Описание программы TRANSPORT applICATION

**СОДЕРЖАНИЕ**

[Определения, обозначения и сокращения 3](#_Toc514409506)

[Аннотация 4](#_Toc514409507)

[1 Общие сведения 5](#_Toc514409508)

[2 Функциональное назначение 6](#_Toc514409509)

[3 Описание логической структуры 7](#_Toc514409510)

[3.1 Алгоритм программы 7](#_Toc514409511)

[3.2 Используемые методы 7](#_Toc514409512)

[3.3 Структура программы с описанием функций составных частей 7](#_Toc514409513)

[3.4 Связи программы с другими программами 10](#_Toc514409514)

[4 Входные данные 11](#_Toc514409515)

[5 Выходные данные 12](#_Toc514409516)

[6 Перечень таблиц 13](#_Toc514409517)

[7 Список рисунков 14](#_Toc514409518)

[A. Приложение А. Алгоритм программы Transport Application по обработке основного потока данных 15](#_Toc514409519)

Определения, обозначения и сокращения

| Термин/сокращения | Определение |
| --- | --- |
| HTTP | От англ. HyperText Transfer Protocol – «протокол передачи гипертекста» – протокол прикладного уровня передачи данных (изначально – в виде гипертекстовых документов в формате HTML, в настоящий момент используется для передачи произвольных данных) |
| SOAP | От англ. Simple Object Access Protocol – Простой протокол доступа к объектам, протокол SOAP |
| TLS | От [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Transport Layer Security – Протокол защиты транспортного уровня |
| ТСР/IP | Сокр. англ. Transmission control protocol/Internet protocol – Протокол управления передачей/Межсетевой протокол |
| WSDL | Язык описания [веб-сервисов](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81" \o "Веб-сервис) и доступа к ним, основанный на языке [XML](https://ru.wikipedia.org/wiki/XML) |
| XML | eXtensible Markup Language (Расширяемый язык разметки) |
| ИС ЭП | Информационная система «Электронная Путевка» |
| ИС | Информационная система |
| НСИ | Нормативно-справочная информация |
| СКЗИ | Средство криптографической защиты информации |
| ЭП | Электронная путёвка |

Аннотация

В данном документе приведено описание программы Transport Application.

В разделе «Общие сведения» указаны обозначение и наименование программы, программное обеспечение, необходимое для функционирования программы, языки программирования, на которых написана программа.

В разделе «Функциональное назначение» указаны классы решаемых задач и (или) назначение программы и сведения о функциональных ограничениях на применение.

В разделе «Описание логической структуры» указаны алгоритм программы, структура программы с описанием функций составных частей и связи между ними, связи программы с другими программами.

В разделах «Входные данные» и «Выходные данные» указаны характер, организация и предварительная подготовка данных, формат, описание и способ кодирования данных.

# Общие сведения

Наименование системы: Программа Transport Application.

Условное обозначение системы: Transport Application.

Для функционирования Transport Application необходимо следующее программное обеспечение:

* CentOS 7.x;
* Apache Tomcat 9.0.1;
* Apache Cassandra 3.0.9;
* СКЗИ КриптоПро JCP версии 2.0.39014.

Программа написана на языке программирования Java.

# Функциональное назначение

Программа Transport Application предназначена для поддержки и обеспечения функциональности ИС ЭП, в части, касающейся регистрации и изменения состояния ЭП, ведение НСИ, и т.п.

Обмен данными с программой Transport Application осуществляется методами transport‑сервиса с использованием протокола SOAP. Данные передаются в составе сообщения, тело которого состоит из следующих структурных элементов.

В качестве технологии информационного обмена используются transport‑службы стиля document/literal. Стороны взаимодействуют, обмениваясь SOAP‑сообщениями. SOAP‑сообщения должны удовлетворять требованиям спецификации на SOAP версии 1.2 и состоять из двух частей.

Описание всех операций, их входных и выходных параметров, а также возможных исключений приводится в WSDL‑документах, удовлетворяющих спецификации WSDL версии 2.0.

Используя WSDL‑описание, разработчики прикладных программных средств самостоятельно реализуют клиентскую часть программного обеспечения, реализующую необходимую функциональность.

# Описание логической структуры

* 1. Алгоритм программы

Программа Transport Application принимает, регистрирует полученное SOAP сообщение‑запрос (далее – запрос), проверяет полномочия источника запроса, валидирует, обрабатывает запрос и далее передает его посредством платформы RabbitMQ, реализующей систему обмена сообщениями между компонентами ИС ЭП, в программу Core Application для формирования содержательного ответа.

При получении запроса программой Core Application, начинается фаза парсинга и логической обработки запроса. В случае успеха, формируется содержательный ответ, в ином случае формируется ответ о невозможности обработать запрос (с подробным указанием причины).

Далее, подготовленный ответ на запрос посредством вызова соответствующего метода веб‑сервиса отправляется в программу Transport Application, где происходит его обработка и сохранение.

На данном этапе обработка сообщения‑запроса считается завершенной и ответ на него может быть получен источником первоначального запроса посредством обращения к соответствующему методу веб-сервиса программы Transport Application.

Алгоритм программы Transport Application по обработке основного потока данных приведён в приложении А.

* 1. Структура программы с описанием функций составных частей

Перечень используемых классов в Transport Application указан в таблице 1.

Таблица 1 – Перечень используемых классов в Transport Application

| Название | Описание |
| --- | --- |
| ru.russiatourism.domain.MessageExchange | Класс-сущность MessageExchange |
| ru.russiatourism.domain.RequestType | Класс-сущность RequestType |
| ru.russiatourism.repository.MessageExchangeRepository | Класс-репозиторий для сущности MessageExchange |
| ru.russiatourism.repository.RequestTypeRepository | Класс-репозиторий для сущности RequestType |
| ru.russiatourism.config.ApplicationProperties | Класс-конфигуратор основных настроек Transport Application |
| ru.russiatourism.config.CassandraConfiguration | Класс-конфигуратор NoSQL хранилища Apache Cassandra |
| ru.russiatourism.config.Constants | Класс, определяющий основные константы программы |
| ru.russiatourism.config.DefaultProfileUtility | Класс, определяющий профиль загрузки программы по умолчанию |
| ru.russiatourism.service.crypto.SignatureValidatorService | Класс-сервис, предназначенный для валидации электронной подписи SOAP сообщения‑запроса |
| ru.russiatourism.service.helper.DatabaseInitializer | Класс-хэлпер, предназначенный для инициализации хранилища стандартным набором данных при запуске Transport Application |
| ru.russiatourism.service.helper.RequestStatus | Класс-хэлпер, определяющий набор именованных констант, описывающих предусмотренные в ИС ЭП статусы (состояния) сообщений-запросов |
| ru.russiatourism.service.helper.RequestValidationResult | Класс-обертка, описывающий структуру результата проверки сообщений-запросов на соответствие формально-логическим требованиям, предъявляемым к таким сообщениям |
| ru.russiatourism.service.helper.RequestValidator | Класс, предназначенный для проверки сообщений-запросов на соответствие формально-логическим требованиям, предъявляемым к таким сообщениям |
| ru.russiatourism.service.helper.RequestValidatorErrorHandler | Класс-обработчик, предназначенный для обработки исключений, возникающих при несоответствии сообщений-запросов формально-логическим требованиям, предъявляемым к таким сообщениям |
| ru.russiatourism.service.helper.Utils | Класс, определяющий различные вспомогательные утилиты, часто используемые в процессе работы Transport Application |
| ru.russiatourism.service.helper.ValidationError | Класс-обертка, описывающий структуру ошибки результата проверки сообщений-запросов на соответствие формально-логическим требованиям, предъявляемым к таким сообщениям |
| ru.russiatourism.service.queue.QueuePublisher | Класс-сервис, предназначенный для передачи сообщений в программу Core Application посредством платформы RabbitMQ, реализующей систему обмена сообщениями между компонентами ИС ЭП, для формирования ответа |
| ru.russiatourism.service.MessageExchangeService | Класс-сервис для сущности MessageExchange |
| ru.russiatourism.service.RequestTypeService | Класс-сервис для сущности RequestType |
| ru.russiatourism.ws.\* | Пакет классов, определяющий и описывающий формат и структуру SOAP сообщений-запросов, а также различные аспекты электронного взаимодействия с Transport Application посредством веб-сервисов |
| ru.russiatourism.EpTransportApplication | Основной класс программы Transport Application |

Контроллеры предоставления данных с использованием веб‑сервисов (SOAP) реализованы с использованием Apache CXF. WSDL описания веб‑сервисов генерируются на основе исходного кода и постоянно находятся в синхронном актуальном состоянии с контроллерами, принимающими внешние запросы. Данная возможность реализована за счет использования Spring framework.

* 1. Связи программы с другими программами

Программа Transport Application взаимодействует с программой Core Application и информационными системами туроператоров посредством обмена XML‑сообщениями по протоколу SOAP через HTTP, а также посредством платформы RabbitMQ, реализующей систему обмена сообщениями между компонентами ИС ЭП.

# Входные данные

Работа программы Transport Application обеспечивается цифровыми каналами связи TCP/IP, обеспечивающими получение данных между внутренними подсистемами ИС ЭП.

Каналы входных сигналов Transport Application указаны в таблице 2.

Таблица 2 – Каналы входных сигналов Transport Application

|  |  |  |
| --- | --- | --- |
| Наименование канала входных сигналов | Транспортный протокол | Определяющий стандарт |
| IN-WSDL-DATA | HTTP, SOAP | Каналообразующий протокол – RFC 2616, Hypertext Transfer Protocol |
| TLS-DATA | TLS | Каналообразующий протокол – RFC 2246, Transport Layer Security (TLS v. 1.0) |

# Выходные данные

Работа Transport Application обеспечивается цифровыми каналами связи TCP/IP, обеспечивающими получение данных между внутренними подсистемами ИС ЭП. Каналы выходных сигналов Transport Application указаны в таблице 3.

Таблица 3 – Каналы выходных сигналов Transport Application

|  |  |  |
| --- | --- | --- |
| Наименование канала выходных сигналов | Транспортный протокол | Определяющий стандарт |
| OUT-WSDL-DATA | HTTP, SOAP | Каналообразующий протокол – RFC 2616, Hypertext Transfer Protocol |

# Перечень таблиц

[Таблица 1 – Перечень используемых классов в Transport Application 7](#_Toc498357139)

[Таблица 2 – Каналы входных сигналов Transport Application 13](#_Toc498357140)

[Таблица 3 – Каналы выходных сигналов Transport Application 14](#_Toc498357141)

# Список рисунков

[Рисунок A.1 – Алгоритм программы Transport Application по обработке основного потока данных 17](#_Toc498357148)

1. Приложение А. Алгоритм программы Transport Application по обработке основного потока данных

Алгоритм программы Transport Application по обработке основного потока данных изображён на рисунке (где «Основное приложение» – программа Core Application, «Транспортное приложение – программа Transport Application»).

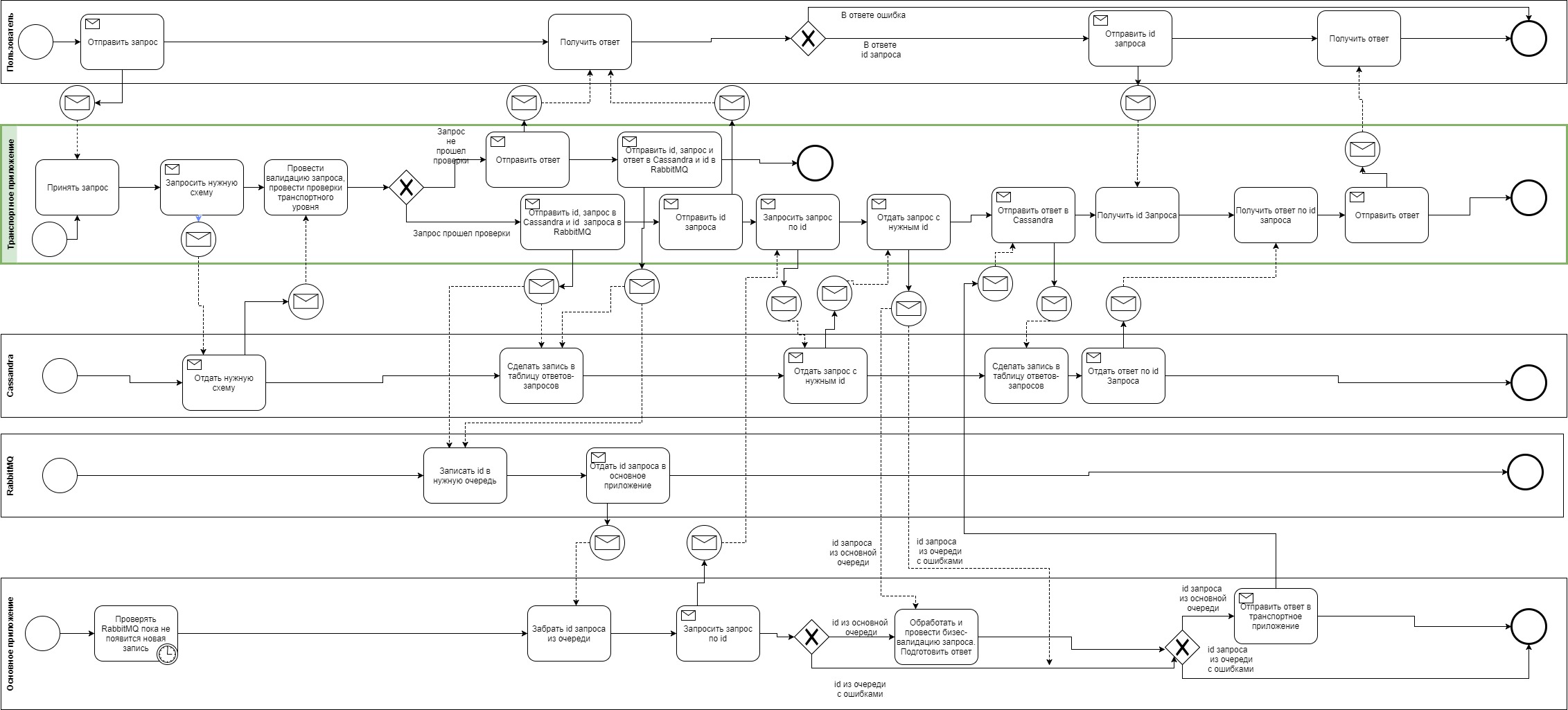


Рисунок A.1 – Алгоритм программы Transport Application по обработке основного потока данных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ***ЮИВЭ.300214.001.ЭД-ИСЭП-3. TransportAppl.13*** | *Лист* |
|  |  |  |  |  | 15 |
| *Изм.* | *Лист* | *№ документа* | *Подпись* | *Дата* |