|  |
| --- |
| **Спецификация «Интеграционные интерфейсы экспорта данных»** |

|  |  |
| --- | --- |
| **Код проекта** | **Наименование проекта** |
|  | Мониторинг технологических отклонений, включая параметры тех. режима и показатели качества. Разработка бизнес-системы «HDP Озеро данных БЛПС» |

|  |  |
| --- | --- |
| **Наименование Бизнес-решения** | **Наименование Бизнес-системы** |
| МО | HDP Озеро данных БЛПС |

|  |  |
| --- | --- |
| **Название документа:** | **Спецификация «Интеграционные интерфейсы экспорта данных»** |
| **Версия документа:** | 1.0 |
| **Дата документа:** | 13.08.2018 |
| **Назначение документа:** | Описание структуры данных и алгоритмов системы |
| **Аудитория:** | Заказчик проекта, Исполнитель работ по проекту (подрядчик), Согласующие подразделения |

**История изменений**

|  |  |  |  |
| --- | --- | --- | --- |
| **Версия** | **Дата** | **Комментарий** | **Автор** |
| 1.0. | 07.08.18 | Первоначальная версия подрядчика ЗАО «КРОК Инкорпорейтед» | Егоров П.В. |
| 1.1 | 16.05.19 | Актуализированная версия | Егоров П.В. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Связанные документы** (этот документ должен читаться вместе с)

|  |  |  |
| --- | --- | --- |
| **Название документа** | **Номер версии / Имя файла** | **Дата** |
| Проектное решение | 1.0 |  |
| Спецификация «Интеграционные интерфейсы импорта данных» | 1.0 | 13.08.2018 |
| Спецификация «Расчет отклонений» | 1.0 | 13.08.2018 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Должность** | **ФИО** | **Роль** | **Дата** | **Подпись** |
| Начальник [Управления технологической эффективности](http://portal.gazprom-neft.local/HR/Employees/Pages/search.aspx#c=1&fl=MPhotoUrl,Name,PublicUrl,NameLink,PublicUrlLink,JobTitle,Company,WorkPhone,MobilePhone,Email,City,Office,Location,DepartmentChain,ExtensionPhone&p=5&flv=,,,,,,,&d=76028) ЦУЭ БЛПС | Рыков Р.В. | Единое ответственное лицо |  |  |
| Главный специалист дирекции корпоративной защиты ДКЗ ББ | Горбунов А.А. | Эксперт |  |  |
| Начальник управления систем автоматизации бизнеса ДСУ БЛПС | Макеенко Д.В. | Куратор от ДСУ БЛПС |  |  |
| Начальник управления промышленной автоматизации БЛПС | Стариков В.А. | Куратор от ДСУ БЛПС |  |  |
| Руководитель направления трансформации бизнес – процессов ДСУ БЛПС | Боровикова Л.В.. | Руководитель проекта от заказчика |  |  |
| Руководитель направления  Инженерный центр, Управление систем операционной деятельности  ООО «Автоматика-сервис» | Еременко А.В. | Руководитель проекта от исполнителя |  |  |
| Руководитель направления ИТ архитектуры ДСУ БЛПС | Кобец А.В. | Архитектор БЛПС |  |  |
| Главный специалист ООО «ИТСК» Дирекция переработки и сбыта | Буткин Р.Ю. | Архитектор ИТ-решения 1С |  |  |
| Начальник управления аналитических исследований и метрологического обеспечения ООО «Автоматика-сервис» | Васбиев Д.Р. | Архитектор ИТ-решения HADOOP |  |  |

Оглавление

[1. Общие положения 4](#_Toc523493563)

[2. Термины и сокращения 5](#_Toc523493564)

[3. Описание задачи 6](#_Toc523493565)

[4. Описание разработанного решения 7](#_Toc523493566)

[4.1 Общая схема потоков 7](#_Toc523493567)

[4.2 Реестр интеграционных процессов экспорта 7](#_Toc523493568)

[4.3 Реестр процессоров разработанного решения 8](#_Toc523493569)

[4.4 Описание процесса «Отправка данных по событиям (показатели качества)» и «Отправка данных по событиям (показатели технологического режима)» 11](#_Toc523493570)

[4.5 Описание процесса экспорта данных по событиям 16](#_Toc523493571)

[4.6 Описание процесса экспорта данных по статистическим показателям 19](#_Toc523493572)

# Общие положения

Настоящий документ является описанием решения по экспорту данных, разработанного в рамках проекта «HDP Озера данных БЛПС».

# Термины и сокращения

|  |  |  |
| --- | --- | --- |
| **№** | **Термин/ сокращение** | **Определение** |
|  | Система МО | Система мониторинга технологических отклонений, включая тех. режимы и качества |
|  | Внешние системы, системы-источники данных | Смежные информационные системы к Системе МО, по которым будут разработаны интеграционные требования (PI, LIMS, лента событий ЦУЭ и др.) и которые будут подключены к Системе МО |
|  | МНПЗ | Московский нефтеперерабатывающий завод |
|  | ОНПЗ | Омский нефтеперерабатывающий завод |
|  | НТР | Нормы технологического режима |
|  | Отклонение | Отклонения фактических параметров от норм, зафиксированных в НМД |
|  | Параметры технологического режима, Параметры | Величина, значение которой, характеризуют технологический процесс: давления, температуры, расходы и т.п. |
|  | PI-System | Система хранения фактических параметров технологических установок на ОНПЗ и на МНПЗ |
|  | LIMS | Система хранения фактических данных лабораторного контроля качества на ОНПЗ и на МНПЗ |

# Описание задачи

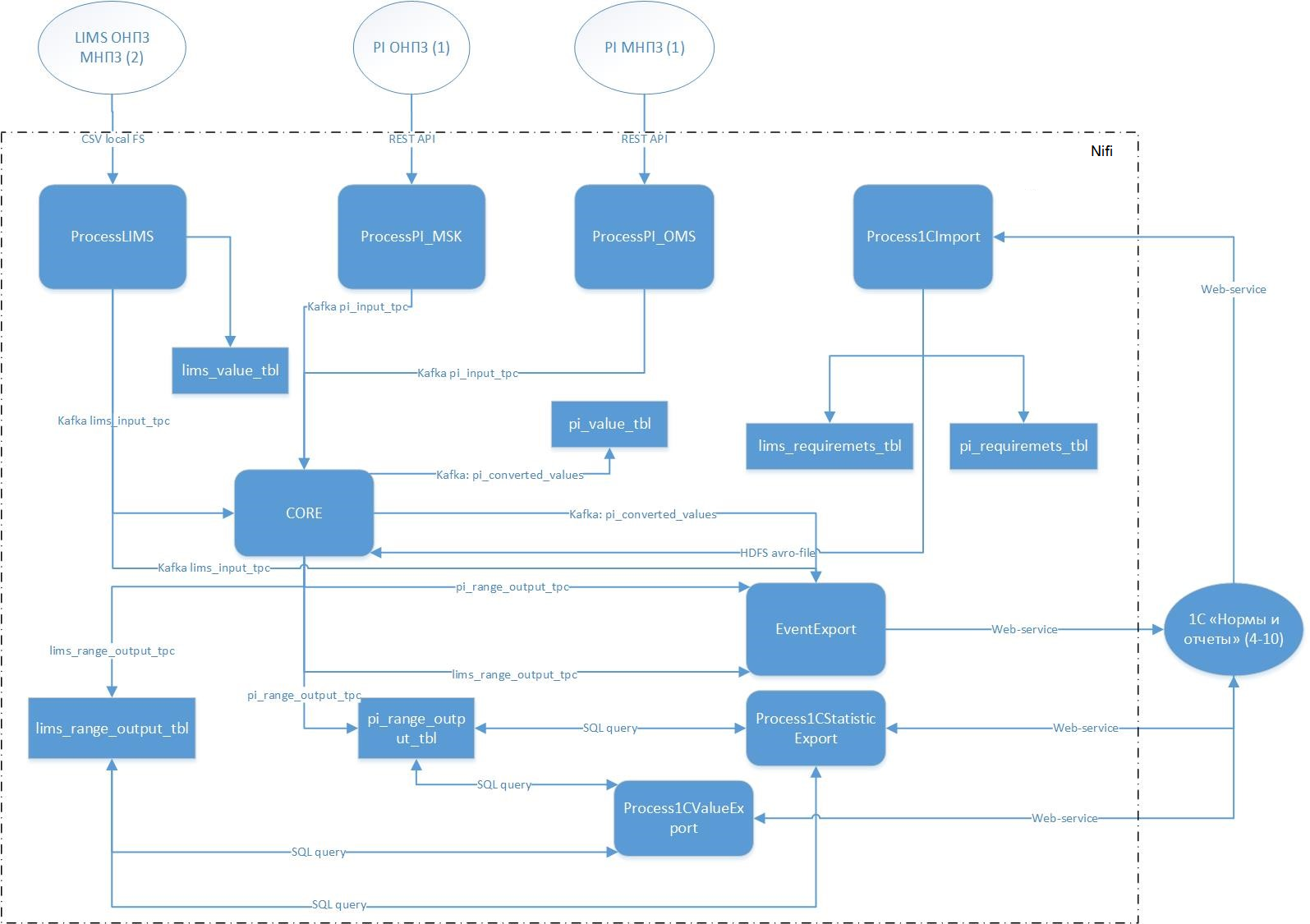
В данном документе описана реализация следующих функций системы:

* Настроить интеграционный поток по выгрузке событий отклонениях технологических показателей и показателей качества в 1С: Нормы и отчеты из kafka по rest сервису, используя nifi.
* Настроить интеграционный поток по выгрузке статистики показателей за период в 1С: Нормы и отчеты из hbase по rest сервису, используя nifi.
* Настроить интеграционный поток по выгрузке сконвертированных значениях показателей за период в 1С: Нормы и отчеты из kafka по rest сервису, используя nifi.

# Описание разработанного решения

Процесс экспорта данных реализован на базе Apache NiFi с использованием Apache Kafka для чтения данных из модуля расчетов (Core). В общем виде процесс представляет из себя поток данных, состоящий из flowFile’ов относящихся к изолированным сущностям интегрируемых данных, которые проходят последовательную обработку, проходя через процессоры NiFi. Процессор представляет из себя функционально независимую часть кода, и имеют специализированное назначение (например, отправка или получение данных через http-запросы, редактирование содержимого flowFile, разделение потока по условию).

## Общая схема потоков



## Реестр интеграционных процессов экспорта

| № | Наименование процесса | Источник | Источник данных | Тип обмена | Формат обмена | Приемник |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | Отправка данных по статическим показателям тех. режима | HDP Nifi | Pi\_range\_output\_tbl | Rest сервис | Json | 1С «Нормы и отчеты» |
| 6 | Отправка данных по статическим показателям качества | HDP Nifi | Lims\_range\_output\_tbl | Rest сервис | Json | 1С «Нормы и отчеты» |
| 7 | Отправка данных по событиям (технологические показатели) | HDP Nifi | Kafka Pi\_converted\_values | Rest сервис | Json | 1С «Нормы и отчеты» |
| 8 | Отправка данных по событиям (показатели качества) | HDP Nifi | Kafka Lims\_input | Rest сервис | Json | 1С «Нормы и отчеты» |
| 9 | Отправка данных по детализированной статистике показателей тех. режима | HDP Nifi | Pi\_range\_output\_tbl | Rest сервис | Json | 1С «Нормы и отчеты» |
| 10 | Отправка данных по детализированной статистике показателей качества | HDP Nifi | Lims\_range\_output\_tbl | Rest сервис | Json | 1С «Нормы и отчеты» |
| 9 | Отправка данных по значениям показателей тех. Режима | HDP Nifi | Kafka Pi\_converted\_values | Rest сервис | Json | 1С «Нормы и отчеты» |
| 10 | Отправка данных по значениям показателей качества | HDP Nifi | Kafka Lims\_input | Rest сервис | Json | 1С «Нормы и отчеты» |
| 11 | Отправка по нормативам | HDP Nifi | Pi\_requirements List\_requirements | Rest сервис | Json | 1С «Нормы и отчеты» |

## Реестр процессоров разработанного решения

| **Название процессора** | **Предназначение** |
| --- | --- |
| [ListFile](https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-standard-nar/1.4.0/org.apache.nifi.processors.standard.ListFTP/index.html) | **Описание:** Составляет список файлов, находящихся в указанной папке локальной файловой системы ноды NiFi. Для каждого файла будет. создан новый файл потока с атрибутом «название файла» заполненным соответственно названию файла в папке. Этот процессор может быть использован совместно с процессором FetchFile для последующей загрузки этих файлов  **Вход:** Отсутствует. Является инициирующим процессором.  **Выход:** Flowfile’ы, количество которых соответствует количеству файлов в заданной папке локальной файловой системы, названия которых соответствуют заданной маске. Контент пустой, в атрибут filename сохранено имя файла. |
| GenerateFlowFile | **Описание:** Создает пустой Flowfile с заданной периодичностью.  **Вход:** Отсутствует. Является инициирующим процессором.  **Выход:** Пустой Flowfile |
| [RouteOnAttribute](https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-standard-nar/1.4.0/org.apache.nifi.processors.standard.RouteOnAttribute/index.html) | **Описание:** Распределяет файлы потока на основании их атрибутов используя Attribute Expression Language  **Вход:** Flowfile с сохраненными атрибутами, на основе которых будет производиться распределение.  **Выход:** Предполагается несколько выходов с указанием правила, в случае выполнения которого,Flowfile направляется по одному из них |
| ExecuteSQL | **Описание:** Получает результат выполнения запроса select к базе данных. Результат формируется в формате avro  **Вход:** Любой Flowfile  **Выход:** Flowfile содержащий в контенте результат выполнения запроса select в формате avro |
| InvokeHTTP | **Описание:** Получает результат выполнения HTTP-запроса к удаленному веб-сервису.  **Вход:** Flowfile, контент которого будет передан web-сервису  **Выход:** Flowfile содержащий в контенте ответ web-сервиса |
| ListenHTTP | **Описание:** Получает данные, отправленные методом POST на хост ноды NiFi с указанными в настройках процессора значений порта и директории  **Вход:** Отсутствует. Является инициирующим процессором.  **Выход:** Flowfile содержащий в контенте сообщение, отправленное на адрес ноды NiFi с указанными значением порта |
| [FetchFile](https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-standard-nar/1.4.0/org.apache.nifi.processors.standard.FetchFTP/index.html) | **Описание:** Получает данные из файла в указанной папке локальной файловой системы ноды NiFi и перезаписывает содержимое входящего файла поток с содержимым файла  **Вход:** Flowfile содержащий атрибут хранящий имя и путь к запрашиваемому файлу локальной файловой системы  **Выход:** Flowfile содержащий в контенте текст запрошенного файла |
| [ReplaceText](https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-standard-nar/1.4.0/org.apache.nifi.processors.standard.ReplaceText/index.html) | **Описание:** Обновляет содержимое файла потока сравнивая с регулярным выражением (regex) и заменяет секцию данных, которые соответствуют регулярному выражению каким-либо альтернативным значением  **Вход:** Любой Flowfile  **Выход:** Flowfile, контент которого изменен в соответствии с правилом |
| [MergeContent](https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-standard-nar/1.4.0/org.apache.nifi.processors.standard.MergeContent/index.html) | **Описание:** Соединяет группы файлов потока вместе на основе стратегии, определенной пользователем в один файл потока. Рекомендуется использовать только с одним входящим потоком  **Вход:** Несколько Flowfile’ов  **Выход:** Один Flowfile, содержащий в себе объединение поданных на вход Flowfile’ов через заданный в процессоре разделитель. |
| [PutHDFS](https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-hadoop-nar/1.4.0/org.apache.nifi.processors.hadoop.PutHDFS/index.html) | **Описание:** Загружает данные файлов потока в HDFS  **Вход:** Flowfile, атрибут filename которого содержит имя файла для создания в HDFS  **Выход:** В HDFS создан файл, имя которого принимает значение атрибутаfilename, а содержимое соответствует контенту Flowfile’а |
| [UpdateAttribute](https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-update-attribute-nar/1.4.0/org.apache.nifi.processors.attributes.UpdateAttribute/index.html) | **Описание:** Обновляет атрибуты файла потока на основании Attribute Expression Language и/или удаляет атрибуты на основании регулярного выражения  **Вход:** Любой Flowfile  **Выход:** Flowfile, атрибуты которого изменены в соответствии с настройками процессора |
| PublishKafka | **Описание:** Отправляет содержимое контента поданного на вход Flowfile в указанный в настройках процессора топик Kafka.  **Вход:** Любой Flowfile  **Выход:** В указанном топике Kafka появилось сообщение повторяющее содержимое контента поданного на входFlowfile |
| SplitJSON | **Описание:** Разделяет поданный на вход Flowfile на несколько, содержащие части исходного json в соответствии с заданным правилом.  **Вход:** Flowfile содержащий json  **Выход:** Flowfile’ы содержащие части поданного на вход json |
| PutSQL | **Описание:** Добавляет строки в HBase на основании содержащейся в контенте команды phoenix.  **Вход:** Flowfile содержащий команду phoenix  **Выход:** В указанную в команде таблицу HBase добавлены указанные значения. |
| EvaluateJsonPath | **Описание:** Сохраняет в атрибуты или контент Flowfile значение полей в соответствии с заданными в настройках процессора правилами.  **Вход:** Flowfile содержащий json  **Выход:** Flowfile, в атрибуты которого добавлены значения указанных в настройках процессора поля |
| ConvertAvroToJSON | **Описание:** Преобразует исходный AVRO-файл в JSON.  **Вход:** Flowfile содержащий avro-файл  **Выход:** Flowfile содержащий json на основе поданного на вход файла |
| JoltTransformJSON | **Описание:** Преобразует исходный JSON-файл в соответствии с правилами JOLT-преобразований по указанной в настройках процессора схеме.  **Вход:** Flowfile содержащий json  **Выход:** Flowfile содержащий преобразованный json |
| ExecuteScript | **Описание:** Выполняет указанный в настройках процессора скрипт на одном из языков программирования: Clojure, ECMAScript, Groovy, Lua, python, ruby  **Вход:** Любой Flowfile  **Выход:** В соответствии с введенным скриптом. |

## Описание процесса «Отправка данных по событиям (показатели качества)» и «Отправка данных по событиям (показатели технологического режима)»

Процесс экспорта производится посредством http-запроса с методом POST. Запрос представляет из себя json, содержащий наименование экспортируемого метода, период времен, за который производится экспорт, код завода и опционально список тэгов для выгрузки.

Адрес Rest сервиса

для PI (показатели тех.режима) <https://spb25-cce-mo1.gazprom-neft.local/BLPS_MO/hs/api/HDP_Deviations_PI_Input>

для LIMS (показатели качества) https://spb25-cce-mo1.gazprom-neft.local/BLPS\_MO/hs/api/HDP\_Deviations\_LIMS\_Input

Метод: Post

Структура сообщения на вход

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Обязательность | Комментарий |
| date\_start | string | Да | Дата начала периода |
| date\_end | string | Да | Дата окончания периода |
| refinery\_id | integer | Да | Код завода (1 - онпз, 2 - мнпз) |
| tags | string | Нет | Имя тэга |

Примеры сообщения:

{"date\_start":"2018-06-16 00:00:00",

"date\_end":"2018-06-18 12:00:00",

"refinery\_id":1,

"tags":[

"19\_3:DK20:Rol15",

"19\_3:EM:QM"]}

{"date\_start":"2018-06-01 00:00:00",

"date\_end":"2018-06-05 12:00:00",

"refinery\_id":2}

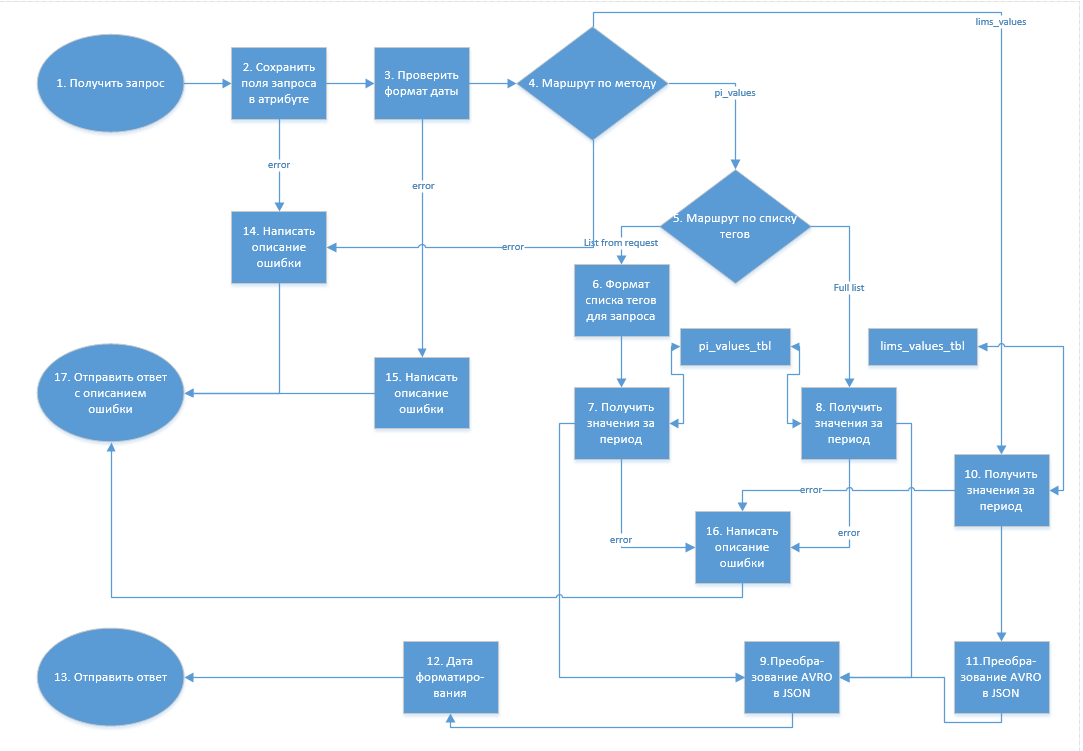


Схема 2. Процесс «Отправка данных по событиям»

Пояснение к схеме 2.

| **№** | **Шаг схемы** | **Название процессора** | **Предназначение** |
| --- | --- | --- | --- |
| 1. | Получить запрос | HandleHttpRequest | Процессор получает запрос, направленный на адрес, в котором в качестве порта используется зафиксированное в настройках процессора значение, а в качестве хоста используется хост NiFi ноды, на которой развернут процессор. На основе каждого полученного сообщения формируется flowFile.  FlowFile передается на шаг 2. |
| 2. | Сохранить поля запроса в атрибуте | EvaluateJsonPath | Значение полей переданного сообщения сохраняются в атрибуты flowFile.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки.  В ином случае передается на шаг 3. |
| 3. | Проверить формат даты | RouteOnAttribute | Производится проверка соответствия полей содержащих значения начала и конца запрашиваемого периода формату даты yyyy-MM-dd HH:mm:ss. В случае корректных значений flowFile передается на шаг 4.  В случае возникновения ошибки flowFile передается на шаг 15 для фиксации ошибки. |
| 4. | Маршрут по методу | RouteOnAttribute | На основе значения запрашиваемого метода, сохраненного на шаге 2, происходит разделение на потоки, соответствующие значениям показателей качества и показателей технологического режима на основании значения поля method.  В случае, если method принимает значение pi\_values flowFile передается на шаг 5 для передачи значений показателей качества.  В случае, если method принимает значение lims\_values flowFile передается на шаг 10 для передачи значений показателей технологического режима.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки. |
| 5. | Маршрут по списку тегов | RouteOnAttribute | На основе значения запрашиваемых тэгов, сохраненных на шаге 2, производится проверка – в случае заполненного необязательного поля tags flowFile передается на шаг 6 формирования выгрузки значений по ограниченному набору тэгов, в случае отсутствия поля tags в полученном сообщении flowFile передается на шаг 8 для выгрузки всех доступных тэгов за период. |
| 6. | Формат списка тегов для запроса | UpdateAttribute | Теги оформляются как список для вставки в запрос.  FlowFile передается на шаг 7. |
| 7. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице pi\_value\_tbl для получения значений тэгов из списка сформированного на шаге 6 полученных за запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 9.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 8. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице pi\_value\_tbl для получения всех актуальных тэгов, соответствующих показателям технологического режима по заводу, и соответствующих им значений, полученных за запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 9.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 9. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированные на шаге 7, 8 и 9 в json.  FlowFile передается на шаг 12. |
| 10. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице lims\_value\_tbl для получения списка всех актуальных показателей качества по заводу, и соответствующих им значений, полученных за запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 11.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 11. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированный на предыдущем шаге в json.  FlowFile передается на шаг 12. |
| 12. | Дата форматирования | ReplaceText | Значения типа Date приводятся к формату yyyy-MM-dd’T’HH:mm:ss’Z’ для стандартизации формата хранения дат на стороне потребителя.  FlowFile передается на шаг 13. |
| 13. | Отправить ответ | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – сформированная выгрузка в формате json, код выполнения – 200.  Обработка завершена. |
| 14. | Написать описание ошибки |  | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong format”}  FlowFile передается на шаг 17. |
| 15. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong date format”}  FlowFile передается на шаг 17. |
| 16. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Selection error”}  FlowFile передается на шаг 17. |
| 17. | Отправить ответ с описанием ошибки | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – текст ошибки в формате json, код выполнения – 400.  Обработка завершена. |

## Описание процесса экспорта данных по событиям

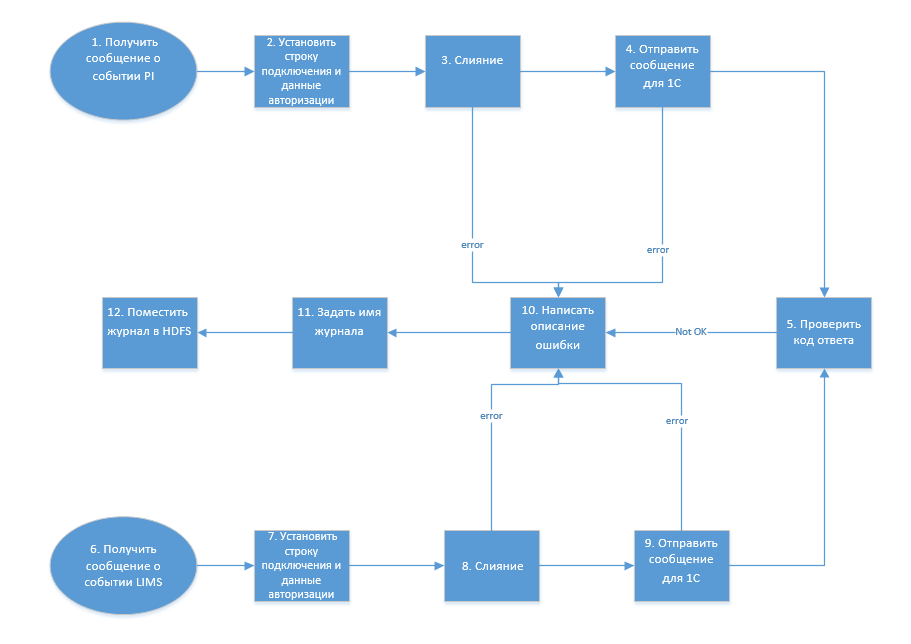


Схема 3. Процесс «Отправка данных по событиям»

Пояснение к схеме 3.

| **№** | **Шаг схемы** | **Название процессора** | **Предназначение** |
| --- | --- | --- | --- |
| 1. | Получить сообщение о событии PI | ConsumeKafka | Процессор получает сообщения от обработчика через топик Kafka pi\_range\_output\_tbc. Сообщения представляют из себя json, содержащий информацию о коде произошедшего события, наименования показателя, по которому произошло отклонение, дата начала и окончания события и максимальное значение отклонения в рамках периода. На основе полученного сообщения формируется flowFile.  FlowFile передается на шаг 2. |
| 2. | Установить строку подключения и данные авторизации | UpdateAttribute | Процессор сохраняет в атрибуты flowFile’а адрес веб-сервиса, на который отправляются значения данных по событиям, и данные для авторизации.  FlowFile передается на шаг 3. |
| 3. | Слияние | MergeContent | Процессор собирает все flowFile’ы пришедшие в течение половины минуты в один, для дальнейшей отправки. Процессор необходим для сокращения нагрузки на веб-сервис приемник при обработке большого количества исходящих отклонений. После успешного слияния flowFile передается на шаг 4, в случае возникновения ошибки выполнения flowFile передается на шаг 10 для фиксации ошибки. |
| 4. | Отправить сообщение для 1C | ExecuteStreamCommand | Процессор исполняет sh-команду (в данном случае Python-скрипт), в ходе которой производится отправка содержимого flowFile как http-запрос POST методом по сохраненному на шаге 2 адресу. В ходе выполнения производится ntlm- аутентификация и установка SSL соединения. Статус ответа сервиса сохраняется в атрибуты исходящего flowFile.  В случае возникновения ошибки выполнения flowFile передается на шаг 10 для фиксации ошибки.  В ином случае flowFile передается на шаг 5. |
| 5. | Проверить код ответа | RouteOnAttribute | Производится проверка значения кода ответа сервиса. В случае получения кода ответа отличного от 200, flowFile передается на шаг 10 для фиксации ошибки.  В ином случае обработка завершена. |
| 6. | Получить сообщение о событии LIMS | ConsumeKafka | Процессор получает сообщения от обработчика через топик Kafka lims\_range\_output\_tbc. Сообщения представляют из себя json, содержащий информацию о коде произошедшего события, наименования показателя, по которому произошло отклонение, дата начала и окончания события и максимальное значение отклонения в рамках периода. На основе полученного сообщения формируется flowFile.  FlowFile передается на шаг 7. |
| 7. | Установить строку подключения и данные авторизации | UpdateAttribute | Процессор сохраняет в атрибуты flowFile’а адрес веб-сервиса, на который отправляются значения данных по событиям, и данные для авторизации.  FlowFile передается на шаг 8. |
| 8. | Слияние | MergeContent | Процессор собирает все flowFile’ы пришедшие в течение половины минуты в один, для дальнейшей отправки. Процессор необходим для сокращения нагрузки на веб-сервис приемник при обработке большого количества исходящих отклонений.  В случае возникновения ошибки выполнения flowFile передается на шаг 10 для фиксации ошибки.  В ином случае передается на шаг 9. |
| 9. | Отправить сообщение для 1C | ExecuteStreamCommand | Процессор исполняет sh-команду (в данном случае Python-скрипт), в ходе которой производится отправка содержимого flowFile как http-запрос POST методом по сохраненному на шаге 7 адресу. В ходе выполнения производится ntlm- аутентификация и установка SSL соединения. Статус ответа сервиса сохраняется в атрибуты исходящего flowFile.  FlowFile передается на шаг 10. |
| 10. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  [Date] [Logging level] [Error code] – message  В случае перехода с шага 3, 4, 5, 8, 9 это будет сообщение:  [%Date%] [ERROR] [Export error] – Can’t export MO values  FlowFile передается на шаг 11. |
| 11. | Задать имя журнала | UpdateAttribute | Процессор задает имя файла в HDFS, в котором хранится лог системы МО.  FlowFile передается на шаг 12. |
| 12. | Поместить журнал в HDFS | PutHDFS | Процессор сохраняет содержимое контента в файле в HDFS, имя которого задано на шаге 11. В случае отсутствия файла, он будет создан. При наличии – строка будет добавлена в конец файла.  Обработка завершена. |

## Описание процесса экспорта данных по статистическим показателям

Процесс экспорта производится посредством http-запроса с методом POST. Запрос представляет из себя json, содержащий наименование экспортируемого метода, период времен, за который производится экспорт, код завода и опционально список тэгов для выгрузки.

Адрес для подключения к Rest сервису  [http://spb99-t-hdpnf1.gazprom-neft.local:9590/](http://spb25-nifi-t-1.gazprom-neft.local:9590/)

Метод: post

Структура сообщения на вход

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Обязательность | Комментарий |
| date\_start | string | да | Дата начала периода |
| date\_end | string | да | Дата окончания периода |
| refinery\_id | integer | да | Код завода (1 - онпз, 2 - мнпз) |
| tags | string | нет | Имя тэга |
| method | string | да | Функция (pi\_metrics\_statistics\_export и lims\_metrics\_statistics\_export) |

Примеры сообщения:

{"date\_start":"2018-06-16 00:00:00",

"date\_end":"2018-06-18 12:00:00",

"refinery\_id":1,

"tags":[

"19\_3:DK20:Rol15",

"19\_3:EM:QM"],

"method":"pi\_metrics\_statistics\_export"}

{"date\_start":"2018-06-01 00:00:00",

"date\_end":"2018-06-05 12:00:00",

"refinery\_id":2,

"method":"lims\_metrics\_statistics\_export"}

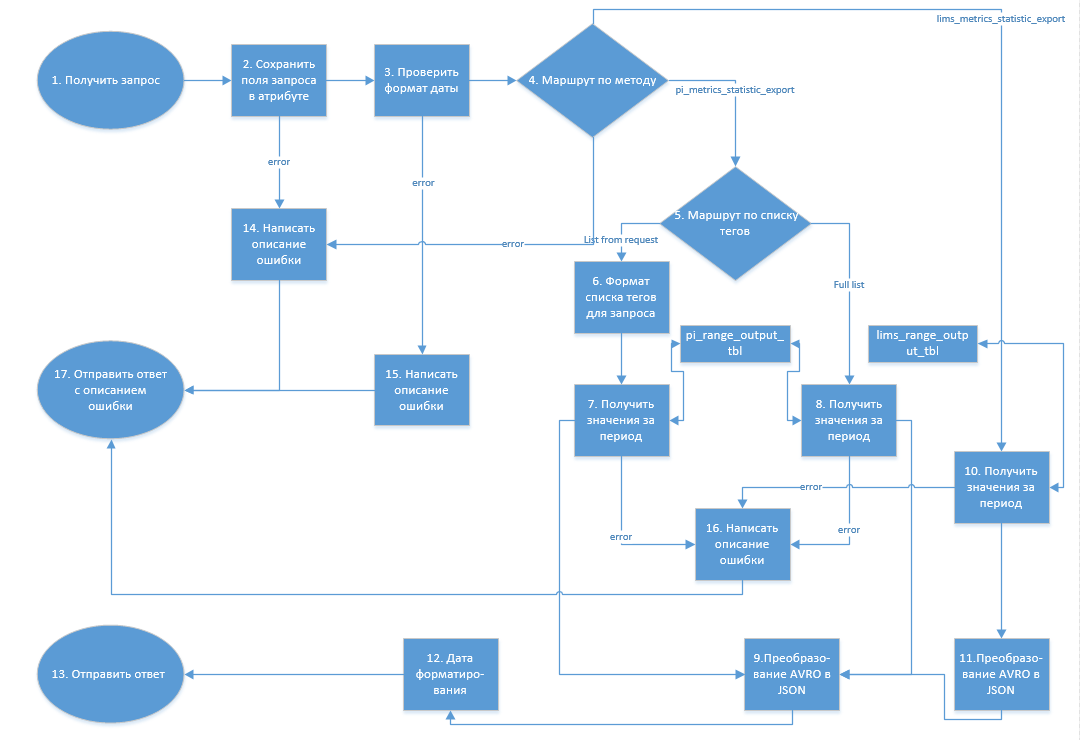


Схема 4. Процесс «Отправка статистики по событиям»

Пояснение к схеме 4.

| **№** | **Шаг схемы** | **Название процессора** | **Предназначение** |
| --- | --- | --- | --- |
| 1. | Получить запрос | HandleHttpRequest | Процессор получает запрос, направленный на адрес, в котором в качестве порта используется зафиксированное в настройках процессора значение, а в качестве хоста используется хост NiFi ноды, на которой развернут процессор. На основе каждого полученного сообщения формируется flowFile. Отправка данных осуществляется по защищенному SSL соединению. В качестве SSLContextService используется SSL keystore service for SMOTR (тип трастстора: JKS, протокол: TLS).  FlowFile передается на шаг 2. |
| 2. | Сохранить поля запроса в атрибуте | EvaluateJsonPath | Значение полей переданного сообщения сохраняются в атрибуты flowFile.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки.  В ином случае передается на шаг 3. |
| 3. | Проверить формат даты | RouteOnAttribute | Производится проверка соответствия полей содержащих значения начала и конца запрашиваемого периода формату даты yyyy-MM-dd HH:mm:ss. В случае корректных значений flowFile передается на шаг 4.  В случае возникновения ошибки flowFile передается на шаг 15 для фиксации ошибки. |
| 4. | Маршрут по методу | RouteOnAttribute | На основе значения запрашиваемого метода, сохраненного на шаге 2, происходит разделение на потоки, соответствующие запросам на получение статистики по показателем качества и показателям технологического режима на основании значения поля method.  В случае, если method принимает значение pi\_metrics\_statistic\_export flowFile передается на шаг 5 для передачи значений показателей качества.  В случае, если method принимает значение lims\_metrics\_statistic\_export flowFile передается на шаг 10 для передачи значений показателей технологического режима.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки. |
| 5. | Маршрут по списку тегов | RouteOnAttribute | На основе значения запрашиваемых тэгов, сохраненных на шаге 2, производится проверка – в случае заполненного необязательного поля tags flowFile передается на шаг 6 формирования выгрузки значений по ограниченному набору тэгов, в случае отсутствия поля tags в полученном сообщении flowFile передается на шаг 8 для выгрузки всех доступных тэгов за период. |
| 6. | Формат списка тегов для запроса | UpdateAttribute | Теги оформляются как список для вставки в запрос.  FlowFile передается на шаг 7. |
| 7. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице pi\_range\_output\_tbl для получения статистики по тэгам из списка сформированного на шаге 6 полученным за запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 9.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 8. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице pi\_range\_output\_tbl для получения всех актуальных тэгов, соответствующих показателям технологического режима по заводу, и соответствующих им статистики, полученной за запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 9.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 9. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированные на шаге 7, 8 и 9 в json.  FlowFile передается на шаг 12. |
| 10. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице lims\_range\_output\_tbl для получения списка всех актуальных показателей качества по заводу, и соответствующих им статистики, полученной запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 11.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 11. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированный на предыдущем шаге в json.  FlowFile передается на шаг 12. |
| 12. | Дата форматиро-вания | ReplaceText | Значения типа Date приводятся к формату yyyy-MM-dd’T’HH:mm:ss’Z’ для стандартизации формата хранения дат на стороне потребителя.  FlowFile передается на шаг 13. |
| 13. | Отправить ответ | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – сформированная выгрузка в формате json, код выполнения – 200.  Обработка завершена. |
| 14. | Написать описание ошибки |  | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong format”}  FlowFile передается на шаг 17. |
| 15. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong date format”}  FlowFile передается на шаг 17. |
| 16. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Selection error”}  FlowFile передается на шаг 17. |
| 17. | Отправить ответ с описанием ошибки | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – текст ошибки в формате json, код выполнения – 400.  Обработка завершена. |

## Описание процесса экспорта данных в таблицу по событиям (PI)

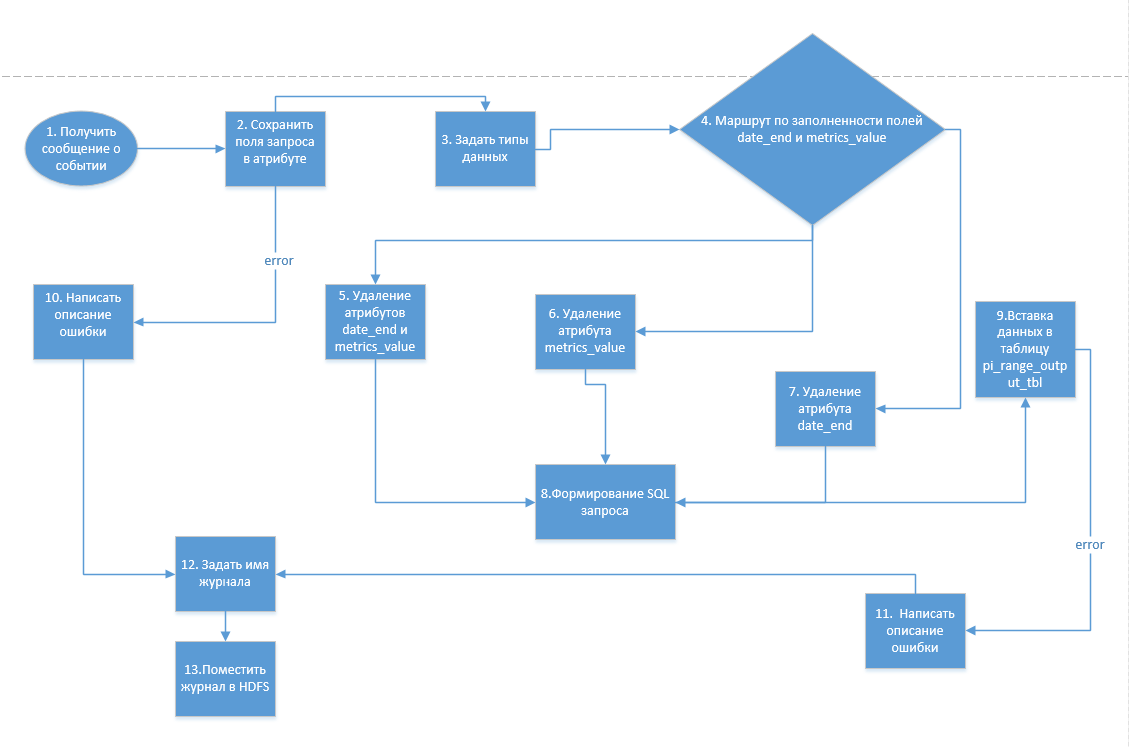


Схема 5. Процесс «Отправка данных в таблицу по событиям»

Пояснение к схеме 3.

| **№** | **Шаг схемы** | **Название процессора** | **Предназначение** |
| --- | --- | --- | --- |
| 1. | Получить сообщение о событии PI | ConsumeKafka | Процессор получает сообщения от обработчика через топик Kafka pi\_range\_output\_tbc. Сообщения представляют из себя json, содержащий информацию о коде произошедшего события, наименования показателя, по которому произошло отклонение, дата начала и окончания события и максимальное значение отклонения в рамках периода. На основе полученного сообщения формируется flowFile.  FlowFile передается на шаг 2. |
| 2. | Сохранить поля запроса в атрибуте | EvaluateJsonPath | Значение полей переданного сообщения сохраняются в атрибуты flowFile.  В случае возникновения ошибки flowFile передается на шаг 10 для фиксации ошибки.  В ином случае передается на шаг 3. |
| 3. | Задать типы данных | UpdateAttribute | Процессор задает каждому атрибуту, сохраненному на шаге 2, тип данных в соответствии с конвенцией jdbc.  FlowFile передается на шаг 4. |
| 4. | Маршрут по заполненности полей date\_end и metrics\_value | RouteOnAttribute | На основе значений (null/not null) полей date\_end и metrics\_value происходит разделение на потоки.  В случае если отсутствует значение и date\_end, и metrics\_value, flowFile передается на шаг 5.  В случае если отсутствует значение metrics\_value, flowFile передается на шаг 6.  В случае если отсутствует значение date\_end, flowFile передается на шаг 7.  В ином случае flowFile передается на шаг 8. |
| 5. | Удаление атрибутов date\_end и metrics\_value | UpdateAttribute | В соответствии с регулярным выражением удаляются атрибуты date\_end и metrics\_value . FlowFile передается на шаг 8. |
| 6. | Удаление атрибута metrics\_value | UpdateAttribute | В соответствии с регулярным выражением удаляется атрибут metrics\_value . FlowFile передается на шаг 8. |
| 7. | Удаление атрибута date\_end | UpdateAttribute | В соответствии с регулярным выражением удаляется атрибут date\_end. FlowFile передается на шаг 8. |
| 8. | Формирование SQL запроса | ReplaceText | Формируется запрос на вставку в таблицу pi\_range\_output\_tbl. FlowFile передается на шаг 9. |
| 9. | Вставка данных в таблицу pi\_range\_output\_tbl | PutSQL | Выполняется запрос на вставку в таблицу pi\_range\_output\_tbl (в батч режиме).  В случае ошибки flowFile передается на шаг 11. |
| 10. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  [Date] [Logging level] [Error code] – message  В случае перехода с шага 2 это будет сообщение:  [%Date%] ] [HBase error] - Can't get list of tags  FlowFile передается на шаг 12. |
| 11. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  [Date] [Logging level] [Error code] – message  В случае перехода с шага 9 это будет сообщение:  [%Date%] ] [HBase error] - Can't get list of tags  FlowFile передается на шаг 12. |
| 12. | Задать имя журнала | UpdateAttribute | Процессор задает имя файла в HDFS, в котором хранится лог системы.  FlowFile передается на шаг 13. |
| 13. | Поместить журнал в HDFS | PutHDFS | Процессор сохраняет содержимое контента в файле в HDFS, имя которого задано на шаге 12. В случае отсутствия файла, он будет создан. При наличии – строка будет добавлена в конец файла.  Обработка завершена. |

## Описание процесса экспорта данных в таблицу по событиям (Lims)

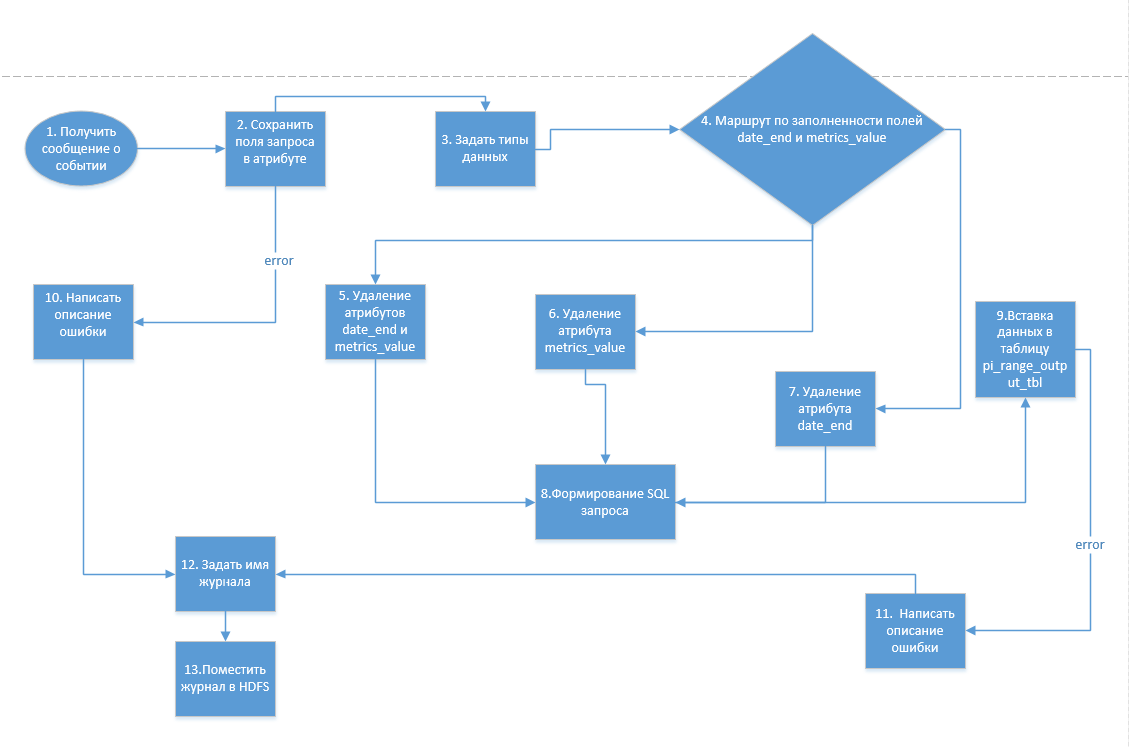


Схема 6. Процесс «Отправка данных в таблицу по событиям»

Пояснение к схеме 3.

| **№** | **Шаг схемы** | **Название процессора** | **Предназначение** |
| --- | --- | --- | --- |
| 1. | Получить сообщение о событии Lims | ConsumeKafka | Процессор получает сообщения от обработчика через топик Kafka lims\_range\_output\_tbc. Сообщения представляют из себя json, содержащий информацию о коде произошедшего события, наименования показателя, по которому произошло отклонение, дата начала и окончания события и максимальное значение отклонения в рамках периода. На основе полученного сообщения формируется flowFile.  FlowFile передается на шаг 2. |
| 2. | Сохранить поля запроса в атрибуте | EvaluateJsonPath | Значение полей переданного сообщения сохраняются в атрибуты flowFile.  В случае возникновения ошибки flowFile передается на шаг 10 для фиксации ошибки.  В ином случае передается на шаг 3. |
| 3. | Задать типы данных | UpdateAttribute | Процессор задает каждому атрибуту, сохраненному на шаге 2, тип данных в соответствии с конвенцией jdbc.  FlowFile передается на шаг 4. |
| 4. | Маршрут по заполненности полей date\_end и metrics\_value | RouteOnAttribute | На основе значений (null/not null) полей date\_end и metrics\_value происходит разделение на потоки.  В случае если отсутствует значение и date\_end, и metrics\_value, flowFile передается на шаг 5.  В случае если отсутствует значение metrics\_value, flowFile передается на шаг 6.  В случае если отсутствует значение date\_end, flowFile передается на шаг 7.  В ином случае flowFile передается на шаг 8. |
| 5. | Удаление атрибутов date\_end и metrics\_value | UpdateAttribute | В соответствии с регулярным выражением удаляются атрибуты date\_end и metrics\_value . FlowFile передается на шаг 8. |
| 6. | Удаление атрибута metrics\_value | UpdateAttribute | В соответствии с регулярным выражением удаляется атрибут metrics\_value . FlowFile передается на шаг 8. |
| 7. | Удаление атрибута date\_end | UpdateAttribute | В соответствии с регулярным выражением удаляется атрибут date\_end. FlowFile передается на шаг 8. |
| 8. | Формирование SQL запроса | ReplaceText | Формируется запрос на вставку в таблицу lims\_range\_output\_tbl. FlowFile передается на шаг 9. |
| 9. | Вставка данных в таблицу lims\_range\_output\_tbl | PutSQL | Выполняется запрос на вставку в таблицу lims\_range\_output\_tbl (в батч режиме).  В случае ошибки flowFile передается на шаг 11. |
| 10. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  [Date] [Logging level] [Error code] – message  В случае перехода с шага 2 это будет сообщение:  [%Date%] ] [HBase error] - Can't get list of tags  FlowFile передается на шаг 12. |
| 11. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  [Date] [Logging level] [Error code] – message  В случае перехода с шага 9 это будет сообщение:  [%Date%] ] [HBase error] - Can't get list of tags  FlowFile передается на шаг 12. |
| 12. | Задать имя журнала | UpdateAttribute | Процессор задает имя файла в HDFS, в котором хранится лог системы.  FlowFile передается на шаг 13. |
| 13. | Поместить журнал в HDFS | PutHDFS | Процессор сохраняет содержимое контента в файле в HDFS, имя которого задано на шаге 12. В случае отсутствия файла, он будет создан. При наличии – строка будет добавлена в конец файла.  Обработка завершена. |

## Описание процесса экспорта данных по детализированной статистике

Процесс экспорта производится посредством http-запроса с методом POST. Запрос представляет из себя json, содержащий наименование экспортируемого метода, период времен, за который производится экспорт, код завода и опционально список тэгов для выгрузки.

Адрес для подключения к Rest сервису <http://spb99-t-hdpnf1.gazprom-neft.local:9591/>

Метод: post

Структура сообщения на вход

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Обязательность | Комментарий |
| date\_start | string | да | Дата начала периода |
| date\_end | string | да | Дата окончания периода |
| refinery\_id | integer | да | Код завода (1 - онпз, 2 - мнпз) |
| tags | string | нет | Имя тэга |
| method | string | да | Функция (pi\_metrics\_statistics\_export и lims\_metrics\_statistics\_export) |

Примеры сообщения:

{"date\_start":"2018-06-16 00:00:00",

"date\_end":"2018-06-18 12:00:00",

"refinery\_id":1,

"tags":[

"19\_3:DK20:Rol15",

"19\_3:EM:QM"],

"method":"pi\_metrics\_statistics\_export"}

{"date\_start":"2018-06-01 00:00:00",

"date\_end":"2018-06-05 12:00:00",

"refinery\_id":2,

"method":"lims\_metrics\_statistics\_export"}

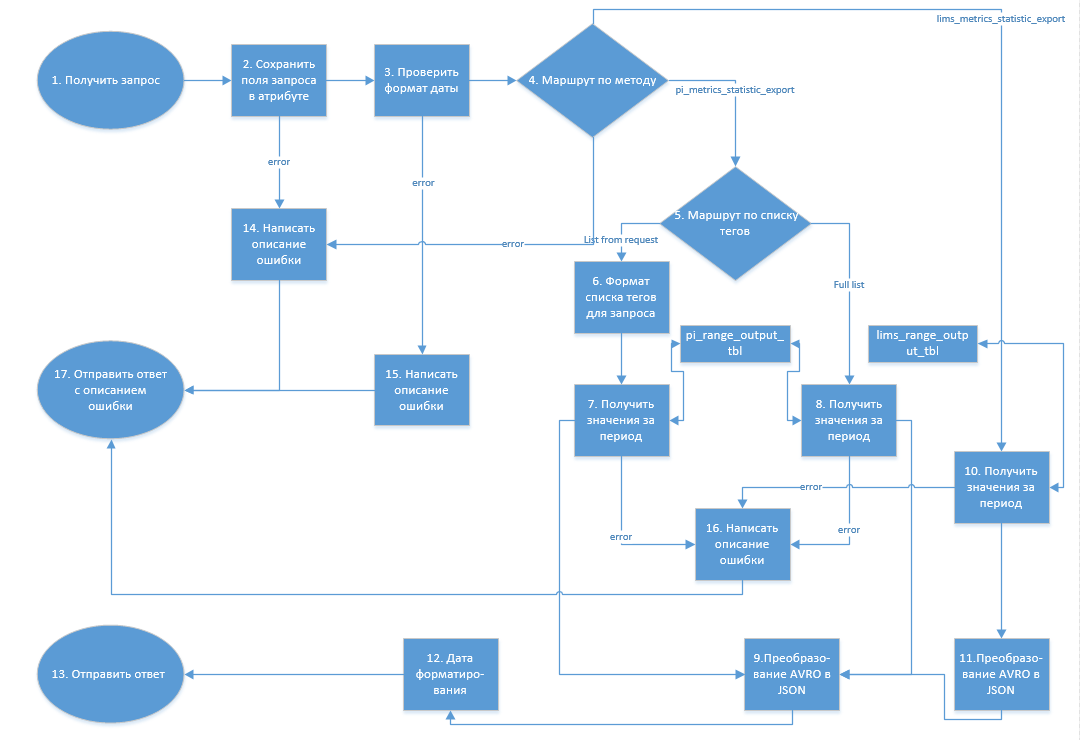


Схема 7. Процесс «Отправка детализированной статистики по событиям»

Пояснение к схеме 7.

| **№** | **Шаг схемы** | **Название процессора** | **Предназначение** |
| --- | --- | --- | --- |
| 1. | Получить запрос | HandleHttpRequest | Процессор получает запрос, направленный на адрес, в котором в качестве порта используется зафиксированное в настройках процессора значение, а в качестве хоста используется хост NiFi ноды, на которой развернут процессор. На основе каждого полученного сообщения формируется flowFile. Отправка данных осуществляется по защищенному SSL соединению. В качестве SSLContextService используется SSL keystore service for SMOTR (тип трастстора: JKS, протокол: TLS).  FlowFile передается на шаг 2. |
| 2. | Сохранить поля запроса в атрибуте | EvaluateJsonPath | Значение полей переданного сообщения сохраняются в атрибуты flowFile.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки.  В ином случае передается на шаг 3. |
| 3. | Проверить формат даты | RouteOnAttribute | Производится проверка соответствия полей содержащих значения начала и конца запрашиваемого периода формату даты yyyy-MM-dd HH:mm:ss. В случае корректных значений flowFile передается на шаг 4.  В случае возникновения ошибки flowFile передается на шаг 15 для фиксации ошибки. |
| 4. | Маршрут по методу | RouteOnAttribute | На основе значения запрашиваемого метода, сохраненного на шаге 2, происходит разделение на потоки, соответствующие запросам на получение статистики по показателем качества и показателям технологического режима на основании значения поля method.  В случае, если method принимает значение pi\_metrics\_statistic\_export flowFile передается на шаг 5 для передачи значений показателей качества.  В случае, если method принимает значение lims\_metrics\_statistic\_export flowFile передается на шаг 10 для передачи значений показателей технологического режима.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки. |
| 5. | Маршрут по списку тегов | RouteOnAttribute | На основе значения запрашиваемых тэгов, сохраненных на шаге 2, производится проверка – в случае заполненного необязательного поля tags flowFile передается на шаг 6 формирования выгрузки значений по ограниченному набору тэгов, в случае отсутствия поля tags в полученном сообщении flowFile передается на шаг 8 для выгрузки всех доступных тэгов за период. |
| 6. | Формат списка тегов для запроса | UpdateAttribute | Теги оформляются как список для вставки в запрос.  FlowFile передается на шаг 7. |
| 7. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице pi\_range\_output\_tbl для получения статистики по тэгам из списка сформированного на шаге 6 полученным за запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 9.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 8. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице pi\_range\_output\_tbl для получения всех актуальных тэгов, соответствующих показателям технологического режима по заводу, и соответствующих им статистики, полученной за запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 9.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 9. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированные на шаге 7, 8 и 9 в json.  FlowFile передается на шаг 12. |
| 10. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице lims\_range\_output\_tbl для получения списка всех актуальных показателей качества по заводу, и соответствующих им статистики, полученной запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 11.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 11. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированный на предыдущем шаге в json.  FlowFile передается на шаг 12. |
| 12. | Дата форматиро-вания | ReplaceText | Значения типа Date приводятся к формату yyyy-MM-dd’T’HH:mm:ss’Z’ для стандартизации формата хранения дат на стороне потребителя.  FlowFile передается на шаг 13. |
| 13. | Отправить ответ | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – сформированная выгрузка в формате json, код выполнения – 200.  Обработка завершена. |
| 14. | Написать описание ошибки |  | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong format”}  FlowFile передается на шаг 17. |
| 15. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong date format”}  FlowFile передается на шаг 17. |
| 16. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Selection error”}  FlowFile передается на шаг 17. |
| 17. | Отправить ответ с описанием ошибки | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – текст ошибки в формате json, код выполнения – 400.  Обработка завершена. |

## Описание процесса экспорта данных по значениям показателей

Процесс экспорта производится посредством http-запроса с методом POST. Запрос представляет из себя json, содержащий наименование экспортируемого метода, период времен, за который производится экспорт, код завода и опционально список тэгов для выгрузки.

Адрес для подключения к Rest сервису <http://spb99-t-hdpnf1.gazprom-neft.local:9589/>

Метод: post

Структура сообщения на вход

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Обязательность | Комментарий |
| date\_start | string | да | Дата начала периода |
| date\_end | string | да | Дата окончания периода |
| refinery\_id | integer | да | Код завода (1 - онпз, 2 - мнпз) |
| tags | string | нет | Имя тэга |
| method | string | да | Функция (pi\_values и lims\_values) |

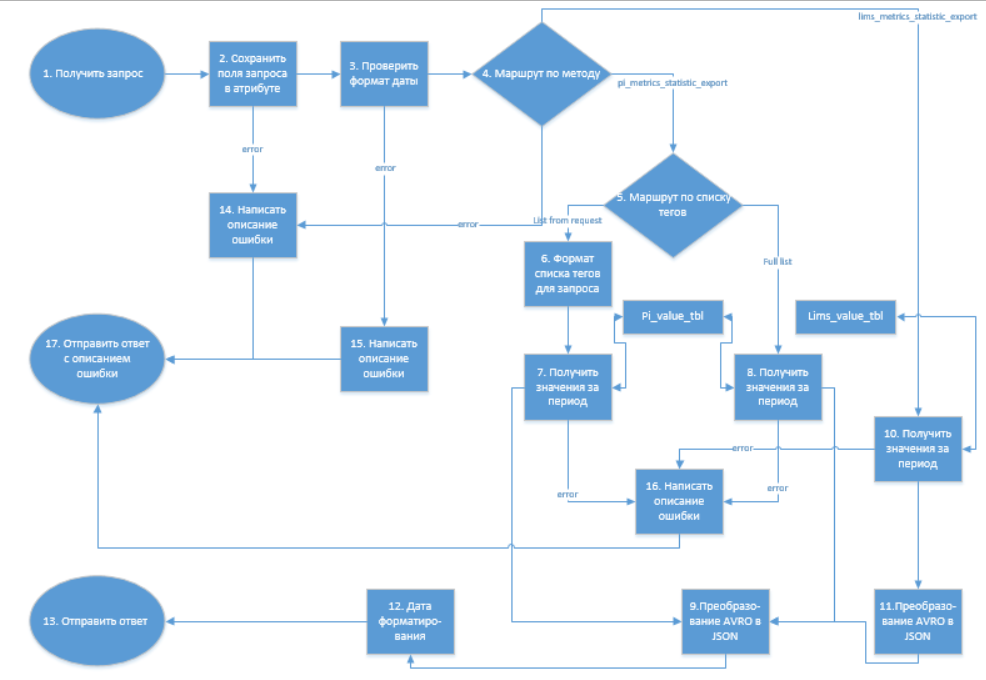


Схема 8. Процесс «Отправка значений показателей»

Пояснение к схеме 8.

| **№** | **Шаг схемы** | **Название процессора** | **Предназначение** |
| --- | --- | --- | --- |
| 1. | Получить запрос | HandleHttpRequest | Процессор получает запрос, направленный на адрес, в котором в качестве порта используется зафиксированное в настройках процессора значение, а в качестве хоста используется хост NiFi ноды, на которой развернут процессор. На основе каждого полученного сообщения формируется flowFile. Отправка данных осуществляется по защищенному SSL соединению. В качестве SSLContextService используется SSL keystore service for SMOTR (тип трастстора: JKS, протокол: TLS).  FlowFile передается на шаг 2. |
| 2. | Сохранить поля запроса в атрибуте | EvaluateJsonPath | Значение полей переданного сообщения сохраняются в атрибуты flowFile.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки.  В ином случае передается на шаг 3. |
| 3. | Проверить формат даты | RouteOnAttribute | Производится проверка соответствия полей содержащих значения начала и конца запрашиваемого периода формату даты yyyy-MM-dd HH:mm:ss. В случае корректных значений flowFile передается на шаг 4.  В случае возникновения ошибки flowFile передается на шаг 15 для фиксации ошибки. |
| 4. | Маршрут по методу | RouteOnAttribute | На основе значения запрашиваемого метода, сохраненного на шаге 2, происходит разделение на потоки, соответствующие запросам на получение статистики по показателем качества и показателям технологического режима на основании значения поля method.  В случае, если method принимает значение pi\_metrics\_statistic\_export flowFile передается на шаг 5 для передачи значений показателей качества.  В случае, если method принимает значение lims\_metrics\_statistic\_export flowFile передается на шаг 10 для передачи значений показателей технологического режима.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки. |
| 5. | Маршрут по списку тегов | RouteOnAttribute | На основе значения запрашиваемых тэгов, сохраненных на шаге 2, производится проверка – в случае заполненного необязательного поля tags flowFile передается на шаг 6 формирования выгрузки значений по ограниченному набору тэгов, в случае отсутствия поля tags в полученном сообщении flowFile передается на шаг 8 для выгрузки всех доступных тэгов за период. |
| 6. | Формат списка тегов для запроса | UpdateAttribute | Теги оформляются как список для вставки в запрос.  FlowFile передается на шаг 7. |
| 7. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице pi\_value\_tbl для получения статистики по тэгам из списка сформированного на шаге 6 полученным за запрашиваемый период. Значения получаются единым файлом в формате avro и передаются на шаг 9.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 8. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице pi\_value\_tbl для получения всех актуальных тэгов, соответствующих показателям технологического режима по заводу. Значения получаются единым файлом в формате avro и передаются на шаг 9.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 9. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированные на шаге 7, 8 и 9 в json.  FlowFile передается на шаг 12. |
| 10. | Получить значения за период | ExecuteSQL | Процессор инициирует запрос к таблице lims\_value\_tbl для получения списка всех актуальных показателей качества по заводу. Значения получаются единым файлом в формате avro и передаются на шаг 11.  В случае возникновения ошибок выполнения flowFile передается на шаг 16 для фиксирования ошибки. |
| 11. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированный на предыдущем шаге в json.  FlowFile передается на шаг 12. |
| 12. | Дата форматиро-вания | ReplaceText | Значения типа Date приводятся к формату yyyy-MM-dd’T’HH:mm:ss’Z’ для стандартизации формата хранения дат на стороне потребителя.  FlowFile передается на шаг 13. |
| 13. | Отправить ответ | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – сформированная выгрузка в формате json, код выполнения – 200.  Обработка завершена. |
| 14. | Написать описание ошибки |  | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong format”}  FlowFile передается на шаг 17. |
| 15. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong date format”}  FlowFile передается на шаг 17. |
| 16. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Selection error”}  FlowFile передается на шаг 17. |
| 17. | Отправить ответ с описанием ошибки | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – текст ошибки в формате json, код выполнения – 400.  Обработка завершена. |

## Описание процесса экспорта данных по нормативам

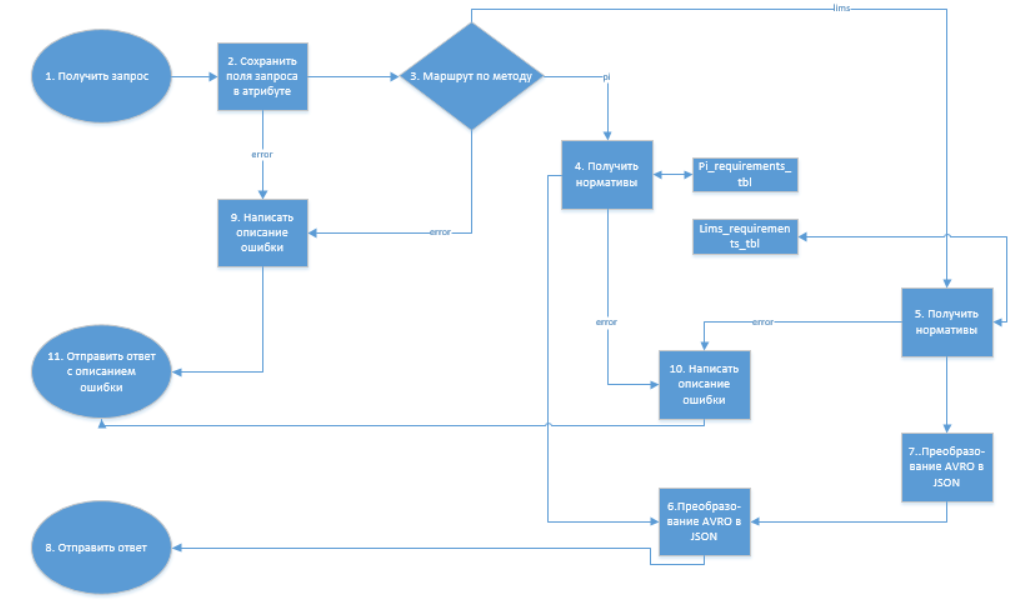


Схема 9. Процесс «Отправка значений показателей»

Пояснение к схеме 9.

| **№** | **Шаг схемы** | **Название процессора** | **Предназначение** |
| --- | --- | --- | --- |
| 1. | Получить запрос | HandleHttpRequest | Процессор получает запрос, направленный на адрес, в котором в качестве порта используется зафиксированное в настройках процессора значение, а в качестве хоста используется хост NiFi ноды, на которой развернут процессор. На основе каждого полученного сообщения формируется flowFile. Отправка данных осуществляется по защищенному SSL соединению. В качестве SSLContextService используется SSL keystore service for SMOTR (тип трастстора: JKS, протокол: TLS).  FlowFile передается на шаг 2. |
| 2. | Сохранить поля запроса в атрибуте | EvaluateJsonPath | Значение полей переданного сообщения сохраняются в атрибуты flowFile.  В случае возникновения ошибки flowFile передается на шаг 14 для фиксации ошибки.  В ином случае передается на шаг 3. |
| 3. | Маршрут по методу | RouteOnAttribute | На основе значения запрашиваемого метода, сохраненного на шаге 2, происходит разделение на потоки, соответствующие запросам на получение нормативов по показателем качества и показателям технологического режима на основании значения поля method.  В случае, если method принимает значение pi flowFile передается на шаг 4 для передачи значений показателей качества.  В случае, если method принимает значение lims flowFile передается на шаг 5 для передачи значений показателей технологического режима.  В случае возникновения ошибки flowFile передается на шаг 9 для фиксации ошибки. |
| 4. | Получить нормативы | ExecuteSQL | Процессор инициирует запрос к таблице pi\_requirements\_tbl для получения нормативов по тэгам. Значения получаются единым файлом в формате avro и передаются на шаг 5.  В случае возникновения ошибок выполнения flowFile передается на шаг 10 для фиксирования ошибки. |
| 5. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл в json. |
| 6. | Получить нормативы | ExecuteSQL | Процессор инициирует запрос к таблице lims\_requirements\_tbl для получения списка всех актуальных показателей качества по заводу. Значения получаются единым файлом в формате avro . |
| 7. | Преобразование AVRO в JSON | ConvertAvroToJSON | Процессор конвертирует avro-файл, сформированный на предыдущем шаге в json. |
| 8. | Отправить ответ | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – сформированная выгрузка в формате json, код выполнения – 200.  Обработка завершена. |
| 9. | Написать описание ошибки |  | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Wrong format”}  FlowFile передается на шаг 17. |
| 10. | Написать описание ошибки | ReplaceText | Процессор заменяет содержимое flowFile на строку об ошибке для вставки в файл логирования. Строка представляет собой запись вида:  {“error\_code”:””,”error\_description”:””}  В случае перехода с шага 3 это будет сообщение:  {“error\_code”:”400”,”error\_description”:”Selection error”}  FlowFile передается на шаг 17. |
| 11. | Отправить ответ с описанием ошибки | HandleHttpResponse | Процессор возвращает ответ на запрос, полученный на шаге 1. В качестве текста ответа – текст ошибки в формате json, код выполнения – 400.  Обработка завершена. |