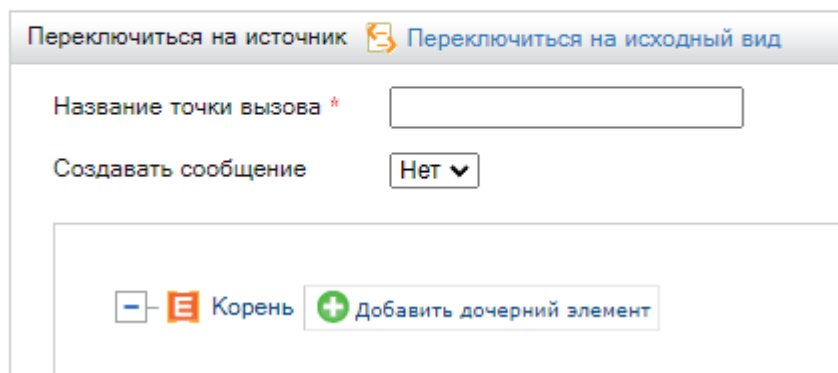
2. Введите необходимые данные в поля. Следующие поля определяют конфигурацию Точки Вызова WSDL
 - Имя: Уникальное имя для точки вызова.
 - WSDL URI: URI WSDL.
 - Сервис: Сервис, выбранный из списка доступных сервисов для данного WSDL.
 - Порт: Порт, выбранный для службы, указанной в поле выше. В WSDL точка вызова привязана к каждому порту внутри каждого сервиса.


Примечание: остальные поля являются общими для точек вызова типа URL, HTTP, По умолчанию и WSDL.

Добавление отказоустойчивой группы

1. В списке **Добавить точку вызова** щёлкните **Отказоустойчивая группа**.
2. В поле **Название точки вызова** введите наименование точки вызова.
3. Щёлкните **Добавить дочерний элемент к Корню**. Появится меню с доступными точками вызова, которые можно добавить в группу.

Отказоустойчивая группа



Переклiчитьcя на источник  Переклiчитьcя на исходный вид

Название точки вызова *

Создавать сообщение


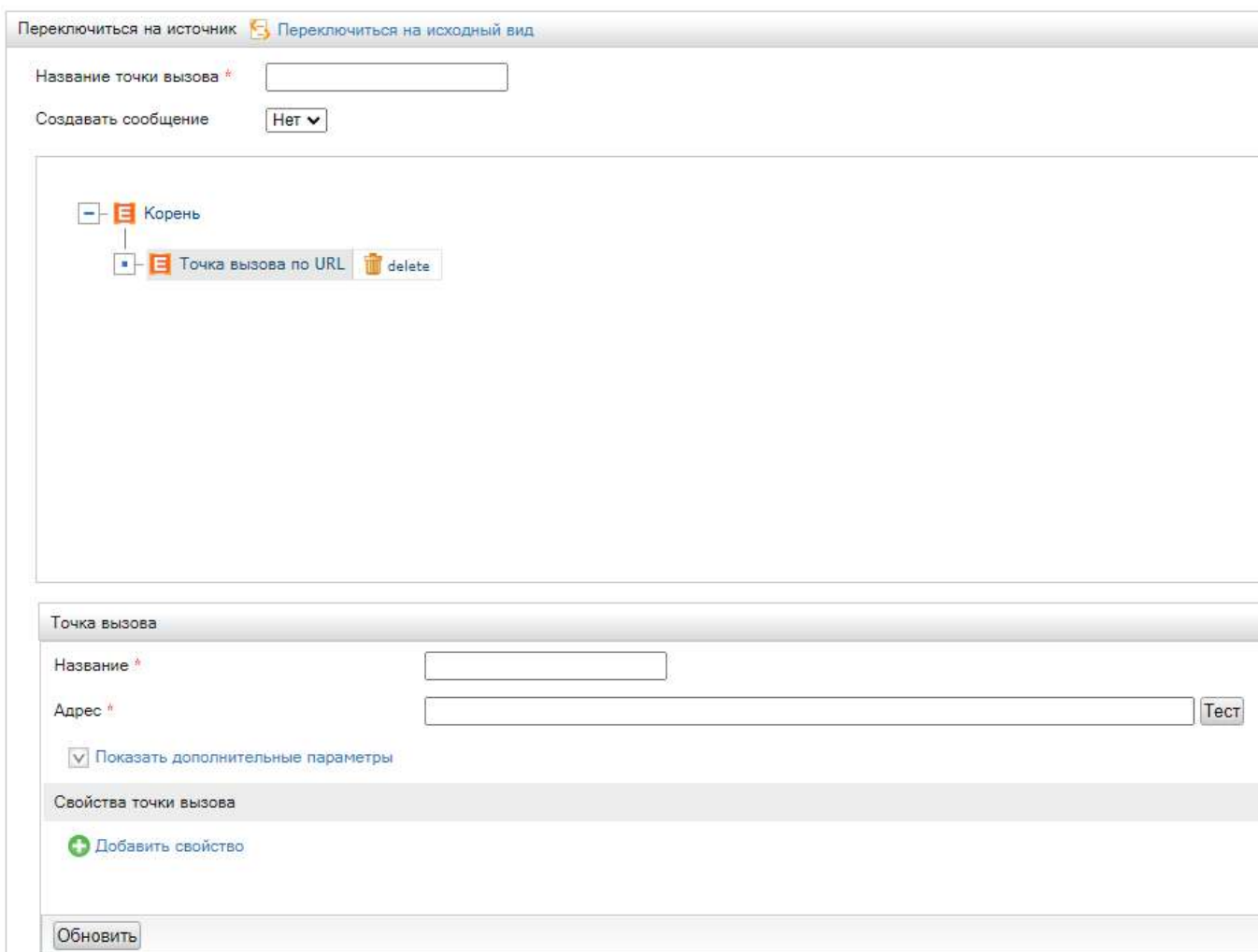

 Корень Добавить дочерний элемент

Рис.31. Добавление отказоустойчивой группы

- Щёлкните по точке вызова, которую вы хотите добавить в отказоустойчивую группу. Появится вкладка с полями, относящимися к выбранной точке вызова.
- Введите данные и щёлкните **Сохранить и Закрyть**. На следующем снимке экрана показано добавление Адресной точки вызова в список отказоустойчивых точек вызова.



Отказоустойчивая группа



Переклiчитьcя на источник  Переклiчитьcя на исходный вид

Название точки вызова *

Создавать сообщение

 Корень
  Точка вызова по URL

Точка вызова

Название *

Адрес *

Показать дополнительные параметры

Свойства точки вызова

Добавить свойство

Рис.32. Настройка отказоустойчивой группы

Вы можете взаимозаменяемо добавлять любое количество уровней отказоустойчивых точек вызова и точек балансировки нагрузки.

Добавление группы балансировки нагрузки

1. В списке **Добавить точку вызова** щёлкните **Точка вызова баланса нагрузки**. Появится страница **Точка Вызова Группа Балансировки Нагрузки**.
2. Введите необходимые данные в поля. Следующие поля определяют конфигурацию Точки Вызова Группа Балансировки Нагрузки.
 - **Имя Точки Вызова:** Уникальное имя для точки вызова.
 - **Управление Сеансом:** Метод управления сеансом из группы балансировки нагрузки. Возможные значения.
 - **Нет:** Управление сеансом не используется.
 - **Транспорт:** Управление сеансом осуществляется на транспортном уровне с использованием файлов cookie HTTP.
 - **SOAP:** Управление сеансом осуществляется с помощью сеансов SOAP.
 - **Идентификатор клиента:** Управление сеансом осуществляется с использованием идентификатора, отправленного клиентом.
 - **Время ожидания сеанса (мс):** время ожидания сеанса в миллисекундах.

[Домашняя страница](#) > [Группа балансировки нагрузки](#)

Группа балансировки нагрузки

Перейти к источнику Перейти к исходному виду

Название точки вызова *

Алгоритм

Создавать сообщение

Управление сеансами

Тайм-аут сеанса (время ожидания)

Корень Добавить дочерний элемент

Рис. 33. Группа балансировки нагрузки

3. Щёлкните **Добавить дочерний элемент** к **Корню**. Появится меню с доступными точками вызова.
4. Щёлкните по точке вызова, которую вы хотите добавить в группу балансировки нагрузки. Появится вкладка с полями, относящимися к выбранной точке вызова.
5. Введите данные и щёлкните **Сохранить и Заккрыть**. Вы можете добавить в группу балансировки нагрузки столько точек вызова, сколько вам нужно.

Добавление точки вызова по шаблону

1. В списке **Добавить Точку Вызова** щёлкните **Шаблонная Точка Вызова**. Появится страница **Шаблонной Точки Вызова** в своём виде по умолчанию.

Точка вызова по шаблону

Шаблон точки вызова [Переключиться на исходный вид](#)

Имя (\$name) *

Адрес (\$uri)

Целевой шаблон *

Доступные шаблоны

[Добавить параметр](#)

Рис. 34. Точка вызова по шаблону

2. Введите необходимые данные в поля. Следующие поля определяют конфигурацию точки вызова по шаблону.
 - Имя: Уникальное имя для шаблонной точки вызова.
 - Адрес: URL точки вызова шаблона. Если часть точки вызова целевого шаблона имеет элемент адреса с объявленным в нем \$uri, то он будет заполнен значением, указанным в этом поле адреса. Как всегда, вы можете проверить доступность данного URL на лету, просто щелкнув *Тестовый Адрес*.
 - Целевой Шаблон: Целевой шаблон для данной точки вызова.

3. Если вы хотите добавить параметры в шаблон Точки Вызова, щёлкните кнопку *Добавить параметр*. После этого появится страница с таблицей параметров. Вы можете добавлять и удалять пары имя/значение параметра.

Точка вызова по шаблону

Шаблон точки вызова [Переключиться на исходный вид](#)

Имя (\$name) *

Адрес (\$uri)

Целевой шаблон *

Доступные шаблоны

Параметры шаблона точки вызова

Название	Значение	Действие
<input type="text"/>	<input type="text"/>	Удалить параметр

[Добавить параметр](#)

Рис. 35. Добавление параметров точки вызова по шаблону

Добавление списка получателей

1. В списке **Добавить Точку Вызова** щёлкните **Точка вызова списка получателей**.
2. В поле **Название точки вызова**, введите имя точки вызова.
3. Щёлкните **Добавить Дочерний Элемент к Корню**. Появится меню с доступными точками вызова.
4. Щёлкните по точке вызова, которую вы хотите добавить в список получателей. Появится вкладка с полями, относящимися к выбранной точке вызова.
5. Введите данные и щёлкните **Сохранить и Заккрыть**.

Домашняя страница > Точка вызова списка получателей

Точка вызова списка получателей

Переклiчить на источник Переклiчить на исходный вид

Название точки вызова *

Корень
 Точка вызова по URL delete

Точка вызова

Название *

Адрес *

Показать дополнительные параметры

Свойства точки вызова

Добавить свойство

Свойства точки вызова списка получателей

Добавить свойство

Рис.36. Настройка списка получателей

Добавление динамических точек вызова

1. Чтобы сохранить точку вызова как динамическую точку вызова, щёлкните **Сохранить в реестре**, после чего отобразится пользовательский интерфейс сохранения динамической точки вызова.
2. Выберите **Реестр управления** или **Реестр конфигурации**.
3. Предоставьте ключ для динамической точки вызова. Этот ключ будет использоваться как путь к ресурсу, который будет добавлен в Реестр.

Домашняя страница > Шаблон точки вызова по URL ? Помощь

Шаблон точки вызова по URL

Точка вызова по URL ? [Переключиться на исходный вид](#)

Название *

Адрес *

Показать дополнительные параметры

Свойства точки вызова

+ [Добавить свойство](#)

Рис.37. Добавление динамической точки вызова

Сохраненная динамическая точка вызова будет отображаться следующим образом

Домашняя страница > endpoints ? Помощь

Управление точками вызова

Динамические точки вызова, сохраненные в реестре

Название точки вызова	Тип	Действие
conf:/ДинамическаяТВ	Шаблон точки вызова по URL	<input type="button" value="Редактировать"/> <input type="button" value="Удалить"/>

Рис.38. Сохраненная динамическая точка вызова

Параметры, общие для точек вызова по URL, HTTP, WSDL и по умолчанию

Таблица 17. Параметры, общие для разных типов точек вызова

Параметр/Группа параметров	Назначение
Состояние приостановки точки вызова	Относятся к ситуации возникновения ошибок при вызове сервиса через endpoint
Коды ошибок приостановки	Если во время вызова сервиса через точку вызова возникнет одна из указанных ошибок, то выполнение будет приостановлено (suspended endpoint)
Начальная длительность	Период времени (в миллисекундах), на который будет приостановлено выполнение точки вызова в случае возникновения ошибок

приостановки выполнения	
Максимальная длительность приостановки выполнения	Максимальный период времени (в миллисекундах), на который будет приостановлено выполнение точки вызова в случае возникновения ошибок
Фактор (коэффициент) приостановки	Если по истечении периода приостановки повторный вызов сервиса через точку вызова также приводит к ошибкам, то при вычислении следующего периода приостановки используется данный коэффициент
Состояние тайм-аута точки вызова	Относятся к ситуации, когда при вызове сервиса через endpoint возникает тайм-аут
Коды ошибок тайм-аута	Если во время вызова сервиса через endpoint возникнет одна из указанных ошибок, то ситуация интерпретируется как тайм-аут
Количество попыток повторного вызова	Количество попыток повторного вызова endpoint в случае возникновения тайм-аута (т. е. возникновения ошибок из предыдущего списка)
Период задержки при повторном вызове	Период времени (в миллисекундах) между двумя повторными вызовами endpoint в случае тайм-аута
Длительность тайм-аута	Период времени (в миллисекундах). Если в течение данного периода не получен ответ после вызова сервиса через endpoint (даже если не возникло ошибок), то ситуация интерпретируется как тайм-аут
Обработка возникновения тайм-аута	<p>Действие, которое должна выполнять Polus ESB при получении ответа от точки вызова, находящей в статусе timeout:</p> <ul style="list-style-type: none"> – Не устанавливать тайм-аут – Отменить сообщение – Выполнить последовательность ошибок
QoS (Quality of Service)	Параметры, относящиеся к дополнительным спецификациям Web Services
WS-Addressing	Включение информации о маршрутизации в SOAP-заголовки
WS-Security	Добавляет данные, относящиеся к спецификации WS-Security (проверка целостности, шифрование, аутентификация). При этом соответствующая политика безопасности указывается как элемент реестра
WS-Reliable Messaging	Добавляет данные, относящиеся к спецификации WS-ReliableMessaging (гарантированная доставка сообщений в распределенной среде). При этом соответствующая настройка указывается как элемент реестра

ПОСЛЕДОВАТЕЛЬНОСТИ МЕДИАЦИИ

Используйте страницу *Последовательности медиации*, чтобы добавлять новые или редактировать существующие последовательности. Как показано на рис.38, в разделе управления последовательностями отображается список существующих последовательностей. Вкладка «Определенные последовательности» показывает последовательности, сохраненные в конфигурации Polus ESB. Вкладка «Динамические последовательности» показывает последовательности, сохраненные в реестре.

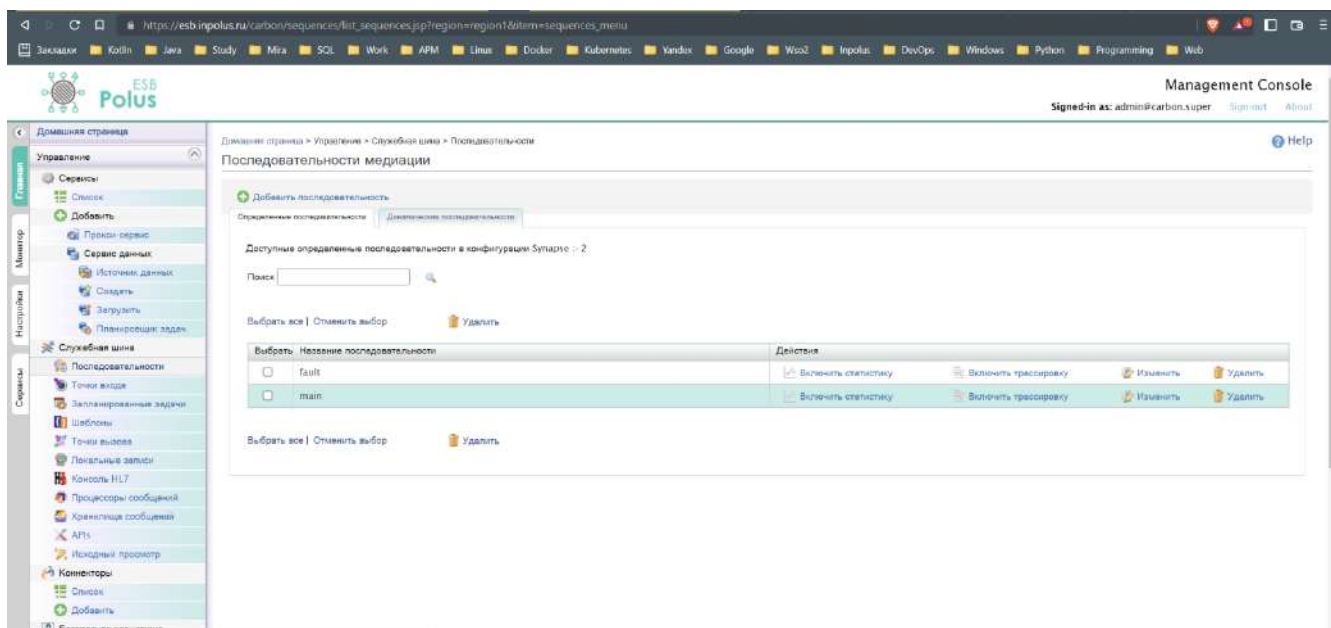


Рис.39. Управление последовательностью медиации

В навигаторе в разделе «Управление/служебная шина» щелкните «Последовательности». Появится страница «Последовательности медиации».

Включить статистику

□ Нажмите «Включить статистику», чтобы собрать информацию, относящуюся к определенной последовательности.

Включить трассировку

□ Нажмите «Включить трассировку», чтобы включить все сообщения трассировки для соответствующей последовательности.

Вышеупомянутые параметры отключены по умолчанию, поскольку они вызывают серьезное снижение производительности. Используйте эти параметры только в ситуации, когда вам нужно отладить конкретную проблему. Динамические последовательности не поддерживают «Включить статистику» и «Включить трассировку».

Редактировать

□ Нажмите «Изменить», чтобы открыть последовательность в проектном представлении для редактирования.

Удалить

- Щелкните «Удалить», чтобы удалить определенную последовательность из системы.

Добавление последовательности

- На странице «Последовательности медиации» нажмите «Добавить последовательность». Появится страница «Проектирование последовательности».
- Введите название последовательности.
- Щелкните «Добавить дочернюю» и выберите необходимые узлы для вашей последовательности. Каждый вариант описан в разделе «Описания полей».

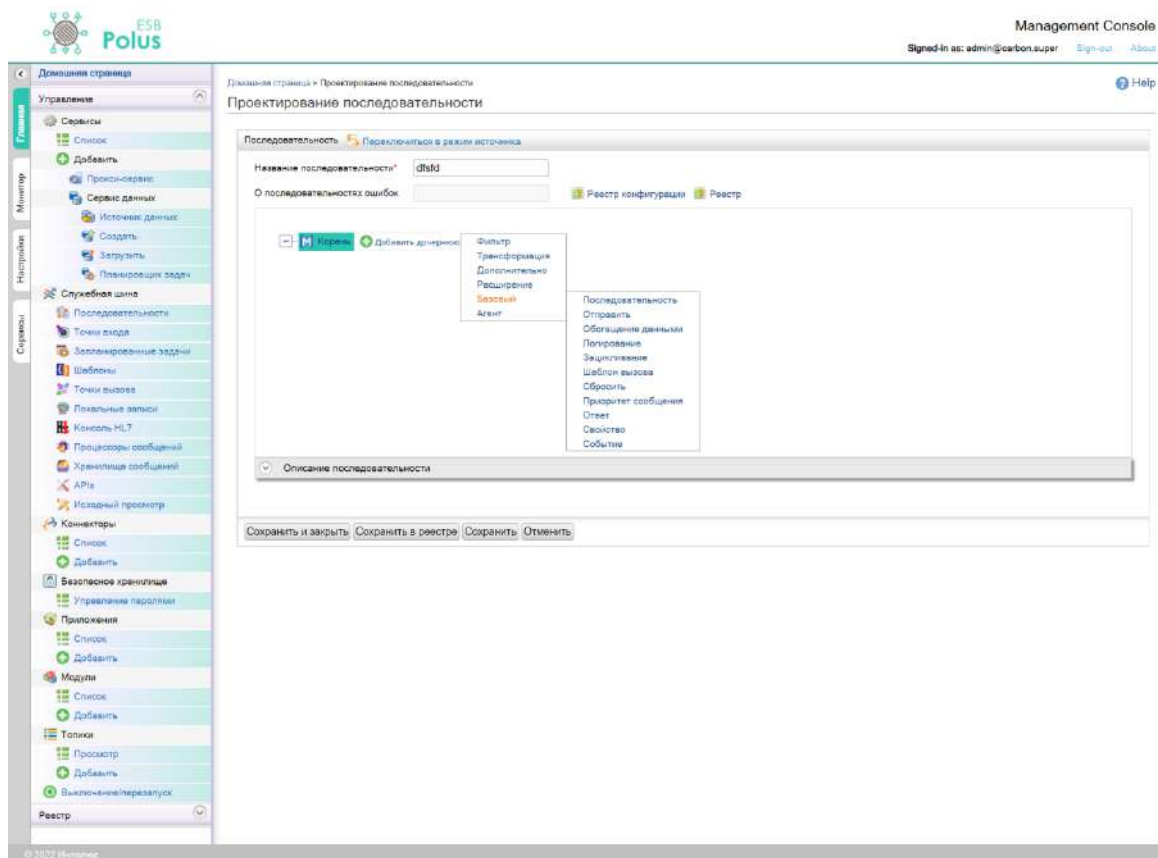


Рис.40. Добавление последовательности - режим конструктора

Добавление динамической последовательности

- На странице последовательности проектирования нажмите кнопку «Сохранить в реестре» - отобразится блок «Сохранение в реестре Synapse».
- Выберите сохранить в «Реестр конфигурации» или «Реестр».
- Укажите ключ динамического ресурса. Этот ключ будет действовать как путь, по которому ресурс будет добавлен в реестр.

Домашняя страница > Проектирование последовательности
Help

Проектирование последовательности

Последовательность Перейти к режиму источника

Название последовательности*

О последовательностях ошибок: Реестр конфигурации Реестр

- [-] Корень
 - [+] Call Добавить горизонтальную родственную связь Удалить

[-] Описание последовательности

Медиатор Перейти к режиму источника [Помощь](#)

Вызов посредника

Выберите тип конечной точки

Не указан

Определение встроенного

Выберите из реестра

XPath

Включить блокировку вызовов

Блокировка:

Сохранить как запись реестра Synapse

Сохранить в: Реестр конфигурации Реестр

Ключ:

© 2022 Интелюс

Рис.41. Добавление динамической последовательности

Сохраненная динамическая последовательность будет отображаться следующим образом

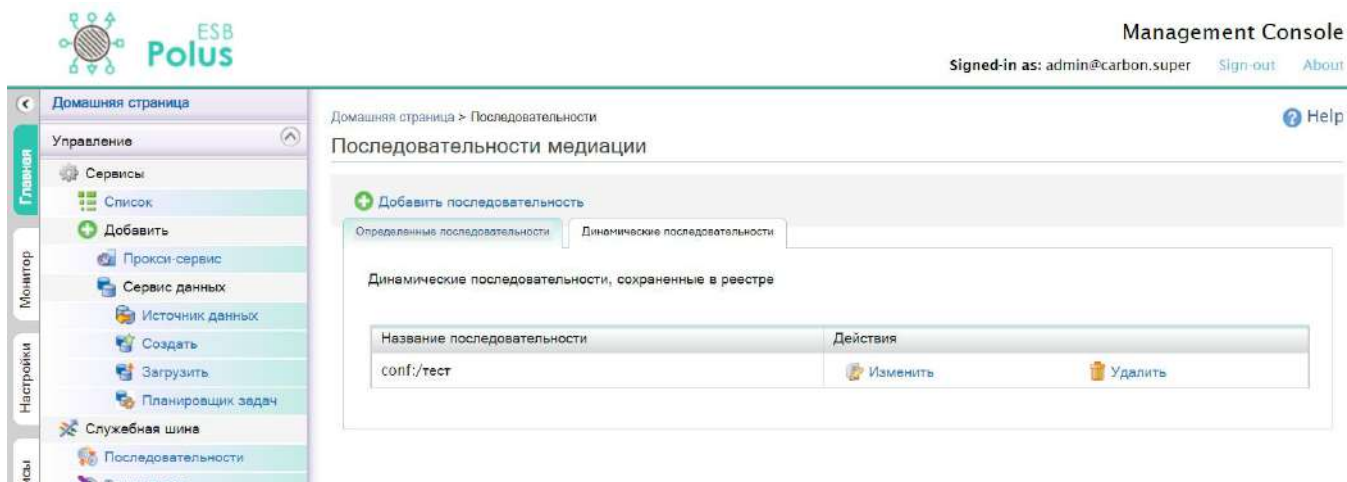


Рис.42. Сохраненная динамическая последовательность

Редактирование последовательности

1. На странице «Последовательности медиации» нажмите «Изменить». Появится страница «Изменить последовательность», отображающая выбранную последовательность.

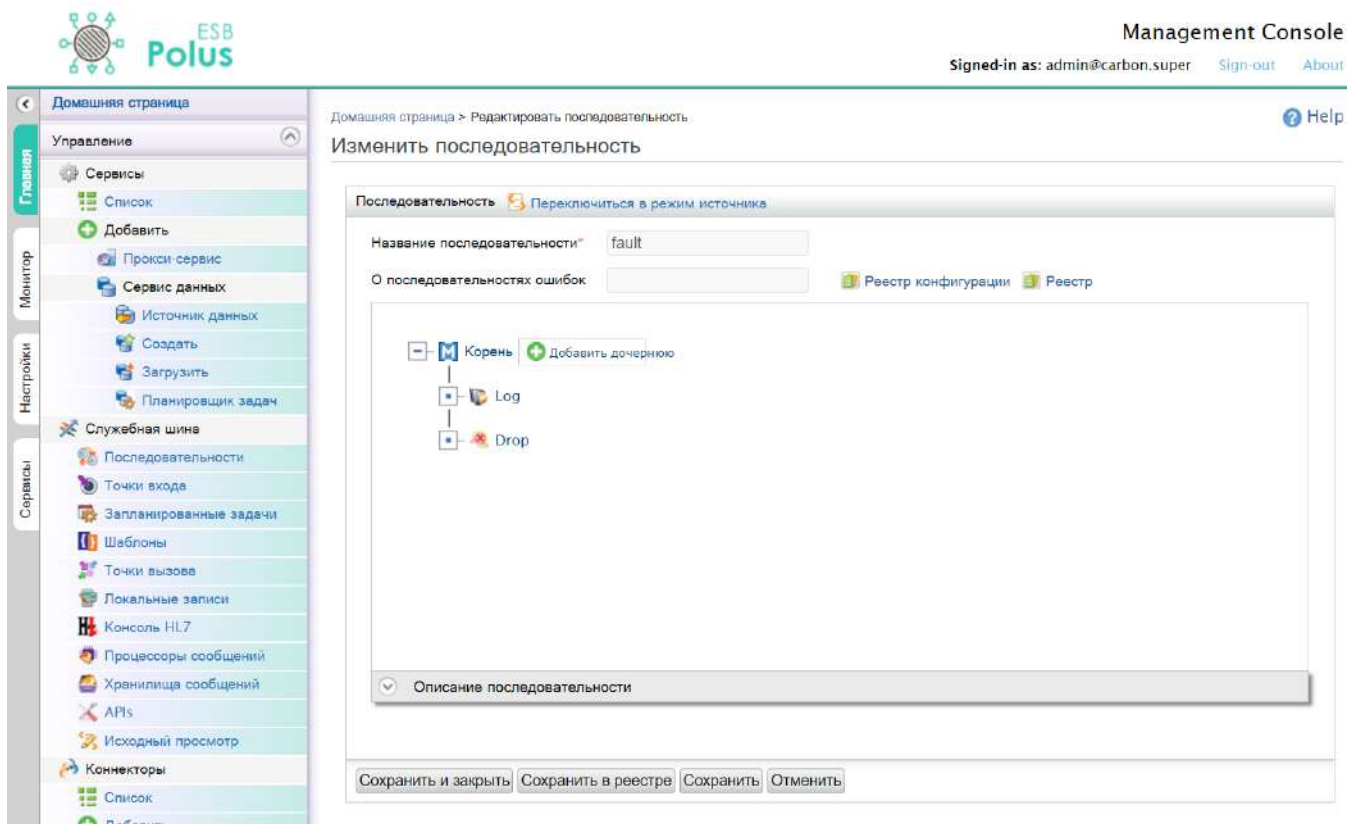


Рис.43. Редактирование последовательности

2. Щелкните узел, который хотите изменить. Свойства этого узла будут показаны ниже.

The screenshot shows the 'Management Console' for ESB Polus. The user is signed in as 'admin@carbon.super'. The main window is titled 'Изменить последовательность' (Change Sequence). The sequence name is 'main'. Below the tree view, the 'Медиатор' (Mediator) configuration is shown with the following details:

- Имя (Name): HTTP_SC
- Действие (Action): Установить (Install) selected, Удалить (Delete) unselected
- Установите действие как (Set action as): Значение (Value) selected, Выражение (Expression) unselected
- Тип (Type): STRING
- Значение (Value): 404
- Шаблон (Template):
- Группа (Group):
- Область действия (Action Area): Axis 2

Buttons at the bottom include 'Сохранить и закрыть' (Save and Close), 'Сохранить в реестре' (Save in Registry), 'Сохранить' (Save), and 'Отменить' (Cancel).

Рис.44. Редактирование свойств узла

- Вы можете редактировать определенные атрибуты узла, отображаемые в проектном представлении.
- Вы можете добавить родственный узел в проектном представлении.
- К некоторым узлам можно добавить только одноуровневые узлы, а к другим можно также добавить дочерний узел.
- Вы можете перемещать выбранный узел вверх или вниз с помощью значков со стрелками.
- Вы также можете удалить узел.

3. Нажмите «**Переключиться в режим источника**», чтобы просмотреть XML конкретной последовательности. Если вы знакомы с языком конфигурации Synapse, вы можете редактировать XML напрямую.

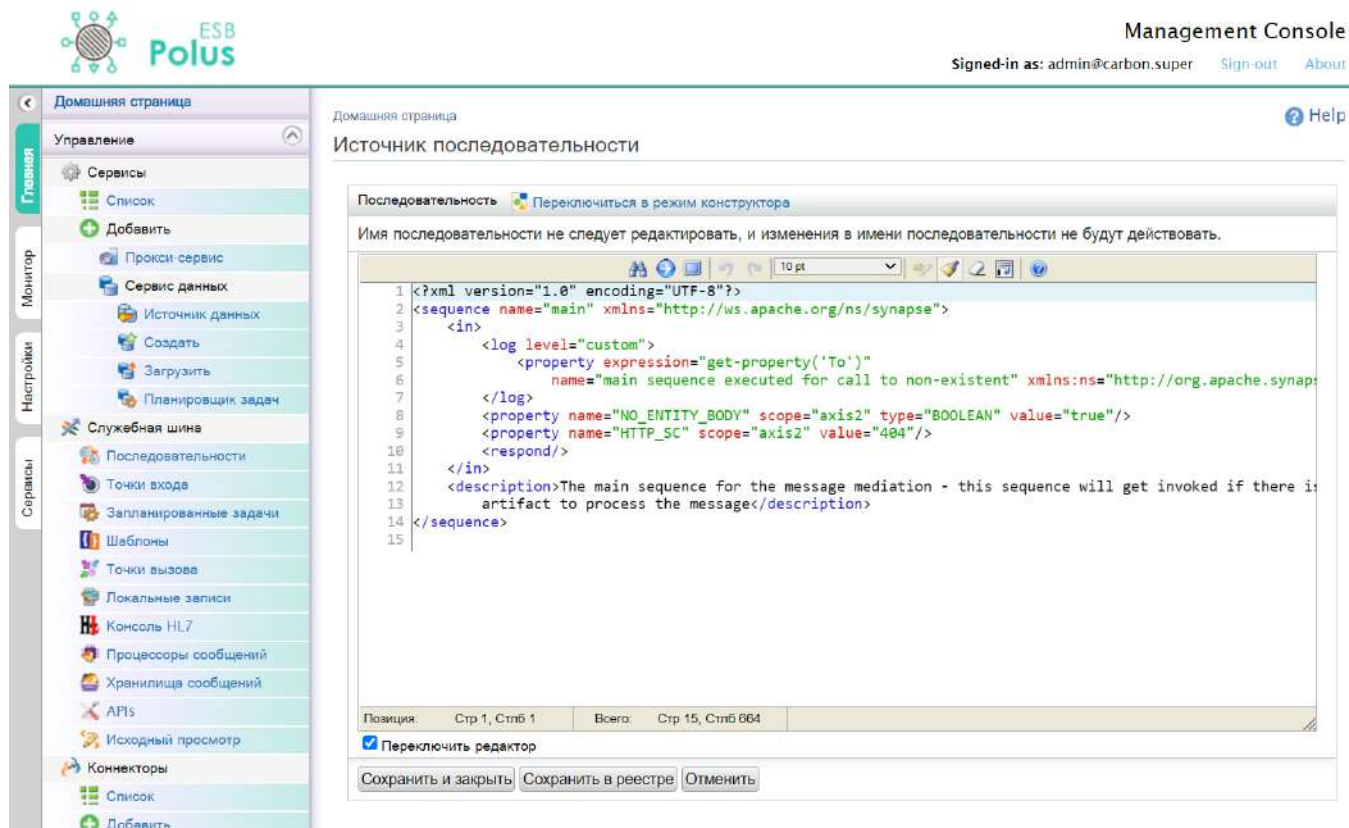


Рис.45. Исходный код последовательности

Типы медиаторов

Таблица 18. Каталог встроенных медиаторов

Тип	Подтип	Описание
Базовые (Core)	Вызов (Call)	Вызов внешнего сервиса (через точку вызова) без окончания обработки сообщения и в синхронном режиме
	Приоритет сообщения (Enqueue)	Установка приоритета сообщения (для того, чем выше приоритет, тем выше обработка)
	Отправка (Send)	Отправка сообщения точке вызова — финальное действие обработки сообщения и передача полученного от внешнего сервиса ответа последовательности обработки ответа

	Защелкивание (Loopback)	Перенаправление сообщения из потока (последовательности) обработки входного запроса в поток обработки ответа
	Последовательность (Sequence)	Вставка дочерней последовательности
	Возврат сообщения (Respond)	Остановка обработки сообщения и отправка его обратно клиенту
	Событие (Event)	Отправка нотификации о событии в определенный топик (см. п.3.1.10 о топиках и событиях)
	Удаление сообщения (Drop)	Удаление сообщения
	Шаблон вызова (Call Template)	Формирование последовательности путем подстановки конкретных значений параметров в шаблон последовательности
	Обогащение данными (Enrich)	Обогащение сообщения дополнительными данными
	Свойство (Property)	Установка значений свойств, связанных с сообщением
	Логирование (Log)	Логирование сообщения
Фильтры (Filters)	Фильтр (Filter)	Фильтрация сообщений с использованием XPath и if-else-логики
	Исходящий (Out)	Медиатор, связанный с глобальным параметром для фильтра исходящих сообщений. Содержит дочерние медиаторы, которые применяются к сообщениям, удовлетворяющим условию глобального фильтра.
	Входящий (In)	Медиатор, связанный с глобальным параметром для фильтра входящих сообщений. Содержит дочерние медиаторы, которые применяются к

		сообщениям, удовлетворяющим условию глобального фильтра.
	Валидация (Validate)	Валидация XML-сообщения — проверка на соответствие XML-схеме
	Переключатель (Switch)	Фильтрация сообщений с использованием XPath и условной логики по типу оператора switch
	Условный маршрутизатор (Conditional Router)	Реализация сложных правил маршрутизации (на основании заголовков сообщения, контента сообщения и т.п.)
Трансформация	XSLT	XSL-преобразование сообщения
	FastXSLT	XSL-преобразование потока (stream) сообщений
	URLRewrite	Модификация элементов URL
	XQuery	Преобразование XQuery
	Заголовок	Установка значений SOAP-заголовков
	Преобразование в ошибку (Fault)	Преобразование текущего сообщения в сообщение об ошибке (SOAP Fault)
	JSON Трансформатор	Трансформация JSON в XML
	PayloadFactory	Трансформация/замена содержимого сообщения при передаче от клиента к сервису первичной обработки
Дополнительные (Advanced)	Кэш (Cache)	Проверка соответствия вновь пришедшего и ранее закэшированного сообщения
	ForEach	Разделяет сообщение на несколько частей, каждая из которых соответствует определенному XPath-выражению, а затем объединяет их в одно новое сообщение
	Клон (Clone)	Клонирует сообщение

Хранилище (Store)	Записывает сообщение в хранилище
Итерация (Iterate)	Порождает дочерние сообщения на основании заданного исходя из определенных правил
Агрегирование (Aggregate)	Агрегирование нескольких сообщений, полученных на предыдущем этапе в результате работы медиаторов <i>Clone</i> или <i>Iterate</i> , в одно
Блокирование вызова (Callout)	Блокирование вызова внешних сервисов в процессе обработки сообщения медиаторами для дальнейшей обработки в одном потоке (thread)
Транзакция (Transaction)	Выполнение набора медиаторов в одной транзакции
Ограничитель обработки (Throttle)	Установка ограничения для определенных ролей и групп пользователей на обработку сообщений
DBReport	Запись данных в БД
DBLookup	Чтение данных из БД
EJB	Вызов внешнего EJB-компонента (Enterprise JavaBean) и сохранение результата вызова в свойствах/содержимом сообщения
Правило (Rule)	Обработка сообщения в соответствии с набором определенных правил
Builder	Формирование SOAP-сообщения из бинарных данных
Проверка полномочий (Entitlement)	Проверка действий пользователя на соответствие политике безопасности, описанной при помощи стандарта XACML
OAuth	Поддержка OAuth-

		аутентификации
	Smooks	Трансформация сообщений с использованием Smooks framework
	Маппер (Data Mapper)	Преобразование сообщения из одного формата данных в другой
Расширения (Extensions)	Bean	Работа с JavaBeans
	Класс	Вызов класса пользовательского медиатора
	Команда	Реализация паттерна «Команда» (Command) – класс с методом public void execute ()
	Скрипт	Вызов медиатора на скриптовом языке
	Spring	Использование Spring Bean в качестве медиатора
Агент	Нотификация о событии (Publish Event)	Формирование событий и публикация их во внешние системы

СЕРВИСЫ ДАННЫХ

Сервисы данных PolusESB — это удобный механизм для предоставления интерфейса веб-сервиса для данных, хранящихся в некоторых источниках. Источники данных, такие как реляционные базы данных, CSV-файлы, файлы Microsoft Excel и электронные таблицы Google, можно легко активировать с помощью Data Services Server. Теперь данные могут быть представлены и доступны безопасным (с помощью WS-Security) и надежным (с помощью WS-ReliableMessaging) способом, а также доступны для объединения с другими веб-сервисами.

Вы можете скачать сервер сервисов данных PolusESB из следующего расположения.

[Скачать Сервер сервисов данных PolusESB]

Установка сервера сервисов данных Polus ESB и доступ к консоли управления

- Если вы загрузили сервер сервисов данных PolusESB по приведенной выше ссылке, извлеките polusesb-X.X.X.zip в каталог в вашей локальной файловой системе.
- В командной строке запустите `bin/ integrator.bat{sh}`.
- После завершения запуска сервера откройте `http://localhost:9443/carbon` в своем любимом браузере. Отобразится страница приветствия сервера сервисов данных PolusESB. Нажмите на ссылку «Войти» в правом верхнем углу страницы. Затем вы можете войти в консоль управления, используя учетные данные администратора по умолчанию (имя пользователя = admin, пароль = admin).

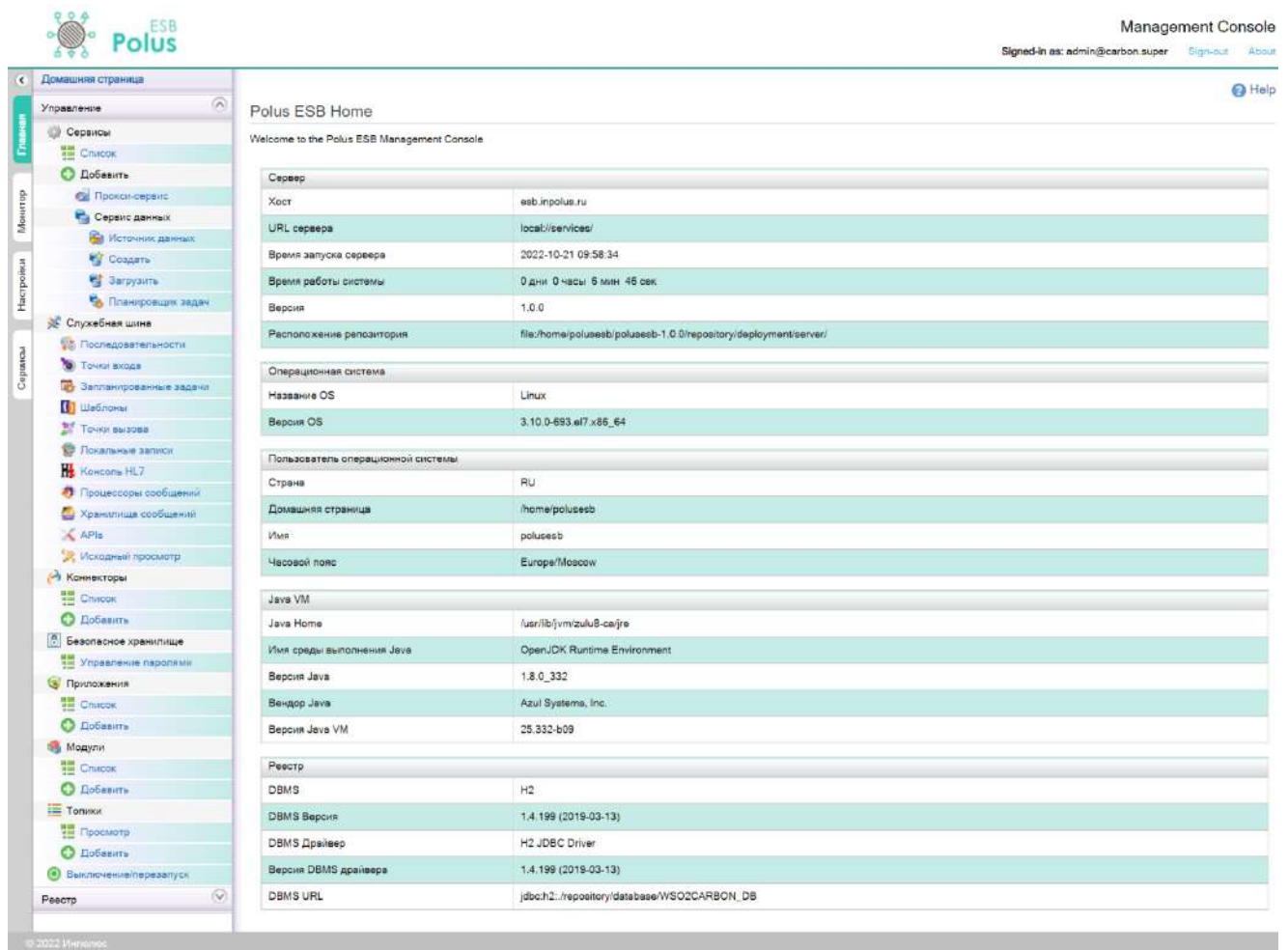


Рис.46. Домашняя страница сервисов данных PolusESB

Создание экземпляра (instance) БД и заполнение базы данными

В этом руководстве предполагается, что PostgreSQL настроен в вашей системе. Если нет, загрузите PostgreSQL 14 или более позднюю версию отсюда.

Инструкция по установке находится здесь.

Давайте создадим простую базу данных с одной таблицей. Откройте командную строку и введите «*psql -Upostgres*», чтобы получить доступ к командной строке PostgreSQL. Введите следующие команды, чтобы создать образец базы данных, создать таблицу и заполнить образцы данных.

```
postgres=# CREATE DATABASE employeedb;
postgres=# \c employeedb;
employeedb=# create table employee(id VARCHAR(10) NOT NULL PRIMARY KEY, name
VARCHAR(100), address VARCHAR(100));
employeedb=# insert into employee values('01','john','Boston');
employeedb=# insert into employee values('02','Micheal','Dallas');
employeedb=# insert into employee values('03','richard','Chicago');
```

```
employeeedb=# exit
```

Поскольку мы используем базу данных PostgreSQL для нашей демонстрации, мы должны скопировать драйвер PostgreSQL JDBC в каталог `polusesb-X.X.X\lib`

После копирования необходимого драйвера `jdbc` обязательно перезапустите Сервер сервисов данных PolusESB.

Создание сервиса данных

Мы готовы создать наш первый сервис данных с использованием Сервера сервисов данных PolusESB. Мы будем использовать пример базы данных PostgreSQL «employeeedb», которую мы создали на предыдущем шаге.

- Выберите «Создать» в левом навигационном меню консоли управления Сервера сервисов данных PolusESB. Отобразится Шаг-1 сервиса данных.
- Укажите название сервиса данных (например - EmployeeDataService) и при желании можете дать описание.
- Нажмите «Далее», как только вы введете данные.

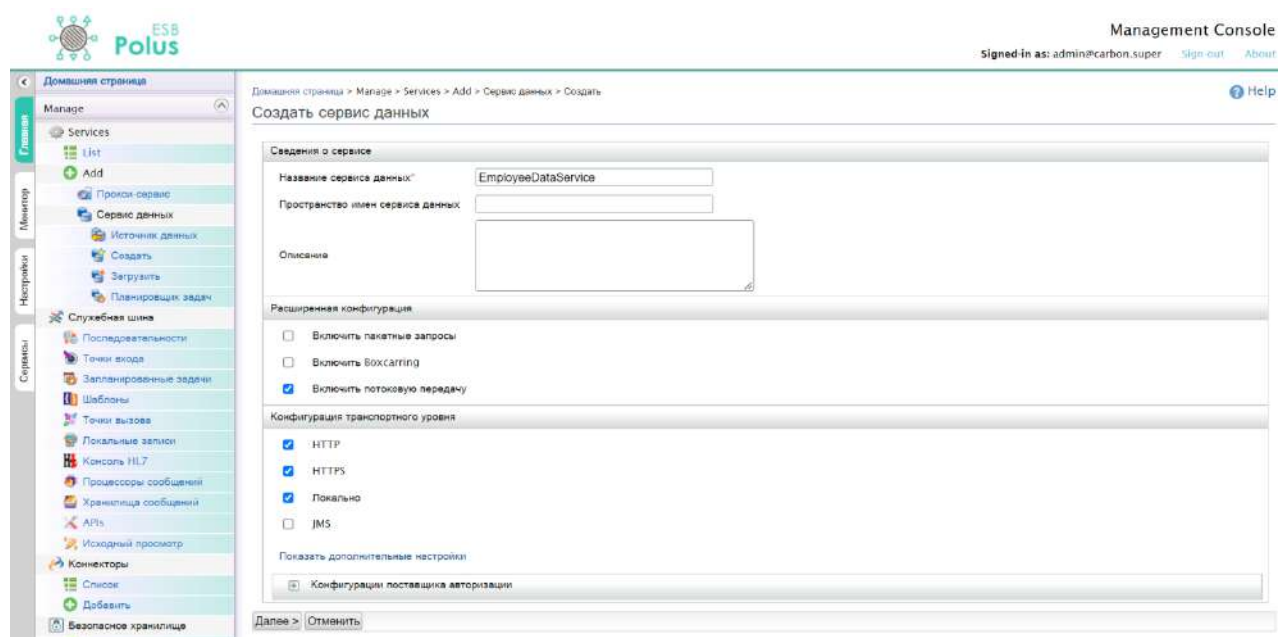


Рис. 47. Страница создания сервиса данных PolusESB

Теперь мы можем ввести данные источника данных на этом шаге. Начнем с того, что дадим источнику данных идентификатор «DataSourceEmp».

- Выберите тип источника данных как «RDBMS» и введите следующие значения.
 - Ядро базы данных = PostgreSQL
 - Класс драйвера = `org.postgresql.Driver`

- JDBC URL = jdbc:postgresql://{host}[:port]/[database]
 - Имя пользователя = postgres
- Нажмите кнопку «Тестовое соединение». Вы должны получить сообщение «Соединение с базой данных успешно выполнено с классом драйвера org.postgresql.Driver, jdbc url jdbc:postgresql://localhost:5432/employeeedb и именем пользователя postgres», если соединение с базой данных успешно.

Management Console
Signed-in as: admin@carbon.super [Sign out](#) [About](#)

Домашняя страница > Добавить источник данных [Help](#)

Редактировать источник данных (DataSourceEmp)

Новый источник данных

Идентификатор источника данных* DataSourceEmp

Тип источника данных* RDBMS По умолчанию

Ядро базы данных* PostgreSQL

Класс драйвера* org.postgresql.Driver

URL-адрес* jdbc:postgresql://localhost:5432/employeeedb

Имя пользователя postgres

Пароль ***** Использовать как скрытый псевдоним

Включить OData

Параметры конфигурации источника данных

Динамическая аутентификация пользователя

Тестовое соединение Сохранить Отменить

Рис.48. Создание источника данных

Нажмите кнопку «Сохранить». Вы увидите созданный источник данных в следующем виде.

Домашняя страница

Источники данных

Имя	Действия
DataSourceEmp	Редактировать источник данных Удалить источник данных
Добавить новый источник данных	

< Назад Далее > Завершить Сохранить как черновик Отменить

Рис.49. Список созданных источников данных

- Затем нажмите кнопку «Далее» в источниках данных. Изначально ваш сервис не включает никаких запросов.
- Нажмите «Добавить новый запрос», чтобы добавить новый запрос в наш сервис. Появится окно «Добавить новый запрос». Введите туда следующие значения.
 - Идентификатор запроса = DataQueryEmp

- SQL = select id, name, address from employee
 - Сгруппировано по элементам = employees
 - Имя строки = employee
 - Пространство имен строк = http://test.org
- Теперь нажмите кнопку «Добавление нового отображения вывода» в окне «Добавить новый запрос». Появится окно «Добавить отображение выходных данных».
- Добавьте три выходных сопоставления следующим образом:
1. Тип сопоставления = элемент
 - Имя выходного поля = id
 - Имя столбца источника данных = id
 - Тип схемы = string
 2. Тип сопоставления = element
 - Имя выходного поля = name
 - Имя столбца источника данных = name
 - Тип схемы = string
 3. Тип сопоставления = attribute
 - Имя выходного поля = address
 - Имя столбца источника данных = address
 - Тип схемы = string

Мы добавили один запрос в нашу конфигурацию сервиса данных. В этом примере мы не используем оператор SQL, который принимает входные параметры. Поэтому мы можем игнорировать «Входные сопоставления».

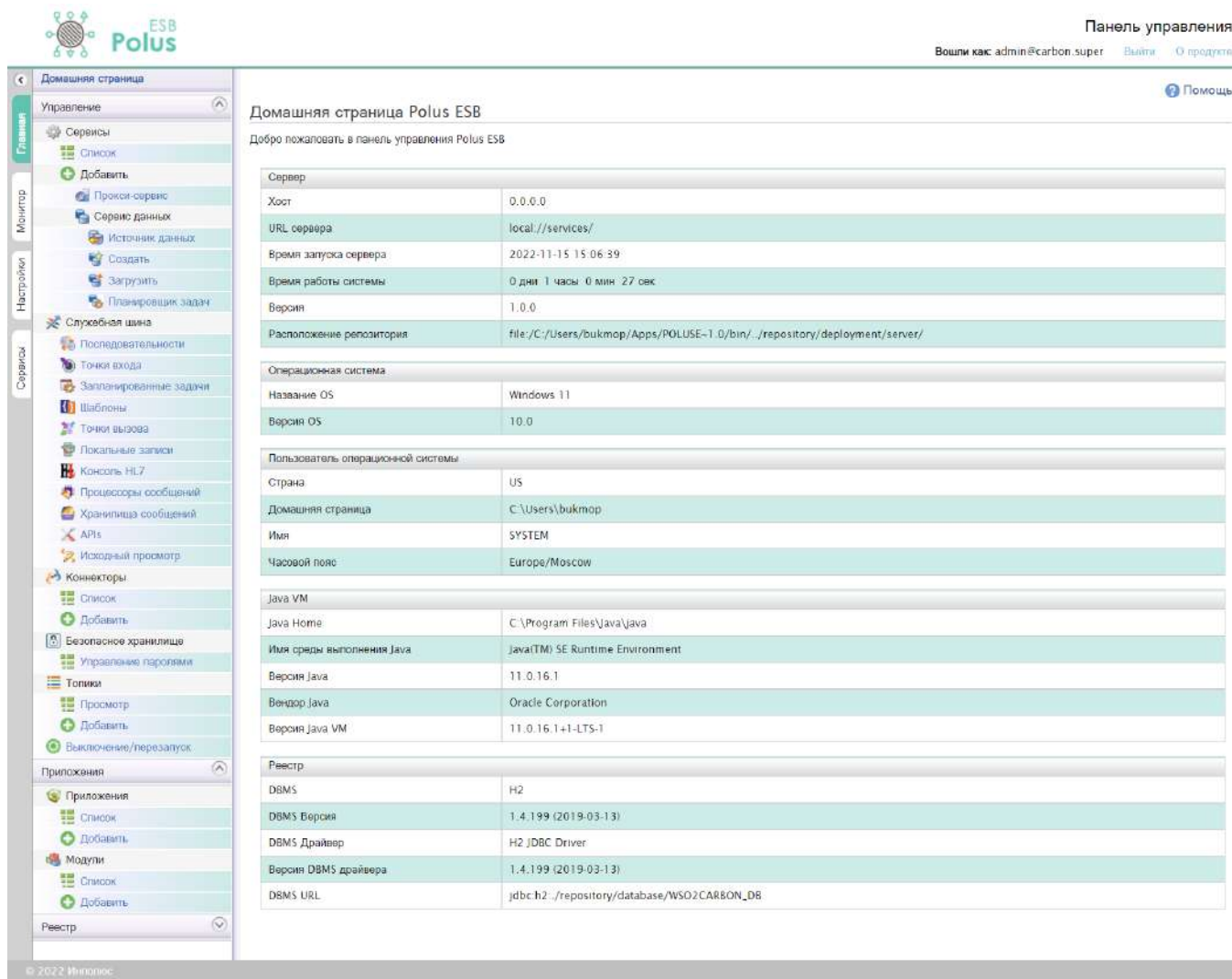


Рис.50. Добавить новый запрос

- Сохраните запрос и нажмите кнопку «Далее» на странице «Запросы». Вы будете перенаправлены на страницу «Операции», где сможете добавить новые операции в конфигурацию сервиса данных.

Домашняя страница > Добавить операцию

Добавить новую операцию (EmployeeDataService)

Операции

Название операции*

Описание

Идентификатор запроса*

+ Добавить параметры запроса в качестве параметров операции.

Включить потоковую передачу

Сохранить
Отменить

Рис.51. Добавление новых операций

Вы заметите, что запрос, который мы создали на предыдущем шаге, будет показан в раскрывающемся списке «Идентификатор запроса» окна «Добавить новую операцию». Укажите имя для операции (например: — `getEmployees`) и нажмите «Сохранить».

- Нажмите «Завершить», чтобы развернуть сервис. Ваш новый сервис данных будет указан на странице «Развернутые сервисы» следующим образом (для обновления нажать на «[n] групп сервисов»).

□

Домашняя страница > Управление > Сервисы > Список

Помощь

Развёрнутые сервисы

6 активных сервисов. 6 группы сервисов. 2 неисправных сервиса.

Тип сервиса: Все | Сервис:

Выбрать все | Отменить выбор

Удалить

Сервисы						
<input type="checkbox"/>	echo	axis2	Открытый	WSDL 1.1	WSDL 2.0	Попробовать Скачать
<input type="checkbox"/>	EmployeeDataService	data_service	Открытый	WSDL 1.1	WSDL 2.0	Попробовать
<input type="checkbox"/>	PolusLocalTajMS	data_service	Открытый	WSDL 1.1	WSDL 2.0	Попробовать
<input type="checkbox"/>	testH2service	data_service	Открытый	WSDL 1.1	WSDL 2.0	Попробовать
<input type="checkbox"/>	Version	axis2	Открытый	WSDL 1.1	WSDL 2.0	Попробовать Скачать
<input type="checkbox"/>	wso2carbon-sts	sts	Открытый	WSDL 1.1	WSDL 2.0	

Выбрать все | Отменить выбор

Удалить

Рис.52. Развернутые сервисы

Вызов сервиса данных

В этом разделе мы рассмотрим два разных способа вызова нашего сервиса данных без написания кода. Сначала мы вызовем наш сервис данных с помощью утилиты **Try-it**, которая интегрирована в консоль управления сервера сервисов данных PolusESB. затем мы вызовем его с помощью простого HTTP-запроса GET.

- Нажмите «EmployeeDataService» на странице «Развернутые сервисы». Вы будете перенаправлены на страницу «Панель управления сервисом» следующим образом.

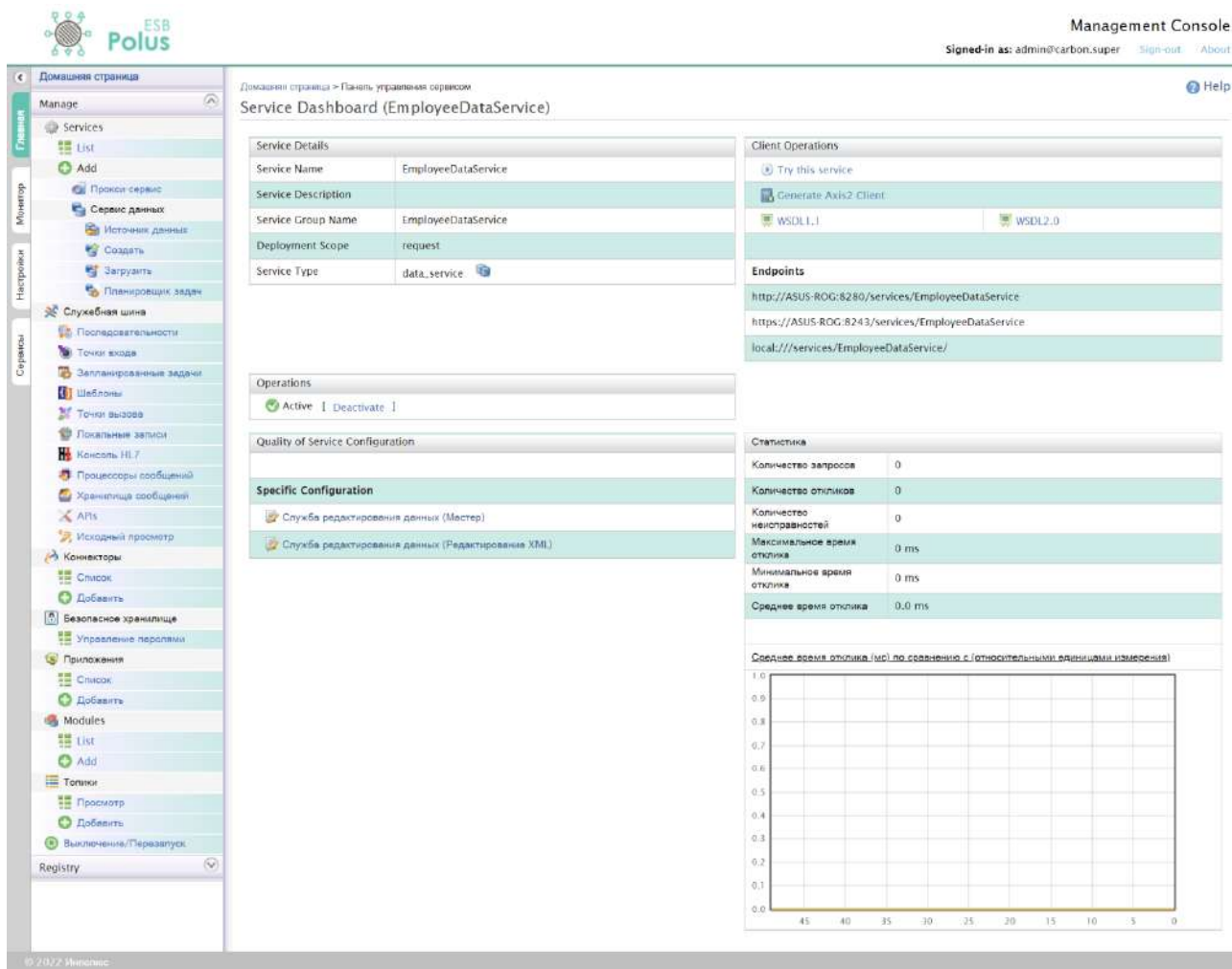


Рис.53. Управление сервисами

- Выберите ссылку «Попробовать этот сервис». Отобразится следующая страница.

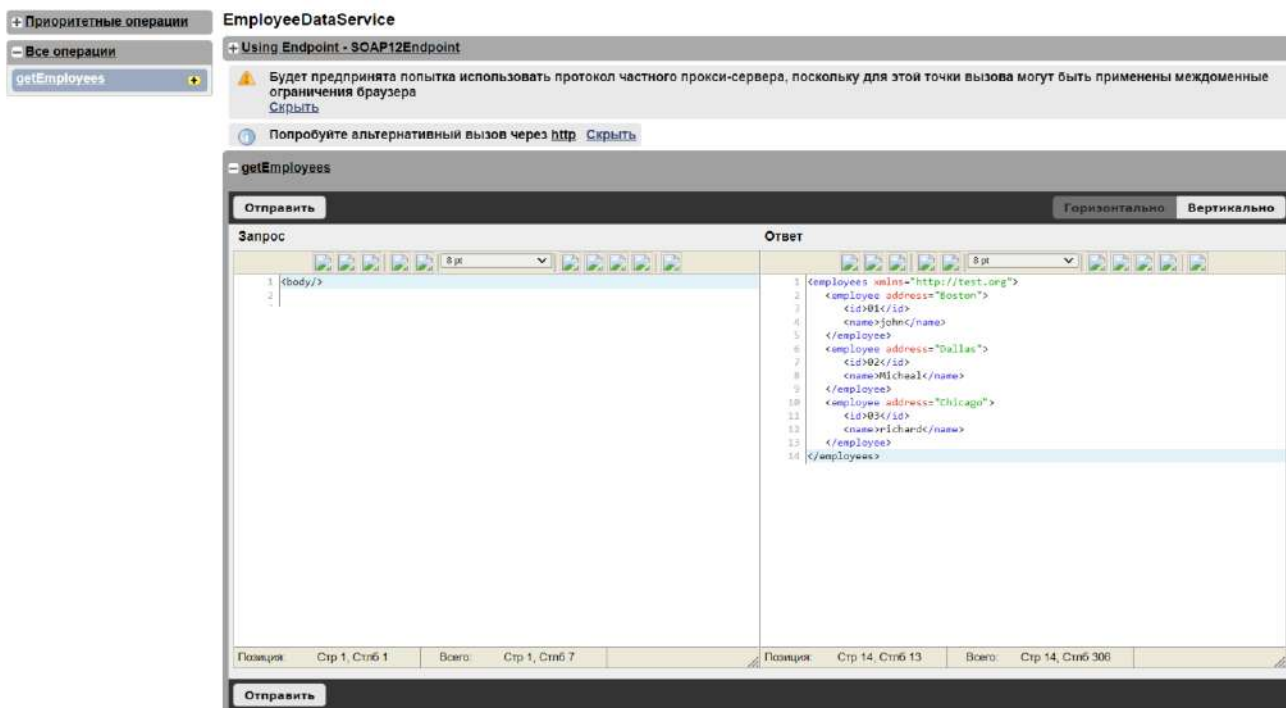


Рис.54. Инструмент Try-it

- Нажмите кнопку во вкладке getAllEmployees «Отправить», чтобы вызвать наш сервис

данных. Вы получите ответное сообщение на той же странице. Обратите внимание, что идентификатор и имя сотрудника будут возвращены в виде элементов xml, а адрес будет заключен в виде атрибута. Это связано с нашей конфигурацией сопоставления выходных данных на шаге 2 мастера. Мы настроили идентификатор и имя в качестве типов сопоставления элементов и адрес в качестве типа сопоставления атрибутов. Вы можете редактировать их и очень легко наблюдать за соответствующими изменениями в ответе с помощью **Try-it**.

- Теперь мы вызовем тот же сервис данных, используя наш второй подход, HTTP-запрос GET. Скопируйте и вставьте URL-адрес в адресную строку браузера и нажмите Enter.

`http://[ipaddress]:8280/services/EmployeeDataService/getEmployees`

Вы получите тот же результат, что и в подходе «**Try-it**».

Генератор автоматизированных сервисов данных

Механизм сервисов данных Polus ESB предоставляет возможность их автоматического создания с использованием заданной структуры базы данных. При создании сервиса генератор сервисов данных будет учитывать структуру таблицы. В соответствии с заданной структурой таблицы операции "SELECT", "INSERT", "UPDATE" и "DELETE" будут созданы автоматически.

Шаг 1 — Создайте источник данных Carbon

Перед созданием сервисов данных следует создать источник данных Carbon в соответствии со свойствами базы данных.

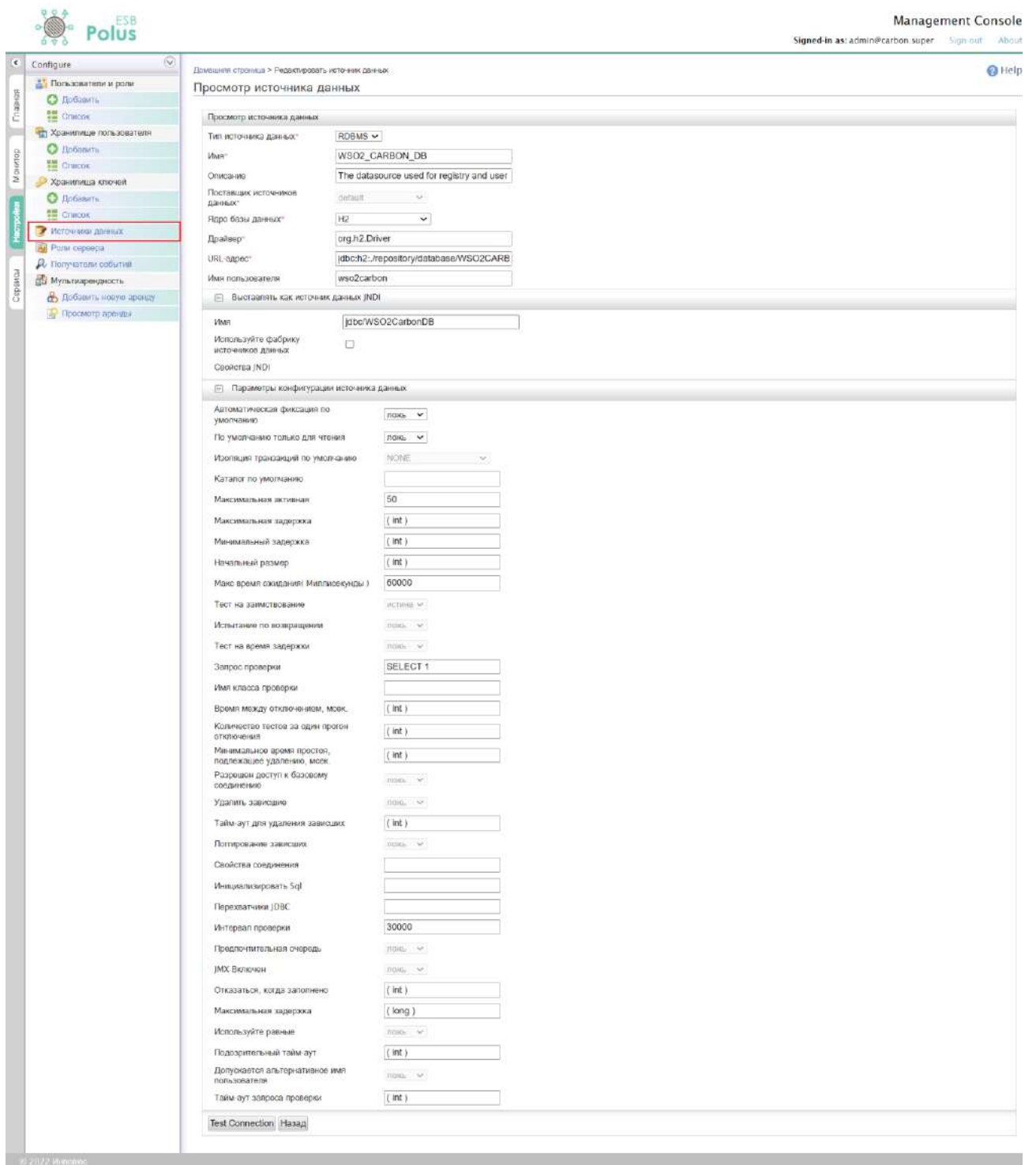


Рис.55. Создание источника данных Carbon

Шаг 2 — выберите источник данных

Чтобы начать генерацию сервисов данных, нажмите ссылку «Источник данных» в левом меню.

На первом появившемся экране выберите источник данных Carbon и имя базы данных, как показано ниже.

Выберите источник данных

Сведения об источнике Carbon

Источник(ы) данных Carbon* WSO2_CARBON_DB ▾

Имя базы данных* DATA_SERV_SAMP

Далее > Отменить

Рис.56. Выбор источника данных

Шаг 3 - Выберите схему базы данных

После выбора источника данных необходимо выбрать схему БД, чтобы определить необходимые таблицы.

Домашняя страница

Список схем

выберите схему

Имя схемы

INFORMATION_SCHEMA PUBLIC

< Назад Далее > Отменить

Рис.57. Выбор схемы БД

Шаг 4 — Выберите таблицы

После выбора схемы структура таблицы будет отображаться в соответствии с заданной схемой. Выберите предпочтительные таблицы для создания сервиса данных.

Настройте генерацию сервисов

Выбрать все | Отменить выбор

Выберите таблицу(ы)

<input type="checkbox"/> REG_ASSOCIATION	<input type="checkbox"/> REG_CLUSTER_LOCK	<input type="checkbox"/> REG_COMMENT
<input checked="" type="checkbox"/> REG_CONTENT	<input type="checkbox"/> REG_CONTENT_HISTORY	<input type="checkbox"/> REG_LOG
<input type="checkbox"/> REG_PATH	<input checked="" type="checkbox"/> REG_PROPERTY	<input type="checkbox"/> REG_RATING
<input type="checkbox"/> REG_RESOURCE	<input type="checkbox"/> REG_RESOURCE_COMMENT	<input type="checkbox"/> REG_RESOURCE_HISTORY
<input type="checkbox"/> REG_RESOURCE_PROPERTY	<input type="checkbox"/> REG_RESOURCE_RATING	<input type="checkbox"/> REG_RESOURCE_TAG
<input type="checkbox"/> REG_SNAPSHOT	<input type="checkbox"/> REG_TAG	<input type="checkbox"/> UM_ACCOUNT_MAPPING
<input type="checkbox"/> UM_CLAIM	<input type="checkbox"/> UM_CLAIM_BEHAVIOR	<input type="checkbox"/> UM_DIALECT
<input type="checkbox"/> UM_DOMAIN	<input type="checkbox"/> UM_HYBRID_REMEMBER_ME	<input type="checkbox"/> UM_HYBRID_ROLE
<input type="checkbox"/> UM_HYBRID_USER_ROLE	<input type="checkbox"/> UM_MODULE	<input type="checkbox"/> UM_MODULE_ACTIONS
<input type="checkbox"/> UM_PERMISSION	<input type="checkbox"/> UM_PROFILE_CONFIG	<input checked="" type="checkbox"/> UM_ROLE
<input type="checkbox"/> UM_ROLE_PERMISSION	<input type="checkbox"/> UM_SHARED_USER_ROLE	<input type="checkbox"/> UM_SYSTEM_ROLE
<input type="checkbox"/> UM_SYSTEM_USER	<input type="checkbox"/> UM_SYSTEM_USER_ROLE	<input type="checkbox"/> UM_TENANT
<input checked="" type="checkbox"/> UM_USER	<input type="checkbox"/> UM_USER_ATTRIBUTE	<input type="checkbox"/> UM_USER_PERMISSION
<input type="checkbox"/> UM_USER_ROLE		

< Назад Далее > Отменить

Рис.58. Выбор таблиц БД

Шаг 5 - Выберите режим генерации сервиса

После определения таблиц необходимо указать режим генерации сервиса. Существует два варианта создания сервиса данных. 1. как единый сервис - где все операции для каждой таблицы приведены в одном сервисе данных. 2. Несколько сервисов для каждой таблицы - где для данной таблицы будут указаны только операции, связанные с этой таблицей. В следующем примере создается один сервис для всех таблиц, как показано ниже.

Домашняя страница > Создание службы

Создание службы

Выберите режим создания сервиса

Единый сервис - создает один сервис для всех выбранных таблиц

Мульти сервис - создает сервис для каждой таблицы

Пространство имен сервиса данных:

Название сервиса данных*:

< Назад Далее > Отменить

Рис.59. Выбор режима генерации сервисов

После выбора режима создания сервиса будут развернуты сервисы данных. Нажмите

«Далее», чтобы просмотреть развернутые сервисы, и нажмите «Завершить».

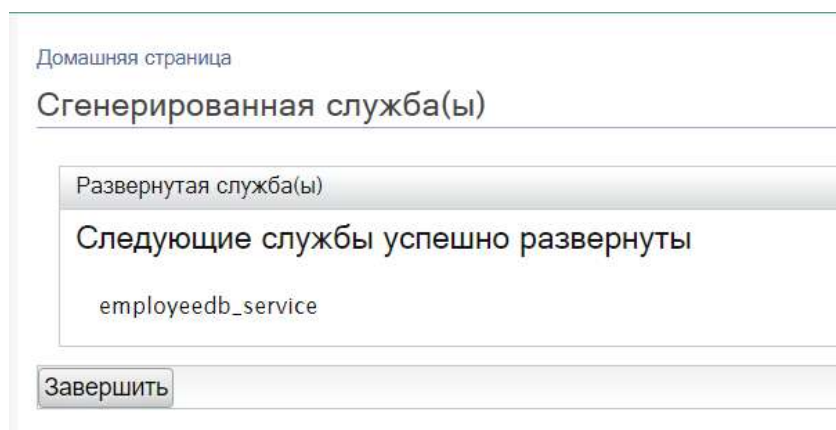


Рис.60. Сервисы созданы

Шаг 6. Сервис развернут

Теперь мы создали сервис данных. Вы можете перейти к списку развернутых сервисов, который находится в разделе «Управление/Сервисы/Список». Список созданных сервисов будет выглядеть следующим образом. Если выбран режим нескольких сервисов, будет создано несколько сервисов в соответствии с выбранным количеством таблиц.

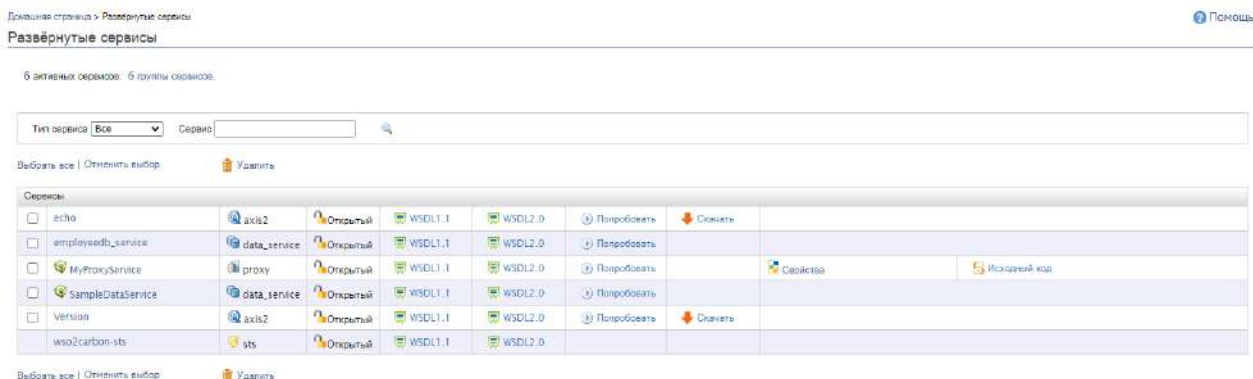


Рис.61. Развернутые сервисы, включая сервис данных

Шаг 7. Попробуйте свой сервис — режим единого сервиса

Нажмите ссылку «Попробовать», чтобы вызвать сервис. Как видно на рис.62, операции (выбрать, обновить, вставить и удалить) для всех таблиц встроены в один и тот же сервис данных.

Операции

Название операции	Запрос	Действия
select_with_key_REG_CONTENT_operation	select_with_key_REG_CONTENT_query	Операция редактирования Операция удаления
select_all_REG_PROPERTY_operation	select_all_REG_PROPERTY_query	Операция редактирования Операция удаления
update_REG_CONTENT_operation	update_REG_CONTENT_query	Операция редактирования Операция удаления
select_with_key_UM_USER_operation	select_with_key_UM_USER_query	Операция редактирования Операция удаления
select_with_key_UM_ROLE_operation	select_with_key_UM_ROLE_query	Операция редактирования Операция удаления
insert_REG_PROPERTY_operation	insert_REG_PROPERTY_query	Операция редактирования Операция удаления
insert_UM_USER_operation	insert_UM_USER_query	Операция редактирования Операция удаления
insert_REG_CONTENT_operation	insert_REG_CONTENT_query	Операция редактирования Операция удаления
insert_UM_ROLE_operation	insert_UM_ROLE_query	Операция редактирования Операция удаления
insert__operation	insert__query	Операция редактирования Операция удаления
delete_REG_PROPERTY_operation	delete_REG_PROPERTY_query	Операция редактирования Операция удаления
delete_UM_USER_operation	delete_UM_USER_query	Операция редактирования Операция удаления
update_REG_PROPERTY_operation	update_REG_PROPERTY_query	Операция редактирования Операция удаления
select_all_REG_CONTENT_operation	select_all_REG_CONTENT_query	Операция редактирования Операция удаления
update_UM_USER_operation	update_UM_USER_query	Операция редактирования Операция удаления
select_all__operation	select_all__query	Операция редактирования Операция удаления
select_all_UM_ROLE_operation	select_all_UM_ROLE_query	Операция редактирования Операция удаления
delete_REG_CONTENT_operation	delete_REG_CONTENT_query	Операция редактирования Операция удаления
select_all_UM_USER_operation	select_all_UM_USER_query	Операция редактирования Операция удаления
update_UM_ROLE_operation	update_UM_ROLE_query	Операция редактирования Операция удаления
delete_UM_ROLE_operation	delete_UM_ROLE_query	Операция редактирования Операция удаления
select_with_key_REG_PROPERTY_operation	select_with_key_REG_PROPERTY_query	Операция редактирования Операция удаления

[+ Добавить новую операцию](#)

[< Назад](#) [Далее >](#) [Завершить](#) [Сохранить как черновик](#) [Отменить](#)

Рис.62. Режим одиночного сервиса

Шаг 8. Попробуйте свой сервис - Режим нескольких сервисов

Если выбран режим нескольких сервисов, для каждой таблицы будут созданы разные сервисы данных.

Развёрнутые сервисы

11 активных сервисов. 11 группы сервисов.

Тип сервиса: Все Сервис:

Выбрать все | Отменить выбор Удалить

Сервисы	Иконка	Статус	WSDL1.1	WSDL2.0	Действия	Дополнительно
<input type="checkbox"/> echo		Открытый			Попробовать	Скачать
<input type="checkbox"/> EmployeeDataService		Открытый			Попробовать	
<input type="checkbox"/> employeeedb_service		Открытый			Попробовать	
<input type="checkbox"/> MyProxyService		Открытый			Попробовать	Свойства
<input type="checkbox"/> REG_CONTENT_DataService		Открытый			Попробовать	
<input type="checkbox"/> REG_PROPERTY_DataService		Открытый			Попробовать	
<input type="checkbox"/> sampleDataService		Открытый			Попробовать	
<input type="checkbox"/> UM_ROLE_DataService		Открытый			Попробовать	
<input type="checkbox"/> UM_USER_DataService		Открытый			Попробовать	
<input type="checkbox"/> Version		Открытый			Попробовать	Скачать
<input type="checkbox"/> wso2carbon-sts		Открытый				

Выбрать все | Отменить выбор Удалить

Рис.63. Режим нескольких сервисов

Редактирование сервиса данных

Вы можете изменить развернутый сервис, щелкнув на имени сервиса в списке развернутых сервисов. Он предоставляет два варианта редактирования сервиса: «Редактировать сервис с помощью мастера» и «Редактировать XML».

The screenshot displays the ESB Polus management interface. The main content area is titled "Панель управления сервисом (EmployeeDataService)". It contains several sections:

- Информация о сервисе:** A table with the following data:

Название сервиса	EmployeeDataService
Описание сервиса	
Группа сервисов	EmployeeDataService
Область развертывания	request
Тип сервиса	data_service
- Клиентские операции:** Includes buttons for "Попробовать", "Создать клиент Axis2", and "WSDL 1.1" / "WSDL 2.0".
- Точки входа:** Lists endpoints:
 - http://ASUS-ROG-8280/services/EmployeeDataService
 - https://ASUS-ROG-8243/services/EmployeeDataService
 - local://services/EmployeeDataService/
- Статистика:** A table showing performance metrics:

Количество запросов	0
Количество откликов	0
Количество неисправностей	0
Максимальное время отклика	0 ms
Минимальное время отклика	0 ms
Среднее время отклика	0.0 ms
- Специфическая конфигурация:** Lists configuration options:
 - Служба редактирования данных (Мастер)
 - Служба редактирования данных (Редактирование XML)

At the bottom right, there is a graph titled "Среднее время отклика (мс) по сравнению с (относительными единицами измерения)" with a y-axis from 0.0 to 1.0 and an x-axis from 45 to 0.

Рис.64. Редактирование сервиса данных

Вариант 1. Редактировать сервис данных с помощью мастера.

Если вы выберете опцию «Редактировать сервис данных (мастер)», вы будете перенаправлены к мастеру сервиса данных. Там вы можете отредактировать необходимую информацию и продолжить, нажав «Далее». Наконец, нажав «Завершить», можно сохранить изменения. Это будет аналогичный мастер, используемый при создании сервиса.

Служба редактирования данных (EmployeeDataService)

Сведения о сервисе	
Название сервиса данных*	EmployeeDataService
Пространство имен сервиса данных	
Описание	
Расширенная конфигурация	
<input type="checkbox"/>	Включить пакетные запросы
<input type="checkbox"/>	Включить Woxcarring
<input checked="" type="checkbox"/>	Включить потоковую передачу
Конфигурация транспортного уровня	
<input checked="" type="checkbox"/>	HTTP
<input checked="" type="checkbox"/>	HTTPS
<input checked="" type="checkbox"/>	Локально
<input type="checkbox"/>	JMS
Показать дополнительные настройки	
+ Конфигурации поставщика авторизации	
<input type="button" value="Далее >"/> <input type="button" value="Отменить"/>	

Рис.65. Служба редактирования сервиса данных

После нажатия «Далее», на рисунке выше, вы можете редактировать источник данных, нажав ссылку «Редактировать источник данных».

Источники данных




Имя	Действия
DataSourceEmp	 Редактировать источник данных  Удалить источник данных
 Добавить новый источник данных	
<input type="button" value="Назад"/> <input type="button" value="Далее >"/> <input type="button" value="Завершить"/> <input type="button" value="Сохранить как черновик"/> <input type="button" value="Отменить"/>	

Рис.66. Редактировать источник данных

Измените необходимые значения на странице мастера «Редактировать источник данных» и нажмите «Сохранить».

Домашняя страница > Добавить источник данных

Редактировать источник данных (DataSourceEmp)

Новый источник данных

Идентификатор источника данных* DataSourceEmp

Тип источника данных* RDBMS По умолчанию

Ядро базы данных* PostgreSQL

Класс драйвера* org.postgresql.Driver

URL-адрес* jdbc:postgresql://localhost:5432/employeedb

Имя пользователя postgres

Пароль ***** Использовать как скрытый псевдоним

Включить OData

Параметры конфигурации источника данных

Динамическая аутентификация пользователя

Тестовое соединение Сохранить Отменить

Рис. 67. Сохранение изменений параметров источника данных

Теперь вы окажетесь на странице «Источники данных». Нажмите «Далее», чтобы изменить запросы. Нажмите ссылку «Редактировать запрос», измените необходимую информацию и нажмите «Сохранить».

Домашняя страница

Запросы

Запрос	Тип	Действия
DataQueryEmp	DataSourceEmp	<input type="button" value="Редактировать запрос"/> <input type="button" value="Удалить запрос"/>

Рис.68. Редактирование запроса

Нажав «Далее» на странице «Запросы», вы перейдете на страницу «Операции». Щелкните ссылку «Редактировать операцию», измените необходимую информацию и нажмите «Сохранить».

Домашняя страница > SOAP операции

Операции

Название операции	Запрос	Действия
getEmployees	DataQueryEmp	<input type="button" value="Операция редактирования"/> <input type="button" value="Операция удаления"/>

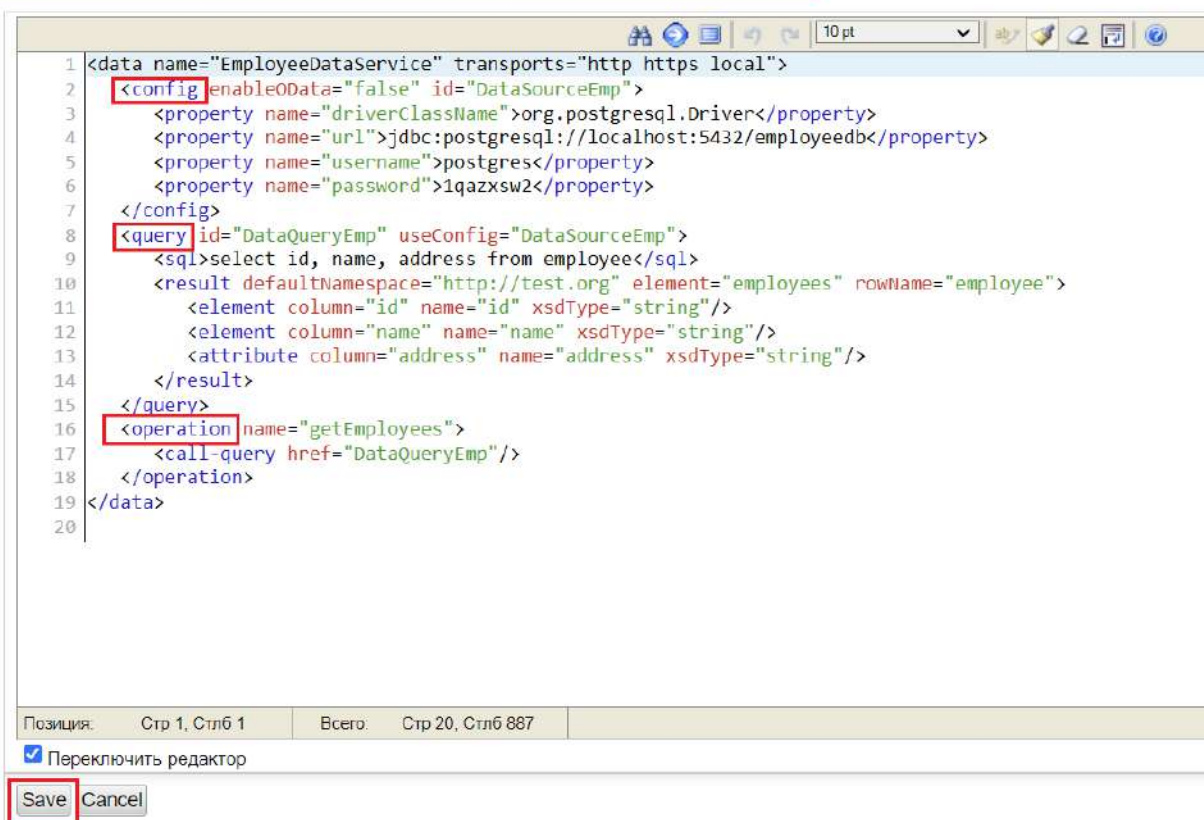
Рис.69. Редактирование операции

Вариант 2. Редактирование сервиса данных с помощью редактора XML

Если вы выберете параметр «Редактировать сервис данных (редактировать XML)», вы будете перенаправлены в XML-редактор сервиса данных. Если вы знакомы с языком конфигурации сервиса данных, вы можете напрямую редактировать необходимые конфигурации с помощью этого редактора. Элемент «config» определяет конфигурацию источника данных. Элемент данных описывает сервис данных или ресурс. Элемент «query» определяет запрос, а элемент «operation» определяет операцию. У нас может быть любое количество элементов конфигурации, запросов и операций. Далее мы можем определить элемент «ресурс», чтобы предоставлять данные как ресурсы RESTful.

Домашняя страница > XML-редактор службы передачи данных

XML-редактор службы передачи данных(EmployeeDataService)



```
1 <data name="EmployeeDataService" transports="http https local">
2   <config enableOData="false" id="DataSourceEmp">
3     <property name="driverClassName">org.postgresql.Driver</property>
4     <property name="url">jdbc:postgresql://localhost:5432/employeeedb</property>
5     <property name="username">postgres</property>
6     <property name="password">1qazxsw2</property>
7   </config>
8   <query id="DataQueryEmp" useConfig="DataSourceEmp">
9     <sql>select id, name, address from employee</sql>
10    <result defaultNamespace="http://test.org" element="employees" rowName="employee">
11      <element column="id" name="id" xsdType="string"/>
12      <element column="name" name="name" xsdType="string"/>
13      <attribute column="address" name="address" xsdType="string"/>
14    </result>
15  </query>
16  <operation name="getEmployees">
17    <call-query href="DataQueryEmp"/>
18  </operation>
19 </data>
20
```

Позиция: Стр 1, Стлб 1 Всего: Стр 20, Стлб 887

Переключить редактор

Save Cancel

Рис.70. Редактирование сервиса данных с помощью редактора XML

Загрузка сервиса данных

Шаг 1

Вы можете легко загрузить созданный сервис данных, щелкнув ссылку «Загрузить» в левом меню «Управление/Сервисы/Добавить/Сервис данных». После того, как вы нажмете на ссылку, появится следующая страница.

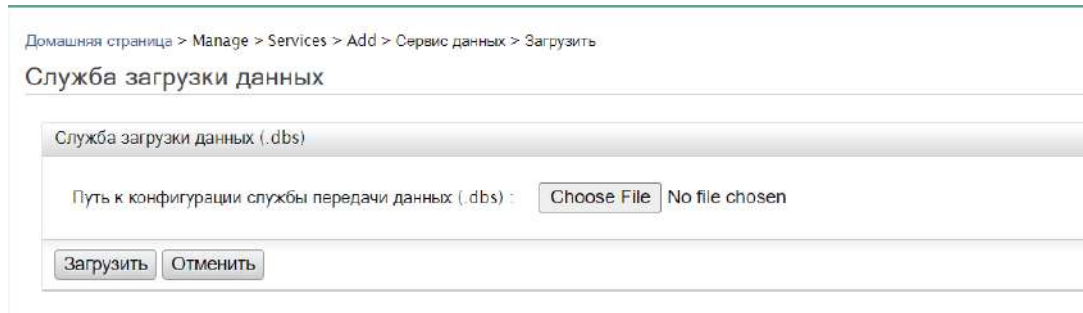


Рис.71: Мастер загрузки сервиса данных

Шаг 2

Выберите файл сервиса данных (.dbs), который вы предпочитаете загрузить.

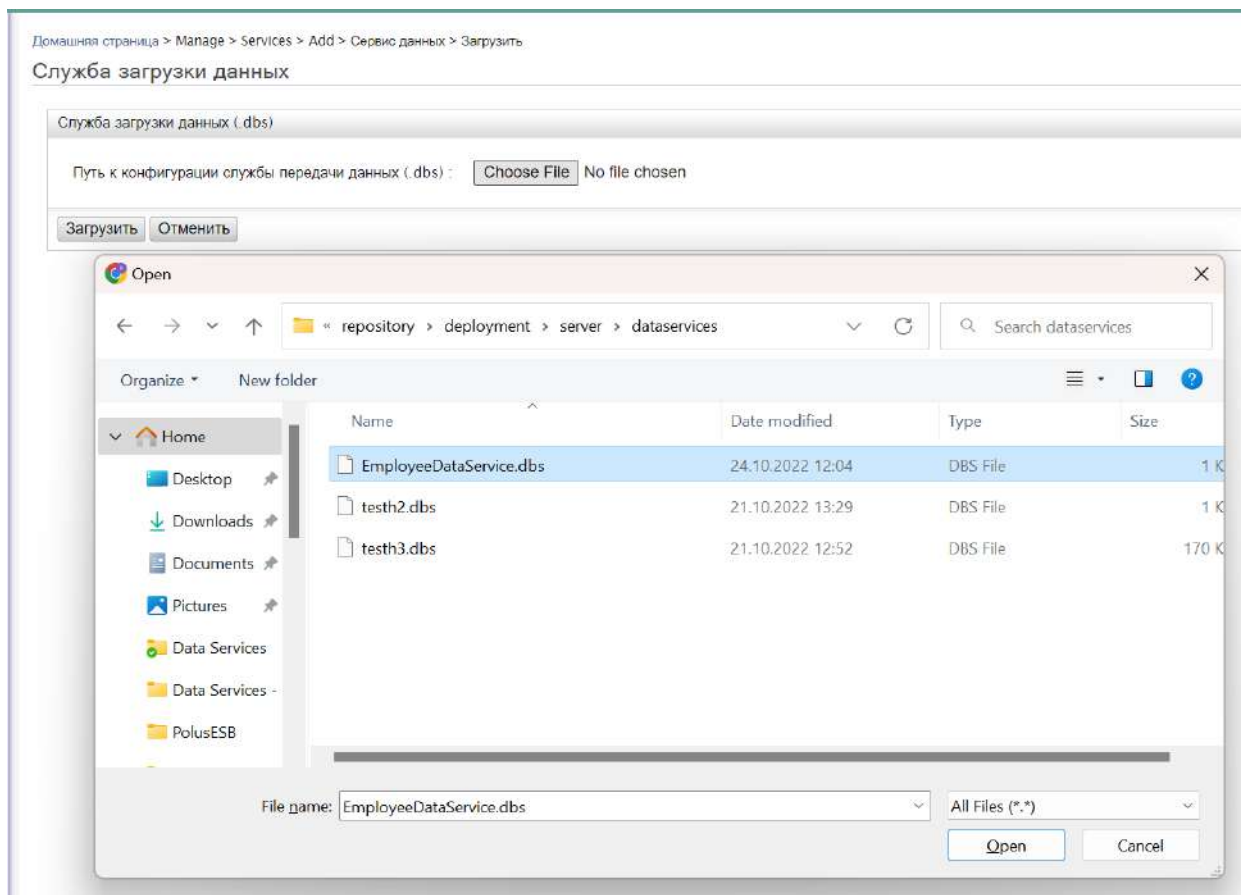


Рис.72. Выбор файла сервиса данных для загрузки

Шаг 3

После выбора файла сервиса данных он будет отображаться следующим образом.

Домашняя страница > Manage > Services > Add > Сервис данных > Загрузить

Служба загрузки данных

Служба загрузки данных (.dbs)

Путь к конфигурации службы передачи данных (.dbs): EmployeeDataService.dbs

Рис.73. Выбранный файл сервиса данных

Шаг 4

Выбранный сервис данных будет развернут и указан в списке развернутых сервисов, как показано ниже.

Домашняя страница > Управление > Сервисы > Список

[Помощь](#)

Развёрнутые сервисы

6 активных сервисов. 6 группы сервисов. [2 неисправных сервиса](#).

Тип сервиса: Сервис:

Сервисы							
<input type="checkbox"/>	echo	axis2	Открытый	WSDL 1.1	WSDL 2.0	<input type="button" value="Попробовать"/>	<input type="button" value="Скачать"/>
<input type="checkbox"/>	EmployeeDataService	data_service	Открытый	WSDL 1.1	WSDL 2.0	<input type="button" value="Попробовать"/>	
<input type="checkbox"/>	PolusLocalToJMS	data_service	Открытый	WSDL 1.1	WSDL 2.0	<input type="button" value="Попробовать"/>	
<input type="checkbox"/>	testH2service	data_service	Открытый	WSDL 1.1	WSDL 2.0	<input type="button" value="Попробовать"/>	
<input type="checkbox"/>	Version	axis2	Открытый	WSDL 1.1	WSDL 2.0	<input type="button" value="Попробовать"/>	<input type="button" value="Скачать"/>
<input type="checkbox"/>	wso2carbon-sts	sts	Открытый	WSDL 1.1	WSDL 2.0		

Рис.74. Развернутый сервис данных

Удаление сервиса данных

Выберите сервисы, которые вы хотите удалить, из списка сервисов и щелкните ссылку удаления, расположенную над таблицей «Группы сервисов», чтобы удалить сервисы. Если сервис был развернут как неисправный сервис, он будет отображаться на странице «Развернутые сервисы» как «**[n] неисправных сервисов:**». Как и для развернутых сервисов, вы можете выбрать соответствующие неисправные сервисы, отметив их галочкой и нажав «Удалить».

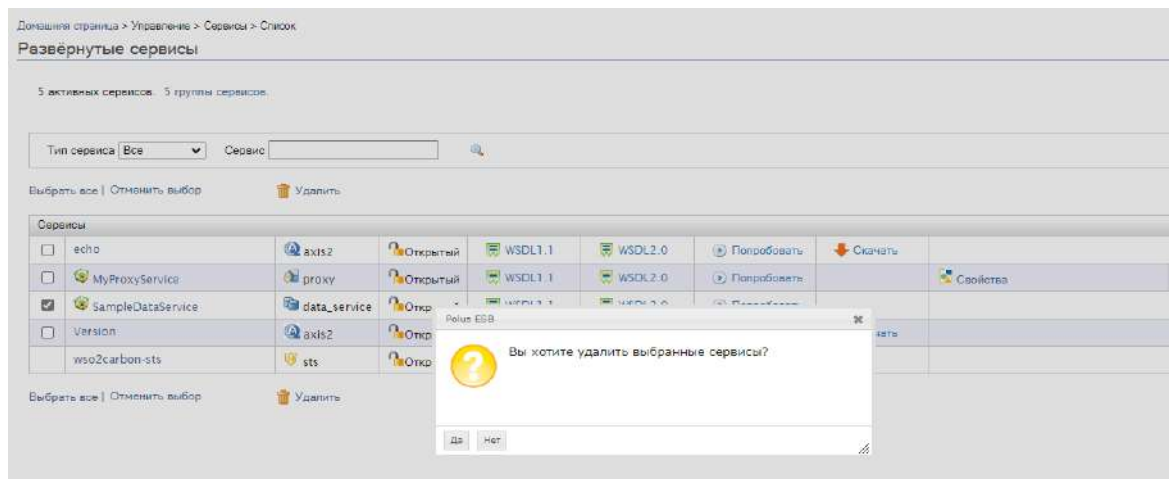


Рис.75. Удаление сервиса

ПРИЛОЖЕНИЯ

Приложение — это набор артефактов, которые можно развернуть на экземпляре Polus ESB, чтобы легко перенести решение на основе веб-сервисов из одной среды в другую.

При разработке решений с использованием платформы Polus ESB в ней может использоваться множество типов артефактов. Например, сервисы Axis2, артефакты Synapse (точки вызова, прокси-сервисы, медиаторы и т.д.), ресурсы реестра, сервисы данных и т.д. Обычно эти артефакты разрабатываются в среде разработки, а затем один за другим перемещаются в промежуточную и производственную среды. При каждой настройке вы должны вручную настраивать эти артефакты, чтобы создавать все решения целиком. Если ваше решение содержит много артефактов, этот процесс может занять много времени и усилий.

Таким образом, концепция приложения пытается решить эту проблему, предоставляя возможность экспортировать все ваше решение как единый артефакт приложения, и просто развернуть его. Инструментальную среду разработки веб-сервисов можно использовать для разработки ваших отдельных артефактов и экспорта их как единого артефакта carbon-приложения. Затем вы можете развернуть его на любом сервере, который имеет возможности развертывания приложений.

Настройка роли сервера

Во время разработки приложения вы должны указать роль сервера для каждого артефакта в приложении. Это связано с тем, что во время разработки мы не знаем, на каком физическом сервере будет развернут тот или иной артефакт. Таким образом, роль логического сервера назначается каждому артефакту.

Пример: Если вы разрабатываете сервис Axis2 и собираетесь развернуть все свои сервисы в одном экземпляре AppServer в рабочей установке, вы можете установить роль сервера как "appserver1".

Во время развертывания приложения вы должны установить роли сервера для каждого экземпляра Carbon в вашей настройке. Вы можете установить одну или несколько ролей, которые могут выполняться текущим экземпляром Carbon. Имя продукта задается как значение по умолчанию для роли сервера каждого продукта. Для установки роли сервера можно использовать следующие три метода.

Использование пользовательского интерфейса диспетчера ролей сервера

Это самый простой способ установить роль сервера для вашего экземпляра Carbon. Перейдите в «**Настройки**» > «**Серверные роли**», и вы сможете просто добавлять/удалять роли сервера. Подробнее об этом читайте на странице «Помощь». Это наиболее рекомендуемый

способ настройки ролей сервера.

Использование POLUSES_HOME/conf/carbon.xml

В файле carbon.xml вы можете найти следующий элемент.

```
<ServerRoles>
    <Role>ApplicationServer</Role>
</ServerRoles>
```

Здесь вы можете установить несколько элементов Role, чтобы указать роли, исполняемые текущим сервером.

Пример: Если вы хотите, чтобы этот сервер действовал как серверы appserver1 и dataservices1, конфигурация будет следующей. Перед настройкой необходимо убедиться, что текущий сервер поддерживает развертывание сервисов Axis2 и сервисов данных.

```
<ServerRoles>
    <Role>appserver1</Role>
    <Role>dataservices1</Role>
</ServerRoles>
```

Таким образом, когда вы развертываете артефакт carbon-приложения на этом сервере, все артефакты, имеющие две вышеуказанные роли сервера, будут развернуты на текущем экземпляре. Другие будут игнорироваться.

Использование системного свойства

Вы можете использовать системное свойство «serverRoles», чтобы указать роли сервера, которые могут выполняться текущим экземпляром Carbon. Поэтому, когда вы запускаете сервер, передайте роли сервера в виде списка, разделенного запятыми, как показано ниже.

Пример: sh integrator.sh -DserverRoles=appserver1,dataservices1

Примечание. После использования пользовательского интерфейса ролей сервера для настройки ролей сервера вы не сможете изменить эту конфигурацию двумя другими способами. Это связано с тем, что при использовании пользовательского интерфейса роли сервера будут храниться в реестре, и после этого значения в реестре будут иметь приоритет над значениями, установленными с помощью двух других методов.

Развертывание приложения

Вы можете загрузить приложение (файл .car) в работающий экземпляр Carbon с помощью консоли управления. Перейдите в «Управление»> «Приложения»> «Добавить», и вы будете перенаправлены на следующую страницу.



Рис.76. Загрузка приложения

Здесь вы можете выбрать свой файл .car, просматривая файловую систему. После того, как вы загрузите файл .car, вы сможете увидеть журнал на своей консоли о статусе вашего приложения. Если развертывание приложения прошло успешно, перейдите в раздел «Управление» -> «Приложения» -> «Список». Вы сможете увидеть развернутое carbon-приложение на этой странице следующим образом.

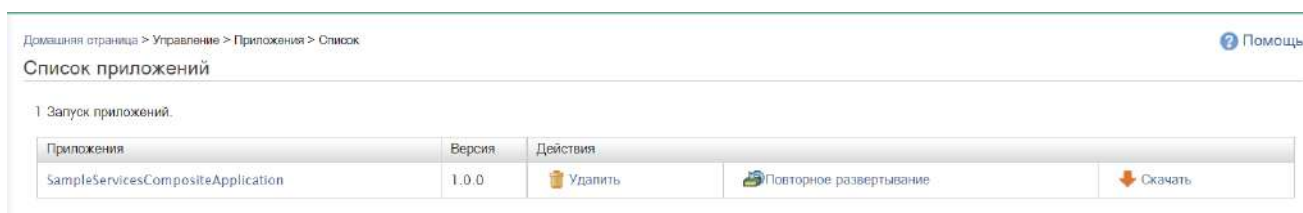


Рис.77. Список развернутых приложений

Если вы нажмете на имя своего приложения, вы будете перенаправлены на страницу панели мониторинга приложения, которая содержит всю информацию о содержимом развернутого приложения. В нем отдельно перечислены все артефакты, и вы можете перейти к информационным страницам этих артефактов, нажав на эти ссылки. Страница панели инструментов приложения показана на следующем изображении.

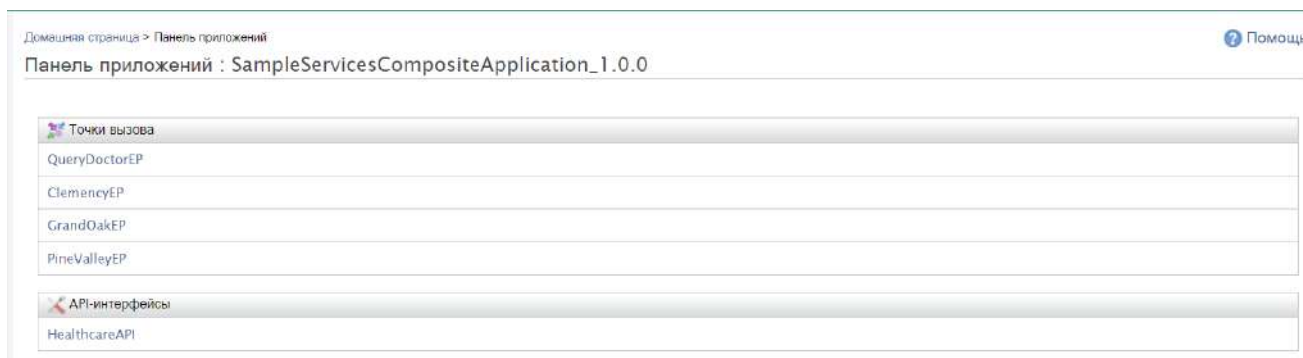


Рис.78. Список артефактов приложения

Примечание. В качестве альтернативы загрузке приложения с помощью консоли управления вы можете вручную скопировать архив .car в папку CARBON_HOME/repository/deployment/server/carbonapps, и он будет развернут в горячем режиме.

Удаление приложения

Не рекомендуется удалять отдельные артефакты приложения. Всё приложение должно быть удалено. Перейдите в «Управление» -> «Приложения» -> «Список» и нажмите значок «Удалить» перед приложением, которое нужно удалить.

Примечание. В качестве альтернативы удалению carbon-приложения с помощью консоли управления вы можете вручную удалить свой архив .car из папки CARBON_HOME/repository/deployment/server/carbonapps, и он не будет развернут.

НАСТРОЙКА ПОЛЬЗОВАТЕЛЕЙ

В этой документации объясняется, как вы можете добавлять пользователей и управлять ими из консоли управления.

Чтобы пользователи могли входить в консоль управления, вы создаете учетные записи пользователей и назначаете им роли, которые представляют собой наборы разрешений. Вы можете добавлять отдельных пользователей или импортировать пользователей массово.

Добавление нового пользователя и назначение ролей

Добавьте свойство GetRoleListOfInternalUserSQL в раздел <Realm> файла /conf/user-mgt.xml, как показано ниже, чтобы избежать проблем с учетом регистра при создании пользователей.

```
<Realm>
<Configuration>
<Property name="GetRoleListOfInternalUserSQL">
    SELECT UM_ROLE_NAME
        FROM UM_HYBRID_USER_ROLE, UM_HYBRID_ROLE
    WHERE UPPER(UM_USER_NAME)=UPPER ( ? )
        AND UM_HYBRID_USER_ROLE.UM_ROLE_ID=UM_HYBRID_ROLE.UM_ID
        AND UM_HYBRID_USER_ROLE.UM_TENANT_ID=?
        AND UM_HYBRID_ROLE.UM_TENANT_ID=?
        AND UM_HYBRID_USER_ROLE.UM_DOMAIN_ID =
            (SELECT UM_DOMAIN_ID
                FROM UM_DOMAIN
                WHERE UM_TENANT_ID=? AND UM_DOMAIN_NAME=?)
</Property>
</Configuration>
</Realm>
```

Следуйте приведенным ниже инструкциям, чтобы добавить новую учетную запись пользователя и настроить ее роль.

На вкладке **«Настройки»** консоли управления нажмите **«Добавить»** в разделе **«Пользователи и роли»** щелкните **«Добавить нового пользователя»**. Откроется следующий экран:

Домашняя страница > Добавить пользователя

Добавить нового пользователя

Шаг 1: Введите имя пользователя и пароль

Введите имя пользователя

Домен	PRIMARY ▾
Имя пользователя*	<input type="text"/>
Пароль*	<input type="password"/>
Подтвердите пароль*	<input type="password"/>

Следующий > Завершить Отменить

Рис.79. Добавление пользователя

Сделайте следующее:

- В списке **Домен** укажите хранилище пользователей, в котором вы хотите создать эту учетную запись пользователя. Этот список включает первичное хранилище пользователей и любые другие вторичные хранилища пользователей, настроенные для вашего профиля EI.
- Введите уникальное имя пользователя и пароль, которые человек будет использовать для входа в систему. По умолчанию пароль должен состоять не менее чем из пяти символов и должен содержать как минимум один символ с заглавной буквой, символы, цифры и специальные символы.
- Нажмите **«Следующий»**. При необходимости выберите роли, которые должны быть у этого пользователя. Если в вашей системе много ролей, вы можете искать их по имени.
- Нажмите **«Завершить»**.

Создается новая учетная запись пользователя с указанными ролями, и она отображается на странице **«Пользователи»**.

Импорт пользователей

Помимо создания пользователей вручную, информацию о пользователях, хранящуюся в файле CSV или Excel, можно массово импортировать в хранилище пользователей, настроенное в вашем продукте WSO2. Эта возможность доступна только в том случае, если для вашего профиля системы настроено хранилище пользователей JDBC.

Прежде чем использовать эту функцию, обратите внимание на следующее:

- Параметр массового импорта пользователей включен в вашем профиле системы по умолчанию. Если нет, вы можете включить его, добавив следующее свойство в хранилище

пользователей JDBC, настроенное в файле user-mgt.xml (хранящемся в каталоге /conf).

```
<Property name="IsBulkImportSupported">true</Property>
```

- Рекомендуется загружать не более 500 000 пользователей одновременно. Если вам нужно загрузить больше пользователей, вы можете загрузить их отдельными пакетами по 500 000 человек.
- Вы также можете указать размер файла, который вы можете загрузить в профиль EI, в файле /conf/carbon.xml с помощью элемента TotalFileSizeLimit, как показано ниже. Это значение указано в МБ.

```
<TotalFileSizeLimit>100</TotalFileSizeLimit>
```

Создание файла с пользователями

Сначала необходимо создать файл CSV или файл Excel с информацией о пользователе. Можно импортировать имя пользователя и пароль непосредственно из CSV/Excel в профиль системы. Вы также можете назначить каждому пользователю несколько ролей. Вот пример CSV-файла:

```
UserName,password,role  
user1,password123,role=admin:developer  
user2,password123,role=admin:tester  
user3,password123,role=admin:developer:tester  
user4,password123,role=devops  
user5,password123,role=devops:tester
```

Убедитесь, что у вас есть роли, которые вы назначаете пользователям, доступным в системе. В противном случае сервер выдаст ошибку.

Помимо импорта пользователей с их паролями и ролями, вы можете импортировать другие атрибуты пользователей, такие как **адрес электронной почты, полное имя, фамилия, мобильный телефон, имя** и т. д., используя требуемые URL-адреса, определенные для атрибутов. Вот пример требуемых URL-адресов, которые вы можете определить для своего профиля системы:

```
http://wso2.org/claims/givenname  
http://wso2.org/claims/lastname  
http://wso2.org/claims/mobile  
http://wso2.org/claims/role
```

Чтобы импортировать пользователей с именем пользователя, паролем, ролями и другими атрибутами (в виде требуемых URL-адресов), создайте файл CSV, как показано в примере ниже:

UserName,password,Claims

user1, password123,http://wso2.org/claims/givenname=myname1,http://wso2.org/claims/lastname=mylastname1,http://wso2.org/claims/mobile=077777777,http://wso2.org/claims/role=admin:developer

user2, password123,http://wso2.org/claims/givenname= myname2,http://wso2.org/claims/lastname=mylastname2,http://wso2.org/claims/mobile=077777777,http://wso2.org/claims/role=admin:devops

user3, password123,http://wso2.org/claims/givenname= myname3,http://wso2.org/claims/lastname=mylastname3,http://wso2.org/claims/mobile=077777777,http://wso2.org/claims/role=developer:devops

Импорт пользователей из файла CSV/Excel

Чтобы импортировать пользователей массово:

- Войдите в консоль управления вашего продукта WSO2.
- В меню «Настройки» в разделе «Пользователи и роли» нажмите «Добавить».

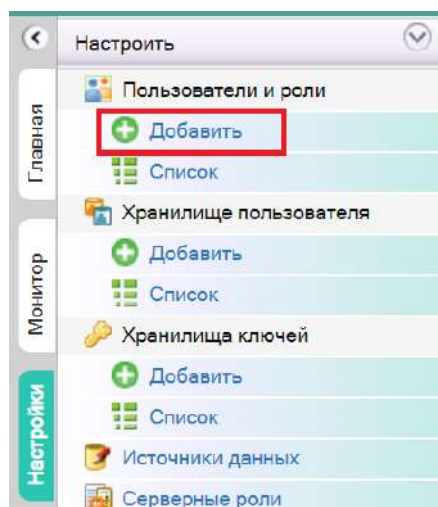


Рис. 80. Добавление пользователей и ролей

- Щелкните «**Пользователи массового импорта**».

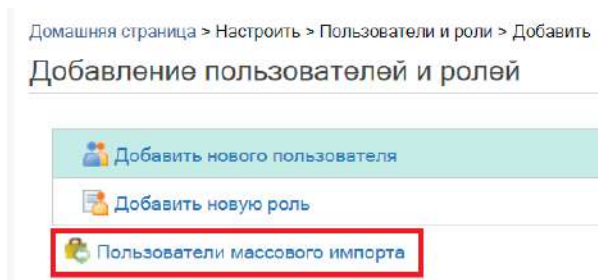


Рис. 81. Пользователи массового импорта

- Пользовательские хранилища, настроенные для вашего профиля EI, будут перечислены в поле Домен. Выберите хранилище пользователей, в которое вы хотите импортировать

пользователей из списка, загрузите таблицу CSV или Excel и нажмите «Завершить».

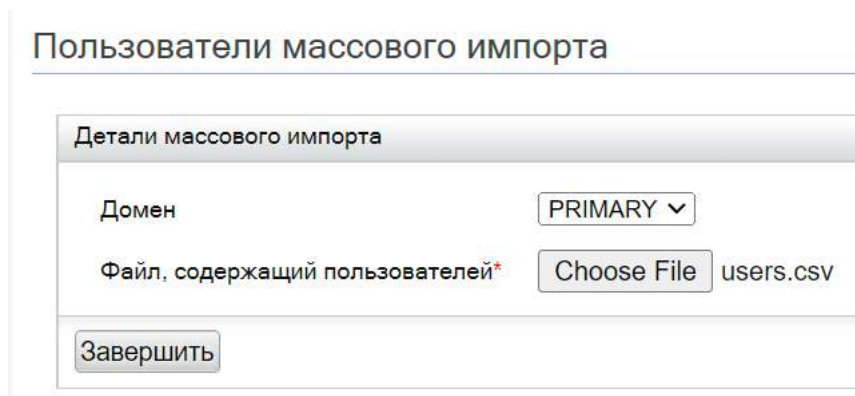


Рис. 82. Массовый импорт — параметры

Пароль по умолчанию для импортированных пользователей действителен только в течение 24 часов. Как системный администратор, вы можете решить проблемы с просроченными паролями, войдя в систему как администратор и изменив пароль пользователя, на странице «Пользователи» в меню «**Настройки**» -> «**Пользователи и роли**» -> «**Пользователи**» -> «**Изменить пароль**». По умолчанию пользователям будет назначена роль "Internal/everyone".

Настройка ролей и разрешений пользователя

Каждая роль определяет набор разрешений, которые пользователь будет иметь при назначении этой роли. После создания пользователя вы можете назначать и удалять роли для этого пользователя, щелкнув «**Назначайте роли**» в столбце «**Действия**». Чтобы увидеть, каким пользователям уже назначена роль, щелкните «**Просмотр пользователей**» рядом с ролью.

Вы также можете настроить, какие разрешения применяются к этому пользователю, щелкнув «**Просмотр ролей**» в столбце «**Действия**» на экране «**Пользователи**», а затем выбрав «**Разрешения**» для каждой роли, которые должны быть у этого пользователя.

Настройка профиля пользователя

У каждого отдельного пользователя есть профиль, который можно обновлять, добавляя различные сведения. Для этого нажмите «**Профиль пользователя**» на экране «**Пользователи**». Внесите необходимые изменения и нажмите «**Обновление**». Вы также можете добавить несколько профилей для пользователя.

Примечание. Вы можете добавлять новые профили, только если вы подключены к хранилищу пользователей JDBC. Вы также должны иметь права администратора.

Чтобы добавить новые профили, выполните следующие действия.

- На вкладке «**Настройки**» в консоли управления нажмите «**Список**» в разделе

«Пользователи и роли».

- Щелкните **«Пользователи»**. Эта ссылка видна только пользователям с ролью администратора.
- Щелкните ссылку **«Профиль пользователя»**.
- Вы можете добавить несколько профилей с помощью ссылки **«Добавить новый профиль»** и создать любое количество профилей для своего пользователя, если пользователь находится в пользовательском хранилище JDBC.

Удаление существующего пользователя

Следуйте приведенным ниже инструкциям, чтобы удалить пользователя.

Удаление пользователя нельзя отменить.

- На вкладке **«Настройки»** в консоли управления нажмите **«Список»** в разделе **«Пользователи и роли»**.
- Щелкните **«Пользователи»**. Эта ссылка видна только пользователям с разрешениями на уровне управления пользователями.
- В списке пользователей щелкните **«Удалить»** рядом с пользователем, которого хотите удалить, а затем щелкните **«Да»**, чтобы подтвердить операцию.

После удаления пользователя вы можете удалить все ссылки на личность удаленного пользователя с помощью инструмента анонимизации личных данных.

НАСТРОЙКА РОЛЕЙ

Данный раздел описывает, как добавлять роли пользователей и управлять ими из консоли управления. Роли содержат разрешения для пользователей на управление сервером. Вы можете создавать разные роли с различными сочетаниями разрешений и назначать их пользователю или группе пользователей. Через консоль управления вы также можете редактировать и удалять существующие роли пользователей.

Polus ESB поддерживает модель аутентификации на основе ролей, в которой привилегии пользователя основаны на выданной ему роли. По умолчанию поддерживаются следующие роли:

- Администратор** — Предоставляет полный доступ ко всем функциям и элементам управления. По умолчанию пользователю с правами администратора назначаются роли **Администратора** и **Все**.
- Все** — Каждый новый пользователь назначается этой роли по умолчанию. По умолчанию она не содержит никаких разрешений.
- Система** — Эта роль не отображается в консоли управления.

Если пользователю назначено несколько ролей, их разрешения складываются.

Добавление роли пользователя

Следуйте приведенным ниже инструкциям, чтобы добавить роль пользователя.

1. На вкладке **Главное** в консоли управления щёлкните **Добавить** в разделе **Пользователи и Роли**.
2. Щёлкните **Роли**. Эта ссылка видна только пользователям с ролью привилегий уровня **Безопасности**. По умолчанию это разрешение включено для пользователя-администратора. Дополнительные сведения о разрешениях смотрите в разделе **Разрешения на основе Ролей**.
3. Щёлкните **Добавить Новую Роль**. Появится следующий экран:

Домашняя страница > Добавить новую роль ? Помощь

Добавить новую роль

Шаг 1: Введите сведения о роли

Введите сведения о роли

Домен PRIMARY ▾

Имя роли*

Рис. 83. Добавление новой роли

4. Выполните следующее:

- В списке **Домен** укажите хранилище пользователей, в котором вы хотите создать эту роль. Этот список включает первичное хранилище пользователей и любые другие вторичные хранилища пользователей, настроенные для вашего профиля ЕІ. Информацию о том, какие хранилища пользователей (то есть репозитории, в которых хранится информация о пользователях и ролях) установлены и настроены, смотрите в разделе **Настройка Хранилищ Пользователей**.
- Введите уникальное имя для этой роли.
- Щёлкните **Далее**.

5. Выберите разрешения, которые должны быть у пользователей с этой ролью. Обратите внимание, что когда вы назначаете выбранную роль пользователю, вы можете переопределить разрешения ролей и настроить их для пользователя.

6. Выберите существующих пользователей, которым должна быть назначена эта роль. Вы также можете назначить эту роль пользователям позже, но если вы создаете эту роль во внешнем хранилище пользователей, которое не допускает пустых ролей, вы должны назначить ее хотя бы одному пользователю. Вы можете найти пользователя по имени или просмотреть всех пользователей, введя * в поле поиска.

7. Щёлкните **Готово**.

Роль создана и указана на странице **Роли**. Теперь вы можете редактировать роль по мере необходимости.

Редактирование или удаление роли

Если вам нужно внести изменения в роль, выберите домен (хранилище пользователей), в котором находится роль, а затем используйте соответствующие ссылки в столбце **Действия** на экране **Роли**:

- Переименовать роль;
- Изменить разрешения по умолчанию, связанные с этой ролью;
- Назначить выбранную роль пользователям;
- Просмотреть пользователей, которым назначена эта роль;
- Удалить роль, если она больше не нужна.

Если роль находится во внешнем хранилище пользователей, к которому вы подключены в режиме только для чтения, вы сможете просматривать существующие роли, но не можете редактировать или удалять их. Однако вы по-прежнему можете создавать новые редактируемые роли.

Обновление имен ролей

Если вам нужно внести изменения в имена ролей, вам необходимо выполнить одно из следующих действий:

Обновление перед первым запуском (рекомендуется)

Имена ролей по умолчанию (администратор и все) можно изменить перед запуском системы, отредактировав файл /conf/user-mgt.xml.

```
<Configuration>
  <AdminRole>admin</AdminRole>
  <AdminUser>
    <UserName>admin</UserName>
    <Password>admin</Password>
  </AdminUser>
  <EveryoneRoleName>everyone</EveryoneRoleName>
  <!-- By default users in this role sees the registry root -->
  <Property name="dataSource">jdbc/WSO2CarbonDB</Property>
  <Property name="MultiTenantRealmConfigBuilder">
    org.wso2.carbon.user.core.config.multitenancy.SimpleRealmConfigBuilder
  </Property>
</Configuration>
```

Ниже приведены изменения, которые необходимо внести в вышеуказанные конфигурации:

- Замените `<AdminRole>admin</AdminRole>` на `<AdminRole>администратор</AdminRole>`.
- Замените `<EveryoneRoleName>everyone</EveryoneRoleName>` на `<EveryoneRoleName>Ваша роль</EveryoneRoleName>`.

Обновление после использования профиля EI в течение некоторого времени

Вам не обязательно делать это при обновлении перед первым запуском. Следующие шаги помогут вам обновить имена ролей:

1. Внесите изменения в конфигурацию, указанные в предыдущем разделе.
2. Вам необходимо внести следующие изменения на уровне хранилища пользователей для существующих пользователей, если вы изменили имена ролей, как упоминалось ранее.

Если вы подключены к `JDBCUserStoreManager`, вам необходимо обновить таблицу `UM_USER_ROLE` с существующими пользователями после изменения имен ролей администратора и всех. Кроме того, если вы изменили разрешение для роли все, `UM_ROLE_PERMISSION` необходимо обновить разрешениями для новой роли.

Схему можно найти, обратившись к источнику данных, указанному в файле `user-mgt.xml`. Определение источника данных можно найти в `/conf/datasources/master-datasources.xml`.

Если вы подключены к `ReadWriteLdapUserStoreManager`, вам необходимо заполнить членов предыдущей роли администратора новой ролью в Группках. Дополнительные сведения смотрите в разделе `Настройка Пользовательских Хранилищ`.

3. После внесения изменений перезапустите сервер.

ЗАПЛАНИРОВАННЫЕ ЗАДАЧИ

Задача — это специально разработанный класс Java, который реализует интерфейс **org.apache.synapse.startup.Task**. Этот интерфейс определяет единственный метод с именем **execute**. Задачи могут быть запланированы для периодического выполнения на Polus ESB. При необходимости задачу можно настроить на запуск ограниченное количество раз.

Прежде чем задачу можно будет запланировать, ее необходимо скомпилировать и упаковать в файл jar. Затем этот файл jar следует поместить в каталог «repository/components/lib» Polus ESB. Затем необходимо перезапустить Polus ESB, чтобы jar, содержащий реализацию задачи, был подобран средой выполнения сервера. Пакет OSGi, содержащий реализацию задачи, будет создан автоматически и развернут на сервере.

Развернув реализацию задачи в среде выполнения Polus ESB, можно использовать консоль управления для планирования различных экземпляров задачи. Можно запланировать задачу, указав количество раз, которое она должна выполняться вместе с частотой, или вы можете использовать Крон-синтаксис.

Планирование задачи

1. В навигаторе в закладке «Главная» в разделе «Управление» -> «Служебная шина» щелкните «Запланированные задачи». Появится страница запланированных задач.

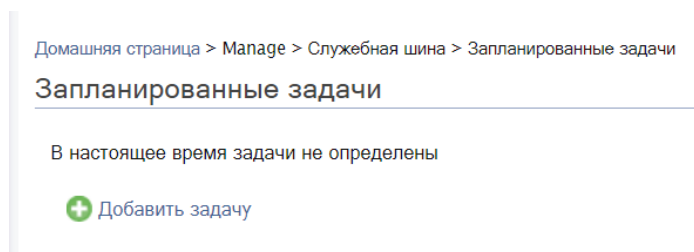


Рис.84. Панель запланированных задач

2. Щелкните «Добавить задачу».
3. Введите необходимые данные и нажмите «Запланировать».

Домашняя страница

Новая запланированная задача

Новая запланированная задача

Название задачи*

Группа задач*

Реализация задачи* [Загрузить свойства задачи](#)

Триггерная информация о задаче

Тип триггера Простой Крон

Количество запусков

Интервал (в секундах)*

Различная информация

Закрепленные серверы

(разделенные через запятую или пробел)

Рис.85. Панель настройки новой запланированной задачи

Таблица 19. Описание полей запланированной задачи

Имя поля	Описание
Название задачи	Название запланированной задачи
Группа задач	Имя группы для группировки задач. Имя группы «synapse.simple.quartz» объединяет задачи, принадлежащие ESB — Synapse. Все доступные группы отображаются в виде выпадающего меню. Если есть задачи, принадлежащие другим доменам, то они будут показаны здесь как отдельные имена групп.
Реализация задачи	Класс реализации задачи. Существует реализация задачи по умолчанию, которая поставляется вместе с ESB. Это <code>org.apache.synapse.startup.tasks.MessageInjector</code> , который просто вводит указанное сообщение в среду Synapse. Образец ESB <code>synapse_sample_300</code> демонстрирует это.
Тип триггера	Тип триггера для задачи. Это может быть «Простой» или «Крон». Простой триггер определяется путем указания «Количества запусков» и «Интервала», подразумевая, что задача будет выполняться «Количество запусков» раз с указанными интервалами. Триггер «Крон» определяется с помощью Крон-выражения.
Количество запусков	Сколько раз будет выполняться задание.
Интервал	Интервал между последовательными выполнениями задачи
Закрепленные серверы	Предоставьте список имен серверов ESB, на которых эта задача должна быть запущена для значения закрепленного сервера.

Пример

В этом примере показано, как указать класс реализации задачи и указать свойства, необходимые для этой реализации задачи.

Класс реализации задачи, используемый в этом примере, называется «**org.apache.synapse.startup.tasks.MessageInjector**». Этот класс доступен в стандартном дистрибутиве Polus ESB и, следовательно, может использоваться сразу после установки без развертывания каких-либо сторонних библиотек или пользовательских jar-файлов. На рис. 3 показана страница в консоли управления после нажатия кнопки «Загрузить класс». Он показывает свойства экземпляра реализации задачи **MessageInjector**. (Дополнительную информацию см. в примере ESB synapse_sample_300)

Домашняя страница Help

Изменить задачу : MyTask

Изменить задачу

Название задачи*

Группа задач*

Реализация задачи* [Перезагрузить свойства задачи](#)

Название свойства	Тип свойства	Значение свойства	Действие
<input type="text" value="to"/>	Литерал	<input type="text" value="http://localhost:9000/services/SimpleStock"/>	Удалить
<input type="text" value="message"/>	XML	<input type="text" value="<m0:getQuote xmlns:m0='http://services.samples'"/>	Удалить
<input type="text" value="format"/>	Литерал	<input type="text" value="soap11"/>	Удалить
<input type="text" value="soapAction"/>	Литерал	<input type="text" value="urn:getQuote"/>	Удалить

Триггерная информация о задаче

Тип триггера Простой Крон

Количество запусков

Интервал (в секундах)*

Различная информация

Закрепленные серверы (разделенные через запятую или пробел)

Рис.86. Пример настройки запланированной задачи

ШАБЛОНЫ

В Polus ESB предусмотрен механизм шаблонов последовательностей и шаблонов точек вызова. Как показано на рисунках 87 и 88, в разделе управления шаблонами отображается список существующих шаблонов. На вкладках шаблонов реестра отображаются шаблоны, сохраненные в реестре, на остальных - шаблоны, сохраненные в конфигурации Synapse.

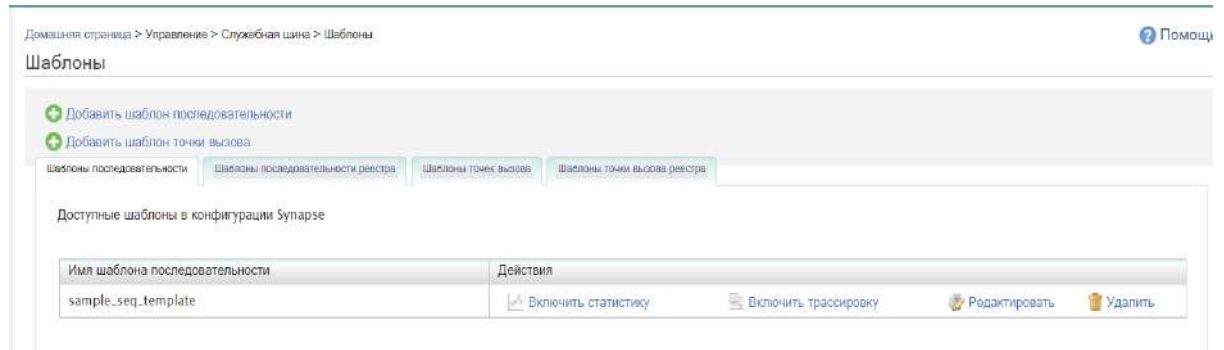


Рис.87. Управление шаблоном последовательности

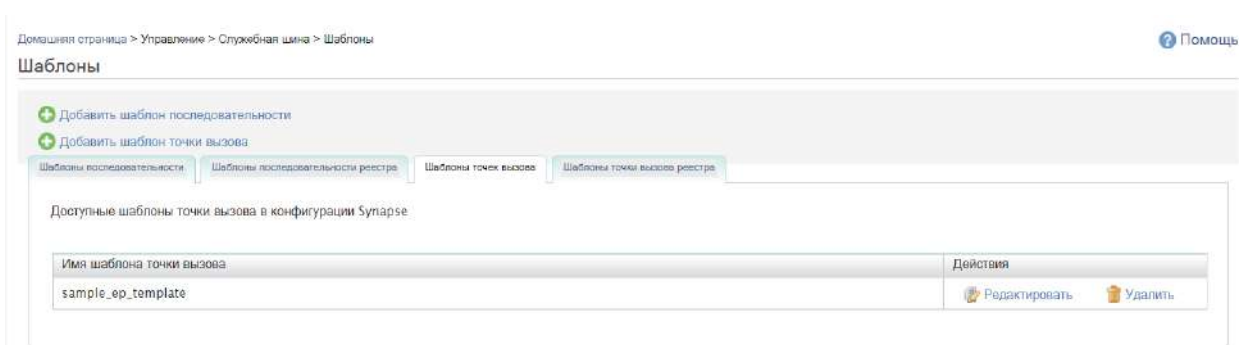


Рис.88. Управление шаблоном точки вызова

В навигаторе в разделе «Управление/Службная шина» нажмите «**Шаблоны**». Появится страница «**Шаблоны**».

Управление шаблонами последовательностей

- Нажмите «Включить статистику», чтобы собрать информацию, относящуюся к конкретному шаблону последовательности.
- Нажмите «Включить трассировку», чтобы включить все сообщения трассировки для соответствующего шаблона последовательности.

Вышеупомянутые параметры отключены по умолчанию, поскольку они вызывают серьезное снижение производительности. Используйте эти параметры только в ситуации, когда вам нужно отладить конкретную проблему. Динамические последовательности не поддерживают «Включить статистику» и «Включить трассировку».

- Нажмите **«Редактировать»**, чтобы открыть шаблон последовательности в проектном представлении для редактирования.
- Нажмите **«Удалить»**, чтобы удалить конкретный шаблон последовательности из системы.

Добавление шаблона последовательности

1. На странице **«Шаблоны»** нажмите **«Добавить шаблон последовательности»**. Появится страница **«Проектирование шаблона»**.
2. Введите имя шаблона последовательности.
3. Щелкните **«Добавить дочерний элемент»** и выберите необходимые узлы для шаблона последовательности.

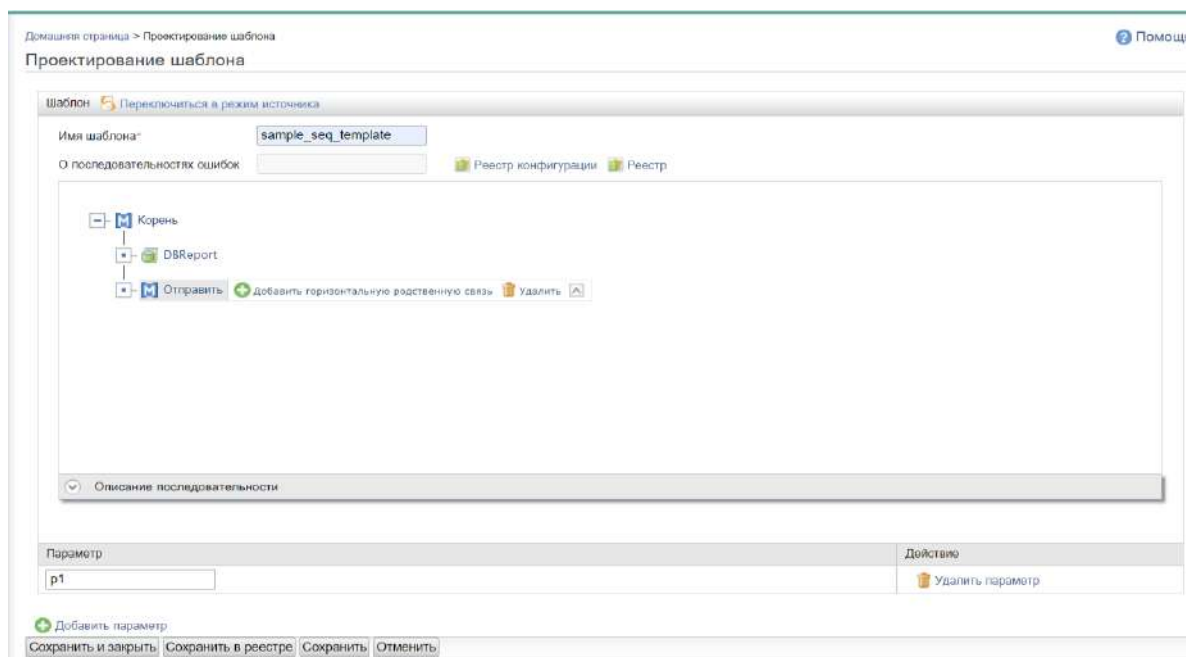


Рис.89. Добавление шаблона последовательности — режим конструктора

Добавление параметров в шаблон последовательности

Если вы хотите добавить параметры в шаблоны последовательности, нажмите кнопку **«Добавить параметр»**. Затем появится страница с таблицей параметров. Вы можете добавлять и удалять имена параметров, а также изменять и добавлять свойства узлов

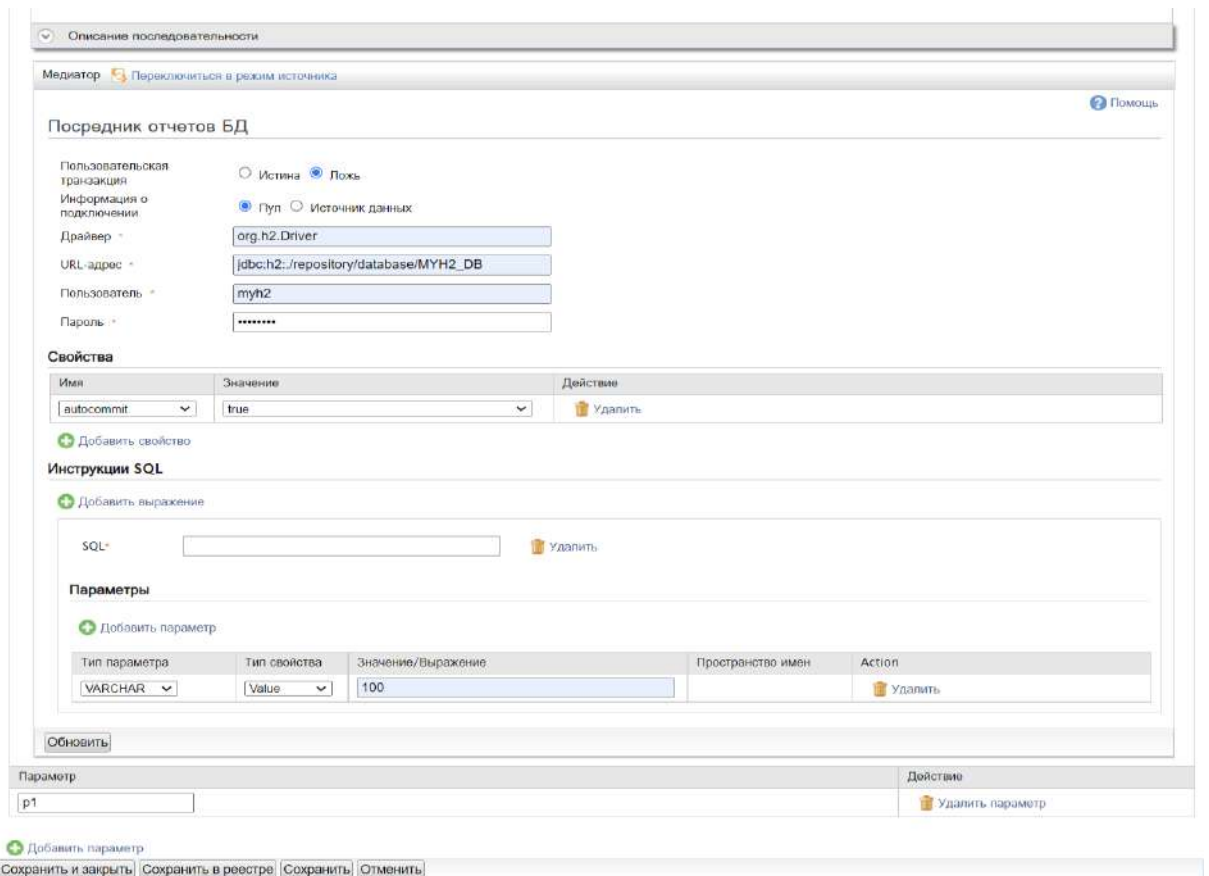


Рис.90. Параметры шаблона последовательности

Нажмите «Переключиться на исходное представление», чтобы просмотреть XML-файл конкретного шаблона последовательности. Если вы знакомы с языком конфигурации Synapse, вы можете напрямую редактировать XML и сохранять конфигурацию, используя это представление.

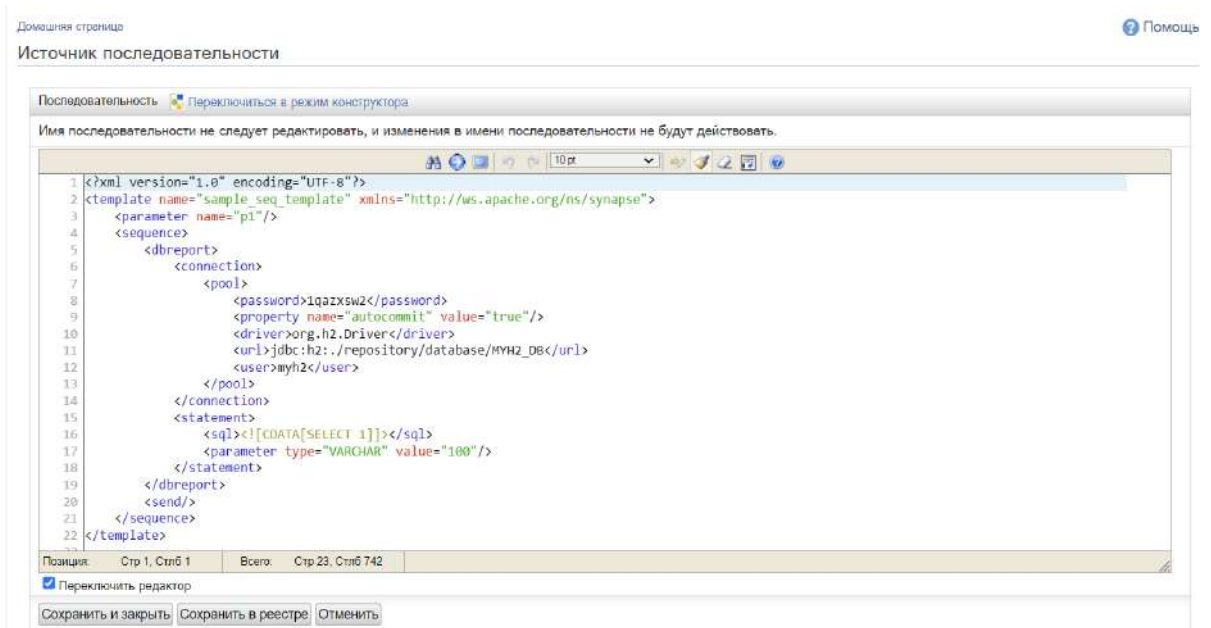


Рис.91. Представление исходного кода шаблона последовательности

Управление шаблонами точек вызова

- Нажмите **«Редактировать»**, чтобы открыть шаблон точки вызова в проектном представлении для редактирования.
- Щелкните **«Удалить»**, чтобы удалить конкретный шаблон точки вызова из системы.
- Обратите внимание, что параметры статистики/отслеживания недоступны для шаблонов точек вызова.

Добавление шаблона точки вызова

1. На странице **«Шаблоны»** щелкните **«Добавить шаблон точки вызова»**. Появится страница добавления шаблонов точек вызова.
2. Выберите тип шаблона точки вызова. В настоящее время поддерживаются четыре типа:
 - Шаблон точки вызова по URL
 - Шаблон точки вызова по умолчанию
 - Шаблон точки вызова HTTP
 - Шаблон точки вызова WSDL

[Домашняя страница](#) > [Добавление шаблонов точек вызова](#)

Добавление шаблонов точек вызова

Выберите тип точки вызова для добавления	
<input type="checkbox"/> Шаблон точки вызова адреса по URL	Определяет прямой URL сервиса
<input type="checkbox"/> Шаблон точки вызова по умолчанию	Определяет дополнительную конфигурацию для целевого объекта по умолчанию
<input type="checkbox"/> Шаблон точки вызова HTTP	Определяет точку вызова REST сервиса на основе шаблона URI.
<input type="checkbox"/> Шаблон точки вызова WSDL	Определяет WSDL, сервис и порт

Рис.92. Типы шаблонов точек вызова

[Домашняя страница](#) > [Точка вызова по URL](#)

Шаблон точки вызова адреса по URL

Точка вызова по URL [Перейти к исходному виду](#)

Название шаблона *

Название *

Адрес *

Параметр	Действие
<input type="text" value="p_id"/>	<input type="button" value="Удалить параметр"/>
<input type="text" value="p_name"/>	<input type="button" value="Удалить параметр"/>

Добавить параметр

Показать дополнительные параметры

Свойства точки вызова

Добавить свойства

Рис.93. Шаблон добавления точки вызова по URL – режим конструктора

Введите необходимые данные в поля. Как показано выше, поля относятся к **«Шаблону»**

точки вызова по URL».

- Название шаблона - Уникальное имя шаблона точки вызова
- Название - имя встроенной точки вызова
- Адрес - URL точки вызова шаблона. Это может быть параметризованное значение, такое как \$uri. Если это реальная точка вызова, вы можете проверить доступность данного URL-адреса на лету, просто щелкнув «Тест».

Добавление параметров в шаблон точки вызова

Вы можете добавить параметры в шаблоны точек вызова, нажмите кнопку «Добавить параметр», появится страница с таблицей параметров. Вы можете добавлять и удалять имена параметров. Доступ к этим именам параметров можно получить в различных атрибутах редактируемого в данный момент шаблона точки вызова. Доступ к параметрам осуществляется с префиксом \$. Для шаблонов точки вызова есть два встроенных параметра «name» и «uri», к которым можно получить доступ с помощью \$name и \$uri. В примере показано, как получить доступ к параметрам *uri*, *p_id* и *p_name* для атрибутов «Шаблон точки вызова по URL», «Коды ошибок приостановки» и «Начальная продолжительность приостановки» соответственно.

The screenshot displays a configuration page for a call point template. It is organized into several sections:

- Параметр (Parameter):** Contains two input fields for 'p_id' and 'p_name'. Below them is a green '+ Добавить параметр' (Add parameter) button and a blue 'Скрыть дополнительные параметры' (Hide additional parameters) link.
- Содержание сообщения (Message content):** Includes dropdown menus for 'Форматировать' (Formatted) and 'Оптимизировать' (Optimized), both set to 'Оставить как есть' (Keep as is).
- Состояние приостановки точки вызова (Call point suspension state):** Features input fields for 'Коды ошибок' (Error codes), 'Начальная продолжительность (миллисекунды)' (Initial duration in milliseconds), 'Максимальная продолжительность (миллисекунды)' (Maximum duration in milliseconds), and 'Фактор' (Factor) set to 1.0. A 'Выберите коды ошибок' (Select error codes) button is also present.
- Состояние тайм-аута точки вызова (Call point timeout state):** Includes input fields for 'Коды ошибок' (Error codes), 'Повторные попытки перед приостановкой' (Retries before suspension) set to 0, and 'Задержка повторной попытки (миллисекунды)' (Retry delay in milliseconds) set to 0. A 'Выберите коды ошибок' (Select error codes) button is also present.
- Время ожидания соединения (Connection wait time):** Includes a dropdown for 'Действие' (Action) set to 'Не устанавливать тайм-аут' (Do not set timeout) and an input field for 'Максимальное время ожидания (миллисекунды)' (Maximum wait time in milliseconds).
- QoS (QoS):** Contains three checkboxes: 'WS-Адресация' (WS Addressing), 'WS-Безопасность' (WS Security), and 'WS-RM' (WS-RM), all currently unchecked.
- Описание точки вызова (Call point description):** A section at the bottom, partially visible.

Рис.94. Добавленные параметры и дополнительные атрибуты шаблона точки вызова

Щелкните «Переключится на исходный вид», чтобы просмотреть XML-файл

конкретного шаблона точки вызова. Как указывалось ранее, если вы знакомы с языком конфигурации Synapse, вы можете напрямую редактировать XML и сохранять конфигурацию, используя это представление.

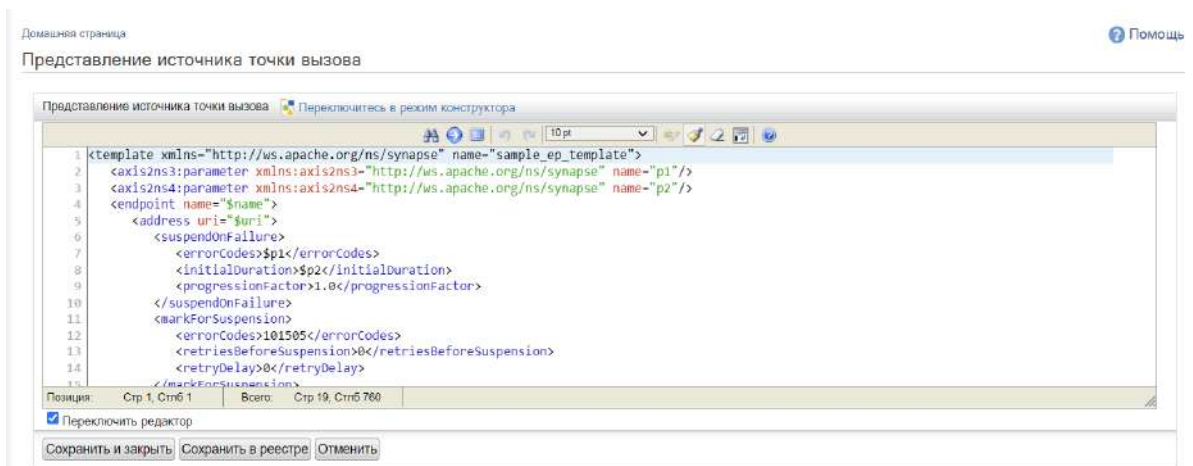


Рис.95. Представление исходного кода шаблона точки вызова

УПРАВЛЕНИЕ ХРАНИЛИЩАМИ СООБЩЕНИЙ

Хранилища сообщений используются для хранения промежуточного состояния сообщений в процессе их обработки. Для реализации данной возможности Polus ESB содержит соответствующий встроенный Медиатор хранилища.

1. В навигаторе в разделе Управление/Службная Шина щёлкните **Хранилища Сообщений**. Появится страница *Управление Хранилищами Сообщений*.

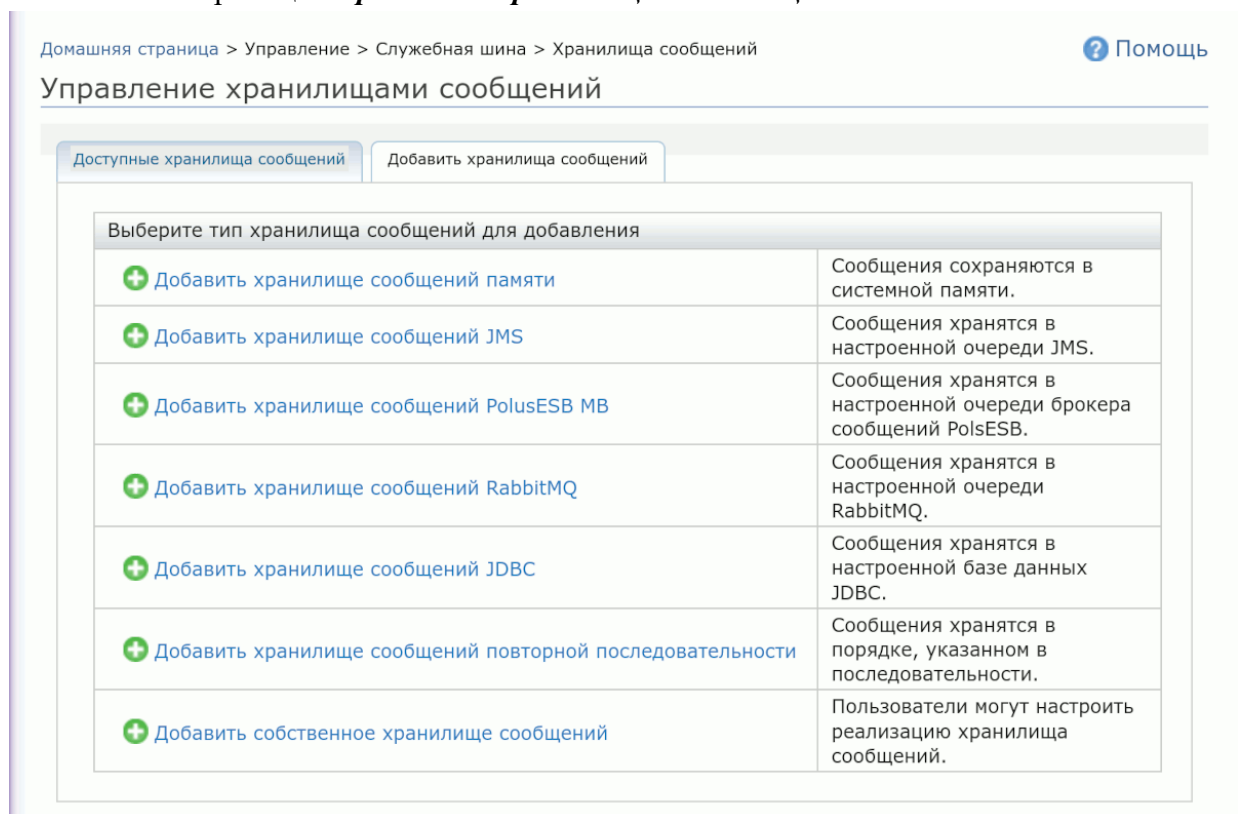


Рис.96. Типы хранилищ сообщений

Добавление хранилища сообщений в памяти

1. Щёлкните ссылку **Добавить хранилище сообщений памяти** - появится соответствующая страница.

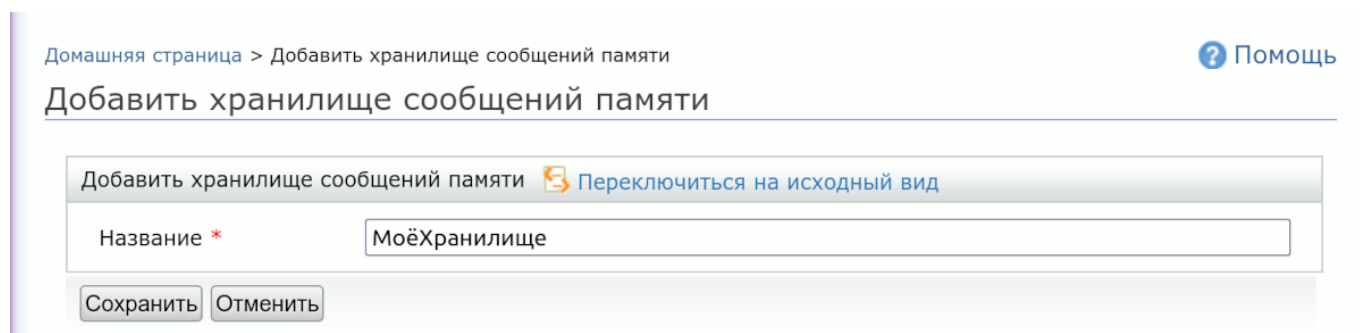


Рис.97. Добавление Хранилища Сообщений в Памяти

2. Введите необходимые данные и щёлкните **Сохранить**. На странице **Управление хранилищами сообщений** отобразится добавленное хранилище сообщений.

Домашняя страница > Message Stores ? Помощь

Управление хранилищами сообщений

Доступные хранилища сообщений Добавить хранилища сообщений

Имя хранилища сообщений	Тип	Сообщения	Действия
test	org.apache.synapse.message.store.impl.memory.InMemoryStore	0	Редактировать Удалить
МоёХранилище	org.apache.synapse.message.store.impl.memory.InMemoryStore	0	Редактировать Удалить

Рис.98. Список хранилищ сообщений

Добавление хранилища сообщений JMS

1. Щелкните ссылку **Добавить хранилище сообщений JMS** - появится соответствующая страница.

Домашняя страница > Добавить хранилище сообщений JMS ? Помощь

Добавить хранилище сообщений JMS

Добавить хранилище сообщений JMS Перекл. на исходный вид

Название *

Фабрика исходного контекста *

URL-адрес поставщика *

скрыть дополнительные параметры

Скрыть параметры гарантированной доставки

Дополнительные параметры

Имя очереди JNDI

Фабрика соединений

Имя пользователя

Пароль

Версия спецификации JMS API

Параметры конфигурации аварийного переключения

Включить гарантированную доставку производителя

Отказоустойчивое хранилище сообщений

Рис.99. Добавление хранилища сообщений JMS

2. Введите необходимые данные и щёлкните **Сохранить**. На странице **Управление хранилищами сообщений** отобразится добавленное хранилище сообщений.

Добавление пользовательского хранилища сообщений

1. Щёлкните ссылку **Добавить собственное хранилище сообщений** - появится соответствующая страница. Вы можете добавить параметры в хранилище сообщений, задав **Имя** и **Значение** для **параметров хранилища сообщений** и нажав кнопку **Добавить Параметр**.

Домашняя страница > Добавить собственное хранилище сообщений ? Помощь

Добавить собственное хранилище сообщений

Добавить собственное хранилище сообщений [Переключиться на исходный вид](#)

Название *

Класс провайдера *

Параметры хранилища сообщений

Название: Значение: [+ Добавить Параметр](#)

Название	Значение	Действие
Название	Значение	Удалить

Рис.100. Добавление пользовательского хранилища сообщений

2. Введите необходимые данные и щёлкните **Сохранить**. На странице **Управление хранилищами сообщений** отобразится добавленное хранилище сообщений.

Возможные действия в хранилище сообщений

1. Щёлчок по добавленному элементу списка хранилища сообщений направляет вас на страницу, где отображаются сообщения в списке. Вы можете **удалить** сообщение или **удалить все сообщения** в хранилище.

Домашняя страница > Message Store ? Помощь

Управление хранилищем сообщений

Имя хранилища сообщений	Тип	Сообщения
InMemoryStore	org.apache.synapse.message.store.impl.memory.InMemoryStore	0

Сообщения

Идентификатор сообщения	Действия
Сообщения не найдены	

Удалить все сообщения

Рис.101. Действия в хранилище сообщений

2. Если щелкнуть **показать содержимое** сообщения, отобразится содержимое сообщения.

The screenshot shows the 'Manage Message Store' interface. At the top, there is a breadcrumb navigation: 'Home > Manage > Service Bus > Message Stores > Message Store'. A 'Help' icon is in the top right corner. Below the breadcrumb is the title 'Manage Message Store'. A table lists message stores:

Message Store Name	Type	Messages
MyStore1	org.apache.synapse.message.store.InMemoryMessageStore	1

Below the table is the 'Message Content' section, which contains a text area with the following XML content:

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
3   <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
4     <wsa:To>http://localhost:9000/services/SimpleStockQuoteService</wsa:To>
5     <wsa:ReplyTo>
6       <wsa:Address>http://www.w3.org/2005/08/addressing/none</wsa:Address>
7     </wsa:ReplyTo>
8     <wsa:MessageID>urn:uuid:516c27f3-8bb8-413a-bf83-9a18af660bd4</wsa:MessageID>
9     <wsa:Action>urn:placeOrder</wsa:Action>
10   </soapenv:Header>
11   <soapenv:Body>
12     <m0:placeOrder xmlns:m0="http://services.samples">
13       <m0:order>
14         <m0:price>85.75609588839981</m0:price>
15         <m0:quantity>18034</m0:quantity>
16       </m0:order>
17     </m0:placeOrder>
18   </soapenv:Body>
19 </soapenv:Envelope>
```

At the bottom of the text area, there are 'Delete' and 'Cancel' buttons.

Рис.102. Содержимое сообщения

УПРАВЛЕНИЕ ПРОЦЕССОРАМИ СООБЩЕНИЙ

Процессоры сообщений предназначены для обработки сообщений Polus ESB, записанных в хранилища сообщений.

1. В навигаторе в разделе «Управление/Шина Сервисов» щелкните **Процессоры сообщений**. Появится страница **Управление процессорами сообщений**.

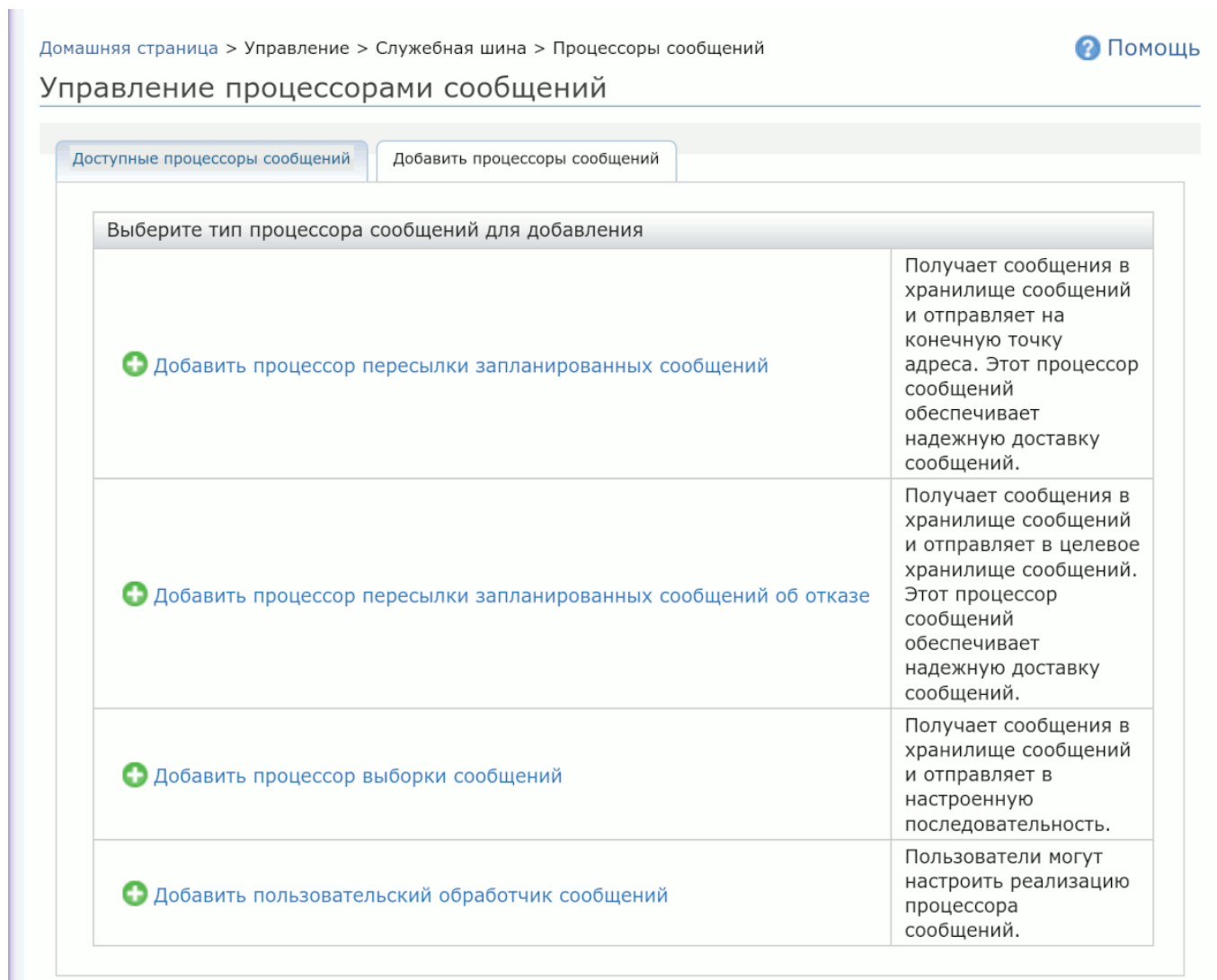


Рис.103. Типы процессоров сообщений

Добавление процессора пересылки запланированных сообщений

1. Щёлкните **Добавить Процессор Переадресации Запланированных Сообщений**, Появится страница **Добавить Процессор Пересылки Запланированных Сообщений**.

Добавить процессор пересылки запланированных сообщений

Добавить процессор пересылки запланированных сообщений [Переключиться на исходный вид](#)

Название *

Название точки вызова * [Реестр конфигурации](#)
[Реестр](#)

Хранилище сообщений *

[Скрыть дополнительные параметры](#)

Параметры процессора пересылки сообщений

Состояние процессора *

Интервал переадресации (миллис)

Интервал повтора (миллис)

Коды состояния HTTP без повторных попыток (например, - 304, 305)

Максимальное количество попыток повторной доставки

Удаление сообщения после максимального количества попыток доставки

Максимальное количество попыток подключения к хранилищу

Интервал попытки подключения к хранилищу (миллисекунды)

Репозиторий клиента Axis2

Конфигурация Axis2

Название последовательности ответов [Реестр конфигурации](#) [Реестр](#)

Название последовательности ошибок [Реестр конфигурации](#) [Реестр](#)

Деактивировать название последовательности [Реестр конфигурации](#) [Реестр](#)

Сбой хранения сообщений

Путь к конфигурационному файлу Quartz

Срок

Счетчик задач (кластерный режим)

Рис.104. Добавление процессора пересылки запланированных сообщений

- Введите необходимые данные и щёлкните **Сохранить**. На странице **Управление процессорами сообщений** отобразятся добавленные процессоры сообщений.

Управление процессорами сообщений

[Доступные процессоры сообщений](#) [Добавить процессоры сообщений](#)

Название обработчика сообщений	Тип	Действия
ПримерПроцессора	Scheduled Message Forwarding Processor	Редактировать Удалить Активный [Деактивировать]

Рис.105. Список процессоров сообщений

Добавление Процессора Выборки Сообщений

- Щелкните **Добавить процессор выборки сообщений**

Добавить процессор выборки сообщений

Добавить процессор выборки сообщений [Перекл. на исходный вид](#)

Название *

Последовательность *

Хранилище сообщений *

[^](#) Скрыть дополнительные параметры

Параметры процессора выборки сообщений

Состояние процессора *

Интервал выборки (миллис)

Параллелизм выборки

Максимальное количество попыток подключения к хранилищу

Интервал попытки подключения к хранилищу (миллисекунды)

Путь к конфигурационному файлу Quartz

Срон

Рис.106. Добавление процессора выборки сообщений

- Введите необходимые данные и щёлкните **Сохранить**. На странице **Управление Процессорами Сообщений** отобразятся добавленные Процессоры Сообщений.

Добавление Пользовательского Процессора Сообщений

- Щёлкните **Добавить Пользовательский Процессор Сообщений**, появится страница **Добавить Пользовательский Процессор Сообщений**.

Домашняя страница > Message Processor [? Помощь](#)

Добавить пользовательский обработчик сообщений

Добавить пользовательский обработчик сообщений [Перекл. на исходный вид](#)

Название *

Полное доменное имя класса поставщика процессора сообщений *

Хранилище сообщений *

Параметры процессора сообщений

Название: Значение: [+ Добавить параметр](#)

Имя	Значение	Действие
Название	Значение	Удалить

Рис.107. Добавление пользовательского процессора сообщений

- Введите необходимые данные и щёлкните **Сохранить**. На странице **Управление процессорами сообщений** отобразятся добавленные процессоры сообщений.

Действия с процессорами сообщений

1. Пользователи могут деактивировать определенные процессоры сообщений, перечисленные на странице **Управление процессорами сообщений**.

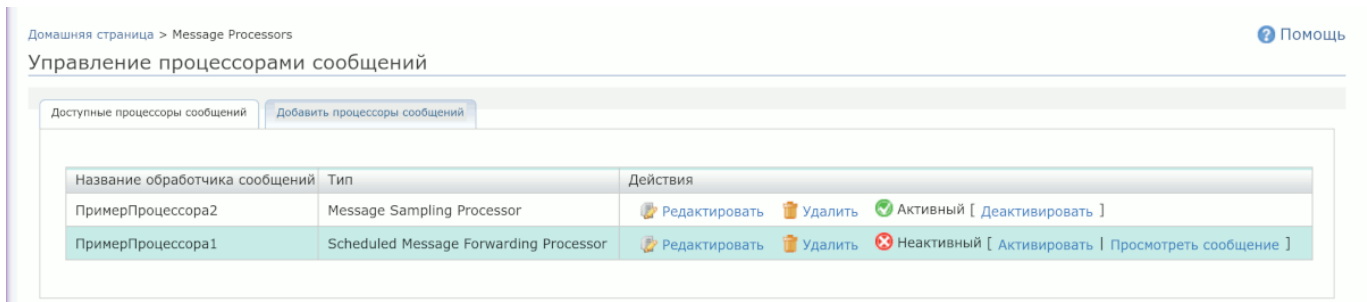


Рис.108: Деактивированные процессоры Сообщений

2. Процессор запланированных сообщений поставляется с пользовательским интерфейсом, чтобы предоставить пользователю некоторый ручной контроль над операциями пересылки сообщений. С помощью интерфейса пользователь может **удалить, повторно отправить, удалить все и повторно отправить все** сообщения в процессоре сообщений.

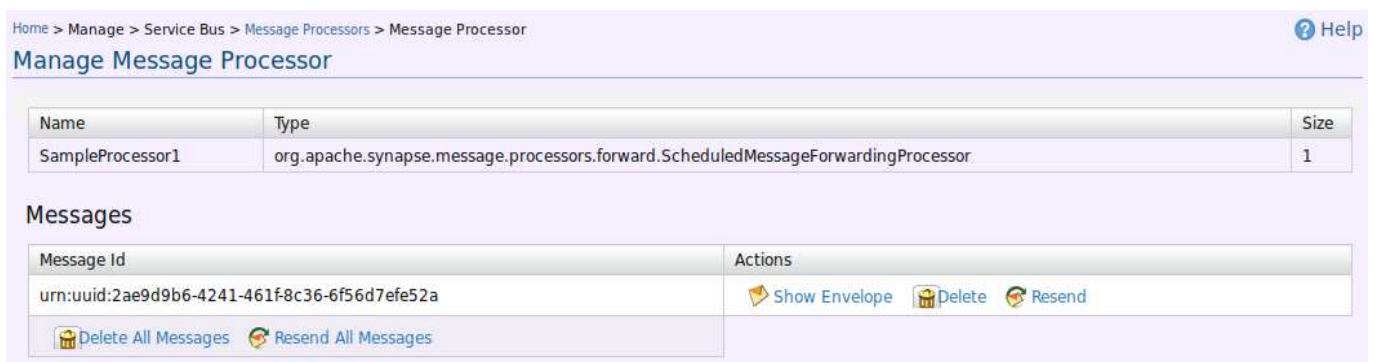


Рис.109. Действия над сообщениями

3. Если щелкнуть **Показать Содержимое** сообщения, отобразится содержимое сообщения.

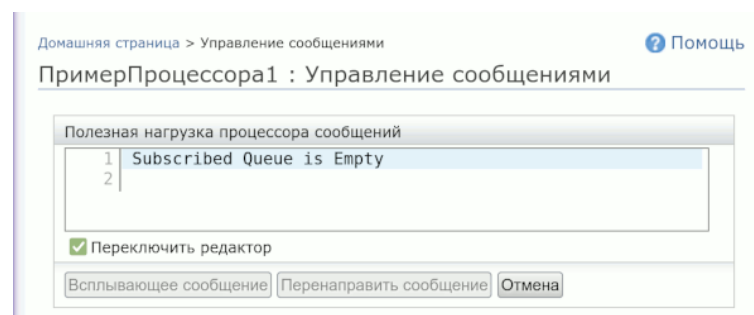


Рис. 110. Содержимое сообщения

КОННЕКТОРЫ

Коннектор позволяет взаимодействовать с функциями стороннего продукта и данными из вашего потока сообщений. Например, predefined коннекторы PolusESB позволяют подключаться и взаимодействовать с интерфейсами API таких сервисов, как Twilio, Salesforce и JIRA. Например, если вы включили коннекторы Twitter и Google Spreadsheet в своем экземпляре ESB, ваш поток сообщений может получать запросы, содержащие имя и пароль пользователя в Twitter, входить в учетную запись пользователя в Twitter, получать список подписчиков пользователя и вносить эту информацию в таблицу Google. Каждый коннектор предоставляет набор операций, которые вы вызываете из своих прокси-сервисов, последовательностей и REST API для взаимодействия с этим продуктом. Например, коннектор Twitter предоставляет операции, которые вызывают интерфейсы API Twitter для получения и отправки напрямую сообщений, получения идентификаторов друзей и подписчиков, обновления статуса, ретвита статуса других пользователей и так далее.

Добавление Коннекторов

Чтобы добавить коннектор к выбранному экземпляру ESB, щёлкните **Управление > Коннекторы > Добавить**.

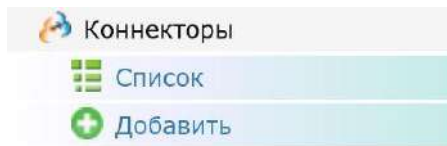


Рис.111. Меню - коннекторы

Затем вы можете загрузить файл ZIP коннектора.

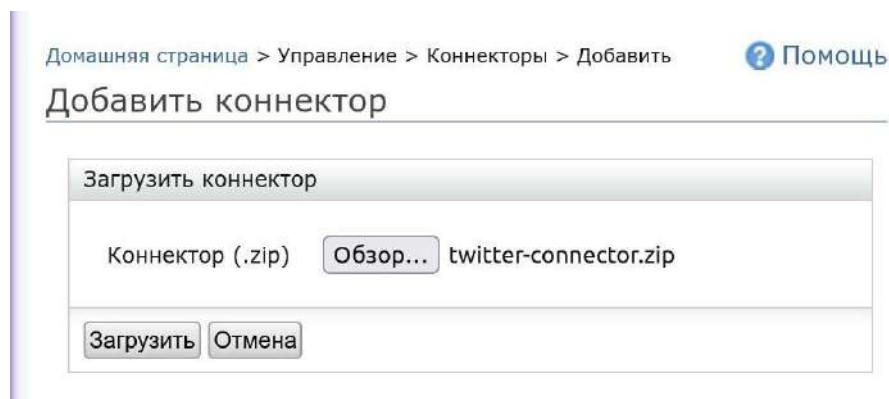


Рис.112. Загрузка коннекторов

Для просмотра списка развернутых коннекторов, щёлкните **Управление > Коннекторы > Список**.

Список коннекторов

Имя библиотеки	Упаковка	Описание	Статус	Действия
twitter	org.wso2.carbon.connectors	synapse library for twitter connector	Отключено	Удалить Скачать

Рис.113. Список коннекторов

Включение и Выключение Коннекторов

Прежде чем вы сможете использовать коннектор, вы должны включить его, щелкнув ссылку **Включить** для этого соединителя в списке. Чтобы сделать соединитель временно недоступным в экземпляре ESB, щёлкните **Отключить**.

Список коннекторов

Имя библиотеки	Упаковка	Описание	Статус	Действия
twitter	org.wso2.carbon.connectors	synapse library for twitter connector	Включено	Удалить Скачать

Рис. 114. Список коннекторов

Удаление Коннектора

Чтобы удалить коннектор из экземпляра Polus ESB, щелкните ссылку **Удалить** для соответствующего коннектора в списке.

ТОПИКИ И ПОДПИСКИ

Polus ESB предоставляет внутренний механизм обмена событиями по типу «Публикация-подписка»: нотификация о событии публикуется в топике, на который возможна подписка для чтения нотификаций из топика. Polus ESB предоставляет браузер топиков для просмотра всех топиков, доступных на сервере, в виде дерева. Пользователь может выполнять все операции с топиком через это представление. Изначально в этом древовидном представлении не отображаются никакие операции, доступные для топика. Когда пользователь щёлкает по топика, отображаются все операции, доступные для этой выбранного топика.

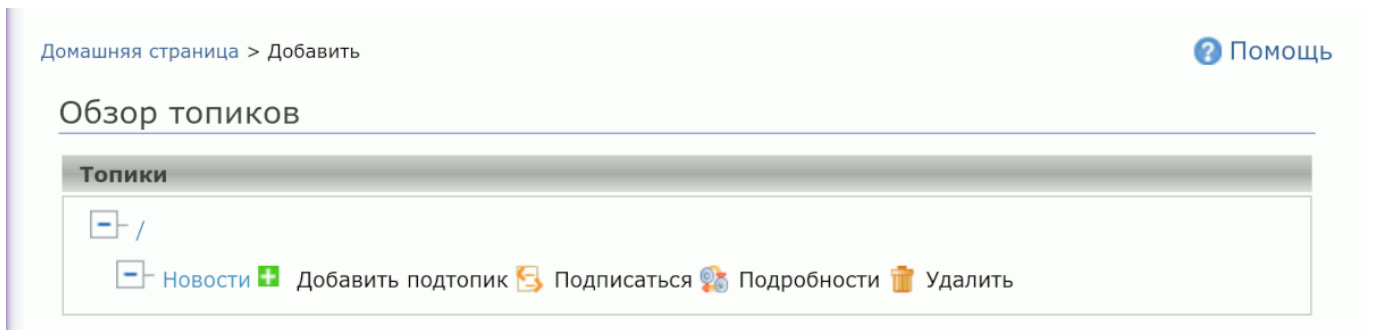


Рис.115. Браузер топиков

Операции, доступные для выбранного топика:

- Добавить Подтопик
- Подписаться
- Показать Подробности
- Удалить

Добавление подтопика

Роль	Подписаться	Публиковать
Internal/everyone	<input type="checkbox"/>	<input type="checkbox"/>

Рис.116. Добавить подтопик

Функция добавления подтопика предоставляет пользователю возможность добавить топик в

существующий топик. При добавлении нового топика в существующий топик пользователь может указать имя нового топика и установить разрешения для публикации и подписки на этот топик.

После добавления подтопика он будет помещена в дерево Просмотра Топиков.

Если пользователь хочет добавить топик непосредственно в корень, он может щелкнуть пункт подменю «Добавить» в меню топиков на левой панели.

Подписка

Домашняя страница > Добавить подписку ? Помощь

Подписаться

Введите данные подписки

Топик*

Режим подписки :*

URL-адрес приемника события*

Дата окончания действия

Время:

Рис.117. Подписка

Все подписки на определенный топик делаются на этой странице. Как только пользователь нажмет на ссылку подписки в браузере топиков, он будет перенаправлен на эту страницу. Чтобы подписаться на этот топик, пользователь должен указать значения следующих параметров:

- Топик** - Пользователю не нужно указывать здесь топик, так как он устанавливается автоматически.
- Режим Подписки** — одно из трех значений:
 - Режим подписки по умолчанию — «Только топик». В этом режиме пользователь создает подписку только на топик. В этом режиме подписчики получают только те события, которые опубликованы только в этом топике.
 - Режим «Топик и ближайший дочерний». В этом режиме подписчики топика получают события, опубликованные не только в указанном топике, но и в ближайшем дочернем элементе этого топика.
 - Режим «Топик и Дочерние». В этом режиме подписчики указанного топика будут получать события, опубликованные в указанном топике и во всех ее дочерних элементах.

– **URL приёмника событий** - это URL, который подписчик должен предоставить для получения опубликованных событий. Когда события публикуются в топике, они отправляются на

указанный здесь URL.

Дата окончания действия - здесь пользователь может указать срок действия подписки. Это необязательный параметр, и если пользователь оставит его незаполненным, срок действия подписки никогда не истечет.

Детали Топика

Домашняя страница > Детали топика ? Помощь

Детали Топика

Название топика:

Сведения о разрешении

Роль	Подписаться	Публиковать
Internal/T_Новости	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Internal/everyone	<input type="checkbox"/>	<input type="checkbox"/>

WS Сведения о подписке

Подписки WS не определены

JMS Сведения о подписке

Подписки на JMS не определены

Публиковать

Топик:

XML Сообщение:

Рис.118. Детали топика

Все относящиеся к топику подробности можно посмотреть на этой странице:

Сведения о разрешении

Здесь можно посмотреть относящиеся к топику разрешения и если пользователь захочет их изменить, то может изменить галочки в предоставленных флажках и нажать кнопку обновления

разрешений.

Подробности Подписки WS

Здесь перечислены все подписки веб-сервисов на этот топик и все его дочерние элементы. Если пользователь хочет отписаться от топика, он может просто щелкнуть ссылку отписки в последнем столбце каждой строки.

Подробности Подписки JMS

Здесь перечислены все постоянные и временные подписки JMS.

– Опубликовать

Эта опция публикации предоставляет пользователю возможность опубликовать образец сообщения XML в топике. После того, как на топик будет оформлен подписка, URL приемника событий получит это опубликованное сообщение XML после нажатия кнопки публикации после размещения сообщения XML в предоставленном пространстве.

Удаление топика

Чтобы удалить топик, количество подписок для этого топика и его дочерних элементов должно быть равно нулю. В противном случае появится сообщение об ошибке, указывающее, что для топика или его дочерних элементов есть подписки.

РАЗВЕРНУТЫЕ СЕРВИСЫ

Страница **Развернутые сервисы** позволяет управлять различными видами сервисов и групп сервисов. Если ваш сервис развернут успешно, он появится на этой странице. Если нет, это будет показано ссылкой «Неисправные сервисные группы» (которая отображается красным цветом).

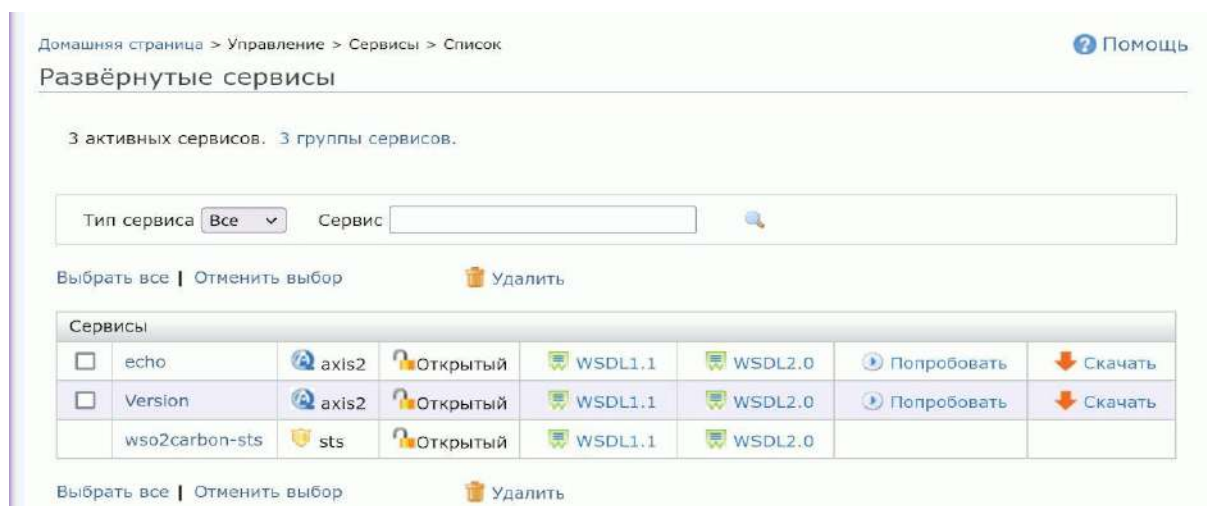


Рис.119. Развёрнутые сервисы

Polus ESB предоставляет множество средств для управления сервисами, которые были успешно развернуты. При развертывании отдельного сервиса в его архиве имя файла архива всегда будет использоваться в качестве имени файла сервиса, если у вас нет имени, связанного с файлом сервиса. Если имя файла архива сервиса Test.aar, то имя сервиса будет Test.

На странице «Панель управления сервисом» представлена подробная информация о заданном сервисе.

Домашняя страница > Панель управления сервисом ? Помощь

Панель управления сервисом (echo)

Информация о сервисе		Клиентские операции	
Название сервиса	echo	Попробовать Создать клиент Axis2 WSDL1.1 WSDL2.0	
Описание сервиса	echo		
Группа сервисов	Echo		
Область развертывания	request		
Тип сервиса	axis2	Точки входа https://esb.inpolus.ru:8243/services/echo http://esb.inpolus.ru:8280/services/echo	

Операции	
	Активен [Деактивировать]

Статистика	
Количество запросов	0
Количество откликов	0
Количество неисправностей	0
Максимальное время отклика	0 ms
Минимальное время отклика	0 ms
Среднее время отклика	0.0 ms

Среднее время отклика (мс) по сравнению с (относительными единицами измерения)

Рис.120. Панель управления сервисом

Панель «**Информация о сервисе**» содержит следующие сведения:

- Имя Сервиса;
- Описание Сервиса;
- Группа сервиса;
- Пространство развёртывания;
- Тип сервиса.


Информация о сервисе	
Название сервиса	echo
Описание сервиса	echo
Группа сервисов	Echo
Область развертывания	request
Тип сервиса	axis2 

Рис.121. Информация о сервисе

Панель «**Статистика**» предоставляет следующую статистику по сервису:

- Количество Запросов;
- Количество Ответов;
- Количество Ошибок;
- Максимальное, Минимальное, Среднее Время Отклика.

Статистика	
Количество запросов	0
Количество откликов	0
Количество неисправностей	0
Максимальное время отклика	0 ms
Минимальное время отклика	0 ms
Среднее время отклика	0.0 ms

Рис.122. Статистика по сервису

Polus ESB предоставляет графическое представление времени отклика системы.



Рис.123. Время отклика

Точки Вызова

Polus ESB по умолчанию настроен на две точки вызова.

- <https://10.100.1.147:9443/services/HelloService/>
- <http://10.100.1.147:9763/services/HelloService/>

Клиентские Операции

Polus ESB обеспечивает две клиентские операции. Это «Попробовать» и «Создать клиент Axis2».

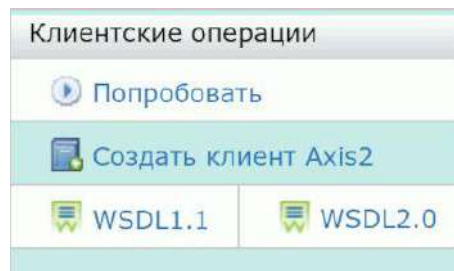


Рис.124. Клиентские Операции

Вы можете проверить, обеспечивает ли ваш сервис желаемый вывод, используя опцию **Try-It**. Когда вы щёлкните **Попробовать**, вы будете перенаправлены на страницу, где будут отображаться операции, доступные для вашего сервиса.

Генерация клиента для вашего сервиса

Вы можете легко создать клиент для своего сервиса с помощью - опции кода WSDL2 доступны на странице **WSDL2Java**. Вы увидите ссылки на файлы WSDL 1.1 и WSDL 2.0 в браузере.

Группы Сервисов

Группа сервисов — это удобный способ развертывания нескольких сервисов в одном файле архива сервисов. Между сервисами во время выполнения существует логическая связь. Единственная разница в `services.xml` для группы сервисов и отдельного сервиса заключается в его корневом элементе. Для группы сервисов корневым элементом является элемент `serviceGroup` и есть несколько элементов сервисов внутри элемента `serviceGroup`.

Например:

- `<serviceGroup>`
- `<service name=Test1>`
- -----
- `<service>`
- `<service name=Test2>`
- -----

- `</service>`
- `</serviceGroup>`

Щелкните ссылку развернутых групп сервисов на странице со списком сервисов, чтобы отобразить группы сервисов.