

Описание прототипа инструмента сверки схем баз данных

Центр разработки PostgreSQL

Exported on 10/21/2020

Table of Contents

1	Общие сведения.....	8
2	Файлы.....	9
3	Предварительные требования.....	10
4	Установка.....	11
5	Описание модели сверки	12
6	Настройка правил сверки.....	13
7	API для запуска задач сверки	16
7.1	Сверка записей HASH.....	16
7.1.1	Формат	16
7.1.2	Параметры запроса	16
7.1.3	Тело запроса	16
7.1.4	Возвращаемое значение	17
7.1.5	Описание/Алгоритм	17
7.1.6	Пример.....	17
7.2	Инкрементальная сверка	17
7.2.1	Формат	18
7.2.2	Параметры запроса	18
7.2.3	Тело запроса	18
7.2.4	Возвращаемое значение	18
7.2.5	Описание/Алгоритм	18
7.2.6	Пример.....	19
7.3	Полная сверка по интервалам времени.....	19
7.3.1	Формат	19
7.3.2	Параметры запроса	19
7.3.3	Тело запроса	19
7.3.4	Возвращаемое значение	20
7.3.5	Описание/Алгоритм	20
7.3.6	Пример.....	20
7.4	Детальная сверка таблицы.....	21
7.4.1	Формат	21
7.4.2	Параметры запроса	21

7.4.3	Тело запроса	21
7.4.4	Возвращаемое значение	22
7.4.5	Описание/Алгоритм	23
7.4.6	Пример	23
8	REST API сверки	24
8.1	Источники данных (datasource)	24
8.1.1	Получить список источников данных	24
8.1.1.1	Формат	24
8.1.1.2	Возвращаемое значение	24
8.1.1.3	Описание/Алгоритм	24
8.1.1.4	Пример	24
8.1.2	Информация по источнику данных	24
8.1.2.1	Формат	24
8.1.2.2	Параметры запроса	24
8.1.2.3	Возвращаемое значение	25
8.1.2.4	Описание/Алгоритм	25
8.1.2.5	Пример	25
8.1.3	Добавление источника данных	25
8.1.3.1	Формат	25
8.1.3.2	Параметры запроса	25
8.1.3.3	Тело запроса	25
8.1.3.4	Возвращаемое значение	26
8.1.3.5	Описание/Алгоритм	26
8.1.3.6	Пример	26
8.1.4	Удаление источника данных	26
8.1.4.1	Формат	26
8.1.4.2	Параметры запроса	26
8.1.4.3	Возвращаемое значение	27
8.1.4.4	Описание/Алгоритм	27
8.1.4.5	Пример	27
8.1.5	Изменение источника данных	27
8.1.5.1	Формат	27
8.1.5.2	Параметры запроса	27
8.1.5.3	Тело запроса	27
8.1.5.4	Возвращаемое значение	28

8.1.5.5 Описание/Алгоритм	28
8.1.5.6 Пример	28
8.2 Схемы	28
8.2.1 Список схем	28
8.2.1.1 Формат	28
8.2.1.2 Параметры запроса	29
8.2.1.3 Возвращаемое значение	29
8.2.1.4 Описание/Алгоритм	29
8.2.1.5 Пример	29
8.2.2 Информация по схеме	29
8.2.2.1 Формат	29
8.2.2.2 Параметры запроса	29
8.2.2.3 Возвращаемое значение	29
8.2.2.4 Описание/Алгоритм	30
8.2.2.5 Пример	30
8.2.3 Добавление схемы	30
8.2.3.1 Формат	30
8.2.3.2 Параметры запроса	30
8.2.3.3 Тело запроса	30
8.2.3.4 Возвращаемое значение	31
8.2.3.5 Описание/Алгоритм	31
8.2.3.6 Пример	31
8.2.4 Удаление схемы	31
8.2.4.1 Формат	31
8.2.4.2 Параметры запроса	32
8.2.4.3 Возвращаемое значение	32
8.2.4.4 Описание/Алгоритм	32
8.2.4.5 Пример	32
8.2.5 Изменение схемы	32
8.2.5.1 Формат	32
8.2.5.2 Параметры запроса	32
8.2.5.3 Тело запроса	32
8.2.5.4 Возвращаемое значение	33
8.2.5.5 Описание/Алгоритм	33
8.2.5.6 Пример	33

8.2.6	Генерация описания таблиц схемы	33
8.2.6.1	Формат	33
8.2.6.2	Параметры запроса	33
8.2.6.3	Тело запроса	33
8.2.6.4	Возвращаемое значение	34
8.2.6.5	Описание/Алгоритм	34
8.2.6.6	Пример	34
8.3	Исключения схем.....	34
8.3.1	Список исключений для схемы.....	34
8.3.1.1	Формат	34
8.3.1.2	Параметры запроса	34
8.3.1.3	Возвращаемое значение	34
8.3.1.4	Описание/Алгоритм	34
8.3.1.5	Пример	34
8.3.2	Добавление исключения	35
8.3.2.1	Формат	35
8.3.2.2	Параметры запроса	35
8.3.2.3	Тело запроса	35
8.3.2.4	Возвращаемое значение	36
8.3.2.5	Описание/Алгоритм	36
8.3.2.6	Пример	36
8.3.3	Удаление исключения	36
8.3.3.1	Формат	36
8.3.3.2	Параметры запроса	36
8.3.3.3	Возвращаемое значение	36
8.3.3.4	Описание/Алгоритм	36
8.3.3.5	Пример	36
8.4	Правила	37
8.4.1	Список правил.....	37
8.4.1.1	Формат	37
8.4.1.2	Параметры запроса	37
8.4.1.3	Возвращаемое значение	37
8.4.1.4	Описание/Алгоритм	37
8.4.1.5	Пример	37
8.4.2	Информация по правилу	37

8.4.2.1	Формат	37
8.4.2.2	Параметры запроса	37
8.4.2.3	Возвращаемое значение	38
8.4.2.4	Описание/Алгоритм	38
8.4.2.5	Пример	38
8.4.3	Добавление правила	38
8.4.3.1	Формат	38
8.4.3.2	Параметры запроса	38
8.4.3.3	Тело запроса	38
8.4.3.4	Возвращаемое значение	40
8.4.3.5	Описание/Алгоритм	40
8.4.3.6	Пример	40
8.4.4	Удаление правила	40
8.4.4.1	Формат	40
8.4.4.2	Параметры запроса	41
8.4.4.3	Возвращаемое значение	41
8.4.4.4	Описание/Алгоритм	41
8.4.4.5	Пример	41
8.4.5	Изменение правила	41
8.4.5.1	Формат	41
8.4.5.2	Параметры запроса	41
8.4.5.3	Возвращаемое значение	41
8.4.5.4	Описание/Алгоритм	41
8.4.5.5	Пример	42
8.5	Задачи	42
8.5.1	Список задач	42
8.5.1.1	Формат	42
8.5.1.2	Параметры запроса	42
8.5.1.3	Возвращаемое значение	42
8.5.1.4	Описание/Алгоритм	42
8.5.1.5	Пример	42
8.5.2	Информация по задаче	43
8.5.2.1	Формат	43
8.5.2.2	Параметры запроса	43
8.5.2.3	Возвращаемое значение	43

8.5.2.4 Описание/Алгоритм	46
8.5.2.5 Пример	46
8.5.3 Прерывание задачи	46
8.5.3.1 Формат	46
8.5.3.2 Параметры запроса	46
8.5.3.3 Тело запроса	46
8.5.3.4 Возвращаемое значение	46
8.5.3.5 Описание/Алгоритм	46
8.5.3.6 Пример	47
8.5.4 Возобновление задачи.....	47
8.5.4.1 Формат	47
8.5.4.2 Параметры запроса	47
8.5.4.3 Тело запроса	47
8.5.4.4 Возвращаемое значение	47
8.5.4.5 Описание/Алгоритм	47
8.5.4.6 Пример	48
9 Рекомендации по выполнению полной сверки баз после миграции	49

1 Общие сведения

Прототип инструмента сверки данных изначально был разработан для внутренних целей SberPlatform и по просьбе разработчиков был включен в состав дистрибутива PostgreSQL SE.

Данный прототип является одним из возможных вариантов инструментов сверки данных. Окончательный выбор инструмента и его использование находится в зоне ответственности продуктовой команды осуществляющей переход с Oracle на PostgreSQL.

Инструмент сверки – war-сервис для проверки данных при миграции с Oracle на PostgreSQL. Для работы необходим сервер PostgreSQL SE.

В разделе «Файлы» даны ссылки на инструмент и другие необходимые для выполнения инструкции файлы.

2 Файлы

Файлы расположены в дистрибутиве в папке migration_tools/db-data-comparator:

- db-data-comparator.zip – архив с ? инструмента. Содержит каталог db-data-comparator:
 - 001.00 – каталог с данными для PostgreSQL, используемым для работы сервиса:
 - dictionaries – ?;
 - others – ?;
 - schema/schema.sql – скрипт создания схемы базы данных для работы сервиса.
 - db-data-comparator.xml – файл набора изменений для liquibase (см. раздел «Установка», шаг 1);
 - db-data-comparator-schema.sh – пример запуска liquibase (см. раздел «Установка», шаг 1).
- db-data-comparator-0.8-beta.war – исполняемый код модуля, используется в WildFly (см. раздел «Установка», шаг 7.с);
- make-postgresql-hash-func.sql – функции, используемые сверкой. Устанавливается в целевую схему PG (см. раздел «Настройка правил сверки», шаг 1);
- application.properties – файл с настройками инструмента в WildFly (см. раздел «Установка», шаг 4);
- module.xml – конфигурация WildFly модуля инструмента сверки (см. раздел «Установка», шаг 2.а).

3 Предварительные требования


Установлено ПО:

1. PostgreSQL SE версии 4.2.3 для работы инструмента.
2. Wildfly. Рекомендуемая версия – 10.1.0.Final.
3. liquibase последней версии.

Выполните настройки:

1. В PostgreSQL SE для работы сервиса создайте:
 - a. Базу данных и схему. Рекомендация – создайте пустую схему.
 - b. Техническую учетную запись (ТУЗ) с правами на создание схемы и объектов в схеме, выполнение DML операций над объектами схемы.
2. В проверяемой PostgreSQL создайте ТУЗ с правами на чтение таблиц и View из проверяемой схемы.
3. В проверяемой Oracle создайте ТУЗ с правами на чтение таблиц и View из проверяемой схемы.


4 Установка

 `${WILDFLY_PATH}` – директория, куда установлен WildFly.

1. Выполните заполнение схемы с помощью liquibase. Файл набора изменений – `db-data-comparator.xml`. Пример вызова приведен в `db-data-comparator-schema.sh`.
2. Поместите конфигурацию модуля сервиса в WildFly:
 - a. В директории инсталляции WildFly создайте директорию `${WILDFLY_PATH}/modules/system/layers/base/ru/sbt/configuration/main`.
 - b. В директорию `${WILDFLY_PATH}/modules/system/layers/base/ru/sbt/configuration/main` поместите файл `module.xml`

3. Опционально. Добавьте или настройте пользователя для управления WildFly.
4. В файле

`application.properties`
`(${WILDFLY_PATH}/modules/system/layers/base/ru/sbt/configuration/main)` добавьте параметр `dataSource`, в котором укажите

 Упоминаемые `jndi` должны соответствовать `jndi` для `datasources` WildFly, создание которых указано ниже по пунктам

5. Установите объём кучи WildFly не меньше объёма данных. При запуске через `${WILDFLY_PATH}/bin/standalone.sh` размер кучи меняется в скрипте `${WILDFLY_PATH}/bin/standalone.conf.ps1`.
6. Запустите WildFly.
7. В консоли WildFly:
 - a. Выполните `deployment` развертывание/установку драйверов `jdbc` для Oracle и PostgreSQL. Рекомендуемые версии:
 - `ojdbc7` – 12.1.0.2;
 - `postgresql` – 42.2.8.
 - b. В разделе *Configuration System properties* добавьте свойство с ключем `oracle.jdbc.J2EE13Compliant` и значением `true`.
 - c. Выполните `deployment` развертывание/установку `war`-файла сервиса.


5 Описание модели сверки

Сущности модели сверки данных схем баз данных:

- источник данных – зарегистрированный источник данных, описывающий наименование источника данных в сервисе сверки:
 - идентификатор источника данных (id источника);
 - имя источника (name, имя источника);
 - jndi-имя в datasource WildFly, используемое для обращения к сверяемой схеме БД (uri);
 - тип источника данных – обозначение используемой БД (type);
 - размера пула соединений (poolSize).
 - схема – описание сверяемой схемы:
 - идентификатор схемы (id схемы);
 - описание схемы – отображаемое (идентифицирующее в сервисе) наименование схемы (schemaDesc);
 - имя схемы в базе данных (schemaName);
 - ? привязку к зарегистрированному источнику данных;
 - структура сверяемых таблиц и полей таблиц (datasource).
 - правило – зарегистрированная пара сверяемых схем, в привязке к параметрам сверки и опциональными правилами соотнесения разноименных (в сверяемых схемах) таблиц и полей таблиц:
 - идентификатор правила (id правила);
 - имя правила (name);
 - описания схем Oracle (firstSchemaDesc) и PostgreSQL (secondSchemaDesc);
 - (matching);
 - размер пачки сверки (batchSize);
 - максимальное количество повторов сверки (maxRetries).
 - исключение – правило исключения объектов схемы (таблиц или полей) из описания схемы при генерации описания схемы через вызов API сервиса:
 - идентификатор исключения (id исключения);
 - описание схемы Oracle (schemaDesc);
 - исключаемый объект: таблица или столбец таблицы (type);
 - маска имен таблиц, в формате для LIKE (tablePattern);
 - маска имен полей, в формате для LIKE (columnPattern).
- Регистрируется в привязке к схеме, на которую действует;
- задача – зарегистрированный запуск задачи сверки схем по определенному правилу с определенным типом сверки и параметрами:
 - идентификатор задачи (id задачи);
 - тип задачи (taskType):
 - HASH – сверка записей HASH;
 - INCREMENT – инкрементальная сверка;
 - OVERALL – полная сверка по интервалам времени;
 - VERBOSE – детальная сверка таблицы.
 - дата и время начала (fromDateTime) и окончания (toDateTime) задачи.

6 Настройка правил сверки

1. Из под ТУЗ в проверяемой схеме PostgreSQL выполните в pgAdmin скрипт `scripts/make-postgresql-hash-func.sql`
2. В консоли WildFly создайте non-XA datasource для БД под Oracle с параметрами подключения к схеме под Oracle. Подключение должно выполняться из под ТУЗ для Oracle.
3. В консоли WildFly создайте non-XA datasource для БД под PostgreSQL с параметрами подключения к схеме под PostgreSQL. Подключение должно выполняться из под ТУЗ для PostgreSQL.
4. Настройте пулы соединений datasources для Oracle и PostgreSQL: количество соединений больше минимального размера пула для схем (параметр `poolSize` в запросах). Пример:

 Во всех запросах необходимо использовать одно и тоже значение параметра `poolSize`.

5. Выполните запрос регистрации datasource для Oracle, пример:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/datasource" -d '{"type":"ORACLE", "name": "имя источника данных Oracle", "uri":"jndi источника данных Oracle в WildFly", "poolSize": 5}'
```

6. Выполните запрос регистрации datasource для PostgreSQL, пример:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/datasource" -d '{"type":"POSTGRESQL", "name": "имя источника данных PostgreSQL", "uri":"jndi источника данных PostgreSQL в WildFly", "poolSize": 5}'
```

7. Выполните регистрацию схемы для Oracle, пример:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/schema" -d '{"schemaDesc":"Описание схемы Oracle, идентифицирует схемы", "schemaName": "имя схемы", "dataSource":{"type":"ORACLE", "name": "имя источника данных Oracle", "uri":"jndi источника данных Oracle в WildFly", "poolSize": 5}}'
```

8. Выполните регистрацию схемы для PostgreSQL, пример:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/schema" -d '{"schemaDesc":"Описание схемы PostgreSQL, идентифицирует схемы", "schemaName": "имя схемы", "dataSource":{"type":"POSTGRESQL", "name": "имя источника данных PostgreSQL", "uri":"jndi источника данных PostgreSQL в WildFly", "poolSize": 5}}'
```

9. Добавьте ограничения на не требующие проверки объекты, выбираемые при формировании описаний схемы Oracle.

Для таблиц:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/exclusion" -d '{"schemaDesc":"Описание схемы Oracle", "type":"TABLE", "tablePattern":"Маска имен таблиц, в формате для LIKE"}'
```

Для полей:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/exclusion" -d '{"schemaDesc":"Описание
схемы Oracle", "type":"COLUMN","tablePattern":"Маска имен таблиц, в формате для
LIKE","columnPattern":"Маска имен полей, в формате для LIKE"}'
```

ограничения можно удалять через вызовы методов, указанных в описании API.

10. Добавьте ограничения на не требующие проверки объекты, выбираемые при формировании описаний схемы PostgreSQL.

Для таблиц:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/exclusion" -d '{"schemaDesc":"Описание
схемы PostgreSQL", "type":"TABLE","tablePattern":"Маска имен таблиц, в формате для LIKE"}'
```

Для полей:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/exclusion" -d '{"schemaDesc":"Описание
схемы PostgreSQL", "type":"COLUMN","tablePattern":"Маска имен таблиц, в формате для
LIKE","columnPattern":"Маска имен полей, в формате для LIKE"}'
```

ограничения можно удалять через вызовы методов, указанных в описании API.

11. Выполните генерацию описания схемы Oracle, пример:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/schema/desc/<описание схемы Oracle>/
generate" -d '{}'
```

12. Выполните генерацию описания схемы PostgreSQL, пример:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/schema/desc/<описание схемы PostgreSQL>/
generate" -d '{}'
```

13. При необходимости выполните редактирование и сохранение описания схемы Oracle. Для редактирования:

- a. Получите текущее описание схемы:

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/schema/desc/<описание схемы Oracle>"
```

- b. Выполните правки описания в любом редакторе.
- c. Выполните запрос на изменение:

```
curl -v -X PUT "http://127.0.0.1:8080/db-data-comparator/schema/desc/<описание схемы Oracle>"
-d 'json-документ описания схемы'
```

14. При необходимости выполните редактирование и сохранение описания схемы PostgreSQL. Для редактирования:

- a. Получите текущее описание схемы:

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/schema/desc/<описание схемы PostgreSQL>"
```

- b. Выполните правки описания в любом редакторе.
- c. Выполните запрос на изменение:

```
curl -v -X PUT "http://127.0.0.1:8080/db-data-comparator/schema/desc/<описание схемы PostgreSQL>" -d 'json-документ описания схемы'
```

15. Зарегистрируйте правило сверки, при необходимости настройте правила соответствия таблиц и полей схем. Пример:

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/rule" -d '{"name":"имя правила, идентифицирует правило", "firstSchemaDesc":"описание схемы Oracle", "secondSchemaDesc":"описание схемы PostgreSQL", "matching":null, "batchSize":"размер пачки сверки", "maxRetries":"максимальное количество повторов сверки"}'
```

поле matching заполняется при изменениях именования объектов между сверяемыми схемами, по шаблону:

```
"matching":[{"tableOne":{"fields":["имя поля 1", ...]}, "tableTwo":{"fields":["имя поля 1", ...]}}], ...]
```

где соответствующие поля таблиц должны находиться в соответствующих позициях в списках.

Изменение описаний схем, настроек datasources и параметров правила могут быть выполнены в любой момент времени с помощью REST-вызовов с методом PUT.

7 API для запуска задач сверки

HTTP RC

404 – не найдено, в теле пояснение, что именно

409 – конфликт, в теле пояснение

500 – внутренняя ошибка, когда все плохо

200, 201, 204 – все ок

7.1 Сверка записей HASH

Сверка каждой записи для пар таблиц по хэсам.


7.1.1 Формат

POST /db-data-comparator

7.1.2 Параметры запроса

7.1.3 Тело запроса

```
{
  "ruleName"="имя правила",
  "taskType"="тип задачи",
  "fromDateTime"="дата и время начала",
  "toDateTime"="дата и время начала"
}
```

 М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
ruleName	string		М	Имя исполняемого правила	

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
taskType	string		М	Применяемый алгоритм сверки. Для данного сценария только HASH.	HASH
fromDateTime	string	YYYY-MM-DDTHH24:MI:SS.MS+TZ	О	Начало интервала	
toDateTime	string	YYYY-MM-DDTHH24:MI:SS.MS+TZ	О	Окончание интервала	

7.1.4 Возвращаемое значение

Вызов возвращает ID созданной задачи

7.1.5 Описание/Алгоритм

При несоответствии описаний таблиц схем правила, либо при отсутствии в какой-либо из таблиц схем ключевого поля - задача инициирована не будет и будет возвращено описание проблемы в конфигурации.

Выполняет последовательную сверку записей, отсортированных по первичному или уникальному ключу, заданному в конфигурации схемы для таблицы. Сверка производится по хэшу, вычисленному по всем полям, заданным в конфигурации для таблицы, кроме ключевых. Сверка больших LOB полей не рекомендована, т.к. происходит очень медленно.

7.1.6 Пример

Без ограничения интервала

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/" -d '{"ruleName": "ruleExample", "taskType": "HASH"}'
```

С ограничением интервала

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/" -d '{"ruleName": "ruleExample", "taskType": "HASH", "fromDateTime": "1970-01-01T01:15:51.10+00:00", "toDateTime": "1970-02-01T05:40:21.04+00:00"}'
```

7.2 Инкрементальная сверка

краткое назначение


7.2.1 Формат

POST /db-data-comparator

7.2.2 Параметры запроса

7.2.3 Тело запроса

```
{
  "ruleName":"имя правила",
  "taskType":"тип задачи"
}
```

 М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
ruleName	string		М	Имя исполняемого правила	
taskType	string		М	Применяемый алгоритм сверки. Для данного сценария только INCREMENT.	INCREMENT

7.2.4 Возвращаемое значение

Вызов возвращает ID созданной задачи.

7.2.5 Описание/Алгоритм

При несоответствии описаний таблиц схем правила, либо при отсутствии в какой-либо из таблиц схем ключевого поля или поля определяющего временную метку записи - задача инициирована не будет и будет возвращено описание проблемы в конфигурации.

Для определения нижней границы времени для создаваемой задачи учитываются только успешно завершённые задачи полной сверки (не по отдельным таблицам)

Будет создана задача с границами времени от максимальной верхней границы интервала времени успешно завершенных задач (без расхождений) до момента времени вызова.

Выполняет последовательную сверку записей, отсортированных по первичному или уникальному ключу, заданному в конфигурации схемы для таблицы. Сверка производится по хэшу, вычисленному по всем полям, заданным в конфигурации для таблицы, кроме ключевых. Сверка больших LOB полей не рекомендована, т.к. происходит очень медленно.

7.2.6 Пример

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/" -d '{"ruleName": "ruleExample", "taskType": "INCREMENT"}'
```

7.3 Полная сверка по интервалам времени

7.3.1 Формат

POST /db-data-comparator

7.3.2 Параметры запроса

7.3.3 Тело запроса

```
{
  "ruleName"="имя правила",
  "taskType"="тип задачи",
  "fromDateTime"="дата и время начала",
  "toDateTime"="дата и время начала"
}
```

i М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
ruleName	string		М	Имя исполняемого правила	

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
taskType	string		M	Применяемый алгоритм сверки. Для данного сценария только OVERALL.	OVERALL
fromDateTime	string	YYYY-MM-DDTHH24:MI:SS.MS+TZ	O	Начало интервала	
toDateTime	string	YYYY-MM-DDTHH24:MI:SS.MS+TZ	O	Окончание интервала	

7.3.4

Возвращаемое значение

Вызов возвращает ID созданной задачи.

7.3.5 Описание/Алгоритм

При несоответствии описаний таблиц схем правила, либо при отсутствии в какой-либо из таблиц схем поля определяющего временную метку записи - задача инициирована не будет и будет возвращено описание проблемы в конфигурации.

Выполняется сверка каждой записи для пар таблиц по хэшам, вычисляемым по всем записям для интервала времени

Интервал времени, за который есть данные в таблицах сверяемой пары таблиц, попадающие в интервал времени, заданный параметрами задачи,

разбивается на 10 равных под интервалов, и каждый подинтервал считается как отдельный блок для сверки. Так сделано для возможности локализации

интервалов времени, на которых есть расхождения в данных

Сверка больших LOB полей не рекомендована, т.к. происходит очень медленно..

7.3.6 Пример

Без ограничения интервала

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/" -d '{"ruleName": "ruleExample", "taskType": "OVERALL"}'
```

С ограничением интервала

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/" -d '{"ruleName": "ruleExample", "taskType": "OVERALL", "fromDateTime": "1970-01-01T01:15:51.10+00:00", "toDateTime": "1970-02-01T05:40:21.04+00:00"}'
```

7.4 Детальная сверка таблицы

краткое назначение

7.4.1 Формат

POST /db-data-comparator

7.4.2 Параметры запроса

7.4.3 Тело запроса

```
{
  "ruleName"="имя правила",
  "taskType"="тип задачи",
  "fromDateTime"="дата и время начала",
  "toDateTime"="дата и время начала"
}
```

i М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
ruleName	string		М	Имя исполняемого правила	
taskType	string		М	Применяемый алгоритм сверки. Для данного сценария только VERBOSE.	VERBOSE
fromDateTime	string	YYYY-MM-DDTHH24:MI:SS.MS+TZ	О	Начало интервала	

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
toDateTime	string	YYYY-MM-DDTHH24:MI:SS.MS+TZ	O	Окончание интервала	

7.4.4 Возвращаемое значение

Вызов возвращает ID созданной задачи.

При отсутствии ключа в одной из таблиц с соответствующей стороны от <-> будет пусто.

Результаты выполнения задачи могут быть получены из информации по задаче (см. API для задач).

info

Самое интересное находится в поле “message” json’a.

Содержит это поле записи вида

Ключ в таблице в Oracle <-> Ключ в таблице в PostgreSQL : (<отличающееся значение поля 1 в таблице в Oracle> <-> <отличающееся значение поля 1 в таблице в PostgreSQL>, ..., <отличающееся значение поля N в таблице в Oracle> <-> <отличающееся значение поля N в таблице в PostgreSQL>)

Например:

- Запись

<-> object_id:6729149752086136832: ()

Говорит о том что в таблице на стороне PostgreSQL найдена запись с ключом по полю object_id и значением 6729149752086136832, при этом в таблице в Oracle запись с таким ключом отсутствует

- Запись

object_id:6814702018578457600 <->: ()

Говорит о том что в таблице на стороне Oracle найдена запись с ключом по полю object_id и значением 6814702018578457600, при этом в таблице в PostgreSQL запись с таким ключом отсутствует

- Запись

object_id:6927793286421882880 <-> object_id:6927793286421882880: (<owner_id:1341671445132619497> <-> <owner_id:1199888763024125949>

<chgcnt:0> <-> <chgcnt:1>

<sys_affinityrootid:1341671445132619497> <-> <sys_affinityrootid:1199888763024125949>)

Говорит о том что для записей в таблице на стороне Oracle и на стороне PostgreSQL с ключами по полю object_id и значением 6927793286421882880 найдены расхождения в значениях по полям owner_id, chgcnt и sys_affinityrootid. Значения расходящихся полей указаны через :

7.4.5 Описание/Алгоритм

При несоответствии описаний таблиц схем правила, либо при отсутствии в какой-либо из таблиц схем ключевого поля - задача инициирована не будет и будет возвращено описание проблемы в конфигурации.

Имя таблицы является обязательным для задач детальной сверки

Формат границ времени YYYY-MM-DDTHH24:MI:SS.MS+TZ

Выполняется сверка для каждой записи пары таблиц, заданных параметром, на заданном (или без ограничений) интервале времени. Сверка производится по каждому полю таблиц.

7.4.6 Пример

Без ограничения интервала

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/" -d '{"ruleName": "ruleExample", "taskType": "VERBOSE", "tableName": "OracleTableNameExample"}'
```

С ограничением интервала

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/" -d '{"ruleName": "ruleExample", "taskType": "VERBOSE", "tableName": "OracleTableNameExample", "fromDateTime": "1970-01-01T01:15:51.10+00:00", "toDateTime": "1970-02-01T05:40:21.04+00:00"}'
```

8 REST API сверки

8.1 Источники данных (datasource)

8.1.1 Получить список источников данных

8.1.1.1 Формат

GET /db-data-comparator/datasource

8.1.1.2 Возвращаемое значение

В результирующем сообщении задачи (message результате в GET запроса) выводится список пар таблиц с указанием количества расхождений для пары таблиц

8.1.1.3 Описание/Алгоритм

8.1.1.4 Пример

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/datasource"
```


8.1.2 Информация по источнику данных

8.1.2.1 Формат

GET /db-data-comparator/datasource/<id источника>

GET /db-data-comparator/datasource/name/<имя источника>

8.1.2.2 Параметры запроса

 [O] – не обязательное поле.

- id источника – ;
- имя источника – .

8.1.2.3 Возвращаемое значение

В результирующем сообщении задачи (message результате в GET запроса) выводится список пар таблиц с указанием количества расхождений для пары таблиц

8.1.2.4 Описание/Алгоритм

8.1.2.5 Пример

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/datasource/105"
```

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/datasource/name/OracleExample"
```

8.1.3 Добавление источника данных

8.1.3.1 Формат

POST /db-data-comparator/datasource

8.1.3.2 Параметры запроса

8.1.3.3 Тело запроса

```
{
  "type"="тип источника: ORACLE или POSTGRESQL",
  "name"="имя источника данных",
  "uri"="jndi источника данных в WildFly",
  "poolSize"="размер пула"
}
```

i М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
type	enum	(ORACLE) (POSTGRES QL)		Тип источника: ORACLE или POSTGRESQL.	

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
name	string			Имя источника данных.	
uri	string			jndi источника данных в WildFly.	
poolSize	integer			Размер пула.	

8.1.3.4 Возвращаемое значение

8.1.3.5 Описание/Алгоритм

8.1.3.6 Пример

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/datasource" -d '{"type":"тип источника: ORACLE или POSTGRESQL", "name": "имя источника данных", "uri":"jndi источника данных в WildFly", "poolSize": <размер пула>}'
```


8.1.4 Удаление источника данных

8.1.4.1 Формат

DELETE /db-data-comparator/datasource/<id источника>

DELETE /db-data-comparator/datasource/name/<имя источника>

8.1.4.2 Параметры запроса

 [O] – не обязательное поле.

- id источника – ;
- имя источника – .

8.1.4.3 Возвращаемое значение

8.1.4.4 Описание/Алгоритм

Источник не будет удален если он используется какой-либо схемой

8.1.4.5 Пример

```
curl -v -X DELETE "http://127.0.0.1:8080/db-data-comparator/datasource/85"
```

```
curl -v -X DELETE "http://127.0.0.1:8080/db-data-comparator/datasource/name/PostgreSQLExample"
```

8.1.5 Изменение источника данных

8.1.5.1 Формат

PUT /db-data-comparator/datasource/<id источника>

PUT /db-data-comparator/datasource/name/<имя источника>

8.1.5.2 Параметры запроса

- id источника – ;
- имя источника – .

8.1.5.3 Тело запроса

```
{
  "type"="тип источника: ORACLE или POSTGRESQL",
  "name"="имя источника данных",
  "uri"="jndi источника данных в WildFly",
  "poolSize"="размер пула"
}
```

- ❗ М (Mandatory) – обязательный атрибут.
- О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
type	enum	(ORACLE) (POSTGRESQL)		Тип источника: ORACLE или POSTGRESQL.	
name	string			Имя источника данных.	
uri	string			jndi источника данных в WildFly.	
poolSize	integer			Размер пула.	

8.1.5.4 Возвращаемое значение

8.1.5.5 Описание/Алгоритм

8.1.5.6 Пример

```
curl -v -X PUT "http://127.0.0.1:8080/db-data-comparator/datasource/<id источника>" -d'{"type":"тип источника: ORACLE или POSTGRESQL", "uri":"jndi источника данных в WildFly", "poolSize": <размер пула>}'
```

```
curl -v -X PUT "http://127.0.0.1:8080/db-data-comparator/datasource/name/<имя источника>" -d'{"type":"тип источника: ORACLE или POSTGRESQL", "uri":"jndi источника данных в WildFly", "poolSize": <размер пула>}'
```


8.2 Схемы

8.2.1 Список схем

8.2.1.1 Формат

GET /db-data-comparator/schema

8.2.1.2 Параметры запроса

 [O] – не обязательное поле.

8.2.1.3 Возвращаемое значение

В результирующем сообщении задачи (message результате в GET запроса) выводится список пар таблиц с указанием количества расхождений для пары таблиц

8.2.1.4 Описание/Алгоритм

8.2.1.5 Пример

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/schema"
```


8.2.2 Информация по схеме

8.2.2.1 Формат

GET /db-data-comparator/schema/<id схемы>

GET /db-data-comparator/schema/desc/<описание схемы>

8.2.2.2 Параметры запроса

 [O] – не обязательное поле.

- id схемы – ;
- описание схемы – .

8.2.2.3 Возвращаемое значение

В результирующем сообщении задачи (message результате в GET запроса) выводится список пар таблиц с указанием количества расхождений для пары таблиц

8.2.2.4 Описание/Алгоритм

8.2.2.5 Пример

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/schema/105"
```

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/schema/desc/OracleExample"
```

8.2.3 Добавление схемы

8.2.3.1 Формат

POST /db-data-comparator/schema

8.2.3.2 Параметры запроса

8.2.3.3 Тело запроса

```
{
  "schemaDesc"="описание схемы, идентифицирует схемы",
  "schemaName"="имя схемы",
  "dataSource"= {
    "type"="тип источника: ORACLE или POSTGRESQL",
    "name"="имя источника данных",
    "uri"="jndi источника данных в WildFly",
    "poolSize"="размер пула"
  }
}
```

i М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
schemaDesc	string		М	Описание схемы, идентифици рует схемы	

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
schemaName	string		М	Имя схемы	
dataSource	object		М	Структура с описанием источника данных	
dataSource.type	enum	(ORACLE) (POSTGRES QL)	М	Тип источника: ORACLE или POSTGRESQL.	
dataSource.name	string		М	Имя источника данных.	
dataSource.uri	string		М	jndi источника данных в WildFly.	
dataSource.poolSize	integer		М	Размер пула.	

8.2.3.4 Возвращаемое значение

8.2.3.5 Описание/Алгоритм

8.2.3.6 Пример

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/schema" -d '{"schemaDesc":"OracleExample",
"schemaName": "OracleSchemaExample", "dataSource":{"type":"ORACLE", "name": "OracleExample", "uri":"java:/
OracleDS", "poolSize": 3}}'
```


8.2.4 Удаление схемы

8.2.4.1 Формат

DELETE /db-data-comparator/schema/<id схемы>

DELETE /db-data-comparator/schema/desc/<описание схемы>

8.2.4.2 Параметры запроса

 [O] – не обязательное поле.

- id схемы – ;
- описание схемы – .

8.2.4.3 Возвращаемое значение

8.2.4.4 Описание/Алгоритм

Схема не будет удалена если она используется каким-либо правилом или для нее задано исключение

8.2.4.5 Пример

```
curl -v -X DELETE "http://127.0.0.1:8080/db-data-comparator/schema/105"
```


```
curl -v -X DELETE "http://127.0.0.1:8080/db-data-comparator/schema/desc/PostgreSQLExample"
```

8.2.5 Изменение схемы

8.2.5.1 Формат

PUT db-data-comparator/schema/desc/<описание схемы>

8.2.5.2 Параметры запроса

 [O] – не обязательное поле.

описание схемы –

8.2.5.3 Тело запроса

json-документ описания схемы

8.2.5.4 Возвращаемое значение

8.2.5.5 Описание/Алгоритм

8.2.5.6 Пример


```
curl -v -X PUT "http://127.0.0.1:8080/db-data-comparator/schema/desc/OracleExample" -d '{"tables":
[{"keyFields":["fieldOne"],"timeField":"fieldFour","fields":
[{"fieldName":"fieldOne","fieldType":"VARCHAR2"}, {"fieldName":"fieldTwo","fieldType":"DATE"},
{"fieldName":"fieldThree","fieldType":"NUMBER"},
{"fieldName":"fieldFour","fieldType":"DATE"}], "tableName":"tableOne"}, {"keyFields":
[{"fieldOne"}, {"timeField":"fieldTwo"}, {"fields":[{"fieldName":"fieldOne","fieldType":"NUMBER"},
{"fieldName":"fieldTwo","fieldType":"DATE"},
{"fieldName":"fieldThree","fieldType":"NUMBER"}], "tableName":"tableTwo"}, {"keyFields":[], "fields":
[{"fieldName":"fieldOne","fieldType":"NUMBER"},
{"fieldName":"fieldTwo","fieldType":"NUMBER"}], "tableName":"tableThree"}], "schemaId":
105, "schemaType":"ORACLE", "schemaDesc":"OracleExample", "schemaName":"OracleSchemaExample", "dataSource":
{"poolSize":3, "name":"OracleExample", "type":"ORACLE", "uri":"java:/OracleDS"}}'
```

8.2.6 Генерация описания таблиц схемы

8.2.6.1 Формат

POST /db-data-comparator/schema/desc/<описание схемы>/generate

8.2.6.2 Параметры запроса

 [O] – не обязательное поле.

описание схемы –

8.2.6.3 Тело запроса

8.2.6.4 Возвращаемое значение

8.2.6.5 Описание/Алгоритм

8.2.6.6 Пример

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/schema/desc/OracleExample/generate" -d '{}'
```


8.3 Исключения схем

8.3.1 Список исключений для схемы

8.3.1.1 Формат

GET /db-data-comparator/exclusion/schema/desc/<описание схемы>

8.3.1.2 Параметры запроса

 [0] – не обязательное поле.

описание схемы –

8.3.1.3 Возвращаемое значение

В результирующем сообщении задачи (message результате в GET запроса) выводится список пар таблиц с указанием количества расхождений для пары таблиц

8.3.1.4 Описание/Алгоритм

также можно увидеть в поле exclusions документа информации по схеме

8.3.1.5 Пример

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/exclusion/schema/desc/PostgreSQLExample"
```

8.3.2 Добавление исключения

8.3.2.1 Формат

POST /db-data-comparator/exclusion

8.3.2.2 Параметры запроса

8.3.2.3 Тело запроса

```
{
  "schemaDesc"="описание схемы",
  "type"="TABLE или COLUMN",
  "tablePattern"="паттерн имен таблиц в формате для LIKE",
  "columnPattern"="паттерн имен полей в формате для LIKE, для типа TABLE не задается"
}
```

i М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
schemaDesc	enum	(ORACLE) (POSTGRESQL)	М	описание схемы.	
type	string	(TABLE) (COLUMN)	М	TABLE или COLUMN.	
tablePattern	string		М	паттерн имен таблиц в формате для LIKE.	
columnPattern	integer		О	паттерн имен полей в формате для LIKE, для типа TABLE не задается.	

8.3.2.4 Возвращаемое значение

8.3.2.5 Описание/Алгоритм

8.3.2.6 Пример

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/exclusion" -d '{"schemaDesc":"OracleExample",  
"type":"COLUMN","tablePattern":"%examp%","columnPattern":"%3%"}'
```


8.3.3 Удаление исключения

8.3.3.1 Формат

DELETE /db-data-comparator/exclusion/<id исключения>

DELETE /db-data-comparator/exclusion/schema/desc/<описание схемы>

8.3.3.2 Параметры запроса

 [O] – не обязательное поле.

- id исключения – ;
- описание схемы – .

8.3.3.3 Возвращаемое значение

8.3.3.4 Описание/Алгоритм

8.3.3.5 Пример

```
curl -v -X DELETE "http://127.0.0.1:8080/db-data-comparator/exclusion/105"
```

```
curl -v -X DELETE "http://127.0.0.1:8080/db-data-comparator/exclusion/schema/desc/PostgreSQLExample"
```


8.4 Правила

8.4.1 Список правил

8.4.1.1 Формат

GET /db-data-comparator/rule

8.4.1.2 Параметры запроса

 [O] – не обязательное поле.

8.4.1.3 Возвращаемое значение

В результирующем сообщении задачи (message результате в GET запроса) выводится список пар таблиц с указанием количества расхождений для пары таблиц

8.4.1.4 Описание/Алгоритм

8.4.1.5 Пример

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/rule"
```


8.4.2 Информация по правилу

8.4.2.1 Формат

GET /db-data-comparator/rule/id правила

GET /db-data-comparator/rule/name/имя правила

8.4.2.2 Параметры запроса

 [O] – не обязательное поле.

- id правила – ;
- имя правила – .

8.4.2.3 Возвращаемое значение

В результирующем сообщении задачи (message результате в GET запроса) выводится список пар таблиц с указанием количества расхождений для пары таблиц

8.4.2.4 Описание/Алгоритм

8.4.2.5 Пример

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/rule/15"
```

```
curl -v -X GET "http://127.0.0.1:8080/db-data-comparator/rule/name/ruleExample"
```

8.4.3 Добавление правила

8.4.3.1 Формат

POST /db-data-comparator/rule

8.4.3.2 Параметры запроса

8.4.3.3 Тело запроса

```
{
  "name"="имя правила",
  "firstSchemaDesc"="описание схемы Oracle",
  "secondSchemaDesc"="описание схемы PostgreSQL",
  "matching"= [
    {
      "имя таблицы в схеме firstSchemaDesc":{"fields":["имя поля 1", ...]},
      "имя второй в схеме secondSchemaDesc":{"fields":["имя поля 1", ...]}
    },
    {
      "имя таблицы в схеме firstSchemaDesc":{"fields":["имя поля 1", ...]},
      "имя второй в схеме secondSchemaDesc":{"fields":["имя поля 1", ...]}
    },
    ...
  ]
}
```

❗ М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
name	string		М	Имя правила, идентифицирует правило.	
firstSchemaDesc	string		М	Имя сверяемой PostgreSQL схемы.	
secondSchemaDesc	string		М	Имя сверяемой Oracle схемы.	
matching	array		О	Массив с соответствием имён таблиц для схем, указанных в атрибутах firstSchemaDesc и secondSchemaDesc.	null
matching.<имя таблицы в схеме firstSchemaDesc>	string		М	Уникальное имя таблицы в схеме, указанной в атрибуте firstSchemaDesc.	

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
matching.<имя таблицы в схеме secondSchemaDesc>	string		М	Уникальное имя таблицы в схеме, указанной в атрибуте secondSchemaDesc.	
matching.fields	enum		М	Перечисление с названиями полей таблицы.	

8.4.3.4 Возвращаемое значение

8.4.3.5 Описание/Алгоритм

при необходимости установления соответствия между таблицами и полями схем поле matching заполняется следующим образом:

```
"matching":{
  "имя первой таблицы":{
    "fields":["имя поля 1", ...],
    "имя второй таблицы":{
      "fields":["имя поля 1", ...]}, ...]
}
```

где соответствующие поля таблиц должны находиться в соответствующих позициях в списках

8.4.3.6 Пример

```
curl -v -X POST "http://127.0.0.1:8080/db-data-comparator/rule" -d '{"name":"ruleExample",
"firstSchemaDesc":"OracleExample","secondSchemaDesc":"PostgreSQLExample","matching":null,"batchSize":50000,
"maxRetries":2}'
```


8.4.4 Удаление правила

8.4.4.1 Формат

DELETE /db-data-comparator/rule/<id правила>

DELETE /db-data-comparator/rule/name/<имя правила>

8.4.4.2 Параметры запроса

 [O] – не обязательное поле.

- id правила – ;
- имя правила – .

8.4.4.3 Возвращаемое значение

8.4.4.4 Описание/Алгоритм

8.4.4.5 Пример

```
curl -v -X DELETE "http://127.0.0.1:8080/db-data-comparator/rule/15"
```

```
curl -v -X DELETE "http://127.0.0.1:8080/db-data-comparator/rule/name/ruleExample"
```

8.4.5 Изменение правила

8.4.5.1 Формат

PUT /db-data-comparator/rule/<id правила>

PUT /db-data-comparator/rule/name/<имя правила>

8.4.5.2 Параметры запроса

- id правила – ;
- имя правила – .

8.4.5.3 Возвращаемое значение

8.4.5.4 Описание/Алгоритм

формирование matching выполняется по тем же правилам что и для случая добавления

8.4.5.5 Пример

```
curl -v -X PUT "http://127.0.0.1:8080/db-data-comparator/rule/15"
```

```
curl -v -X PUT "http://127.0.0.1:8080/db-data-comparator/rule/name/ruleExample"
```


8.5 Задачи

8.5.1 Список задач

8.5.1.1 Формат

GET /db-data-comparator

8.5.1.2 Параметры запроса

 [O] – не обязательное поле.

8.5.1.3 Возвращаемое значение

формат ответа - массив json записей формата, описанного в получении информации по задаче
запрос возвращает последние 50 задач

8.5.1.4 Описание/Алгоритм

8.5.1.5 Пример


```
curl -v -X GET "http://адпес WildFly:8080/db-data-comparator"
```

8.5.2 Информация по задаче

8.5.2.1 Формат

GET /db-data-comparator/<id задачи>

8.5.2.2 Параметры запроса

 [O] – не обязательное поле.

id задачи – .

8.5.2.3 Возвращаемое значение

Вызов возвращает состояние указанной задачи на момент вызова, в виде json

```
{
  "id": "id задачи",
  "ruleId": "идентификатор правила",
  "taskState": "статус задачи - WAITING, RUNNING, ERROR или FINISHED",
  "result": "сообщение с результатами выполнения задачи",
  "errorMessage": "сообщение об ошибке, если статус - ошибка",
  "taskType": "тип задачи",
  "tableName": "имя таблицы для детальной сверки по таблице, опционально. Для полных и инкрементальных сверок отсутствует",
  "fromDateTime": "строковое представление нижней границы интервала сверки для задачи, опционально. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00",
  "toDateTime": "строковое представление верхней границы интервала сверки для задачи, опционально. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00",
  "stateDateTime": "строковое представление даты и времени последней смены статуса задачи. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00",
  "createDateTime": "строковое представление даты и времени создания задачи. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00",
  "finishDateTime": "строковое представление даты и времени окончания задачи. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00",
  "maxRetries": "максимальное количество итераций для задачи",
  "retry": "номер текущей итерации задачи",
  "tasks": "общее количество задач в текущей итерации",
  "completed": "количество завершенных задач в итерации"
}
```

Атрибут	Тип	Описание
id	integer	Уникальное имя таблицы.

Атрибут	Тип	Описание
ruleId	integer	Перечисление с названиями полей таблицы.
taskState	string	Статус задачи: WAITING RUNNING ERROR FINISHED
result	string	сообщение с результатом выполнения задачи
errorMessage	string	сообщение об ошибке, если статус - ошибка
taskType	string	тип задачи
tableName	string	имя таблицы для детальной сверки по таблице, опционально. Для полных и инкрементальных сверок отсутствует
fromDateTime	string	строковое представление нижней границы интервала сверки для задачи, опционально. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00
toDateTime	string	строковое представление верхней границы интервала сверки для задачи, опционально. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00
stateDateTime	string	строковое представление даты и времени последней смены статуса задачи. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00
createDateTime	string	строковое представление даты и времени создания задачи. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00
finishDateTime	string	строковое представление даты и времени окончания задачи. Формат YYYY-MM-DDTHH24:MI:SS.MS+TZ, например 2019-11-01T10:31:30.566+03:00
maxRetries	integer	максимальное количество итераций для задачи
retry	integer	номер текущей итерации задачи
tasks	integer	общее количество задач в текущей итерации

Атрибут	Тип	Описание
completed	integer	количество завершенных задач в итерации

содержимое поля result зависит от типа выполненной задачи.

Для задачи типа **verbose** - содержит записи вида

Ключ в таблице в первой схеме <-> Ключ в таблице во второй схеме : (<отличающееся значение поля 1 в таблице в первой схеме> <-> <отличающееся значение поля 1 в таблице во второй схеме>, ..., <отличающееся значение поля N в таблице в первой схеме> <-> <отличающееся значение поля N в таблице во второй схеме>)

Например:

1. Запись

<-> object_id:6729149752086136832: ()

Говорит о том что в таблице на стороне второй схемы найдена запись с ключом по полю object_id и значением 6729149752086136832, при этом в таблице в первой схеме запись с таким ключом отсутствует

1. Запись

object_id:6814702018578457600 <->: ()

Говорит о том что в таблице на стороне первой схемы найдена запись с ключом по полю object_id и значением 6814702018578457600, при этом в таблице во второй схеме запись с таким ключом отсутствует

1. Запись

object_id:6927793286421882880 <-> object_id:6927793286421882880: (<owner_id:1341671445132619497> <-> <owner_id:1199888763024125949>

<chgcnt:0> <-> <chgcnt:1>

<sys_affinityrootid:1341671445132619497> <-> <sys_affinityrootid:1199888763024125949>)

Говорит о том что для записей в таблице на стороне первой схемы и на стороне второй схемы с ключами по полю object_id и значением 6927793286421882880 найдены расхождения в значениях по полям owner_id, chgcnt и sys_affinityrootid. Значения расходящихся полей указаны через :

Для задачи типа **overall** - содержит записи вида

Таблица в первой схеме <-> таблица во второй схеме : (<интервал времени с расхождениями>, ..., <интервал времени с расхождениями>).

Надо понимать, что задачи типа overall служат для грубого определения наличия расхождений в данных при первоначальной сверке и определения интервалов времени, на которых есть расхождения, для дальнейшего уточнения задачами других типов.

Для остальных задач - формат схож с форматом для задачи типа verbose, но без детализации расхождений по полям.

8.5.2.4 Описание/Алгоритм

8.5.2.5 Пример

```
curl -v -X GET "http://адрес WildFly:8080/db-data-comparator/36"
```

8.5.3 Прерывание задачи


8.5.3.1 Формат

POST /db-data-comparator/terminate

8.5.3.2 Параметры запроса

8.5.3.3 Тело запроса

```
{
  "id"="id задачи"
}
```

 М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
id	integer		М	id задачи.	

8.5.3.4 Возвращаемое значение

Вызов возвращает описание прерванной задачи

8.5.3.5 Описание/Алгоритм

Вызов сохраняет состояние задачи

8.5.3.6 Пример

```
curl -v -X POST "http://адрес WildFly:8080/db-data-comparator/terminate" -d '{"id":"36"}'
```

8.5.4 Возобновление задачи


8.5.4.1 Формат

POST /db-data-comparator/resume

8.5.4.2 Параметры запроса

8.5.4.3 Тело запроса

```
{
  "id"="id задачи"
}
```

 М (Mandatory) – обязательный атрибут.
О (Optional) – не обязательный атрибут.

Атрибут	Тип	Шаблон	М/О	Описание	Значение по умолчанию
id	integer		М	id задачи.	

8.5.4.4 Возвращаемое значение

Вызов возвращает ID возобновленной задачи

8.5.4.5 Описание/Алгоритм

Выполнение задачи возобновляется только при наличии сохраненного состояния прерванной или завершенной с ошибкой задачи

8.5.4.6 Пример

```
curl -v -X POST "http://адрес WildFly:8080/db-data-comparator/resume" -d '{"id": "36"}'
```


9 Рекомендации по выполнению полной сверки баз после миграции

Выполнить полную сверку по интервалам времени.