CROC INTEGRATION PLATFORM

Руководство по доработке

Листов 11

Содержание

1 Введение 3
2 Требования к коннекторам 4
3 Диагностика коннекторов 7
4 Комментарии к реализации поставляемых с платформой коннекторов 8
4.1 Коннектор входящих соединений 8
4.2 Коннектор исходящих соединений 8
Перечень условных обозначений, терминов и сокращений 10

1 Введение

CROC Integration Platform или Платформа интеграции КРОК (далее платформа) предназначена для построения решений по координации совместной работы информационных систем предприятий и организаций различных сфер деятельности, а также государственных органов власти и местного самоуправления и позволяет обеспечить целостность и надежность доставки данных.

Платформа предоставляет стандартизированный интерфейс для подключения приложений и обеспечивает возможность создания правил маршрутизации и преобразования потоков данных. Встроенная система мониторинга предоставляет инструменты сквозного контроля над функционированием платформы и смежных приложений.

Для эффективной эксплуатации и поддержки платформы необходимы следующие категории персонала:

– администраторы платформы − выполняют администрирование компонентов платформы.

Перед эксплуатацией обслуживающему персоналу необходимо ознакомиться со следующими документами:

- Руководство по развертыванию и настройке CROC Integration Platform;
- Руководство по доработке CROC Integration Platform;
- документация по платформе виртуализации Docker (https://docs.docker.com/).

2 Требования к коннекторам

Коннекторы входящих и исходящих соединений взаимодействуют с потоками обработки обращений и ответов через транспортную подсистему, представляющую из себя менеджер очередей Apache Artemis версии 1.5.6. Каждый коннектор должен быть подключен к очереди входящих и исходящих транспортных сообщений.

Схема взаимодействия коннекторов с потоками обработки обращений и ответов приведена на рисунке 1.

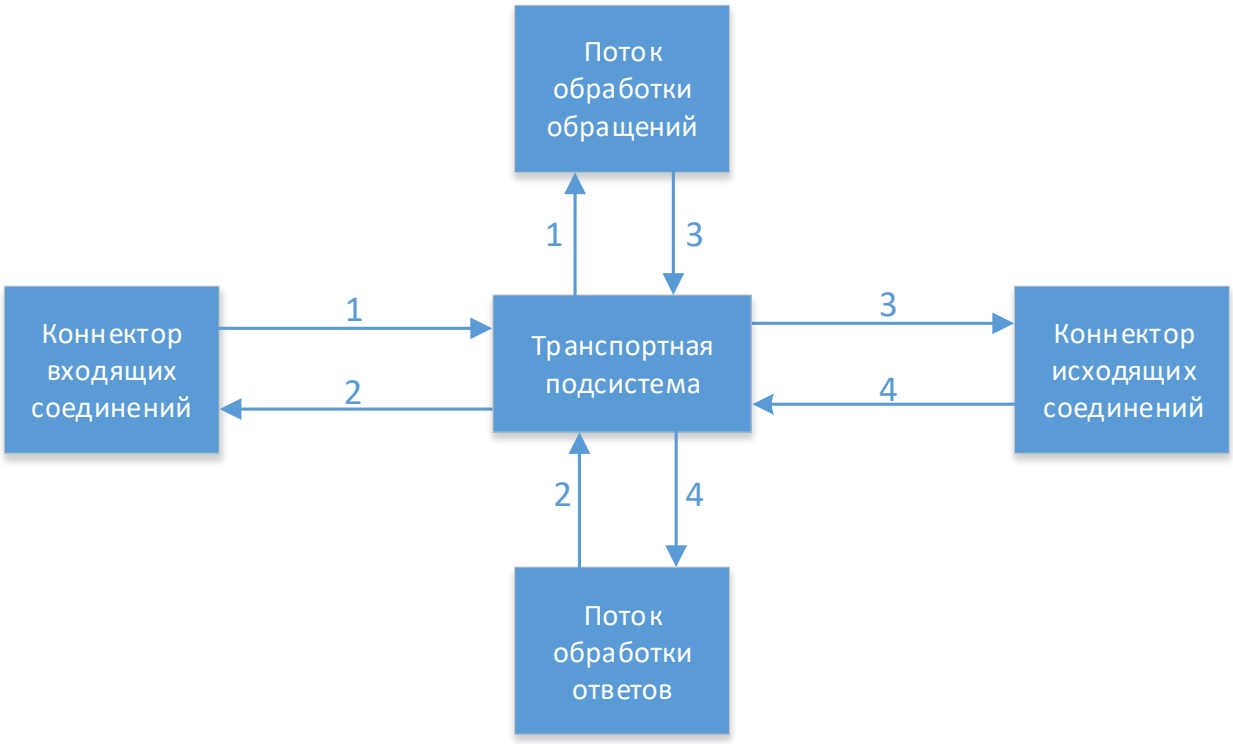

Рисунок 1 – Схема взаимодействия коннекторов с потоками обработки

Требования к взаимодействию между коннекторами и потоками обработки представлены в таблице 1.

Таблица 1 – Требования к взаимодействию между коннекторами и потоками обработки платформы интеграции КРОК

№	Название	Очередь	Требования к сообщениям
1	Взаимодействие между коннектором входящих соединений и потоком обработки обращений	Очередь исходящих сообщений коннектора входящих соединений – в данную очередь поступают преобразованные в транспортные сообщения обращения от системы инициатора	Сообщения должны иметь следующие обязательные JMS заголовки: – contentType –тип данных стандарта HTTP, в значении которого символ «/» заменен на символ «_»; – responseConnectorUri – адрес входящей очереди коннектора в формате jms:queueName; – messageType –тип сообщения (может быть пустым), используется при валидации, трансформации и маршрутизации. Сообщения могут иметь следующие дополнительные заголовки: – JMSCorrelationID – используется для корреляции обращения и ответа системы получателя
2	Взаимодействие между потоком обработки ответов и коннектором входящих соединений	Очередь входящих сообщений коннектора входящих соединений - в данную очередь поступают преобразованные в транспортные сообщения ответы от системы получателя (в случае их наличия)	Сообщения должны иметь следующие обязательные JMS заголовки: – contentType – описание типа данных стандарта HTTP, в котором символ «/» заменен на символ «_»; – JMSCorrelationID – используется для корреляции обращения и ответа системы получателя
3	Взаимодействие между потоком обработки обращений и коннектором исходящих соединений	Очередь входящих сообщений коннектора исходящих соединений - в данную очередь поступают преобразованные в транспортные сообщения обращения от системы инициатора	Сообщения должны следующие обязательные JMS заголовки: – contentType –тип данных стандарта HTTP, в значении которого символ «/» заменен на символ «_»; – communicationType – тип обращения, request или notification; – messageType – тип сообщения (может быть пустым)

№	Название	Очередь	Требования к сообщениям
4	Взаимодействие между коннектором исходящих соединений и потоком обработки ответов	Очередь исходящих сообщений коннектора исходящих соединений - в данную очередь поступают преобразованные в транспортные сообщения ответы от системы получателя (в случае их наличия)	Сообщения должны иметь следующие обязательные JMS заголовки: – contentType –тип данных стандарта HTTP, в значении которого символ «/» заменен на символ «_»; – messageType –тип сообщения (может быть пустым), используется при валидации и трансформации; – adapterName – имя адаптера обработки ответов, который будет обрабатывать ответ системы получателя (в случае его наличия)

3 Диагностика коннекторов

Для диагностики подключения вновь разрабатываемых коннекторов входящих и исходящих соединений с потоками обработки обращений и ответов необходимо:

1) Проверить на наличие ошибок журналы вновь разрабатываемых коннекторов входящих и исходящих соединений в соответствии с их документацией;
2) Проверить на наличие ошибок журналы компонентов платформы как описано в подразделе 7 документа «Руководство по развертыванию и настройке CROC Integration Platform»
3) Проверить корректность настройки маршрутизации, валидации и трансформации обращений в интерфейсе администратора платформы как описано в подразделе 9.3 документа «Руководство по развертыванию и настройке CROC Integration Platform»

4 Комментарии к реализации поставляемых с платформой коннекторов

4.1 Коннектор входящих соединений

Исходный код коннектора находится в директории дистрибутива source, модуль http-inbound проекта connector.

Компонент реализован на языке программирования Java с использованием инструмента сборки Apache Maven и каркаса разработки приложений Spring Boot версии 1.5.15.

Класс InboundController, расположенный по пути:

src/main/java/croc/miniesb/connector/http/inbound/controller/InboundController.java,

обрабатывает поступающие к компоненту HTTP обращения методом POST при помощи методов asyncRequest и syncRequest (в зависимости от URI запроса, /async/ и /sync/ соответственно) и вызывает методы sendNotification или sendRequest (в зависимости от типа обращения, нотификация или запрос соответственно) класса MessageService, расположенного по пути:

src/main/java/croc/miniesb/connector/http/inbound/service/MessageService.java,

передавая в них тело обращения и значения заголовков HTTP Content-Type и Message.

Методы asyncRequest и syncRequest класса MessageService преобразуют обращение в транспортное сообщение и отправляют его в транспортную подсистему, используя метод send класса Sender, расположенного по пути:

src/main/java/croc/miniesb/connector/http/inbound/jms/Sender.java.

Метод sendRequest класса MessageService дополнительно вызывает метод receive класса Receiver, расположенного по пути:

src/main/java/croc/miniesb/connector/http/inbound/jms/Receiver.java,

который ожидает ответа из транспортной подсистемы. При получении ответа он преобразуется из формата из транспортного сообщения и возвращается по цепочке вызовов в метод syncRequest класса InboundController, где преобразуется в ответ системе-инициатору в по протоколу HTTP.

4.2 Коннектор исходящих соединений

Исходный код коннектора находится в директории дистрибутива source, модуль http-outbound проекта connector.

Компонент реализован на языке программирования Java с использованием инструмента сборки Apache Maven и каркаса разработки приложений Spring Boot версии 1.5.15.

Класс Listener, расположенный по пути:

src/main/java/croc/miniesb/connector/http/outbound/jms/Listener.java,

получает сообщения из транспортной подсистемы, преобразует их из формата транспортного сообщения и вызывает методы sendNotification или sendRequest (в зависимости от типа обращения, нотификация или запрос соответственно) класса MessageService, расположенного по пути:

src/main/java/croc/miniesb/connector/http/outbound/service/MessageService.java,

которые инициируют обращение к системе-получателю по протоколу HTTP методом POST.

Метод sendRequest класса MessageService получает ответ от системы-получателя и вызывает метод send класса Sender, расположенного по пути:

src/main/java/croc/miniesb/connector/http/outbound/jms/Sender.java,

который преобразует ответ в формат транспортного сообщения и отправляет в транспортную подсистему.

Перечень условных обозначений, терминов и сокращений

Платформа	– CROC Integration Platform, платформа интеграции КРОК
Docker	– система виртуализации уровня операционной системы
Коннектор входящих соединений	– компонент платформы, получает обращения от информационной системы источника, преобразовывает обращения в транспортные сообщения и отправляет в поток обработки обращений, при наличии ответа преобразовывает его из транспортного сообщения в протокол взаимодействия с информационной системой и возвращает ответ информационной системе источнику
Коннектор исходящих соединений	– компонент платформы, получает транспортные сообщения от потока обработки обращений, преобразовывает их из транспортного сообщения в протокол взаимодействия с информационной системой получателем и производит взаимодействие с информационной системой, в случае наличия ответа преобразовывает его в транспортное сообщение и отправляет в поток обработки ответов
Поток обработки обращений	– компонент платформы, получает от коннектора входящих соединений транспортное сообщение с данными обращения, получает у интерфейса администратора платформы по протоколу SOAP правила по маршрутизации валидации, трансформации, после чего производит операции маршрутизации, валидации и трансформации обращения и отправляет его в коннектор исходящих соединений
SOAP	– Simple Object Access Protocol – протокол обмена структурированными сообщениями в распределённой вычислительной среде
Поток обработки ответов	– компонент платформы, получает от коннектора исходящих соединений транспортное сообщение с данными ответа, производит операции валидации и трансформации ответа и отправляет его в коннектор входящих соединений
Транспортная подсистема	– компонент платформы, менеджер очередей, обеспечивает надежную доставку транспортных сообщений между компонентами платформы интеграции
Менеджер очередей	– программный компонент, предоставляющий сервисы очередей сообщений посредством API
API	– набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах
Очередь	– именованное хранилище сообщений, управляемое посредством менеджера очередей
Транспортное сообщение	– совокупность элементов информации, оформленных в соответствии с требованиями транспортного протокола.
JMS	– Java Message Service, стандарт промежуточного ПО для рассылки сообщений. Используется версия 2.0.
Тип сообщения	– обязательный заголовок транспортного сообщения определяющий

HTTP	− HyperText Transfer Protocol – протокол прикладного уровня передачи данных
Валидация	− процесс проверка соответствия обращения установленным для него требованиям
Маршрутизация	− процесс определения системы получателя обращения поступившего от системы инициатора
Трансформация	− процесс преобразования обращения, например, в другой формат
URI	− Uniform Resource Identifier — унифицированный (единообразный) идентификатор ресурса