

# Программное обеспечение «Jay Data»

Версия 1.0.1: © ООО «Кросстех Солюшнс Групп», 2023.03

# Содержание

Введение .....	1
Наименование и назначение .....	1
Развёртывание .....	1
Развёртывание в минимальном режиме .....	1
Развёртывание в рекомендуемом режиме .....	2
Подготовка инфраструктуры .....	2
Шаги по развёртыванию .....	2
Запуск .....	4
Производительность .....	5
Масштабирование .....	9
Требования к аппаратному обеспечению .....	10
Требования в минимальном режиме .....	10
Требования в рекомендуемом режиме .....	11
Лицензирование системы .....	12
Установка лицензии на сервер приложений .....	14
Проверка лицензии .....	14
Использование .....	19
Вход в систему .....	19
Стартовый экран .....	19
Панель навигации .....	19
Дашборды .....	20
Планы обезличивания по статусам .....	20
Изменение планов обезличивания .....	21
Выполнение планов обезличивания .....	22
Распределение конфиденциальных данных .....	22
Экраны .....	23
Типы экранов .....	23
Кнопки .....	23
Поля .....	24
Списки .....	25
Типы списков .....	25
Количество записей в списках .....	25
Множественный выбор и редактирование .....	25
Сортировка списков .....	26
Фильтры и поиск .....	26
Фильтрация в Системе .....	26
Фильтрация с условием поиска .....	27
Приложение .....	29

Методы маскирования	29
Создание нового метода маскирования	31
Параметры методов маскирования	31
Создание нового параметра	32
Консистентность данных	34
Зависимости от других столбцов БД	35
Валидация в методах маскирования	38
Системные методы маскирования	39
Замена на константу	39
Замена на null	40
Замена по справочнику	40
Валидация	41
Замена N символов на служебные	41
Маскирование числа	42
Добавление к дате/времени константного времени	42
Добавление к дате/времени случайного времени	43
Номер банковской карты	44
Валидация	48
ИНН РФ	48
Замена спец. формата - СНИЛС	49
Международный номер	51
Валидация	53
Сопоставление цифр	53
Валидация	54
Случайная замена	54
Валидация	55
Замена на случайное значение из БД	56
Генератор строки ФИО	56
Случайная замена JSON	56
Случайная замена XML	58
Правила маскирования	59
Подключения к БД	61
Создание нового подключения	62
Политики данных	63
Создание новой политики данных	64
Типы данных	64
Создание нового типа данных	65
Домены	66
Системные домены	67
_V_CARD_NUMBER (Номер банковской карты)	67
_V_PAYMENT_ACCOUNT (Расчетный счет)	68

_C_DATE (Дата)	69
_C_EMAIL (Электронная почта)	69
_C_IMEI (IMEI)	70
_C_IMSI (IMSI)	70
_C_IP (IP-адрес)	71
_C_LOCALITY (Населенный пункт)	71
_C_MAC (MAC-адрес)	71
_C_PHONE (Номер мобильного телефона)	72
_C_STREET (Улица)	72
_D_INTERNATIONAL_PASSPORT (Заграничный паспорт)	72
_D_MEDICAL_POLICY (Полис ОМС)	73
_D_PASSPORT (Паспорт)	73
_D_SNILS (СНИЛС)	73
_D_TIN (ИНН физического лица)	74
_O_NAME (Наименование юридического лица)	74
_O_TIN (ИНН юридического лица)	75
_P_EDUCATION_LEVEL (Уровень образования)	75
_P_FIRST_NAME (Имя)	75
_P_FIRST_NAME_LATIN (Имя (лат.))	76
_P_LAST_NAME (Фамилия)	76
_P_LAST_NAME_LATIN (Фамилия (лат.))	76
_P_MARITAL_STATUS (Семейное положение)	77
_P_NATIONALITY (Национальность)	77
_P_PATRONYMIC (Отчество)	77
_P_PATRONYMIC_LATIN (Отчество (лат.))	78
_P_PHILOSOPHICAL_VIEWS (Философские взгляды)	78
_P_POLITICAL_VIEWS (Политические взгляды)	79
_P_RELIGIOUS_VIEWS (Религиозные взгляды)	79
Пользовательские домены	80
Создание нового пользовательского домена	80
Комплексные домены	80
Справочники	81
Управление справочниками	82
Элементы справочника	83
Действия с элементами справочника	83
Массовая загрузка (импорт) элементов справочника	84
Администрирование	87
Пользователи	87
LDAP-пользователи	87
Создание нового LDAP-пользователя	87
Локальные пользователи	88

Создание нового локального пользователя .....	88
Ролевая модель управления доступом .....	89
Матрица ролей и полномочий .....	89
Сопоставление ролей и групп Active Directory .....	90
LDAP .....	91
Конфигурация .....	91
Email .....	92
Конфигурация .....	92
Аудит действий пользователя .....	93

# Введение

## Наименование и назначение

**Jay Data**— система управления тестовыми данными (далее - Система, англ. Test Data Manager, TDM) предназначена для поиска, классификации и обезличивания чувствительных данных путем их маскирования, что помогает устранить риски передачи конфиденциальной информации третьим лицам. Система позволяет защитить конфиденциальные данные от несанкционированного доступа с помощью алгоритмов маскирования.

**Система** основана на высокоуровневой Java-платформе для создания корпоративных информационных систем Jmix (ранее известный как CUBA Platform) с открытым исходным кодом (Open Source). Основной целью системы является маскирование полученных данных для развертывания сред тестирования в соответствии с настроенными алгоритмами маскирования.

Система решает следующие задачи:

- Автоматизация подготовки тестовых данных.
- Профилирование конфиденциальных данных.
- Устранение рисков передачи конфиденциальных данных с помощью алгоритмов маскирования.
- Маскирование данных в зависимости от типа данных и желаемого результата.
- Сведение статистики по результатам маскирования.

## Развёртывание

В данном разделе описаны основные этапы развёртывания системы, а также требования к подготовке и настройке окружения для стабильной работы.

Развёртывание может происходить в **минимальном** и **рекомендуемом** режимах.

### Развёртывание в минимальном режиме

**Развёртывание в минимальном режиме** - это самый простой вид развёртывания, когда все подсистемы развёртываются на одном сервере. Для такого развёртывания необходима подготовка инфраструктуры в соответствии с требованиями к аппаратному обеспечению [в минимальном режиме](#).

Данный вид развёртывания подходит для:

- демонстрационных целей;
- работы с системой в Trial-режиме;
- работы с системой на небольших объёмах данных;

- работы с системой в условиях ограниченных ресурсов;

## Развёртывание в рекомендуемом режиме

**Развёртывание в рекомендуемом режиме** - это оптимальный вид развёртывания, когда подсистемы устанавливаются и запускаются на разных серверах. Для такого развёртывания необходима подготовка инфраструктуры в соответствии с требованиями к аппаратному обеспечению [в рекомендуемом режиме](#). При данном развёртывании необходимо придерживаться рекомендаций, описанных в разделе [масштабирование](#).

Данный вид развёртывания подходит для:

- работы с несколькими БД одновременно;
- работы с системой в продуктивном режиме;
- работы с системой на больших объёмах данных.

## Подготовка инфраструктуры

Перед развёртыванием системы необходимо определить количество одновременно работающих экземпляров микросервисов каждого модуля (см. документ Архитектура) и развернуть конфигурацию серверов в соответствии с [требованиями к аппаратному обеспечению](#).

Необходимо организовать все необходимые сетевые доступы к целевым БД и предоставить требуемые для работы системы права служебным учётным записям для подключения.

## Шаги по развёртыванию



Все необходимые зависимости и окружение входят в состав дистрибутива. Дополнительной установки компонентов на сервер не требуется.

Для установки системы с помощью дистрибутива на ОС CentOS необходимо выполнить следующие действия:

Скачать или загрузить архив дистрибутива на сервер.

Разархивировать архив с помощью команды `tar xzvf`:

```
tar xzvf c7_tdmsrv_deps_installer.tar.gz
```

```
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$
[dss@c7x64srv ~]$ ls
c7_tdmrsv_deps_installer.tar.gz
[dss@c7x64srv ~]$ tar xzvf c7_tdmrsv_deps_installer.tar.gz
c7_tdmrsv_deps_installer/pgdg_repo/
c7_tdmrsv_deps_installer/c7_repo/
c7_tdmrsv_deps_installer/ansible_tdm/
c7_tdmrsv_deps_installer/tdm_offline_purge.sh
c7_tdmrsv_deps_installer/README.md
c7_tdmrsv_deps_installer/tdm_offline_install.sh
c7_tdmrsv_deps_installer/pgdg_repo/repodata/
c7_tdmrsv_deps_installer/pgdg_repo/postgresql14-libs-14.6-1PGDG.rhel7.x86_64.rpm
c7_tdmrsv_deps_installer/pgdg_repo/postgresql14-14.6-1PGDG.rhel7.x86_64.rpm
c7_tdmrsv_deps_installer/pgdg_repo/postgresql14-server-14.6-1PGDG.rhel7.x86_64.rpm
c7_tdmrsv_deps_installer/pgdg_repo/repodata/af833a099bd99362606d36d1e14e70c9f6ddb657dc3ff506b876f2d459c62e2-other.sqlite.bz2
c7_tdmrsv_deps_installer/pgdg_repo/repodata/ca09edab72d73451794889952e70bba1acd813af1dbd7331e0d0b1182c0f346d-filelists.sqlite.bz2
c7_tdmrsv_deps_installer/pgdg_repo/repodata/d8c5cc3fcec171773f4cdd3cbf18c2016e640868f45f138407e622470c3bfb94-other.xml.gz
```

Просмотреть содержимое директории с помощью команды ls, отобразится содержимое директории с установочными файлами системы:

```
c7_tdmrsv_deps_installer/ansible_tdm/templates/pg_hba.conf
c7_tdmrsv_deps_installer/ansible_tdm/templates/README.md
c7_tdmrsv_deps_installer/ansible_tdm/templates/tdmserver-1.0.0-1.e18.x86_64
c7_tdmrsv_deps_installer/ansible_tdm/templates/redis/redis.service
c7_tdmrsv_deps_installer/ansible_tdm/templates/redis/redis-stack-server-6.2
c7_tdmrsv_deps_installer/ansible_tdm/templates/redis/redis.conf
c7_tdmrsv_deps_installer/ansible_tdm/templates/redis/redis_create_cluster.sh
[dss@c7x64srv ~]$ ls
c7_tdmrsv_deps_installer c7_tdmrsv_deps_installer.tar.gz
[dss@c7x64srv ~]$ ls
```

Перейти в директорию проекта с помощью команды cd:

```
cd c7_tdmrsv_deps_installer
```

```
[dss@c7x64srv c7_tdmrsv_deps_installer]$
[dss@c7x64srv c7_tdmrsv_deps_installer]$
[dss@c7x64srv c7_tdmrsv_deps_installer]$
[dss@c7x64srv c7_tdmrsv_deps_installer]$ cd ..
[dss@c7x64srv ~]$ pwd
/home/dss
[dss@c7x64srv ~]$ ls
c7_tdmrsv_deps_installer c7_tdmrsv_deps_installer.tar.gz
[dss@c7x64srv ~]$ cd c7_tdmrsv_deps_installer
[dss@c7x64srv c7_tdmrsv_deps_installer]$ ls
ansible_tdm c7_repo pgdg_repo README.md tdm_offline_install.sh tdm_offline_purge.sh
[dss@c7x64srv c7_tdmrsv_deps_installer]$
```

Ознакомиться с файлом README.md и выполнить инструкции для запуска инсталлятора:

```
cat README.md
```

```
[dss@c7x64srv ~]$ cd c7_tdmrsv_deps_installer
[dss@c7x64srv c7_tdmrsv_deps_installer]$ ls
ansible_tdm c7_repo pgdg_repo README.md tdm_offline_install.sh tdm_offline_purge.sh
[dss@c7x64srv c7_tdmrsv_deps_installer]$ cat README.md
### Install TDM Server instruction.

Install TDM Server.
$ chmod +x tdm_offline_install.sh
$ sudo ./tdm_offline_install.sh

Uninstall TDM Server.
$ sudo ./tdm_offline_purge.sh[dss@c7x64srv c7_tdmrsv_deps_installer]$
```

Скопировать и выполнить из README.md две команды для начала процесса установки:



```
chmod +x tdm_offline_install.sh
sudo ./tdm_offline_install.sh
```

Задать пароль для базы данных PostgreSQL и REDIS, а также указать количество экземпляров сервисов DATAPROCESSOR и MASKING:

```
Install TDM Server.
$ chmod +x tdm_offline_install.sh
$ sudo ./tdm_offline_install.sh

Uninstall TDM Server.
$ sudo ./tdm_offline_purge.sh[dss@c7x64srv c7_tdm_srv_deps_installer]$ chmod +x tdm_offline_install.sh
[dss@c7x64srv c7_tdm_srv_deps_installer]$ sudo ./tdm_offline_install.sh
Enter a new PostgreSQL Server password:
Enter a new Redis password:
Enter the number of Dataprocessor services: 1
Enter the number of Masking services: 2

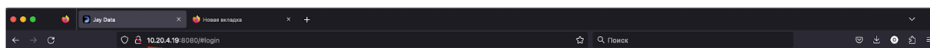
Prepare install.
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
```

Дождаться окончания процесса установки. В конце установщик выдаст URL для доступа в JMIX микросервис системы:

```
slots: (0 slots) slave
replicas 7a0e33fd2fa8a7eed9bdfaf1be22244ebded31cf
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
localhost | CHANGED | rc=0 >>
Created symlink from /etc/systemd/system/multi-user.target.wants/tdmservice.service to /usr/lib/systemd/
localhost | CHANGED | rc=0 >>

TDM Server url: http://10.20.4.19:8080
[dss@c7x64srv c7_tdm_srv_deps_installer]$
```

Запуск микросервисов системы может занимать некоторое время, поэтому в случае отсутствия доступа к микросервису JMIX по указанному URL необходимо перезагрузить страницу позже. Для первого входа в систему ввести учетные данные пользователя **admin**:



## Запуск

После установки системы с помощью дистрибутива запуск произойдет автоматически. Тем не менее, всегда есть возможность запустить/перезапустить/остановить конкретный экземпляр любого микросервиса вручную, работая непосредственно с процессом системы, отвечающей за запуск соответствующего .jar-файла нужного сервиса.

Запуск .jar-файла происходит командой `java -jar` из директории, в которой расположен

данный файл:

```
java -jar jmix.jar
```

## Производительность

В разделах ниже представлена статистическая информация о производительности при развёртывании в разных режимах: [минимальном](#) и [рекомендуемом](#).



Перечисленные показатели производительности и времени работы приведены ориентировочно, так как они зависят от множества параметров, которые в разных инфраструктурах отличаются:

- Скорость чтения/записи в БД-источник и БД-приёмник (см. раздел [Подключения к БД](#)) в зависимости от её типа;
- IOPS (количество операций ввода/вывода) СХД, в которой размещены БД и сама система;
- Выбранные [правила маскирования](#) (на каких [методах маскирования](#) они базируются);
- Значения [параметров](#) в правилах маскирования (например, правила, которые запускаются с параметром сохранения [консистентности данных](#) `saveConsistency = 'true'` выполняются дольше, чем правила без сохранения консистентности);
- Пропускная способность сети;
- Количество служебных объектов, которые необходимо отключать и включать перед маскированием (см. [Служебные SQL-скрипты](#));
- и другие параметры.

Для обезличивания была выбрана база данных PostgreSQL с применением следующих методов маскирования:

- `BANK_CARD_STRING` (Номер банковской карты (строка)) для таблицы `bank_card`;
- `RANDOM_CHANGE_STRING` (Случайная замена (строка)) для таблицы `individual`;
- `REPLACE_BY_DICTIONARY_STRING` (Замена по справочнику (строка)) для таблицы `individual`;
- `SYMBOLS_REPLACE_STRING` (Замена N символов на служебные (строка)) для таблицы `phone_number`, `individual`, `phone_number`, `address`;
- `INTERNATIONAL_NUMBER_STRING` (Международный номер (строка)) для таблицы `phone_number`;
- `SYMBOLS_REPLACE_STRING` (Замена N символов на служебные (строка)) для таблицы `address`;
- `INN_STRING` (ИНН РФ (строка)) для таблицы `inn`;
- `SNILS_RUS_STRING` (СНИЛС РФ (строка)) для талблицы `snils`.

Результаты замеров производительности при разных параметрах развёртывания приведены в таблице ниже.

Таблица 1. Производительность системы

Кол-во планов	Общее кол-во таблиц	Общее кол-во столбцов	Общее кол-во записей	Записей в секунду, М/Р*	Кол-во MC MASKING / REDIS / DATAPROCESSOR	Общее время процесса, М/Р*	Пиковая нагрузка DATAPROCESSOR ЦП, ОЗУ, М/Р*	Пиковая нагрузка MASKING ЦП, ОЗУ, М/Р*
1	6	22	22 млн.	56 700	1 / 1 / 1	6м 28с.	60% ЦП, 70% ОЗУ	78% ЦП, 80% ОЗУ
1	6	22	57,2 млн.	34 050	1 / 1 / 1	28м.	50% ЦП, 50% ОЗУ	70% ЦП, 75% ОЗУ
1	6	22	57,2 млн.	41 450	4 / 4 / 2	23м.	35% ЦП, 40% ОЗУ	40% ЦП, 60% ОЗУ
2	12	44	44 млн.	56 410	1 / 1 / 1	13 мин.	80% ЦП, 90% ОЗУ	80% ЦП, 90% ОЗУ
2	12	44	44 млн.	56 410	2 / 2 / 2	13 мин.	40% ЦП, 70% ОЗУ	20% ЦП, 60% ОЗУ
2	12	44	44 млн.	58 000	4 / 4 / 4	12 мин.	30% ЦП, 70% ОЗУ	20% ЦП, 60% ОЗУ
4	24	88	88 млн.	58 000	4 / 4 / 4	12 мин.	30% ЦП, 60% ОЗУ	20% ЦП, 50% ОЗУ

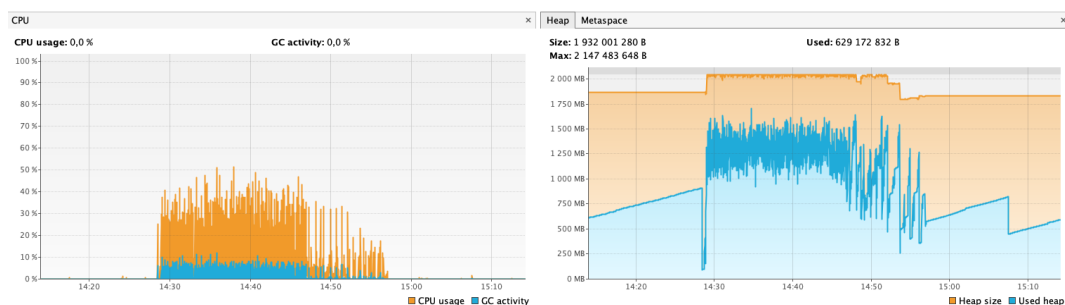
- \* М/Р\* — при **минимальных** и **рекомендуемых** настройках развёртывания.



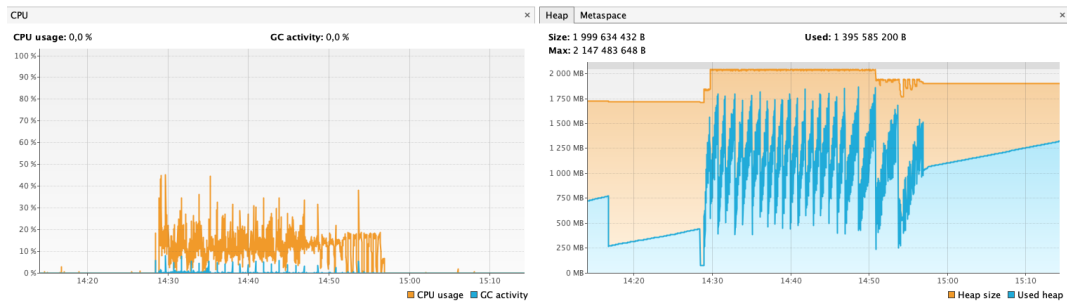
для повышения производительности системы и достижения показателей при **рекомендуемых** параметрах следует придерживаться рекомендаций, приведённых в разделе **Масштабирование**.

Процесс обезличивания на одной виртуальной машине показал следующие результаты нагрузки на микросервисах:

- Результаты нагрузки на DATAPROCESSOR



- Результаты нагрузки на MASKING



В таблице ниже приведена производительность системных методов маскирования (в условных единицах) при прочих равных условиях:

Таблица 2. Производительность системных методов маскирования

Название метода	Код метода	Тип данных	Значения параметров	Скорость в у.е. в секунду, М/Р*
Маскирование числа	ADDITION_NUMBER	NUMBER		
Замена на константу	CONSTANT_REPLACE_STRING	STRING	constantValue = <значение_скрыто> ignoreEmpty = v	
	CONSTANT_REPLACE_CLOB	CLOB	constantValue = <значение_скрыто> ignoreEmpty = v	
Замена N символов на служебные	SYMBOLS_REPLACE_STRING	STRING		
Случайная замена	RANDOM_CHANGE_STRING	STRING		
Случайная замена JSON	RANDOM_CHANGE_JSON	JSON		
	RANDOM_CHANGE_JSON_STRING	STRING		
	RANDOM_CHANGE_JSON_CLOB	CLOB		
Случайная замена XML	RANDOM_CHANGE_XML_CLOB	CLOB		
	RANDOM_CHANGE_XML	XML		
	RANDOM_CHANGE_XML_STRING	STRING		

Название метода	Код метода	Тип данных	Значения параметров	Скорость в у.е. в секунду, М/Р*
Сопоставление цифр	NUMBER_MATCHING_NUMBER	NUMBER		
	NUMBER_MATCHING_STRING	STRING		
Замена по Справочнику	REPLACE_BY_DICTIONARY_STRING	STRING		88 / 70
Генератор строки ФИО	FULL_NAME_GENERATOR_STRING	STRING		50 / 40
Добавление к дате/времени константного времени (дата)	MODIFY_DATE	DATE		24 / 22
Добавление к дате/времени случайного времени (дата)	RANDOM_MODIFY_DATE	DATE		12 / 10
Замена на null	REPLACE_TO_NULL_STRING	STRING	-	10 / 10
	REPLACE_TO_NULL_BLOB	BLOB		
	REPLACE_TO_NULL_CLOB	CLOB		
СНИЛС РФ	SNILS_RUS_NUMBER	NUMBER		
	SNILS_RUS_STRING	STRING		
ИНН РФ	INN_NUMBER	NUMBER		
	INN_STRING	STRING		
Международный номер	INTERNATIONAL_NUMBER_STRING	STRING		
Номер банковской карты	BANK_CARD_STRING	STRING		

Название метода	Код метода	Тип данных	Значения параметров	Скорость в у.е. в секунду, М/Р*
Замена на случайное значение из БД	RANDOM_REPLACE MENT_FROM_DB_STRING	STRING	-	
	RANDOM_REPLACE MENT_FROM_DB_BOOLEAN	BOOLEAN	-	
	RANDOM_REPLACE MENT_FROM_DB_DATE	DATE	-	
	RANDOM_REPLACE MENT_FROM_DB_NUMBER	NUMBER	-	

## Масштабирование

При масштабировании DATAPROCESSOR и MASKING повышается нагрузочная способность в следующих аспектах:

- Масштабирование DATAPROCESSOR даёт возможность запускать несколько **планов обезличивания** одновременно. При масштабировании DATAPROCESSOR выполнение процессов обезличивания распределяется с помощью балансировщика нагрузки и выбирается менее загруженный экземпляр микросервиса, поэтому при необходимости одновременного запуска обезличивания нескольких БД рекомендовано масштабировать именно DATAPROCESSOR. Распаралеливание между экземплярами происходит на уровне вычитки данных из разных таблиц, т.е. вычитка данных из 1 таблицы будет происходить на одном экземпляре DATAPROCESSOR'е, а вычитка данных из другой таблицы может происходить на другом экземпляре.
- Масштабирование MASKING(+REDIS) повышает скорость маскирования одновременно нескольких таблиц БД. Процесс маскирования выполняется параллельно на нескольких экземплярах MASKING(+REDIS), что приводит к увеличению скорости маскирования и нагрузочной способности приложения. Распаралеливание между экземплярами происходит на уровне отправки разных строк таблиц на разные экземпляры MASKING, т.е. на одном экземпляре MASKING маскируется целиком строка с данными (а именно значения всех её полей). Процесс маскирования одной строки таблицы БД не может быть разделён между экземплярами MASKING, но разные строки из одной таблицы могут маскироваться на разных экземплярах. Вместе с MASKING масштабируется REDIS - для исключения конкурентного доступа и образования очередей (когда несколько MASKING записывают и читают данные в один REDIS) каждый MASKING должен использовать свой REDIS.



Существуют 2 основных показателя производительности при

маскировании: скорость процесса маскирования и нагрузочная способность. Скорость маскирования не будет увеличиваться прямо пропорционально увеличению экземпляров MASKING+REDIS, она сильно зависит от вышеупомянутых показателей: выбранных методов маскирования и их параметров (при сохранении консистентности происходит доп. запрос в REDIS), скорости записи данных в БД-приёмник и т.д. Скорость маскирования имеет предельное значение, после которого дальнейшее масштабирование MASKING+REDIS не будет иметь смысла. При этом показатель нагрузочной способности будет увеличиваться прямо пропорционально увеличению экземпляров MASKING+REDIS.

Рекомендации по масштабированию для обеспечения максимальной производительности:

- Экземпляры микросервисов JMIX, DISCOVERY, GATEWAY можно размещать на одном отдельном сервере. Так как данные подсистемы не масштабируются, разносить их на разные серверы не имеет практического смысла;
- Экземпляры микросервисов DATAPROCESSOR необходимо размещать на выделенном сервере с ресурсами только под них. Можно размещать несколько экземпляров DATAPROCESSOR на едином сервере (но не более 2-х экземпляров) или размещать каждый экземпляр DATAPROCESSOR на отдельном сервере;
- Экземпляры микросервисов MASKING вместе с подсистемой промежуточного хранения REDIS необходимо размещать на выделенном сервере с ресурсами только под них. Должно быть поднято столько экземпляров REDIS, сколько экземпляров MASKING развёрнуто — на каждый MASKING свой REDIS (для обеспечения эффективного параллельного доступа MASKING в In memory-хранилище). Можно размещать несколько экземпляров MASKING+REDIS на едином сервере или размещать каждый экземпляр MASKING+REDIS на отдельном сервере;
- Все экземпляры REDIS должны быть объединены в [REDIS-кластер](#) для синхронизации данных между экземплярами REDIS;
- БД должны быть вынесены на отдельный сервер БД.

Описанная конфигурация обеспечивает возможность выполнения процесса обезличивания нескольких БД одновременно за счет снижения нагрузки на каждый микросервис.

## Требования к аппаратному обеспечению

### Требования в минимальном режиме

*Таблица 3. Минимальная конфигурация развёртывания*

Кол-во ВМ	Подсистема	Требования при развёртывании	Окружение
1	JMIX	ОЗУ 16ГБ, Intel® Core™ i7-10510U CPU (8 CPUs), 60ГБ HDD	ОС Windows Server / Linux OpenJDK 11 PostgreSQL 13
	DATAPROCESSOR		
	MASKING		
	REDIS		
	DISCOVERY		
	GATEWAY		
	Сервер БД		

Все подсистемы располагаются на одной виртуальной машине с операционной системой Windows / Linux с указанными характеристиками. БД подсистем могут быть расположены на этой же виртуальной машине.

## Требования в рекомендуемом режиме

Таблица 4. Рекомендуемая конфигурация развёртывания

Кол-во ВМ	Подсистема	Требования к аппаратному обеспечению при развёртывании на разных ВМ*	Системные требования при развёртывании на одной ВМ*	Окружение
1	JMIX	ОЗУ 16ГБ, Intel® Core™ i7-10510U CPU (8 CPUs), 60ГБ HDD		ОС Windows Server / Linux OpenJDK 11
	DISCOVERY			
	GATEWAY			
2	DATAPROCESSOR	ОЗУ 16ГБ, Intel® Core™ i7-10510U CPU (8 CPUs), 40ГБ HDD	ОЗУ 32ГБ, Intel® Core™ i7-10510U CPU (16 CPUs), 60ГБ HDD	ОС Windows Server / Linux OpenJDK 11
4	MASKING	ОЗУ 4ГБ, Intel® Core™ i7-10510U CPU (2 CPUs), 40ГБ HDD	ОЗУ 16ГБ, Intel® Core™ i7-10510U CPU (8 CPUs), 60ГБ HDD	ОС Windows Server / Linux OpenJDK 11
	REDIS			
1	Сервер БД	ОЗУ 16ГБ, Intel® Core™ i7-10510U CPU (8 CPUs), 60ГБ HDD		ОС Windows Server / Linux PostgreSQL 13+



\* **Требования к аппаратному обеспечению** указаны из расчётов, что ВМ с такими характеристиками необходимо разворачивать столько штук, сколько указано в **Кол-ве ВМ**. Например, для MASKING + REDIS необходимо 4 ВМ с указанными характеристиками. Альтернативным равноценным вариантом является развёртывание одной ВМ, но с характеристиками в 4 раза мощнее.

Подсистемы располагаются на разных виртуальных машинах с операционной системой Windows / Linux с указанными характеристиками. БД подсистем должны быть расположены



на отдельной виртуальной машине.

Некоторые подсистемы должны быть расположены на одной виртуальной машине для повышения производительности (см. [масштабирование](#)): ячейки [Характеристики ВМ](#) для таких подсистем объединены.



Теоретически возможно развёртывание всех подсистем на одной ВМ с суммарными характеристиками для всех микросервисов. Для распределения нагрузки в данном случае необходимо ограничивать количество ресурсов, выше которого каждый микросервис не может использовать. Но, в данном варианте, распределение ресурсов между подсистемами ВМ остаётся на операционной системе, что может привести к неоптимальному балансу и использованию ОЗУ или ядер ЦП и, в последствии, к непрогнозируемому снижению производительности.

## Лицензирование системы

Для корректного запуска системы необходима действующая лицензия, выпущенная на сервере лицензирования производителя ООО «Кросстех Солюшнс Групп» для данной организации.

Лицензия на Систему имеет следующие параметры:

- **Дата начала** (необязательный параметр). Дата начала действия лицензии. В случае, если текущая дата < **даты начала**, то лицензия считается невалидной (в информации о лицензии в таком случае параметр будет выделен красным цветом). При попытке входа на UI с ненаступившей датой начала будет выведено сообщение об истечении срока действия лицензии. Если значение параметра не указано, то дата начала лицензии не ограничена (в информации о лицензии указано 'Не ограничено');
- **Дата окончания** (необязательный параметр). Дата окончания действия лицензии. В случае, если текущая дата > **даты окончания**, то лицензия считается невалидной (в информации о лицензии в таком случае параметр будет выделен красным цветом). При попытке входа на UI с истёкшей датой окончания будет выведено сообщение об истечении срока действия лицензии. Если значение параметра не указано, то дата окончания лицензии не ограничена (в информации о лицензии указано 'Не ограничено');
- **Число уникальных БД-источников** (необязательный параметр). Кол-во уникальных [подключений к БД](#), используемых в качестве БД-источника в [планах обезличивания](#). Уникальный БД-источник определяется следующими атрибутами подключения:
  - Хост;
  - Порт;
  - База данных.



Разные схемы в рамках одной и той же БД считаются одним подключением. Одним **использованием** уникального БД-источника считается **первая загрузка списка таблиц из этой БД-источника в плане обезличивания**.

В системе ведётся счётчик, который подсчитывает использования уникальных БД-источников, и, в случае достижения указанного в лицензии количества, делает невозможным использование следующего БД-источника. В информации о лицензии отображается количество уже использованных БД-источников / разрешённое количество БД-источников. При попытке использовать ещё один БД-источник сверх разрешённого количества будет выведено соответствующее уведомление о том, что все подключения к БД-источникам в рамках лицензии были потрачены и в информации о лицензии данный параметр будет выделен красным цветом. Создание подключений к БД и проверка успешности подключения к ним с соответствующего экрана происходит без увеличения счётчика использований. Если значение параметра не указано, то число уникальных БД-источников не ограничено (в информации о лицензии указано 'Не ограничено');

- **Пробная версия.** Признак того, является ли данная сборка системы пробной (Trial) версией. Пробная версия системы ограничивает функционал системы в части выбора количества таблиц и столбцов БД: в одном плане обезличивания после подключения к БД-источнику и загрузки списка таблиц для дальнейшей работы можно выбрать лишь 1 таблицу и 2 столбца БД;
- **Типы БД** (необязательный параметр). Список типов БД, включенных в лицензию (значения перечисляются через запятую). В системе могут создаваться и использоваться подключения к БД только с данными типами БД. Если ранее уже было создано подключение с типом БД, который не поддерживается в текущей лицензии, то с таким подключением невозможно создать/запустить план обезличивания. Возможные доступные типы БД: PostgreSQL, Vertica, ApacheHive, MySQL, MSSQL, ClickHouse, Oracle. Если значение параметра не указано, то доступны все типы БД (в информации о лицензии указано 'Не ограничено');
- **Лицензия выдана.** Юридическое лицо, кому выдана лицензия. В свободной форме;
- **Когда и кем выдана.** Юридическое лицо, выпустившее лицензию (ООО Кросстех Солюшнс Групп) и дата выпуска данной лицензии. В свободной форме.

Для просмотра информации о лицензии в Системе необходимо открыть пункт меню: [Администрирование](#) → [Информация о лицензии](#).

Параметр	Значение
Дата начала лицензии	Не ограничено
Дата окончания лицензии	2023-02-14 15:00:00
Поддерживаемые типы Баз Данных	ApacheHive, ClickHouse, PostgreSQL, MySQL, Vertica, MSSQL, Oracle, Sybase
Число уникальных БД-источников	Не ограничено
Триал	Нет
Лицензия выдана	Кросстех Солюшнс Групп ООО

После нажатия на пункт меню на экране появляется всплывающее окно с информацией по текущей лицензии.

## Установка лицензии на сервер приложений

При развёртывании системы необходимо разместить лицензию **license.lic** и публичный ключ **public.key** по путям, указанным в `application.properties` микросервиса `DATAPROCESSOR`:

*Пример 1. Параметры лицензии `application.properties`:*

```
license.path\=/app
license.file.path\=${license.path}/license.lic
license.publicKey.path\=${license.path}/public.key
```

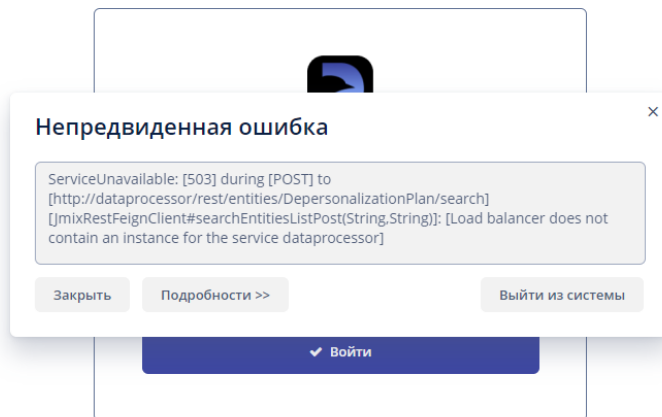
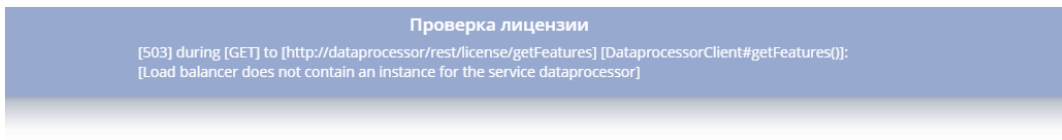
Указаны относительные пути, т.е. путь относительно каталога с `.jar`-файлом микросервиса `DATAPROCESSOR`. Если `dataprocessor.jar` расположен в директории `C:/jay_data/dataprocessor/`, то путь к директории лицензии `C:/jay_data/dataprocessor/app`

## Проверка лицензии

Система проверяет наличие лицензии и её параметры при следующих событиях:

- **Запуск системы.** При запуске `DATAPROCESSOR` система проверяет:
  - Наличие файла лицензии и публичного ключа и их валидность — система проверяет, что `DATAPROCESSOR` был запущен с валидной лицензией;
  - Что **Дата начала** лицензии меньше текущей даты, либо она не ограничена — система проверяет, что `DATAPROCESSOR` был запущен с лицензией с правильной датой начала;
  - Что **Дата окончания** лицензии больше текущей даты, либо она не ограничена — система проверяет, что `DATAPROCESSOR` был запущен с неистёкшей по дате окончания лицензией.

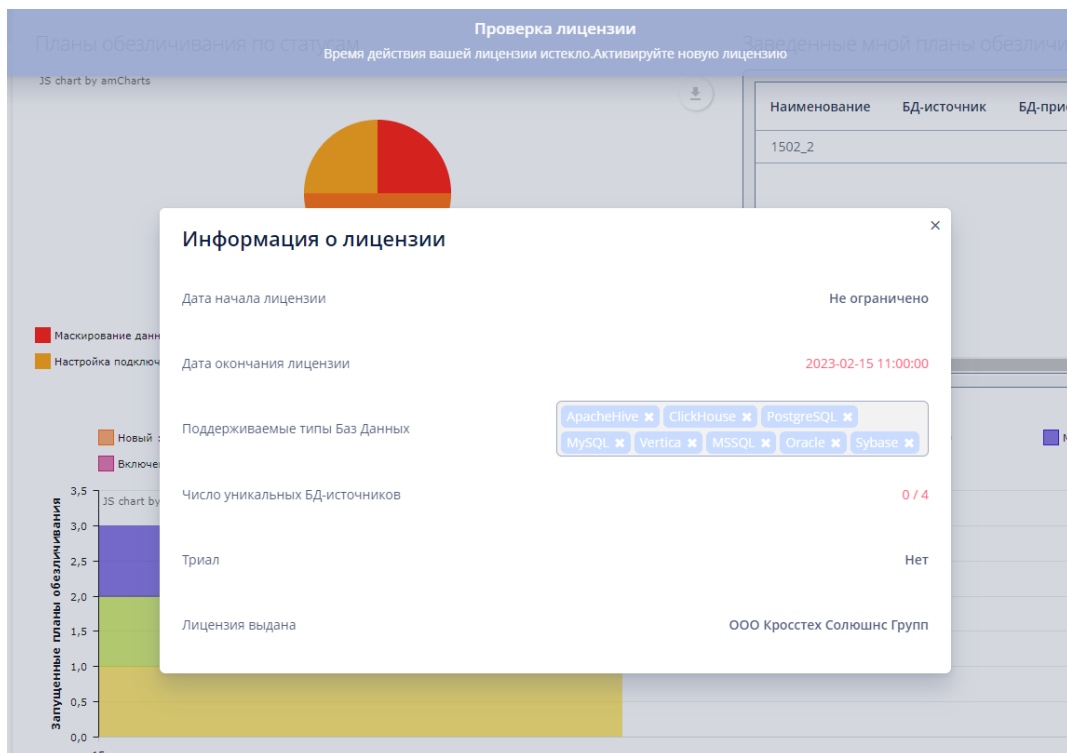
При непрохождении хотя бы одной из данных проверок система не позволит войти в неё с ошибкой запуска `DATAPROCESSOR`:



• **Вход в систему.** При попытке входа в систему она проверяет:

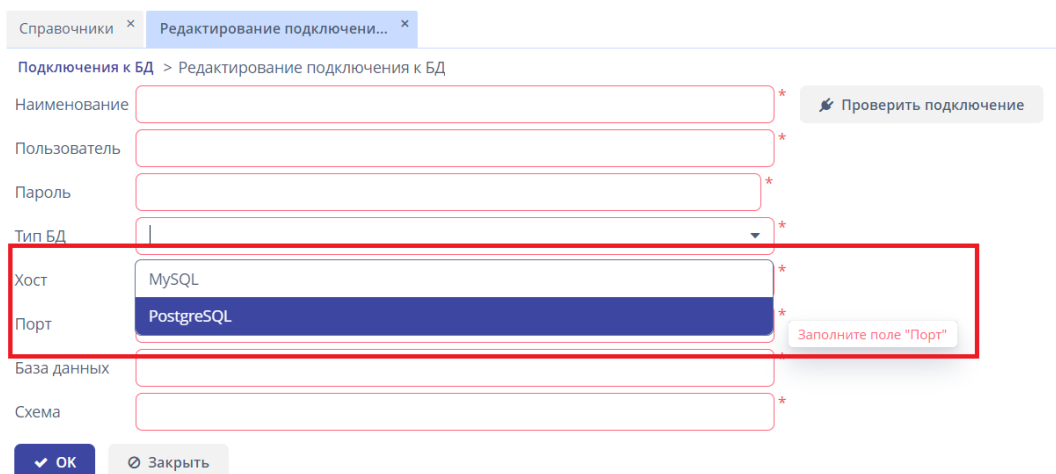
- Что **Дата начала** лицензии меньше текущей даты, либо она не ограничена — система проверяет, что дата начала лицензии уже наступила;
- Что **Дата окончания** лицензии больше текущей даты, либо она не ограничена — система проверяет, что дата окончания лицензии ещё не наступила. Система выдаёт оповещение в виде всплывающей строки за 2 недели до наступления даты окончания лицензии при входе;

При непрохождении хотя бы одной из данных проверок система не позволит войти в неё с оповещением, что лицензия истекла, а также выведет информацию о лицензии с подсвеченной датой, которая истекла:



- **Создание и использование подключений к БД.** При попытке создать или использовать подключение к БД (загрузить список таблиц БД, запустить профилирование, предпросмотр или маскирование) в плане обезличивания система проверяет:
  - Что создаваемое или используемое подключение к БД имеет тип БД из разрешённых лицензией, указанных в **Типы БД**;

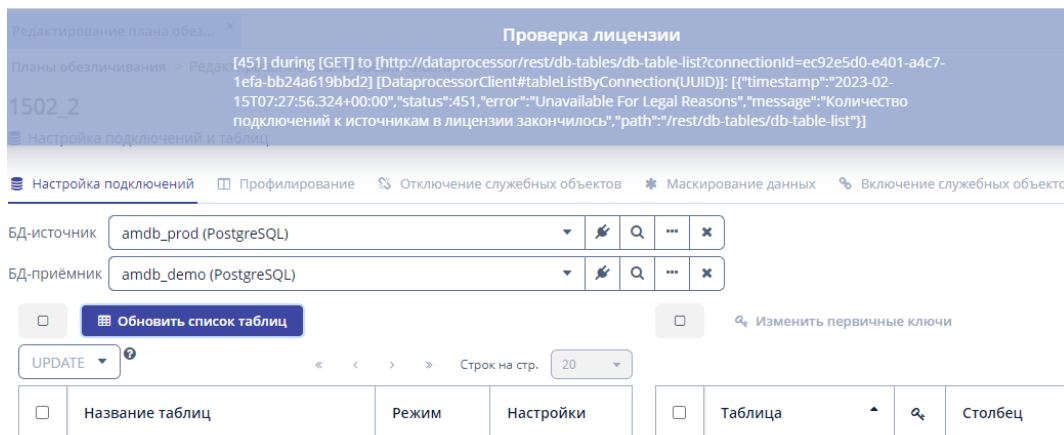
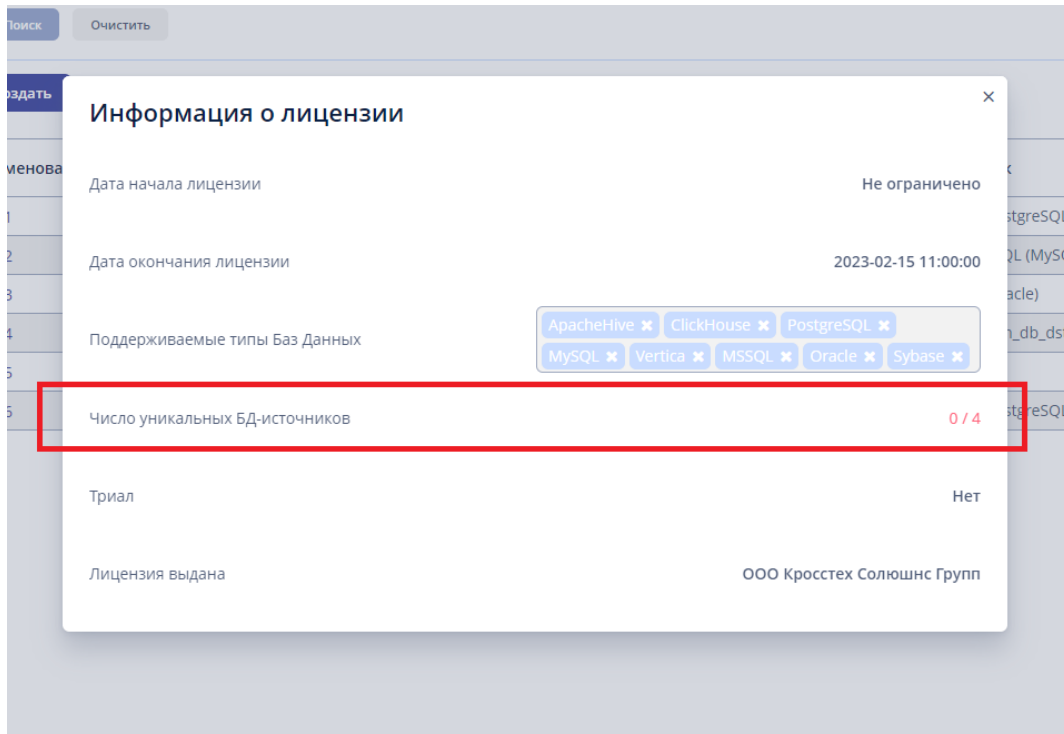
При непрохождении этой проверки система не позволит создать или использовать такое подключение к БД. Дополнительно, все подключения с исключёнными типами БД будут скрыты на UI.



- **Первая загрузка списка таблиц БД.** При попытке первый раз загрузить список таблиц (использовать) подключение к БД в плане обезличивания система проверяет:
  - Что общее количество используемых уникальных подключений к БД-источникам не превышает разрешённое лицензией количество, указанное в **Число уникальных БД-источников**;

При непрохождении этой проверки система не позволит загрузить список таблиц и

столбцов из нового подключения к БД.



- **Запуск профилирования, предпросмотра, маскирования.** При попытке запустить данные процессы система проверяет:
  - Если лицензия является пробной (признак **Trial**), то разрешено запускать данные процессы не более чем для 1 таблицы и 2 столбцов БД. Если лицензия не является пробной, то ограничений нет.

При непрохождении этой проверки система не позволит запустить процесс и оповестит пользователя о превышении количества выбранных таблиц и/или столбцов.

Профилерование   Отключение служебных объектов   Маскирование данных   Включение

ачать профилерование   Скрыть таблицу   В процессе   Ошибка

### Ошибка профилерования

[451] during [POST] to [http://dataprocessor/rest/profiling?planId=248125bed16-9e4c-b886-2c9f53a09067&forceUpdateDomains=false]  
[DataprocessorClient#startProfiling(String,Boolean)]: [{"timestamp":"2023-02-15T09:48:09.981+00:00","status":451,"error":"Unavailable For Legal Reasons","message":"У вас пробная лицензия.Выберите пожалуйста 1 таблиц и 2 столбцов","path":"/rest/profiling"}]

Копировать   **Закреть**

Столб			
middle			име
last_na			
name			
first_n			
login_l			
passw			
passw			
login			
inn	STRING (varchar)		

# Использование

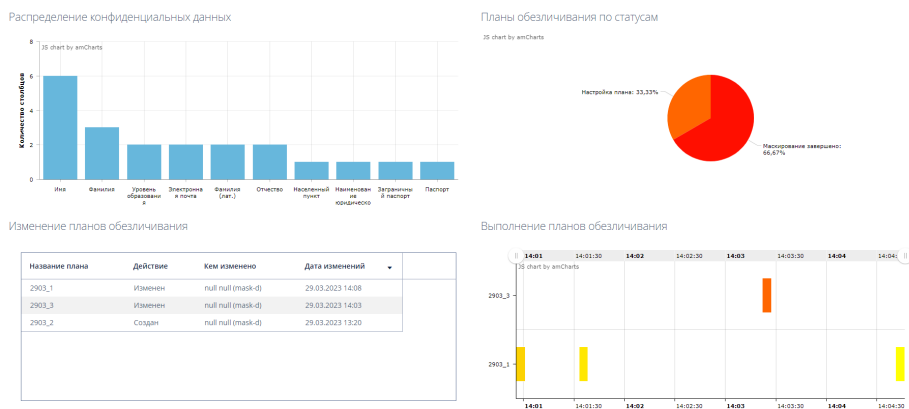
В данном разделе приведено описание общих операций для **пользователя** Системы.

## Вход в систему

Пользователь может войти в систему с помощью доменного логина и пароля или с помощью административной учётной записи **admin**.

## Стартовый экран

После входа в Систему по умолчанию открывается стартовый экран. На нем отображаются заранее настроенные дашборды, которые показывают статистику по планам обезличивания.



Дашборды являются **кликабельными**, т.е. при нажатии на какой-либо из столбцов/секторов диаграммы будет открыт список сущностей, который формируют данный столбец.

## Панель навигации

В левой части экрана расположена Панель навигации, через которую осуществляется доступ к сущностям и экранам Системы.

Пункты панели являются **разделами**, которые содержат в себе соответствующие подразделы. Разделы навигационной панели раскрываются при нажатии на соответствующую секцию. При повторном нажатии раздел сворачивается.

В верхней части панели располагается наименование системы. В нижней части находится логин текущего Сотрудника, часовой пояс, кнопка сворачивания Панели навигации и кнопка выхода из Системы.

Если нажать на кнопку **сворачивания панели**, то Панель навигации будет открываться только при наведении на неё курсора мыши, а при отведении курсора панель будет сворачиваться. Такой эффект предоставляет более широкую рабочую область.

В случае, если необходимо зайти в Систему под другим пользователем (не под своей



текущей доменной записью) или выйти из нее, необходимо нажать на кнопку **Выйти** в правом нижнем углу навигационной панели.



Возможен вход также под доменными учётными данными другого пользователя. Для этого необходимо ввести их в указанные поля логина/пароля (без домена).



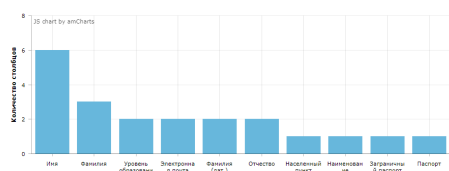
При входе в Систему при вводе логина/пароля вручную (как описано выше) в момент входа будет произведена синхронизация данных учётной записи с Active Directory и при входе будут назначены соответствующие пользовательской роли права.

## Дашборды

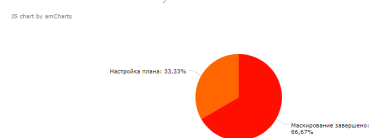
**Дашборд** – компонент, который визуализирует информацию, наборы данных, диаграммы и графики. Доступен только для авторизованных пользователей. Адаптивная компоновка способна подстроить дашборд под различные размеры монитора.

**Виджеты** — графические элементы, из которых состоит дашборд. Система имеет преднастроенный дашборд с виджетами, доступный через пункт меню **Панель мониторинга**, а также на главной странице системы. Панели мониторинга представляет собой сводные отчеты о планах, объектах обезличивания данных и объектах администрирования. В панели мониторинга возможно просмотреть распределение доменов данных, планов и их статусы.

Распределение конфиденциальных данных



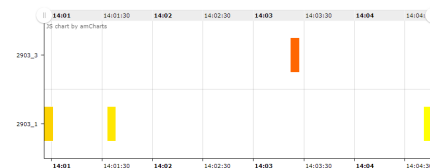
Планы обезличивания по статусам



Изменение планов обезличивания

Название плана	Действие	Кем изменено	Дата изменений
2903_1	Изменен	ruil null (mask-d)	29.03.2023 14:08
2903_3	Изменен	ruil null (mask-d)	29.03.2023 14:03
2903_2	Создан	ruil null (mask-d)	29.03.2023 13:20

Выполнение планов обезличивания

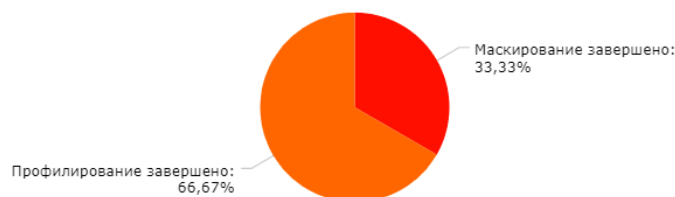


## Планы обезличивания по статусам

Отображается в виде круговой диаграммы, которая показывает, какое количество планов обезличивания заведено в системе в процентах в разбивке по статусам.

## Планы обезличивания по статусам

JS chart by amCharts



При нажатии на статус осуществляется переход на раздел [Планы обезличивания](#) с наложенным фильтром по статусу.

### Изменение планов обезличивания

На графике отображаются 10 последних созданных или изменённых планов обезличивания. Для отображения данных используется простая таблица.

Изменение планов обезличивания

E

Название плана	Действие	Статус	Кем изменено
plan02	Изменен	Профилирование завершено	Екатерина Чернышова (chernysho
plan01	Изменен	Профилирование завершено	Екатерина Чернышова (chernysho

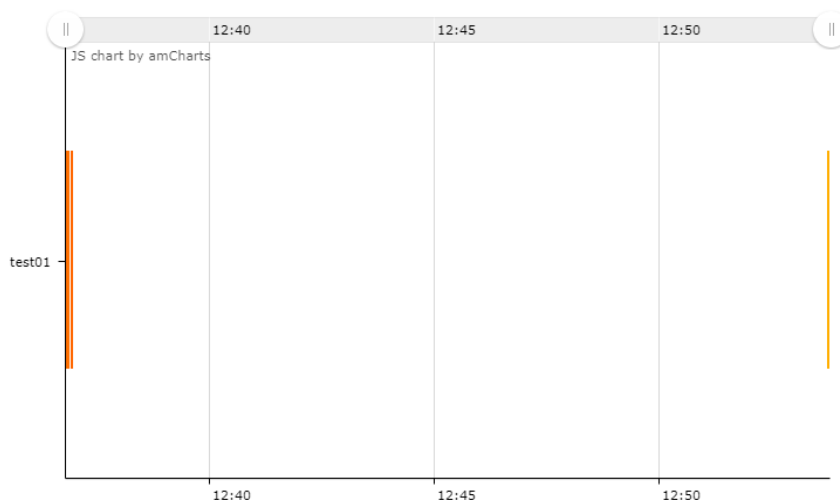
На панели таблицы отображаются следующие столбцы:

- **Название плана** – имя измененного плана;
- **Действие** – тип влияния на план (Изменение, Создание);
- **Статус** – статус плана обезличивания;
- **Кем изменено** - имя пользователя, внесшего изменение;
- **Дата изменения** – дата создания, изменения или удаления плана.

## Выполнение планов обезличивания

На графике отображаются данные о выполненных планах обезличивания в привязке к конкретной дате и времени. График отображает время начала и конца процесса обезличивания данных. На оси x диаграммы Ганта отображено время выполнения за последние 3 дня, на оси y последние 10 активных планов обезличивания. По умолчанию отображается текущая дата. Если необходимо посмотреть информацию в разрезе часов/минут, то для этого реализован специальный ползунок наверху графика. Минимальный шаг шкалы (в случае максимального приближения) ограничен одной минутой.

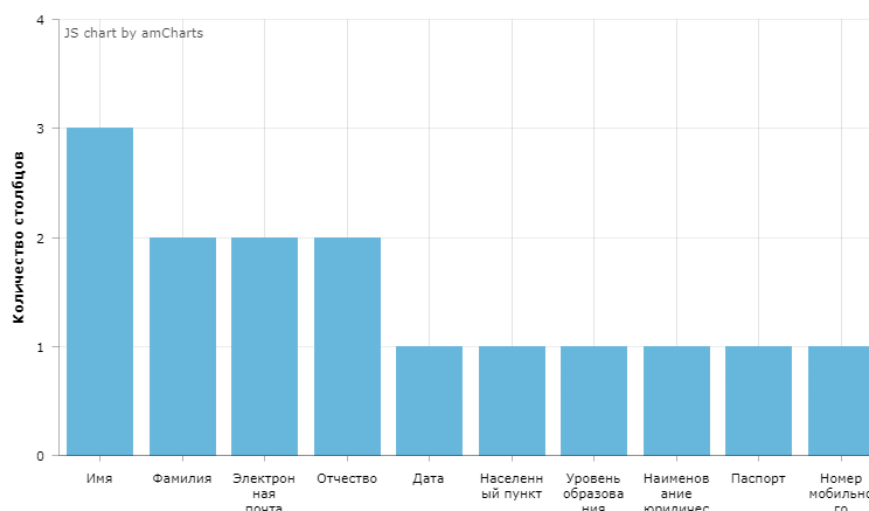
Выполнение планов обезличивания



## Распределение конфиденциальных данных

На графике отображается топ-10 доменов по их количеству во всех заведенных планах в виде столбчатой диаграммы.

Распределение конфиденциальных данных



На оси x отображены домены, на оси y их найденное в процессе профилирования количество во всех планах обезличивания. По умолчанию домены отсортированы в

порядке убывания.

## Экраны

**Экран** – это основной интерфейс взаимодействия пользователя с сущностями в системе. Экраны могут быть разных типов, с разным набором кнопок и компонентов, но имеют общее назначение: вывод и корректировка информации о каких-либо сущностях.

### Типы экранов

Экраны сущностей в Системе могут быть 2-х типов:

- **Карточка.** Отображает детальную информацию об отдельной записи с возможностью отредактировать её атрибуты (при наличии доступа).

Типы данных БД > Редактирование типа данных БД

Наименование  \*

Тип БД  \*

Тип данных  \*

- **Список.** Отображает таблицу записей без возможности отредактировать их атрибуты. Для открытия **Карточки** конкретной записи необходимо выбрать нужную запись галочкой и нажать кнопку **Изменить** в верхней части таблицы. Вторым вариантом открытия карточки является нажатие на значение в кликабельной колонке таблицы (в большинстве экранов это первая колонка).

Фильтр

Наименование  Системное имя  Тип данных

Пользовательские \* Системные

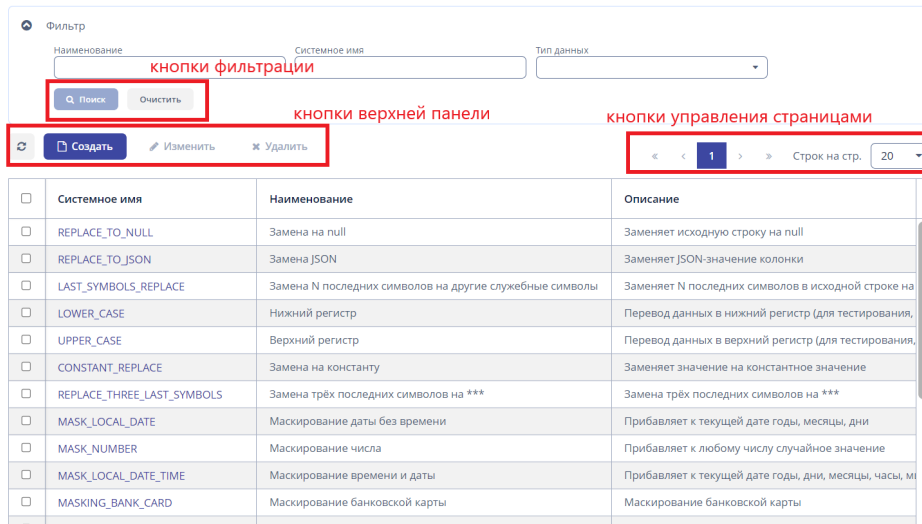
Просмотр

<input type="checkbox"/>	Системное имя	Наименование	Описание
<input type="checkbox"/>	REPLACE_TO_NULL_STRING	Замена на null (строка)	Заменяет исходную строку на null
<input type="checkbox"/>	RANDOM_REPLACEMENT_FROM_DB_STRING	Замена на случайное значение из БД (строка)	Метод, заменяющий значение на случайное из той же колонки базы де
<input type="checkbox"/>	RANDOM_REPLACEMENT_FROM_DB_BOOLEAN	Замена на случайное значение из БД (логический)	Метод, заменяющий значение на случайное из той же колонки базы де
<input type="checkbox"/>	RANDOM_REPLACEMENT_FROM_DB_DATE	Замена на случайное значение из БД (дата)	Метод, заменяющий значение на случайное из той же колонки базы де
<input type="checkbox"/>	RANDOM_REPLACEMENT_FROM_DB_NUMBER	Замена на случайное значение из БД (число)	Метод, заменяющий значение на случайное из той же колонки базы де
<input type="checkbox"/>	REPLACE_TO_NULL_BLOB	Замена BLOB на null	Заменяет BLOB на null

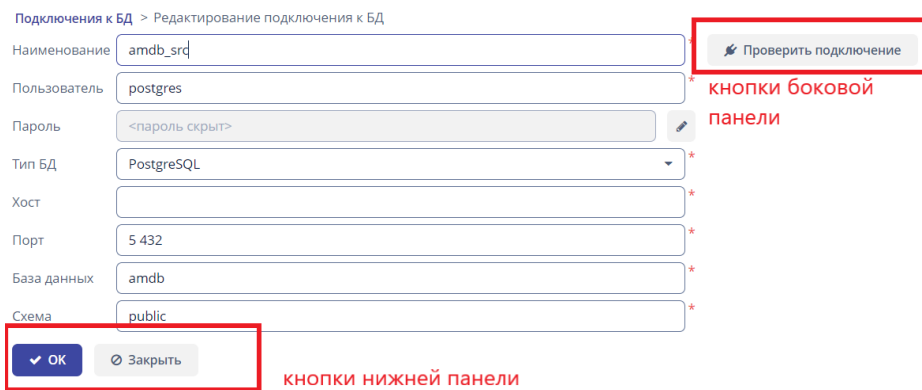
## Кнопки

Экраны могут содержать в себе различные кнопки, выполняющие какие-либо действия. На различных типах экранов кнопки могут располагаться в разных местах и иметь разный тип. Описание типа элемента 'Кнопка' доступно в инструкции **Jmix** по [ссылке](#).

В системе присутствуют свои блоки кнопок, отвечающие за фильтрацию, управление страницами, управление списком записей (кнопки верхней панели).



В карточке редактирования также есть кнопки нижней панели **Сохранить** и **Отмена**, которые сохраняют и сбрасывают изменения соответственно. Есть также кнопки боковой панели.



## Поля

Поля на экранах могут быть разных типов. Описание всех типов полей доступно в инструкции **Jmix** по [ссылке](#).

- Текстовые и числовые поля представляют собой **TextField** (для однострочных) / **TextArea** (для многострочных);
- Текстовые поля, содержащие **Groovy-скрипты** или **JPQL-запросы** представляют собой **SourceCodeEditor**;
- Поля-признаки представляют собой **CheckBox**;
- Поля с перечислениями представляют собой **LookupField**. В данном типе поля всегда работает автозаполнение;
- Поля с датой и временем представляют собой **DateField**;
- Ссылочные поля представляют собой **LookupPickerField** В большинстве случаев такие поля имеют 3 действия слева направо:
  - **Open**. Открыть карточку связанной записи;
  - **Lookup**. Открыть экран 'Список' для поиска и выбора связанной записи;
  - **Clear**. Очистить поле.

**EntityPicker.** Ссылка на созданную сущность в Системе.

У некоторых полей может присутствовать **контекстная подсказка**, которая открывается при наведении на поле.

Системное имя	Наименование
REPLACE_TO_NULL_STRING	Замена на null (строка)
RANDOM_REPLACEMENT_FROM_DB_STRING	Замена на случайное значение из БД (строка)

## Списки

**Список** – это таблица, отображающая набор записей, относящихся к той или иной сущности. Список позволяет искать записи, просматривать, фильтровать и открывать их экраны.

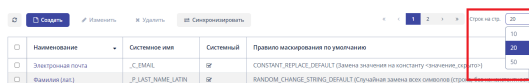
### Типы списков

Списки в Системе являются **плоскими** (не содержащими дочерние элементы). Список отображается в виде плоского перечня записей без иерархии.

Наименование	Системное имя	Метод маскирования по умолчанию
Улица	C_STREET	LAST_SYMBOLS_REPLACE (Заменяет N последних символов в исходной строке на другие уик)
Уровень образования	P_EDUCATION_LEVEL	LAST_SYMBOLS_REPLACE (Заменяет N последних символов в исходной строке на другие уик)
Филозофские взгляды	P_PHILOSOPHICAL_VIEWS	LAST_SYMBOLS_REPLACE (Заменяет N последних символов в исходной строке на другие уик)
Паспорт	D_PASSPORT	LAST_SYMBOLS_REPLACE (Заменяет N последних символов в исходной строке на другие уик)
ИИН физического лица	D_TIN	MASKING_INN_RUS_STRING (Маскирование номера ИИН (РФ))
ИИН юридического лица	D_TIN	MASKING_INN_RUS_STRING (Маскирование номера ИИН (РФ))
Номер мобильного телефона	C_PHONE	LAST_SYMBOLS_REPLACE (Заменяет N последних символов в исходной строке на другие уик)
Адрес электронной почты	C_EMAIL	CONSTANT_REPLACE (Заменяет значение на константное значение)
Отчество	P_PATRONOMIC	LAST_SYMBOLS_REPLACE (Заменяет N последних символов в исходной строке на другие уик)

### Количество записей в списках

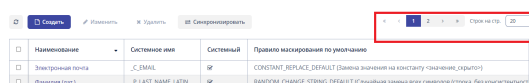
Система отображает определённое количество записей на одной странице в списке при его открытии. Для управление количеством в верхней части экрана в меню необходимо выбрать необходимое числовое значение.



Наименование	Системное имя	Системный	Правило маскирования по умолчанию
Электронная почта	C_EMAIL	☑	CONSTANT_REPLACE_DEFAULT (Замена значения на константу значения_скрыто)
Фамилия (лат)	P_LAST_NAME_LATIN	☑	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов строки)

Если всего в списке записей больше, чем отображается на текущей странице, то при открытии общее количество записей будет отображаться как **?**. Для подсчёта общего числа записей в списке необходимо нажать на данный значок.

Списки поддерживают **постраничное отображение**. Панель перехода на следующую/предыдущую страницу списка выглядит следующим образом:



Наименование	Системное имя	Системный	Правило маскирования по умолчанию
Электронная почта	C_EMAIL	☑	CONSTANT_REPLACE_DEFAULT (Замена значения на константу значения_скрыто)
Фамилия (лат)	P_LAST_NAME_LATIN	☑	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов строки, без константности)

### Множественный выбор и редактирование

Пользователь не может отредактировать сразу несколько элементов в списке со множественным выбором. Для того, чтобы отредактировать несколько элементов в списке, необходимо отредактировать каждый из них по отдельности:

- Поставить галочку напротив необходимого элемента;
- Нажать на кнопку **Изменить**.



Если выбрать несколько элементов во множественном списке кнопка **Изменить** будет недоступна.

## Сортировка списков

В верхней части каждой колонки списка расположена **кнопка** заголовка с именем атрибута, который отображается в данной колонке. Эта кнопка позволяет выполнить сортировку значений списка по указанному атрибуту.

Для сортировки значений колонки в порядке возрастания необходимо один раз нажать на заголовок колонки.

<input type="checkbox"/>	Наименование	Системное имя	Системный	Правило сортировки по умолчанию
<input type="checkbox"/>	id	UUID	<input type="checkbox"/>	REPLACE_TO_NULL_DEFAULT (Заменяет исключенную строку на null)
<input type="checkbox"/>	IMB	_C_IMB	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_FROM_8_CHAR (Случайная замена с 8 символов строка, без констант)
<input type="checkbox"/>	IMSI	_C_IMSI	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_FROM_8_CHAR (Случайная замена с 8 символов строка, без констант)
<input type="checkbox"/>	IP адрес	_C_IP	<input checked="" type="checkbox"/>	CONCATENATE_REPLACE_DEFAULT (Вставка значения на константу (значение_ссылки))
<input type="checkbox"/>	IP адрес	_C_IP	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_FROM_8_CHAR (Случайная замена с 8 символов строка, без констант)
<input type="checkbox"/>	URL	_C_URL	<input type="checkbox"/>	RANDOM_CHANGE_STRING_FROM_8_CHAR (Случайная замена с 8 символов строка, без констант)
<input type="checkbox"/>	Дата	_C_DATE	<input checked="" type="checkbox"/>	RANDOM_MODIFY_DATE (добавление и удаление случайного кол-ва дней от 1 до 10)
<input type="checkbox"/>	Зеркальный паспорт	_D_INTERNATIONAL_PASSPORT	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов строки, без констант)
<input type="checkbox"/>	Имя	_F_FIRST_NAME	<input checked="" type="checkbox"/>	REPLACE_BY_DICTIONARY_STRING_FIRST_NAME (Замена по справочнику имени FN, без констант)
<input type="checkbox"/>	Имя (лат.)	_F_FIRST_NAME_LATIN	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов строки, без констант)
<input type="checkbox"/>	ИИН физического лица	_D_TIN	<input checked="" type="checkbox"/>	INN_RUS_STRING_DEFAULT (ИИН РФ строка, без констант)
<input type="checkbox"/>	ИИН юридического лица	_D_TIN	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов строки, без констант)

Если нажать кнопку заголовка дважды, сортировка будет произведена в порядке убывания (в обратном порядке).

При тройном нажатии сортировка будет сброшена.

## Фильтры и поиск

### Фильтрация в Системе

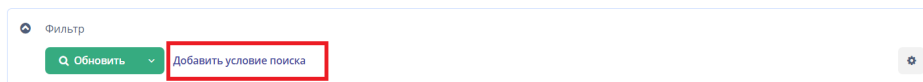
Окно с фильтрами представляет собой набор полей по которым будет осуществляться поиск по Системе. Для каждого подраздела из пункта меню предусмотрено окно с фильтрами. Доступны как поля с ручным вводом значений, так и поля с выпадающим списком, предлагающим варианты для выбора.

Для того чтобы воспользоваться поиском по фильтрам, необходимо ввести/выбрать значение и нажать на кнопку **Поиск**. Для того чтобы убрать значения из полей, нужно нажать на кнопку **Очистить**.

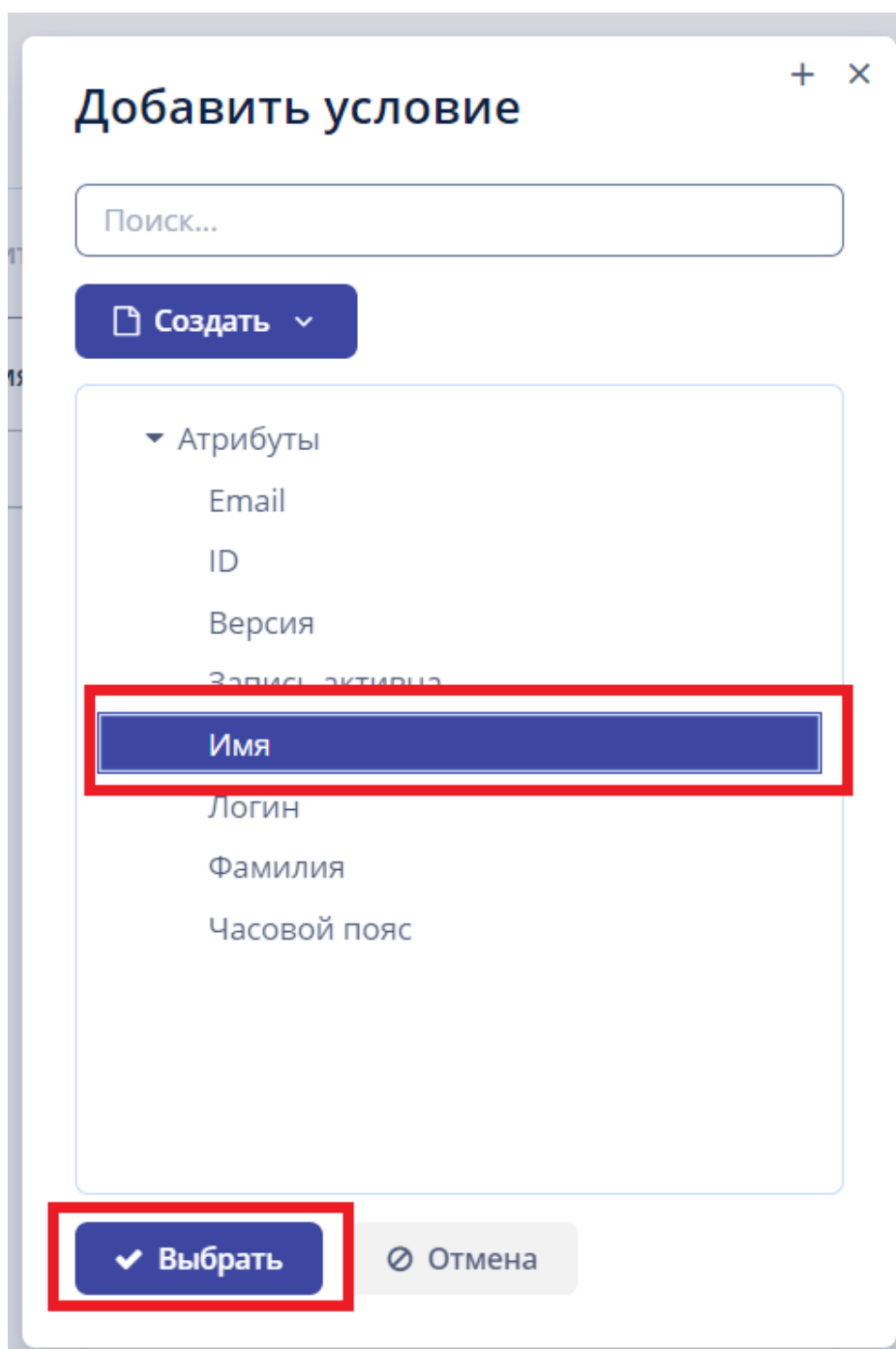
## Фильтрация с условием поиска

В разделе **Пользователи** и **Дашборды** настроены расширенные фильтры, отличные от остальных фильтров в Системе.

В таких фильтрах можно настроить дополнительное условие поиска по кнопке **Добавить условие поиска** в верхней части экрана.



В открывшемся окне необходимо выбрать атрибут, по которому нужно отфильтровать записи и нажать на кнопку **Выбрать**.



В верхней части экрана появляется фильтр по выбранному атрибуту с условием **Содержит**.



Фильтр

Имя   ✕

Добавить условие поиска

Можно добавить несколько условий по вышеописанному принципу.

Для поиска конкретной записи необходимо нажать на кнопку **Обновить**. Если нажать на стрелочку под кнопкой **Обновить**, то появится кнопка, которая сбросит все настроенные фильтры.

Фильтр

Имя   ✕

Добавить условие поиска

Назначения ролей

# Приложение

В данном разделе приведено описание всех пунктов основного меню системы.

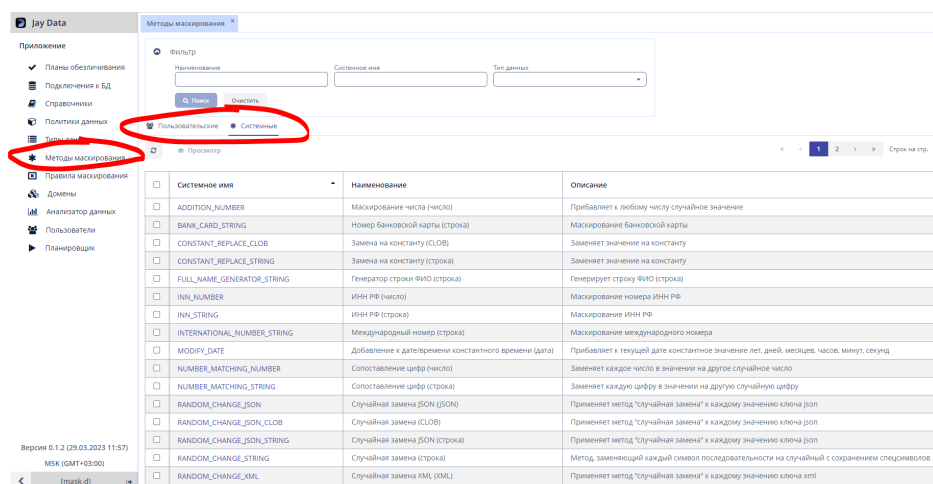
**Маскирование** предполагает создание **плана обезличивания**, в рамках которого выполняется набор действий, требуемый для обезличивания баз данных, хранящихся на физических носителях.

## Методы маскирования

**Метод маскирования** - это сущность, определяющая логику, по которой преобразовываются обезличиваемые данные.

Существуют **системные** (поставляемые вместе с дистрибутивом системы) и пользовательские (создаваемые пользователем системы) методы. Системные методы нельзя удалять или изменять. На основе обоих типов методов можно создавать **правила маскирования**.

Список методов маскирования доступен в разделе меню **Методы маскирования**.



Карточка метода маскирования содержит в себе следующие поля и разделы:

- **Наименование** - название метода маскирования;
- **Системное имя** - системный код метода маскирования;
- **Описание** - описание метода маскирования;
- **Тип данных** - **тип данных**, к которому привязан метод маскирования. Каждый метод маскирования привязан к определённому типу данных. Это означает, что к столбцам БД с определённым типом данных (например, STRING) можно применить только методы маскирования, которые работают с типом данных STRING;
- **Список параметров** - таблица с **параметрами** метода маскирования;
- **Список зависимостей** - таблица с **зависимостями от других столбцов БД**;
- **Скрипт преобразования** - Groovy-скрипт, определяющий логику преобразования исходного значения. Скрипт может использовать все возможности языка Groovy и доступен для изменения только наиболее опытным пользователям Системы, а также её

администраторам.

Скрипт всегда **принимает**:

- **исходное значение** (переменная **value**) - оригинальное значение из столбца БД до маскирования;
- **<переменные\_с\_названием\_параметра>** (определённые в соответствующем блоке) - значения **параметров** доступны в переменных скрипта, которые называются как сам параметр (например, `constantValue`). Тип переменной определяется типом параметра.
- **<переменные\_с\_названием\_другого\_столбца\_БД>** - переменные зависимостей, которые будут получены из других столбцов БД (см. **зависимости от других столбцов БД**). Такие значения доступны по имени заданных переменных. Тип переменной определяется типом столбца БД, из которого будет взято значение;

Скрипт **возвращает** маскированное значение того же типа после ключевого слова **return**.

*Пример 2. Пример скрипта метода **Замена на константу**:*

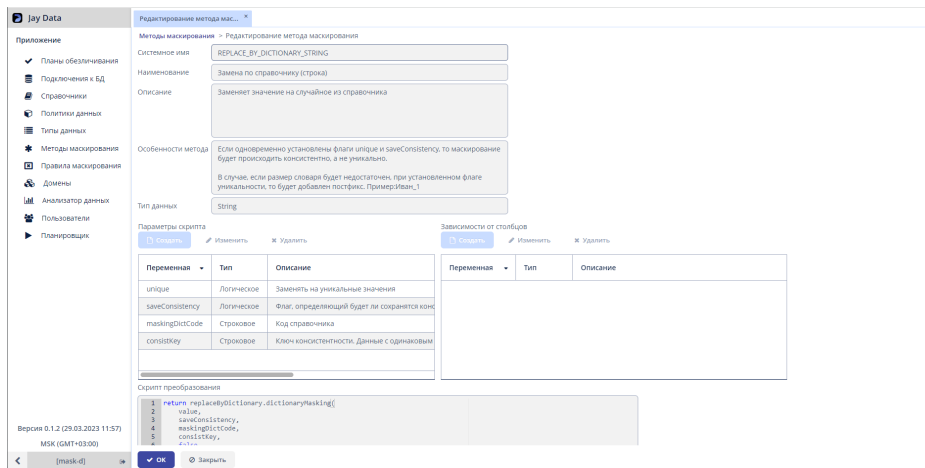
```
import org.apache.commons.lang.*

if (BooleanUtils.isTrue(ignoreEmpty)) {
    if (StringUtils.isBlank(value))
        return value
}

return constantValue
```

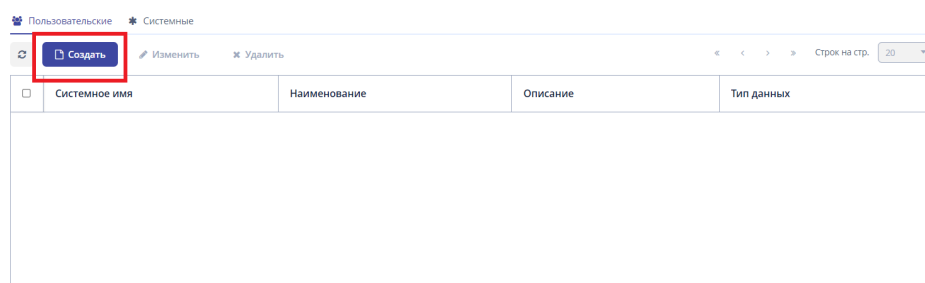
В данном примере проверяется условие, игнорировать ли пустые значения (условие задаётся булевым параметром **ignoreEmpty**). Если условие = 'true', то в случае пустого значения в столбце БД функция возвращает оригинальное (пустое) значение (которое содержится в переменной **value**), но если условие = 'false', то функция возвращает константу, которая задана параметром **constantValue**.

- **Тестирование скрипта маскирования** - блок, отвечающий за проверку работы метода. Содержит в себе поле для ввода оригинального значения и кнопку **Тест**, с помощью которой к нему можно применить скрипт преобразования и во всплывающем окне увидеть результат. В качестве значений параметров при этом будут взяты значения по умолчанию.

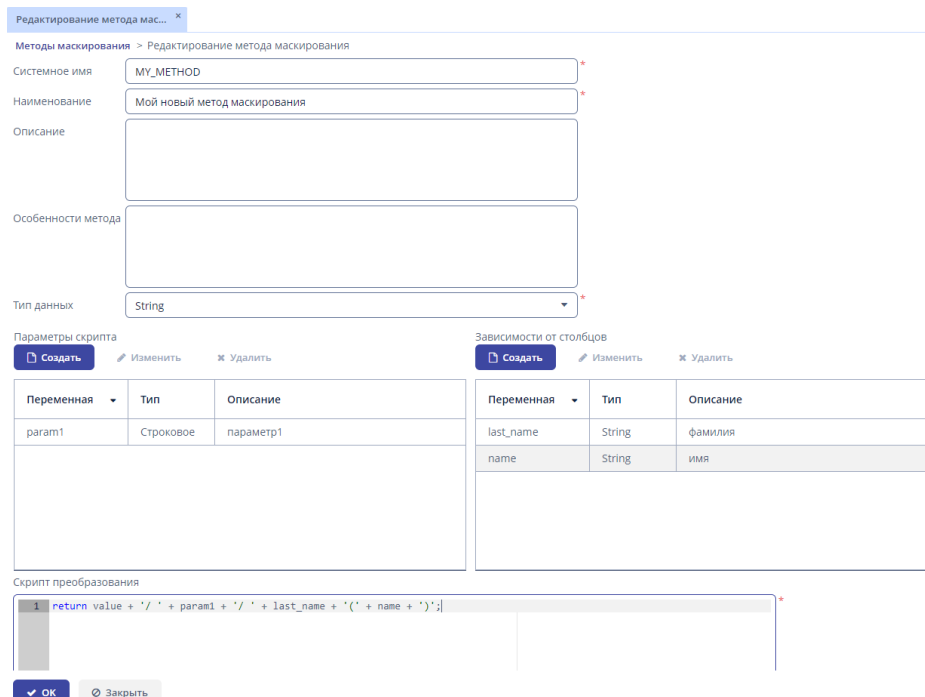


## Создание нового метода маскирования

Для создания нового метода маскирования необходимо перейти в раздел **Методы маскирования** и нажать на кнопку **Создать** под фильтрами.



В открывшемся экране редактирования метода маскирования пользователю необходимо заполнить обязательные поля карточки, указанные в разделе **методы** и нажать кнопку **OK**.



## Параметры методов маскирования

Параметры методов маскирования - переменные, которые могут быть использованы в

Groovy-скрипте Метода маскирования.

В каждый метод маскирования передаётся свой уникальный набор параметров. Набор параметров определяется при создании метода маскирования, а их значения заполняются на этапе [настройки правил маскирования](#) в плане обезличивания. При создании [правила маскирования](#), основанного на методе, необходимо указать значения параметров для данного правила по умолчанию.

### Пример 3. Использование параметров в методах маскирования

При необходимости заменить последние N букв в значении на служебные символы, пользователь может воспользоваться любым из правил, основанном на соответствующем системном методе, например, [Замена N символов на служебные](#). Если значение параметра **количество символов** у этого метода будет установлено = 5, **символ** = ?, параметр **fromEnd** = true, то фамилия 'Иванов' будет преобразована в 'И????'. В этом случае необходимо воспользоваться правилом [замена 5 последних символов на ?](#), где параметры установлены указанным образом (5, ?, true соответственно). А если **количество символов** будет установлено = 2, **символ** = #, параметр **fromEnd** = true, то 'Иванов' **будет преобразовано в 'Иван#'**. В таком случае необходимо воспользоваться правилом [замена 2 последних символов на #](#), где параметры установлены именно так (2, #, true соответственно). Названия правила приведены для примера, каждый пользователь может создать себе собственные правила со своим набором параметров и назвать правила как посчитает необходимым.

## Создание нового параметра

Для создания нового параметра необходимо перейти на вкладку редактирования выбранного метода маскирования и нажать на кнопку **Создать** над таблицей с параметрами.

Редактирование метода мас... x

Методы маскирования > Редактирование метода маскирования

Системное имя: MY\_МЕТОД \*

Наименование: Мой новый метод маскирования \*

Описание:

Особенности метода:

Тип данных: String \*

Параметры скрипта

Переменная	Тип	Описание
param1	Строковое	параметр1

Зависимости от столбцов

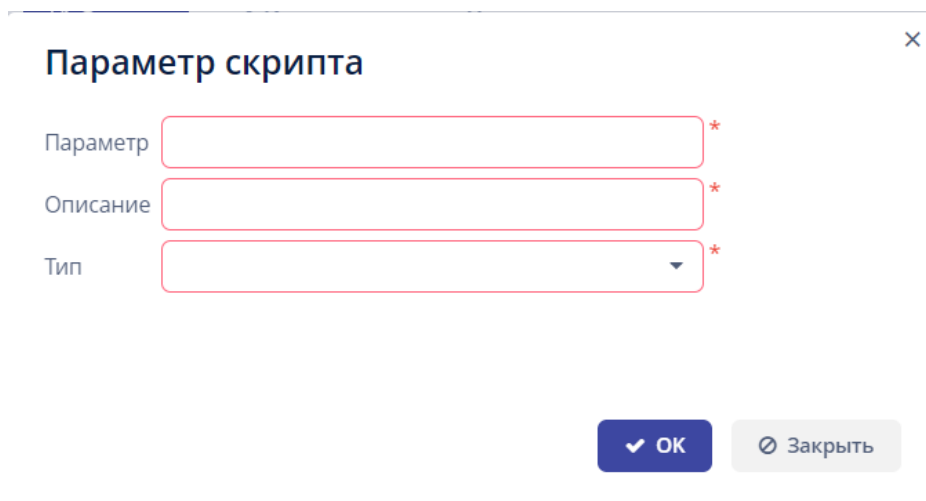
Переменная	Тип	Описание
last_name	String	фамилия
name	String	имя

Скрипт преобразования

```
1 return value + '/' + param1 + '/' + last_name + '(' + name + ')';
```

OK Закрыть

Открывается страница создания параметра скрипта.



Параметр скрипта

Параметр \*

Описание \*

Тип \*

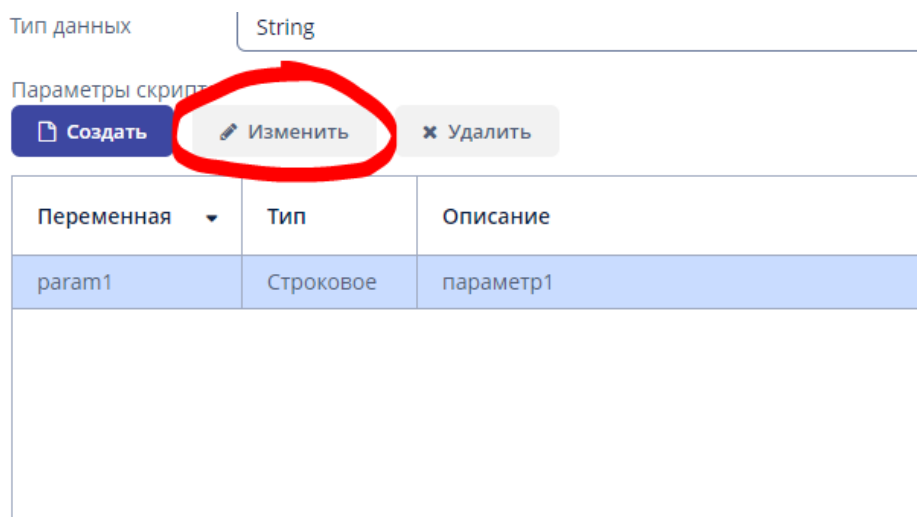
✓ OK    ✕ Закрыть

Необходимо заполнить следующие обязательные поля:

- **Переменная** — переменная параметра скрипта;
- **Описание** — краткое описание параметра;
- **Тип** — выпадающий список для выбора типа данных параметра. Параметры могут быть 3 типов: строковые (STRING), логические (BOOLEAN) и числовые (NUMBER);

Для сохранения нового параметра необходимо нажать **OK**.

Для изменения параметра метода необходимо выбрать нужный параметр и нажать кнопку **Изменить**.



Тип данных String

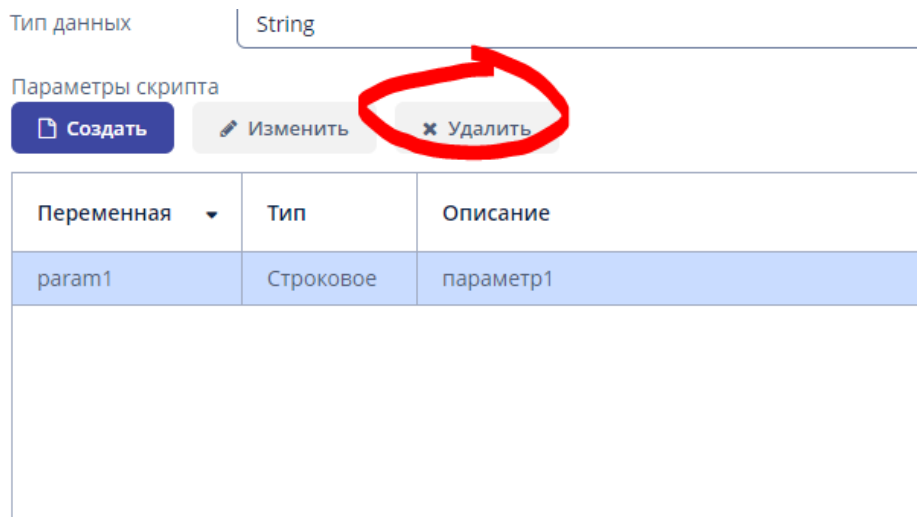
Параметры скрипта

Создать    **Изменить**    Удалить

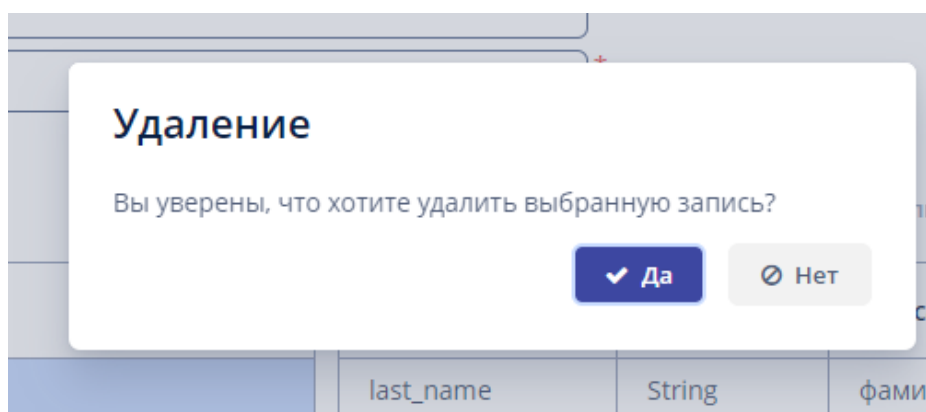
Переменная	Тип	Описание
param1	Строковое	параметр1

Для сохранения изменений необходимо нажать **OK**.

Для удаления параметра метода необходимо выбрать нужный параметр и нажать кнопку **Удалить**.



Во всплывающем окне подтверждения удаления ответить **Да**.



### Консистентность данных

**Консистентность данных** - это один из параметров методов маскирования, который обеспечивает согласованность и целостность маскируемых данных. Используя параметры консистентности, значения, которые повторяются в БД-источнике, маскируются одинаково. Фактически, параметры консистентности гарантируют, что при одном ключе консистентности одинаковые значения А будут единообразно преобразованы в Б и никакие другие значения больше не будут преобразованы в Б.

#### Пример 4. Пример маскирования фамилий с консистентностью

При использовании правила маскирования Замена по словарю российских фамилий для обезличивания фамилий `LAST_NAME` пользователей в таблице `USER` и `EMPLOYEE` можно задать параметр консистентности `saveConsistency = 'Да'` и `consistKey = 'key1'`. Так как у таблиц указан одинаковый ключ консистентности, при таких настройках в обеих таблицах фамилия 'Антонов' замаскируется в 'Иванов', фамилия 'Петров' - в 'Игнатъев' и так далее. Если в столбце БД `LAST_NAME` одной из таблиц БД-источника несколько раз будет встречаться фамилия 'Антонов', она всегда будет маскироваться в 'Иванов', а фамилия 'Петров' - в 'Игнатъев'. Добавим третью и четвертую таблицу `RESIDENT` и `AUTHOR` со столбцом `LAST_NAME`, в которых тоже могут содержаться фамилии. Зададим у правила, указанного для этого столбца параметр консистентности `saveConsistency = 'Да'` и `consistKey = 'key2'`. При таких настройках консистентность будет отдельно соблюдаться в рамках столбца `LAST_NAME` таблиц `USER` и `EMPLOYEE` и между ними ('Антонов' → 'Иванов',

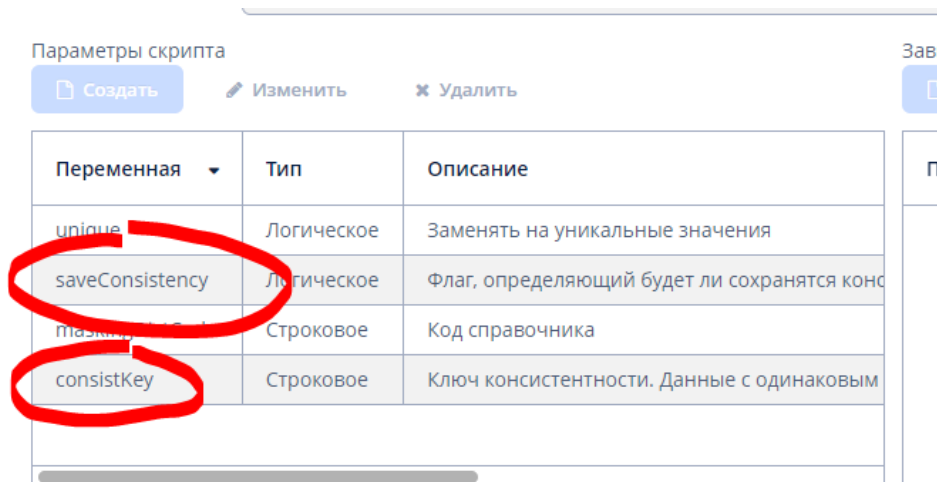
'Петров' → 'Игнатьев') и отдельно в рамках `LAST_NAME` таблиц `RESIDENT` и `AUTHOR` ('Антонов' будет маскироваться уже в другую фамилию, например в 'Селезнёв', а 'Петров' - в 'Енотов').

#### Пример 5. Пример маскирования номеров банковских карт с консистентностью

Параметры консистентности также содержатся в других методах, например [маскирование номера банковской карты](#). Номер карты `1111 1111 1111 1111` система замаскирует в условный `2222 2222 2222 2222`, и если такой номер карты будет встречаться в последующих записях, то он всегда будет маскироваться в `2222 2222 2222 2222` и никакой другой номер в `2222 2222 2222 2222` в рамках ключа консистентности замаскирован не будет.

### Параметры, обеспечивающие консистентность

- `saveConsistency` — признак, определяющий сохранение консистентности в рамках заданного ключа консистентности;
- `consistKey` — ключ консистентности, по которому происходит синхронизация при включенном параметре `saveConsistency`. В параметрах каждого метода маскирования определён свой ключ консистентности. Он хранит информацию о том, между какими столбцами БД сохраняется консистентность.



Параметры скрипта

Создать Изменить Удалить

Переменная	Тип	Описание
unique	Логическое	Заменять на уникальные значения
saveConsistency	Логическое	Флаг, определяющий будет ли сохраняться кон...
maskingCode	Строковое	Код справочника
consistKey	Строковое	Ключ консистентности. Данные с одинаковым

### Зависимости от других столбцов БД

Методы маскирования содержат таблицу зависимостей от маскированных (или в некоторых случаях оригинальных) значений других столбцов БД. Данный функционал полезен для создания методов, которые формируют маскированные значения столбца БД исходя из маскированных (или оригинальных) значений других столбцов БД.

#### Пример 6. Пример вычисления значения столбца ФИО пользователя

**ФИО пользователя** - столбец, значение которого получается в результате конкатинации трёх столбцов через пробел: фамилии, имени и отчества. Чтобы данные получились консистентными и реалистичными, ФИО необходимо формировать после того как значения фамилии, имени и отчества уже были замаскированы, то есть на момент



вычисления маскированного значения **ФИО пользователя** скрипт должен знать, какие замаскированные значения получены в указанных столбцах БД. За это и отвечает механизм зависимостей от других столбцов БД.

В скрипт необходимо передать 3 зависимости, каждая из которых будет соответствовать нужному столбцу БД. Сам скрипт будет выглядеть следующим образом:

```
return String.format("%s %s %s", surname, name, patronymic)
```

где surname, name, patronymic - переменные, в которые будут передаваться значения из соответствующих столбцов БД на шаге [настройки правил маскирования](#).

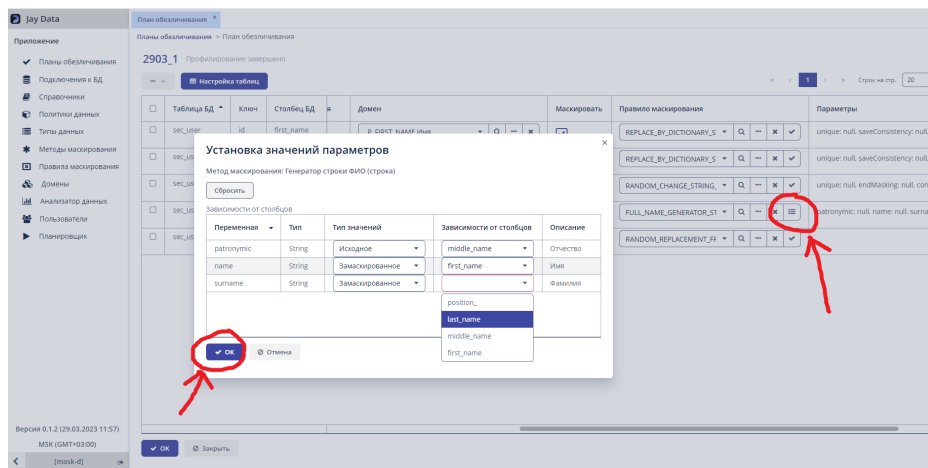
Данная таблица расположена рядом с таблицей параметров. При создании новой такой зависимости заполняются следующие поля:

**Переменная** – наименование переменной, по которой можно получить значение столбца БД, от которого будет зависеть текущий метод; **Тип** – тип данных в столбце БД, из которого берётся значение; **Описание** – описание зависимости.

Переменная	Тип	Описание
patronymic	String	Отчество
name	String	Имя
surname	String	Фамилия

```
1 return String.format("%s %s %s", surname, name, patronymic);
```

В момент настройки правил маскирования в [плане обезличивания](#) данным зависимостям будет необходимо указать конкретные столбцы БД из той же таблицы, из которых система возьмёт значения surname, name, patronymic, а также указать какие именно значения использовать: замаскированные или оригинальные. Во избежание заикливания для выбора допустимы только те столбцы БД, у которых установлено [правило](#), базирующееся на 'простом' методе (то есть таком, у которого в свою очередь нет ни одной зависимости от других столбцов БД).



Все зависимости от столбцов обязательно необходимо указать, в ином случае система не позволит сохранить план обезличивания. После указания всех зависимостей необходимо нажать кнопку **OK**.

Если не все зависимости установлены, иконка напротив правила маскирования будет иметь вид 'списка'.

Планы обезличивания > План обезличивания

2903\_1 Профилирование завершено

Настройка таблиц

Таблица БД	Ключ	Столбец БД	Домен	Маскировать	Правило маскирования	Па
sec_user	id	first_name	._P_FIRST_NAME Имя	<input checked="" type="checkbox"/>	REPLACE_BY_DICTIONARY_S	un
sec_user	id	last_name	._P_LAST_NAME Фамилия	<input checked="" type="checkbox"/>	REPLACE_BY_DICTIONARY_S	un
sec_user	id	middle_name	._P_PATRONYMIC Отчество	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING	un
sec_user	id	name	._P_FIRST_NAME Имя	<input checked="" type="checkbox"/>	FULL_NAME_GENERATOR_ST	pat
sec_user	id	position_	._P_EDUCATION_LEVEL Уровень о	<input checked="" type="checkbox"/>	RANDOM_REPLACEMENT_FF	un

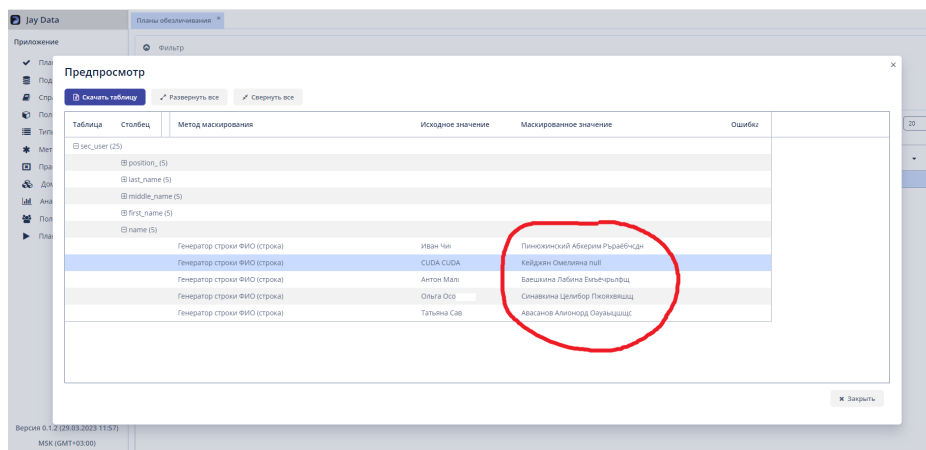
Если все зависимости установлены, иконка напротив правила маскирования примет вид 'галочки'.

2903\_1 Профилирование завершено

Настройка таблиц

Таблица БД	Ключ	Столбец БД	Домен	Маскировать	Правило маскирования	Параметры
sec_user	id	first_name	._P_FIRST_NAME Имя	<input checked="" type="checkbox"/>	REPLACE_BY_DICTIONARY_S	unique: null, saveConsistency: null, ma
sec_user	id	last_name	._P_LAST_NAME Фамилия	<input checked="" type="checkbox"/>	REPLACE_BY_DICTIONARY_S	unique: null, saveConsistency: null, ma
sec_user	id	middle_name	._P_PATRONYMIC Отчество	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING	unique: null, endMasking: null, convert
sec_user	id	name	._P_FIRST_NAME Имя	<input checked="" type="checkbox"/>	FULL_NAME_GENERATOR_ST	patronymic: null, name: null, surname
sec_user	id	position_	._P_EDUCATION_LEVEL Уровень о	<input checked="" type="checkbox"/>	RANDOM_REPLACEMENT_FF	

В результате при маскировании ФИО примет значение, сконкатенированное из значений 3-х других столбцов: last\_name, first\_name, middle\_name



## Валидация в методах маскирования

Пользователь может создавать **правила маскирования** на основе методов, задавая разные значения **параметров**. Может случиться ситуация, когда с указанными значениями параметров применить правило к столбцу БД не представляется возможным по какой-либо причине. Одной из таких причин может быть слишком большое количество уникальных записей в маскируемом столбце БД: в таком случае система оповещает о том, что данное правило не может быть применено к столбцу БД в виду слишком маленькой мощности выборки. Другой причиной может быть недопустимая комбинация взаимоисключающих параметров или некорректные значения этих параметров.

**Мощность выборки** - количество возможных вариантов замаскированных значений при определённом наборе параметров правила.

Каждый метод маскирования имеет свой набор ограничений на значения и комбинацию вводимых параметров в **правилах**, которые на них основаны. Ограничения определяются индивидуально в каждом из методов с помощью Groovy-скрипта в поле **Скрипт валидации**. Валидация имеет скрипт проверки и сообщение об ошибке (в случае непрохождения валидации).

### Пример 7. Пример

Например, пользователь хочет применить правило **Сопоставление 2-х последних цифр**, основанное на методе **сопоставление цифр**. В данном правиле указан следующий набор параметров:

Таблица 5. Полный список параметров для примера

Название переменной	Значение
saveConsistency	Активен
consistKey	numberMaskingConsistKey
convertEmptyValue	Не активен
usingZero	Активен
startMasking	9
endMasking	11

Маскируется числовое поле с телефонами с сохранением **консистентности**. При указанных параметрах (индекс символа начала маскирования - 9, окончания маскирования - 11) будут замаскированы только последние 2 числа, остальные останутся без изменений. При маскировании последних 2 чисел в номере телефона, например, **89161234567** мощность выборки снижается, поскольку вариантов маскирования 2-х последних чисел всего 100 (от 00 до 99). Это создает проблемы, если маскируемых записей в столбце БД с телефонами больше 100, так как консистентность в данном случае не может быть выдержана по определению - вариантов замаскировать запись меньше чем самих записей (повторы неизбежны). При этом, если флаг, отвечающий за консистентность, будет снят, то данное ограничение будет не актуально, так как повторы в таком случае допустимы.

Также, в каждом из методов могут присутствовать дополнительные ограничения, которые связаны со спецификой данных, хранящихся в столбце БД (относящиеся к **домену**). Например, в номерах банковских карт первые четыре цифры отвечают за принадлежность к определенному банку, часто повторяются (+ не могут быть случайными) что изначально сказывается на мощности выборки. Количество возможных вариантов написания ИНН тоже определено математической последовательностью, определённой законом, что может сокращать выборку.

## Системные методы маскирования

В данном разделе приведены системные (встроенные) методы маскирования, поставляемые вместе с дистрибутивом Системы.

### Замена на константу

Позволяет заменять исходные значения на заранее определенную константу. Константой может выступать не только слово или число, но и любой другой символ.

Доступны два отдельных метода замены на константу, в зависимости от типа данных, хранимых в БД-источнике:

- Замена на константу (строка) - замена на NULL строкового типа данных;
- Замена на константу (CLOB) - замена на NULL типа данных CLOB.

**Параметры метода.** В данном методе по умолчанию заведены два параметра:

- **constantValue** — на какое значение константы будет меняться текущее значение.
- **ignoreEmpty** — игнорировать пустые значения в строке или заменять их на константу.

*Пример 8. Применение метода маскирования "Замена на константу"*

Если задать константу **constantValue = значение\_скрыто**, то при наборе параметров:

*Таблица 6. Полный список параметров для примера*

Название переменной	Значение
constantValue	значение_скрыто
ignoreEmpty	x

фамилия **Иванов** будет заменено на **значение\_скрыто**.

## Замена на null

Метод маскирования, который меняет значение исходной строки на значение null.

Доступны три отдельных метода замены на NULL, в зависимости от типа данных, хранимых в БД-источнике:

- Замена на NULL (строка) - замена на NULL строкового типа данных;
- Замена на NULL (CLOB) - замена на NULL типа данных CLOB;
- Замена на NULL (BLOB) - замена на NULL типа данных BLOB.

*Пример 9. Пример для метода "Замена на null"*

Если нужно замаскировать адрес методом "Замена на null", то значение строки 'Москва, Тверская, 7' будет преобразовано в NULL.

## Замена по справочнику

Заменяет какое-то значение на элемент из [справочника](#). Метод удобно использовать, когда нужно заменить исходное значения на реалистичное, но отличное от оригинального.

Работает только со строковыми значениями.

### Особенности метода

- Если одновременно установлены флаги **unique** и **saveConsistency**, то маскирование будет происходить консистентно, а не уникально;
- В случае, если размер словаря будет недостаточен при установленном флаге уникальности, то будет добавлен числовой постфикс. Пример: если имя **Иван** уже было использовано, то будет получено значение **Иван\_1**.

### Параметры метода

- **maskingDictCode** — указание кода [справочника](#) на элементы которого, будут маскироваться исходные данные;
- **saveConsistency** — признак, определяющий сохранение консистентности в рамках ключа консистентности;
- **consistKey** — ключ консистентности. Указывается при активной переменной **saveConsistency**;
- **unique** — всегда заменять исходное значение на уникальное выходное значение.

Данный метод часто используется при маскировании фамилии или имени Пользователей.

*Пример 10. Применение метода маскирования 'Замена по справочнику'*

Имя пользователя 'Иван' в результате маскирования с параметрами: .Полный список параметров для примера

Название переменной	Значение
saveConsistency	v
maskingDictCode	RUS_FIRST_NAME
consistKey	defaultFirstNameConsistKey
unique	v

изменится на 'Андрей'. При активном параметре сохранения консистентности, если в ходе маскирования имя пользователя 'Иван' хотя бы раз заменилось на 'Андрей', то все последующие имена 'Иван' в рамках ключа консистентности будут заменены на 'Андрей'.

#### Валидация

Ниже приведён список проверок в рамках [валидации](#) указанного метода.

**Проверка 1:** Количество записей в справочнике с указанным в параметре `maskingDictCode` кодом должно быть достаточным, чтобы выдержать консистентность. В данном случае валидация проверяет количество записей в справочнике перед запуском самого маскирования и не дает Пользователю осуществить маскирование с консистентностью при количестве уникальных записей в столбце > мощности справочника.

#### Замена N символов на служебные

Заменяет указанное количество символов в исходной строке на указанные служебные символы с начала исходной строки и/или с её конца.

Работает только со строковыми значениями.

*Пример 11. Применение метода "Замена N символов на служебные"*

Строка `Разработка` при наборе параметров:

*Таблица 7. Полный список параметров для примера 1*

Название переменной	Значение
numSymbolsToReplace	3
fromStart	v
fromEnd	x

Название переменной	Значение
symbol	*

будет преобразована в **\*\*\***работка

И при наборе параметров:

Таблица 8. Полный список параметров для примера 2

Название переменной	Значение
numSymbolsToReplace	2
fromStart	v
fromEnd	v
symbol	?

будет преобразована в **??**работ**??**

## Маскирование числа

Прибавляет к любому числу случайное значение из указанного диапазона. Такой метод маскирования может использоваться при необходимости скрыть паспортные данные, телефонные номера или банковские реквизиты.

Работает только с числовыми значениями.

Пример 12. Применение метода "Маскирование числа"

Серия паспорта **4400** после применения данного метода маскирования с параметрами:

Таблица 9. Полный список параметров для примера

Название переменной	Значение
randomRange	100

может быть преобразована в **4430** (было добавлено случайное число 30, в диапазоне от 1 до 100).

## Добавление к дате/времени константного времени

Прибавляет к текущей дате годы, дни, месяцы, часы, минуты и секунды.

Работает с типом данных 'Дата'.

## Параметры метода

- **years** — год в исходной дате увеличивается на указанное значение в параметре. Для

отрицательных значений добавляется -years;

- **months** — **месяц** в исходной дате увеличивается на указанное значение в параметре. Для отрицательных значений добавляется -months;
- **days** — **день** в исходной дате увеличивается на указанное значение в параметре. Для отрицательных значений добавляется -days;
- **hours** — **час** в исходной дате увеличивается на указанное значение в параметре. Для отрицательных значений добавляется -hours;
- **minutes** — **минута** в исходной дате увеличивается на указанное значение в параметре. Для отрицательных значений добавляется -minutes;
- **seconds** — **секунда** в исходной дате увеличивается на указанное значение в параметре. Для отрицательных значений добавляется -seconds.

*Пример 13. Применение метода "Добавление к дате/времени константного времени"*

Например, если **Значение по умолчанию** = 5 для переменной minutes(минуты), то в приемнике к указанной дате будет прибавлено +5 к минутам. Соответственно, дата '10.05.2000 10:10:10' при наборе параметров:

*Таблица 10. Полный список параметров для примера*

Название переменной	Значение
years	0
months	0
days	0
hours	0
minutes	5
seconds	0

будет преобразована в '10.05.2000 10:15:10'.

### **Добавление к дате/времени случайного времени**

Прибавление к текущей дате случайное количество лет, дней, месяцев, часов, минут, секунд.

Работает с типом данных 'Дата'.

### **Параметры метода**

- **years** — добавляется случайное значение лет от 1 до years. Для отрицательных значений от -1 до years;
- **months** — добавляется случайное значение месяцев от 1 до months. Для отрицательных значений от -1 до months;



- **days** — добавляется случайное значение дней от 1 до days. Для отрицательных значений от -1 до days;
- **hours** — добавляется случайное значение часов от 1 до hours. Для отрицательных значений от -1 до hours;
- **minutes** — добавляется случайное значение минут от 1 до minutes. Для отрицательных значений от -1 до minutes;
- **seconds** — добавляется случайное значение дней от 1 до seconds. Для отрицательных значений от -1 до seconds.

Пример 14. Применение метода "Добавление к дате/времени случайного времени"

Например, если **Значение по умолчанию** = 5 для переменной minutes(минуты) и seconds (секунды), то в приемнике к указанной дате будет прибавлено число от 1 до 5 к минутам и секундам. Соответственно, дата '10.05.2000 10:10:10' при наборе параметров:

Таблица 11. Полный список параметров для примера

Название переменной	Значение
years	0
months	0
days	0
hours	0
minutes	5
seconds	5

может быть преобразована в '10.05.2000 10:13:12'.

## Номер банковской карты

Позволяет изменить числа в номере банковской карты в соответствии с заданными параметрами.

Работает только со строковыми значениями.

## Особенности метода

- При маскировании с уникальными значениями, если интервал маскирования недостаточен, то исходное значение будет замаскировано полностью.
- Отсчет индексов начала и конца маскирования начинается с 0.
- Если одновременно установлены флаги **unique** и **saveConsistency**, то маскирование будет происходить консистентно, а не уникально.
- Карта маскируется только при условии, что `endMasking - startMasking >= 1`.
- В случае, если последовательность состоит из менее 15 или более 19 символов или если

строка не является числовой последовательностью, то создается случайная банковская карта из 16 символов.

- В случае, если разделители присутствовали в исходной последовательности и не были удалены из-за неверного их указания в параметре, то такая последовательность будет считаться не числовой.
- Разделители, при их сохранении остаются на прежних местах, что и были. Так, для последовательности `1234_1234_1234_1234` при установлении флага `saveLenCardNumber = 'false'` и при сохранении длины исходной карты возможны подобные результаты: `1234_1234_1234_12345`.
- Разделители не являются индексруемым элементом, например в номере `0123_4567_8901_2345` индекс цифры "4" = 4

## Параметры метода

- `saveLenCardNumber` — признак, отвечающий за сохранение длины номера карты. Если активировать параметр и в исходных данных было 16 символов, столько же будет и в замаскированном значении;
- `saveSeparators` — признак, отвечающий за сохранение разделителей в тех же местах, где и в исходных данных;
- `startMasking` — индекс начала фрагмента маскирования.

### Пример 15. Пример для параметра `startMasking`

Если указать параметр `startMasking = 5`, то номер карты `1111 2222 3333 4444` будет маскироваться начиная с 6-го символа. А первые 5 останутся равными исходному значению (при условии, что `endMasking` не указан).

Таблица 12. Полный список параметров для примера

Название переменной	Значение
<code>saveLenCardNumber</code>	v
<code>saveSeparators</code>	v
<code>startMasking</code>	5
<code>endMasking</code>	0
<code>convertEmptyCardNumber</code>	v
<code>saveConsistency</code>	x
<code>consistKey</code>	<code>defaultBankCardconsistKey</code>
<code>separator</code>	EMPTY
<code>unique</code>	v

Таким образом, номер карты в результате маскирования будет преобразован в `1111 2648 8789 6786`.

- `endMasking` — индекс окончания фрагмента маскирования.

*Пример 16. Пример использования параметра `endMasking`*

Если указать параметр `endMasking = 3`, то в номере карты `1111 2222 3333 4444` замаскируются только первые 3 значения (при условии, что `startMasking` не указан)

*Таблица 13. Полный список параметров для примера*

Название переменной	Значение
<code>saveLenCardNumber</code>	v
<code>saveSeparators</code>	v
<code>startMasking</code>	0
<code>endMasking</code>	3
<code>convertEmptyCardNumber</code>	v
<code>saveConsistency</code>	x
<code>consistKey</code>	<code>defaultBankCardconsistKey</code>
<code>separator</code>	EMPTY
<code>unique</code>	v

номер карты будет преобразован в `6895 2222 3333 4444`.

*Пример 17. Пример использования 2-х параметров `startMasking` и `endMasking`*

Для того чтобы номер карты `1111 1111 1111 1111` был преобразован в `1111 2222 2222 1111`, используя переменные `startMasking` и `endMasking` необходимо указать следующие значения всех параметров:

*Таблица 14. Полный список параметров*

Название переменной	Значение
<code>saveLenCardNumber</code>	v
<code>saveSeparators</code>	v
<code>startMasking</code>	4
<code>endMasking</code>	12
<code>convertEmptyCardNumber</code>	v
<code>saveConsistency</code>	x
<code>consistKey</code>	<code>defaultBankCardconsistKey</code>
<code>separator</code>	EMPTY
<code>unique</code>	v

- `convertEmptyCardNumber` — параметр, определяющий будут ли пустые значения и

последовательности не состоящие из чисел преобразовываться в банковскую карту длиной 16 символов.

*Пример 18. Пример использования параметра convertEmptyCardNumber*

Например, если вместо номера карты указано значение **нет карты**, то в результате маскирования данное значение при наборе параметров:

*Таблица 15. Полный список параметров для примера*

Название переменной	Значение
saveLenCardNumber	v
saveSeparators	v
startMasking	0
endMasking	0
convertEmptyCardNumber	v
saveConsistency	x
consistKey	defaultBankCardconsistKey
separator	EMPTY
unique	v

будет преобразовано в **0101 0102 0203 0304**.

- **saveConsistency** — чекбокс, отвечающий за сохранение консистентности в рамках ключа консистентности. Это означает, что если номер карты в одной строчке **0000 0001 0000 0001** был заменен на **1111 1110 1111 1110**, то все карты с номером **0000 0001 0000 0001** будут заменены на **1111 1110 1111 1110**. Работает при активном параметре saveConsistency.
- **consistKey** — ключ консистентности. Указывается при активированном параметре saveConsistency.
- **separator** — разделитель между числами банковской карты. В качестве разделителя можно использовать как пробел, так и специальный символ, например, **\_**. Значение по умолчанию EMPTY означает отсутствие разделителей.

*Пример 19. Пример для параметра разделитель*

Если для номера карты **0001 0001 0001 0001** указать параметр separator = **\_**, то при наборе параметров:

*Таблица 16. Полный список параметров для примера*

Название переменной	Значение
saveLenCardNumber	v
saveSeparators	v
startMasking	0

Название переменной	Значение
endMasking	0
convertEmptyCardNumber	v
saveConsistency	x
consistKey	defaultBankCardconsistKey
separator	-
unique	v

он будет преобразован в 2110\_3110\_4110\_5110.

- **unique** — всегда заменять исходное значение на уникальное выходное значение.

### Валидация

Ниже приведён список проверок в рамках [валидации](#) указанного метода.

**Проверка 1:** Значения **startMasking** и **endMasking** должны быть  $\geq 0$  и **endMasking** должен быть  $> startMasking$ .

**Проверка 2:** Диапазон значений между **endMasking** и **startMasking** должен быть достаточным, чтобы выдержать консистентность. В данном случае валидация проверяет комбинацию значений указанных параметров правила маскирования перед запуском самого маскирования и не дает Пользователю осуществить маскирование с консистентностью при количестве уникальных записей в столбце  $>$  мощности выборки.

**Проверка 3:** Флаг **convertEmptyValue** не должен быть установлен вместе с флагом **saveConsistency**. По определению **консистентность** не может маскировать одинаковые значения, в том числе NULL, в разные значения. Поэтому комбинация данных флагов не имеет практического смысла.

### ИНН РФ

Значения исходного столбца заменяются на случайно сгенерированное значение формата ИНН РФ.

За маскирование ИНН отвечают два метода в зависимости от типа данных в БД-источнике:

- ИНН РФ (строка) - маскирование ИНН для строкового типа данных.
- ИНН РФ (число) - маскирование ИНН для числового типа данных.

### Особенности метода

- Если одновременно установлены флаги **unique** и **saveConsistency**, то маскирование будет происходить консистентно, а не уникально.
- В случае, если последовательность состоит не из 10 или 12 символов или если строка не является числовой последовательностью, то создается случайный ИНН из 12 символов.

## Параметры метода

- `saveConsistency` — признак, отвечающий за сохранение консистентности в рамках ключа консистентности;
- `consistKey` — ключ консистентности. Указывается при активной переменной `saveConsistency`;
- `keyDictIfns` — код словаря ИФНС. Код словаря берется из раздела [Справочники](#);
- `convertEmptyInn` — признак, отвечающий за преобразование не валидного формата ИНН в ИНН стандартного образца;
- `unique` — всегда заменять исходное значение на уникальное выходное значение.

Пример 20. Пример использования метода маскирования "ИНН РФ"

Если в поле ИНН указано значение `4457-0`, то при наборе параметров:

Таблица 17. Полный список параметров для примера

Название переменной	Значение
<code>saveConsistency</code>	x
<code>consistKey</code>	INN_RUS
<code>keyDictIfns</code>	RUS_IFNS
<code>convertEmptyInn</code>	v
<code>unique</code>	v

замаскированное значение примет вид стандартного ИНН 10-значного номера `1234512345`.

## Замена спец. формата - СНИЛС

Значения исходного столбца заменяются на случайно сгенерированное значение формата СНИЛС РФ с проверкой контрольного числа. СНИЛС имеет вид "XXX-XXX-XXX YY", где "XXX-XXX-XXX" - номер, а "YY" - контрольное число.

Доступны два отдельных метода по замене СНИЛС, в зависимости от типа данных, хранимых в БД-источнике:

- Маскирование СНИЛС (РФ) (строка) - маскирование СНИЛС для строкового типа данных;
- Маскирование СНИЛС (РФ) (число) - маскирование СНИЛС для числового типа данных.

## Особенности метода

- Если одновременно установлены флаги `unique` и `saveConsistency`, то маскирование будет происходить консистентно, а не уникально.
- В случае, если последовательность состоит не из 11 символов или если строка не

является числовой последовательностью, то создается случайный СНИЛС из 11 СИМВОЛОВ.

## Параметры метода

- `convertEmptySnils` — признак, определяющий, будут ли конвертироваться пустые значения.
- `saveSeparator` (только для строковой версии метода) — признак, определяющий будет ли сохранен разделитель от исходного значения.
- `firstSeparator` (только для строковой версии метода) — первый разделитель, который может быть в последовательности;
- `saveConsistency` — признак, определяющий будет ли сохранена консистентность в рамках ключа консистентности;
- `consistKey` — ключ консистентности. Указывается при активной переменной `saveConsistency`;
- `unique` — всегда заменять исходное значение на уникальное выходное значение.

### Пример 21. Пример использования параметра `firstSeparator`

Если указать значение параметра `firstSeparator = #`, то замаскированный СНИЛС примет значение '111#223-333 00'.

Таблица 18. Полный список параметров для примера

Название переменной	Значение
<code>saveConsistency</code>	x
<code>consistKey</code>	SNILS_RUS
<code>convertEmptySnils</code>	x
<code>saveSeparator</code>	v
<code>firstSeparator</code>	#
<code>secondSeparator</code>	DEFAULT
<code>unique</code>	v

- `secondSeparator` — второй разделитель, который может быть в последовательности.

### Пример 22. Пример использования параметра `secondSeparator`

Если указать значение параметра `secondSeparator = &`, то замаскированный СНИЛС примет значение '111-223&333 00', при активном параметре `saveConsistency`.

Таблица 19. Полный список параметров для примера

Название переменной	Значение
saveConsistency	x
consistKey	SNILS_RUS
convertEmptySnils	x
saveSeparator	v
firstSeparator	DEFAULT
secondSeparator	&
unique	v

- **consistKey** — значение ключа консистентности. Указывается при активном параметре **saveConsistency**.

## Международный номер

Международный номер IBAN (International bank account number) заменяется на случайно сгенерированное значение IBAN.

В номере IBAN может содержаться до 34 символов. В каждом государстве устанавливается свой стандарт IBAN. Первые две буквы кода обозначают код страны, следующие два числа — контрольные числа, рассчитываемые по стандарту ISO 7064, далее следует внутригосударственный номер банковского счета, включающий в себя номер счета и номер банка.

## Особенности метода

- Если одновременно установлены флаги **unique** и **saveConsistency**, то маскирование будет происходить консистентно, а не уникально.
- При маскировании с уникальными значениями, если интервал маскирования недостаточен, то исходное значение будет замаскировано полностью.

Работает только со строковыми значениями.

*Пример 23. Пример международного номера IBAN*

```
RU483200000012345864
```

## Параметры метода

- **saveConsistency** — признак, определяющий сохранение консистентности данных в рамках ключа консистентности;
- **startMasking** — индекс начала фрагмента маскирования.

1. Пример использования параметра **startMasking**

Если указать значение параметра **startMasking = 10**, то международный номер будет



маскироваться начиная с 11 символа (при условии, что параметр `endMasking` не указан). Международный номер `RU48320000012345864` в результате маскирования с набором параметров:

Таблица 20. Полный список параметров для примера

Название переменной	Значение
<code>unique</code>	<code>v</code>
<code>saveConsistency</code>	<code>x</code>
<code>consistKey</code>	<code>defaultInternationalNumberConsistKey</code>
<code>startMasking</code>	<code>10</code>
<code>keyDictCountryComponent</code>	<code>INTERNATIONAL_NUMBER_COUNTRY</code>
<code>endMasking</code>	<code>0</code>
<code>convertInternationalNumber</code>	<code>x</code>

будет преобразован в `RU483200008976312482`.

- `consistKey` — ключ консистентности. Указывается при активной переменной `saveConsistency`;
- `keyDictCountryComponent` — ключ словаря, который отвечает за список префиксов стран (первые две буквы в номере);
- `endMasking` — индекс окончания фрагмента маскирования;

#### 1. Пример использования параметра `endMasking`

Если указать значение параметра `endMasking = 8`, то международный номер будет маскироваться до 8 символа (при условии, что параметр `startMasking` не указан).

Таблица 21. Полный список параметров для примера

Название переменной	Значение
<code>unique</code>	<code>v</code>
<code>saveConsistency</code>	<code>x</code>
<code>consistKey</code>	<code>defaultInternationalNumberConsistKey</code>
<code>startMasking</code>	<code>0</code>
<code>keyDictCountryComponent</code>	<code>INTERNATIONAL_NUMBER_COUNTRY</code>
<code>endMasking</code>	<code>8</code>
<code>convertInternationalNumber</code>	<code>x</code>

- `convertInternationalNumber` — признак, определяющий будет ли маскироваться пустой номер. Пустой номер маскируется по стандартному формату международного номера;
- `unique` — всегда заменять исходное значение на уникальное выходной значение.

## Валидация

Ниже приведён список проверок в рамках [валидации](#) указанного метода.

**Проверка 1:** Значения `startMasking` и `endMasking` должны быть  $\geq 0$  и `endMasking` должен быть  $> startMasking$ .

**Проверка 2:** Диапазон значений между `endMasking` и `startMasking` должен быть достаточным, чтобы выдержать консистентность. В данном случае валидация проверяет комбинацию значений указанных параметров правила маскирования перед запуском самого маскирования и не дает Пользователю осуществить маскирование с консистентностью при количестве уникальных записей в столбце  $>$  мощности выборки.

**Проверка 3:** Флаг `convertInternationalNumber` не должен быть установлен вместе с флагом `saveConsistency`. По определению [консистентность](#) не может маскировать одинаковые значения, в том числе NULL, в разные значения. Поэтому комбинация данных флагов не имеет практического смысла.

## Сопоставление цифр

Метод маскирования, который позволяет заменять части значения исходного столбца на случайно сгенерированное значение.

Доступны два отдельных метода сопоставление цифр, в зависимости от типа данных, хранимых в БД-источнике:

- Сопоставление цифр (строка) - метод для сопоставления строкового типа данных;
- Сопоставление цифр (число) - метод для сопоставления числового типа данных.

## Особенности метода

- Если одновременно установлены флаги `unique` и `saveConsistency`, то маскирование будет происходить консистентно, а не уникально.
- При маскировании с уникальными значениями, если интервал маскирования недостаточен, то исходное значение будет замаскировано полностью.

## Параметры метода

- `saveConsistency` — признак, определяющий сохранение консистентности в рамках ключа консистентности;
- `consistKey` — ключ консистентности. Указывается при активной переменной `saveConsistency`;
- `convertEmptyValue` - признак, определяющий маскирование пустых значений;
- `usingZero` — признак, определяющий использование цифры 0 в маскировании;
- `startMasking` — индекс начала фрагмента маскирования;
- `endMasking` — индекс окончания фрагмента маскирования;
- `unique` — всегда заменять исходное значение на уникальное выходное значение.

#### Пример 24. Пример использования метода маскирования "Сопоставление цифр"

Число **56189** в результате применения метода маскирования "Сопоставление цифр" примет вид **52705**.

Таблица 22. Полный список параметров для примера

Название переменной	Значение
saveConsistency	v
consistKey	numberMaskingConsistKey
convertEmptyValue	x
usingZero	v
startMasking	1
endMasking	0
unique	v

#### Валидация

Ниже приведён список проверок в рамках [валидации](#) указанного метода.

**Проверка 1:** Значения **startMasking** и **endMasking** должны быть  $\geq 0$  и **endMasking** должен быть  $> startMasking$ .

**Проверка 2:** Диапазон значений между **endMasking** и **startMasking** должен быть достаточным, чтобы выдержать консистентность. В данном случае валидация проверяет комбинацию значений указанных параметров правила маскирования перед запуском самого маскирования и не дает Пользователю осуществить маскирование с консистентностью при количестве уникальных записей в столбце  $>$  мощности выборки.

**Проверка 3:** Флаг **convertEmptyValue** не должен быть установлен вместе с флагом **saveConsistency**. По определению [консистентность](#) не может маскировать одинаковые значения, в том числе NULL, в разные значения. Поэтому комбинация данных флагов не имеет практического смысла.

#### Случайная замена

Позволяет заменять значения исходного столбца на случайно сгенерированное значение. Новое значение генерируется таким образом, что число заменяется на число, а буква заменяется на букву, при этом спецсимволы остаются без изменений.

Работает только со строковыми значениями.

- Случайная замена (строка) - случайная замена для строкового типа данных;
- Случайная замена (CLOB) - случайная замена для типа данных CLOB.

#### Особенности метода

- Если одновременно установлены флаги `unique` и `saveConsistency`, то маскирование будет происходить консистентно, а не уникально.
- Замена символов происходит по следующим правилам:
  - Заглавная русская буква - случайная заглавная русская буква;
  - Прописная русская буква - случайная прописная русская буква;
  - Заглавная английская буква - случайная заглавная английская буква;
  - Прописная английская буква - случайная прописная английская буква;
  - Цифра - случайная цифра;
  - Иные символы - остаются неизменным.

## Параметры метода

- `saveConsistency` — признак, определяющий сохранение консистентности в рамках ключа консистентности;
- `consistKey` — ключ консистентности. Указывается при активной переменной `saveConsistency`;
- `convertEmptyValue` — признак, определяющий маскирование пустых значений;
- `startMasking` — индекс начала фрагмента маскирования;
- `endMasking` — индекс окончания фрагмента маскирования;
- `unique` — всегда заменять исходное значение на уникальное выходное значение.

*Пример 25. Пример использования метода маскирования "Случайная замена"*

Строка `123_a4b5` в результате маскирования методом "Случайная замена" с параметрами:

*Таблица 23. Полный список параметров для примера*

Название переменной	Значение
<code>saveConsistency</code>	x
<code>consistKey</code>	<code>randomChangeMaskingConsistKey</code>
<code>convertEmptyValue</code>	x
<code>startMasking</code>	0
<code>endMasking</code>	0
<code>unique</code>	v

будет преобразована в `685_e2n1`.

## Валидация

Ниже приведён список проверок в рамках валидации указанного метода.

**Проверка 1:** Значения `startMasking` и `endMasking` должны быть  $\geq 0$  и `endMasking` должен быть  $> startMasking$ .

**Проверка 2:** Диапазон значений между `endMasking` и `startMasking` должен быть достаточным, чтобы выдержать консистентность. В данном случае валидация проверяет комбинацию значений указанных параметров правила маскирования перед запуском самого маскирования и не дает Пользователю осуществить маскирование с консистентностью при количестве уникальных записей в столбце  $>$  мощности выборки.

**Проверка 3:** Флаг `convertEmptyValue` не должен быть установлен вместе с флагом `saveConsistency`. По определению **консистентность** не может маскировать одинаковые значения, в том числе NULL, в разные значения. Поэтому комбинация данных флагов не имеет практического смысла.

### Замена на случайное значение из БД

Значения исходного столбца БД заменяются на другое значение, случайно полученное из этого же столбца БД. Фактически, данный метод является перемешиванием значений в столбце БД. Повторения (замена на одно и то же значение из БД) допускаются.

Доступны 4 отдельных метода замены на случайное значение из БД, в зависимости от типа данных, хранимых в БД-источнике:

- Замена на случайное значение из БД (строка) - замена на случайное значение из БД для строкового типа данных;
- Замена на случайное значение из БД (Логический тип) - замена на случайное значение из БД для логического типа данных;

### Генератор строки ФИО

Данный метод позволяет объединить фамилию, имя, отчество в одну строку, используя **зависимости от других столбцов** при маскировании. Метод удобно использовать, когда необходимо объединить замаскированные данные в одну строку.

**Параметры метода** Для данного метода отсутствуют параметры, но указываются зависимости от столбцов.

- `patronymic` — Отчество
- `name` — Имя
- `surname` — Фамилия

Таким образом, указав необходимые зависимости от столбцов, фамилия, имя и отчество объединяются в одну строку.

### Случайная замена JSON

Позволяет заменять значения формата JSON в связке `ключ:значение` на случайно сгенерированное значение, оставляя ключи оригинальными.

Доступны 3 отдельных метода значения формата JSON, в зависимости от типа данных, хранимых в БД-источнике:

- Случайная замена JSON (JSON) — замена значения формата JSON в связке ключ:значение для столбцов типа данных JSON;
- Случайная замена JSON (строка) — замена значения формата JSON в связке ключ:значение для столбцов типа данных строка;
- Случайная замена JSON (CLOB) — замена значения формата JSON в связке ключ:значение для столбцов типа данных CLOB;

### Параметры метода

- **jsonKeys** — параметр, содержащий значения ключей, перечисляемых через запятую, которые необходимо замаскировать внутри JSON. Остальные значения ключей останутся оригинальными. При пустом значении параметра маскируются все значения ключей.
- **convertJsonToNull** — признак, который преобразует JSON в null. Таким образом, вся строка примет значение 'null'.
- **convertJsonToEmptyObj** — признак, преобразующий JSON в пустой объект {}.
- **convertJsonToEmptyCollection** — признак, преобразующий JSON в пустой объект [].

*Пример 26. Пример использования метода маскирования "Случайная замена JSON"*

Имеется строка JSON:

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "create.tags",
  "params": {
    "access_token": "111aaa567bbc",
    "user_id": "12345",
    "name": "tag1"
  }
}
```

Данная строка в результате маскирования методом "Случайная замена JSON" с параметрами:

*Таблица 24. Полный список параметров для примера*

Название переменной	Значение
jsonKeys	access_token,user_id
convertJsonToNull	x
convertJsonToEmptyObj	x

Название переменной	Значение
convertJsonToEmptyCollection	x

будет преобразована в строку JSON следующего вида:

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "create.tags",
  "params": {
    "access_token": "784erd848des",
    "user_id": "16839",
    "name": "tag1"
  }
}
```

## Случайная замена XML

Позволяют заменять значения формата XML, оставляя тэги оригинальными.

Доступны 3 отдельных метода значения формата XML, в зависимости от типа данных, хранимых в БД-источнике:

- Случайная замена XML (XML) — замена значения формата XML для столбцов типа данных XML;
- Случайная замена XML (строка) — замена значения формата XML для столбцов типа данных строка;
- Случайная замена XML (CLOB) — замена значения формата XML для столбцов типа данных CLOB;

## Параметры метода

- `convertXmlToNull` — признак, который преобразовывает XML в null.
- `xmlKeys` — параметр, содержащий значения тэгов, перечисляемых через запятую, которые необходимо замаскировать внутри XML. Остальные значения тэгов останутся оригинальными. При пустом значении параметра маскируются все значения тэгов.
- `convertXmlToEmpty` — признак, который преобразует XML в пустой XML-файл.

*Пример 27. Пример использования метода маскирования "Случайная замена XML"*

Имеется структура XML:

```
<req>
<name>Anna</name>
<country>Russia</country>
<city>Moscow</city>
```

```
<gender>FEMALE</gender>
</req>
```

с параметрами:

Таблица 25. Полный список параметров для примера

Название переменной	Значение
xmlKeys	name,country
convertXmlToNull	x
convertXmlToEmpty	x

будет преобразована в строку XML следующего вида:

```
<req>
<name>Rsme</name>
<country>Vifrsx</country>
<city>Moscow</city>
<gender>FEMALE</gender>
</req>
```

## Правила маскирования

**Правила маскирования** - сущность, которая определяет **метод маскирования**, применяемый к столбцу БД и содержит в себе дефолтные значения **его параметров**. Для одного метода маскирования может быть создано несколько правил. Если создание методов - это функция ближе к разработке, то создание правил - к администрированию. По сути правила определяют набор параметров для метода, с которыми удобно работать пользователю.

Список правил маскирования доступен в разделе меню **Правила маскирования**.

The screenshot shows the 'Jay Data' application interface. On the left, a sidebar menu has 'Правила маскирования' (Masking Rules) circled in red. The main window displays a table of masking rules. The table has three columns: 'Системное имя' (System Name), 'Наименование' (Name), and 'Метод маскирования' (Masking Method). The table lists various rules such as 'ADDITION\_NUMBER\_10', 'BANK\_CARD\_STRING\_DEFAULT', 'CONSTANT\_REPLACE\_CLOB\_DEFAULT', etc., each with a brief description of its function and the specific masking method it uses.



### Пример 28. Пример правил маскирования

Например, **REPLACE\_BY\_DICTIONARY** (Замена по Справочнику) - это метод маскирования, а Замена по справочнику имён (Россия), Замена по справочнику фамилий (Россия), Замена по справочнику проектов (Россия) - это правила маскирования, основанные на этом методе.

В системе для каждого **системного метода** заранее созданы правила маскирования с наиболее вероятными для использования значениями параметров. Сами правила не делятся на системные и пользовательские: они могут создаваться как по системным, так и по пользовательским методам.

Правила маскирования сгруппированы по методу, на котором они основаны. Это сделано для удобства поиска и фильтрации.

### Пример 29. Пример группировки правил

Правило маскирования Замена на константу 555, будет сгруппировано с другими правилами по методу Замена на константу (**CONSTANT\_REPLACE**). А правило маскирования Замена по справочнику имён (Россия) будет сгруппировано с другими правилами по методу **REPLACE\_BY\_DICTIONARY** (Замена по Справочнику).

В каждом правиле для каждого **параметра метода** устанавливается **Значение**, которое будет присвоено параметру если оно не будет переопределено пользователем системы на этапе **настройка правил маскирования**. Пользователь имеет возможность при назначении правила маскирования **домену** в столбце БД переопределить значения параметров на другие.

В карточке правила присутствует возможность протестировать скрипт метода, на котором оно основано (а также просмотреть его с помощью кнопки **</> Просмотр скрипта**), с указанными в правиле параметрами. Для этого в поле **Значение** необходимо ввести исходное значение, которое необходимо замаскировать и нажать кнопку **Тест**.

Планы обезличивания × Редактирование правила ма...

Правила маскирования > Редактирование правила маскирования

Системное имя REPLACE\_BY\_DICTIONARY\_STRING\_MYRULE \*

Наименование Замена по словарю несуществующих городов \*

Особенности метода Если одновременно установлены флаги unique и saveConsistency, то маскирование будет происходить консистентно, а не уникально.  
В случае, если размер словаря будет недостаточен, при установленном флаге уникальности, то будет добавлен постфикс. Пример:Иван\_1

Метод маскирования REPLACE\_BY\_DICTIONARY\_STRING (Замена по справочнику (строка)) 🔍 ⋮ ✕ \*

Параметры скрипта

Переменная	Тип	Значение	Описание
unique	Логическое	<input checked="" type="checkbox"/>	Заменять на уника...
saveConsistency	Логическое	<input type="checkbox"/>	Флаг, определяю...
maskingDictCode	Строковое	RUS_FANTASY_CI	Код справочника
consistKey	Строковое	REPLACE_BY_DIC	Ключ консистент...

Зависимости от столбцов

Переменная	Тип	Тип значений	Описание
------------	-----	--------------	----------

Тест <> Просмотр скрипта

Если правило основано на методе с **зависимостями от других столбцов**, то в нём необходимо указать какое из значений данных столбцов необходимо использовать по умолчанию: замаскированное или исходное.

Планы обезличивания × Редактирование правила ма...

Правила маскирования > Редактирование правила маскирования

Системное имя: FULL\_NAME\_GENERATOR\_STRING

Наименование: Генератор строки ФИО (строка)

Особенности метода

Метод маскирования: FULL\_NAME\_GENERATOR\_STRING (Генератор строки ФИО (строка))

Параметры скрипта

Переменная	Тип	Значение	Описание
patronymic	String	Замаскированное	Отчество
name	String	Замаскированное	Имя
surname	String	Замаскированное	Фамилия

Тест | Просмотр скрипта

OK | Закрыть

## Подключения к БД

**Подключение к БД** — сущность, которая содержит в себе параметры подключения к определенной базе данных.

Подключения к БД используются на первом этапе создания плана обезличивания (см. описание [Создание нового плана обезличивания](#)).

Создать | Изменить | Дублировать | Удалить

1 2 | Строк на стр. 20

Наименование	Тип БД	Хост	Порт	База данных	Схема	Пользователь
oracle_dst	Oracle	tdm-oracle02.lan	1 521	XE	TDM_USER	system
oracle_src	Oracle	tdm-oracle01.lan	1 521	XE	TDM_USER	system
amdb_src	PostgreSQL	10.20.1.49	5 432	amdb	public	postgres
amdb_dst	PostgreSQL	10.20.1.49	5 432	amdb_dst	public	postgres
source_demo	PostgreSQL	pdb01-drbs.lan	5 432	demo	bookings	postgres
target_demo	PostgreSQL	pdb01-drbs.lan	5 432	demo_target	bookings	postgres
sampler_demo_src	PostgreSQL	pdb01-drbs.lan	5 432	sampler_demo_src	public	postgres
sampler_demo_dst	PostgreSQL	pdb01-drbs.lan	5 432	sampler_demo_dst	public	postgres
tdm_src_Vertica	Vertica	10.20.6.60	5 433	tdm	public	dbadmin
tdm_dst_Vertica	Vertica	10.20.6.60	5 433	tdm	tdm_dst	dbadmin
amdb_demo	PostgreSQL	10.15.3.22	5 432	amdb_demo	public	postgres
amdb_prod	PostgreSQL	10.15.3.22	5 432	amdb	public	postgres
новое	Sybase	localhost	1 234	db	schema	пользователь



Поскольку в результате маскирования, обезличенные данные будут записываться в **БД-приёмник**, необходимо заранее создать подключения как для БД-источника, так и для БД-приёмника.

- **БД-источник** - эта БД является свежей копией оригинальной базы данных. Она содержит в себе незамаскированные чувствительные данные. Данная БД будет использоваться как источник данных, сами данные в ней модифицированы не будут. Рекомендуется снимать копию оригинальной базы данных, которую необходимо обезличить как можно ближе по времени к процессу запуска маскирования, чтобы в копии и оригинале не возникало расхождений. В некоторых случаях допускается (но не рекомендуется) указывать в качестве БД-источника саму оригинальную базу данных.
- **БД-приемник** - это вторая копия оригинальной базы данных. В данную БД в

соответствии с правилами маскирования будут загружаться обезличенные данные (в [режиме INSERT](#)) или данные в ней будут модифицироваться и обновляться (в [режиме UPDATE](#)).

Список поддерживаемых типов БД в системе:

- PostgreSQL;
- MySQL;
- Oracle;
- Vertica;
- ClickHouse;
- MS SQL Server;
- Apache Hive.

В списке подключений отображаются все подключения к базам данных, настроенные пользователями системы.

Подключение к БД содержит в себе следующие поля и разделы:

- **Наименование** - наименование подключения к БД;
- **Пользователь** - наименование учётной записи от БД;
- **Пароль** - пароль учётной записи от БД;
- **Тип БД** - тип базы данных;
- **Хост** - IP-адрес/имя хоста, на которой расположена БД;
- **Порт** - порт БД;
- **База данных** - наименование БД;
- **Схема** - схема БД;
- **Только источник** - данное подключение можно использовать только в качестве БД-источника.

## Создание нового подключения

Для создания нового подключения необходимо нажать на кнопку **Создать** в разделе **Подключения к БД**. Открывается новое окно по созданию нового подключения:

Подключения к БД > Редактирование подключения к БД

Наименование  \*

Пользователь  \*

Пароль  \*

Тип БД  \*

Хост  \*

Порт  \*

База данных  \*

Схема  \*

Далее заполняются обязательные поля. Чтобы узнать параметры подключения к нужной БД, необходимо в СУБД открыть настройки подключения к этой БД (или обратиться к администратору нужной БД).

Для проверки подключения необходимо нажать на кнопку **Проверить подключение** и система отобразит статус подключения: успешно или неуспешно. В случае ошибки подключения будет выведена информация об ошибке.

Подключения к БД > Редактирование подключения к БД

Ключ:

Наименование:

Пользователь:  \*

Пароль:  \*

Тип БД:  \*

Хост:  \*

Порт:  \*

База данных:  \*

Схема:  \*

Также проверить подключение можно, выбрав в списке нужное подключение и нажав на кнопку **⋮ → Проверить подключение**

Подключения к БД

Фильтр

Наименование:  Тип БД:  База данных / Схема / Пользователь:

<input type="checkbox"/>	Проверить подключение	Тип БД	Хост	Порт	База данных
<input type="checkbox"/>		PostgreSQL	10.15...	5 432	amdb_demo
<input checked="" type="checkbox"/>		PostgreSQL	10.20...	5 432	amdb_dst
<input type="checkbox"/>		PostgreSQL	10.15...	5 432	amdb
<input type="checkbox"/>		PostgreSQL	10.20...	5 432	amdb
<input type="checkbox"/>		Apache Hive	10.20...	10 000	tdm_demo
<input type="checkbox"/>		Apache Hive	10.20...	10 000	tdm_demo
<input type="checkbox"/>		ClickHouse	10.20...	8 123	tdm_target
<input type="checkbox"/>		ClickHouse	10.20...	8 123	tdm
<input type="checkbox"/>		PostgreSQL	pdb0...	5 432	demo_db_dst
<input type="checkbox"/>		PostgreSQL	pdb0...	5 432	demo_db_src

## Политики данных

**Политики данных** — сущность, которая группирует **домены** по принадлежности к одной и той же области информации.

*Пример 30. Объединение доменов и политики данных*

Такие домены, как **\_P\_LAST\_NAME** (Фамилия), **\_P\_PATRONYMIC** (Отчество) и **\_P\_NATIONALITY**

(Национальность) объединяются в Политику данных **\_P\_** (Персональная информация).

Политика данных содержит в себе следующие поля и разделы:

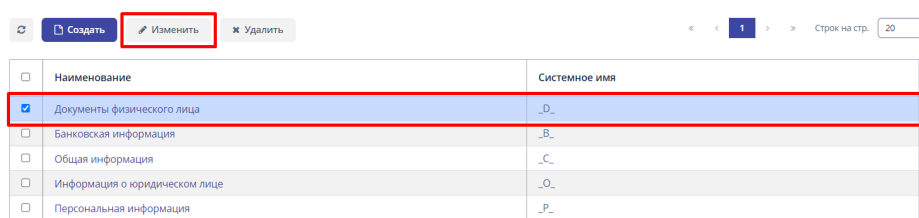
- **Наименование** - наименование Политики данных.
- **Системное имя** - системный код Политики данных латинскими буквами.
- **Список доменов** - список доменов, которые относятся к этой политике данных.

## Создание новой политики данных

Для создания перейти в раздел **Политики данных** и нажать на кнопку **Создать**.



В открывшемся окне редактирования политики данных необходимо заполнить обязательные поля. Для сохранения новой Политики данных необходимо нажать **OK**. Для изменения уже существующей Политики данных, необходимо выбрать нужную Политику данных из списка и нажать на кнопку **Изменить**.

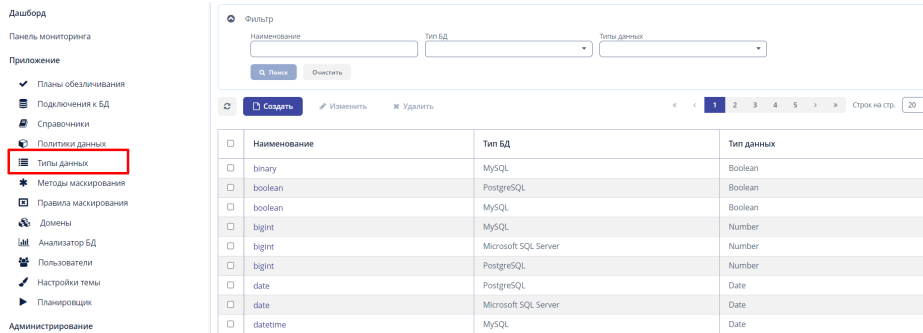


## Типы данных

**Тип данных** - сущность, описывающая соответствие типов данных из поддерживаемых БД эталонному типу данных в системе. В таблице раздела **Типы данных** тип данных каждой БД приводится к одному унифицированному типу данных Java, с которым могут работать **методы маскирования**. Метод маскирования работает только с одним типом данных: он принимает на вход значение этого типа и возвращает преобразованное значение того же типа.

*Пример 31. Пример для типов данных PostgreSQL и ClickHouse*

Например, тип данных для даты и времени **Date**, в PostgreSQL определяется как **date**, в ClickHouse как **Date32**, в MySQL **datetime**. Поэтому все перечисленные типы данных приводится к одному стандартному значению - **Date**.



По умолчанию в Системе заведены следующие унифицированные типы данных:

- String;
- Boolean;
- Number;
- Date;
- CLOB;
- BLOB.

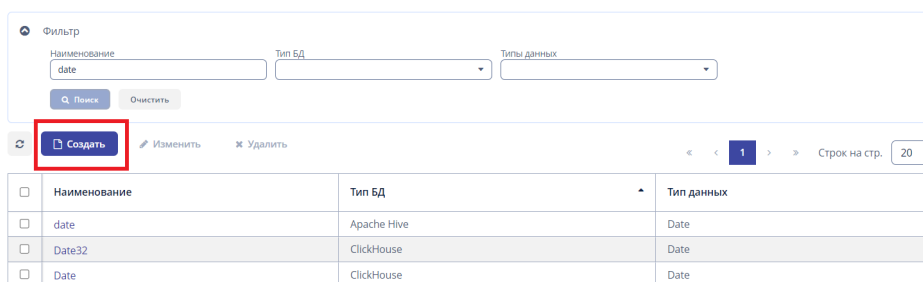
Тип данных содержит в себе следующие поля и разделы:

- **Наименование** — наименование типа данных;
- **Тип БД** — БД, к которой относится тип данных;
- **Тип данных** — эталонный унифицированный тип данных Java, к которому приводится этот тип БД. Для сохранения изменений и перехода на экран типов данных необходимо нажать кнопку **OK**.

Типы данных String, CLOB, BLOB имеют ограничение по загрузке данных из БД: по умолчанию это первые 65536 символов (65536 байт, 64 Kb). Символы, идущие после этого значения, обрезаются.

## Создание нового типа данных

Пользователь может добавить новые типы данных БД при необходимости. Для добавления нового типа данных необходимо перейти по кнопке **Создать** под окном с фильтрами.



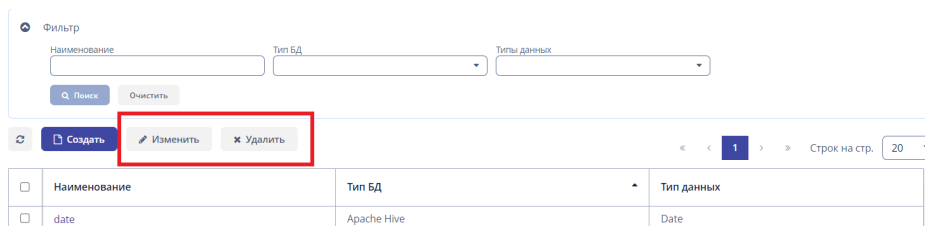
Открывается страница создания типа данных БД:

Наименование  \*

Тип БД  \*

Тип данных  \*

Для создания нового типа данных необходимо заполнить все обязательные поля. При выборе типа данных из таблицы пользователь может также **Изменить** или **Удалить** выбранный тип данных.



## Домены

**Домен** - сущность, определяющая чувствительные данные, являющиеся с точки зрения заказчика секретными, не подлежащей разглашению, которые хранятся в текущей колонке БД. Домены определяют содержимое колонки БД, которая участвует в **профилировании**. Если в процессе профилирования пользователь определил, что для поля **EMAIL** присваивается домен EMAIL, предполагается, что в этом поле лежат данные электронной почты пользователей.

Домены делятся на **системные** и **пользовательские**. Тип домена определяется значением поля **Системный**.

Домен содержит в себе следующие поля и разделы:

- **Наименование** - название домена;
- **Системное имя** - системный код домена;
- **Правило по умолчанию** - **правило**, установленное по умолчанию для данного домена;
- **Политика данных** - **политика данных**, к которой относится данный домен;
- **Описание** - описание домена;
- **Учитывать при профилировании** (только для пользовательских доменов) - при активном признаке отправляется POST-запрос в анализатор БД на создание указанного пользовательского домена;
- **Системный (признак)** - признак, который отвечает за принадлежность домена к типу **Системный**. Для пользовательских доменов данный признак остается неактивным и недоступным для изменений. Домены с неактивным признаком **Системный** считаются пользовательскими;
- **Метод профилирования** - определение того, каким образом происходит поиск конфиденциальных данных.

Поле **Метод профилирования** представляет собой выпадающий список, из которого можно выбрать одно из двух значений: **Регулярное выражение** или **Список значений**. При выборе значения из списка появляется поле для ввода регулярного выражения или списка значений (в зависимости от выбранного метода профилирования).

- **Проверка рег. выражения** - кнопка проверяет соответствие введенного пользователем **Значения для проверки** регулярному выражению или списку значений из текущего домена.
- **Доп. метод проверки** (только для системных доменов) – дополнительный метод проверки метода профилирования. Поскольку системные домены нельзя изменить, данное поле необходимо только для отображения информации о том, что в анализаторе БД для данного домена используется дополнительный метод проверки.

## Системные домены

**Системные домены** - это домены, определяющие данные, формат которых регулируется стандартами (например, законами РФ или данные имеют определённый формат). Системные домены определяют общеизвестные категории конфиденциальной информации, которые могут быть полезны для всех заказчиков, например ФИО, номер телефона, наименование организации.

Системные домены доступны только на чтение. Набор системных доменов является неизменяемым с точки зрения пользователя системы. Наименования системных доменов содержат префиксы, определяющие принадлежность домена какой-либо **политике данных**.

Ниже представлены наименования, описания и логика определения системных доменов, а также дополнительные методы проверки для некоторых из них.

### B\_CARD\_NUMBER (Номер банковской карты)

16-значный номер банковской карты, корректный с точки зрения алгоритма Луна,



платежной системы Мир, American Express, Visa или Mastercard.

**Дополнительный метод проверки:** Алгоритм Луна для номера банковской карты.

**Логика определения:**

- Домен определяет 16-значные номера банковских карт.
- Первая цифра номера указывает на платежную систему. Домен определяет номера, первые цифры которых совпадают со следующими значениями: [2-6]. Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы. Например, домен обнаруживает номер карты в строке “номер\_карты4216111122223333\_”, а в строке “942161111222233339” номер не обнаруживается.
- Домен определяет номера, в которых могут быть разделители “-”, “ ”, “\_” после каждого квартета чисел. Например, домен обнаруживает номер в строках “4216 1111 2222 3333” “4216\_1111\_2222\_3333”, а в строке “42-16 1111.2222\_3\_3\_3\_3” номер обнаруживается.
- Домен осуществляет проверку номера на его корректность с помощью алгоритма Луна.

#### **\_V\_PAYMENT\_ACCOUNT (Расчетный счет)**

20-значный расчетный счет в одной из следующих валют: российский рубль, доллар США, евро, юань, фунт стерлингов, тенге, белорусский рубль, иена.

**Дополнительный метод проверки:** \*Невозможно применить алгоритм проверки контрольной суммы, так как в общем случае неизвестен БИК банка, в котором может быть открыт рассматриваемый счет.

**Логика определения:**

- Домен определяет 20-значные номера банковских (расчетных) счетов.
- Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Домен определяет номера, первые три цифры которых совпадают со следующими значениями: [401-425, 430].
- Цифры на позициях 6, 7, 8 указывают на код валюты той или иной страны. Домен определяет следующие коды валют:
  - 051 (Армения)
  - 156 (Китай)
  - 364 (Иран)
  - 376 (Израиль)
  - 392 (Япония)
  - 398 (Казахстан)
  - 408 (КНДР)
  - 417 (Киргизия)

- 496 (Монголия)
  - 643, 810 (Россия)
  - 784 (ОАЭ)
  - 826 (Великобритания)
  - 840 (США)
  - 860 (Узбекистан)
  - 933 (Беларусь)
  - 934 (Туркменистан)
  - 941 (Сербия)
  - 944 (Азербайджан)
  - 949 (Турция)
  - 972 (Таджикистан)
  - 978 (Евросоюз)
- Домен определяет номера, в которых могут быть разделители “-”, “ ”, “\_” после каждого квартета чисел.

### **\_C\_DATE (Дата)**

Дата в формате дд.мм.гг, дд месяц или их вариациях.

#### **Логика определения:**

- Домен определяет даты в форматах дд.мм.гггг., дд.мм.гг., гггг.мм.дд., гг.мм.дд.
- Слева и справа от даты не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Домен определяет даты, в которых после числа и перед годом могут находиться наименования месяцев. Например, “31 декабря 1991 года”.
- Домен определяет даты, в которых могут быть разделители “/”, “.”, “-”, “ ”, “\_” между числом, месяцем и годом. Например, домен обнаруживает дату строках “31.12.1999”, “1999/12/31”, а в строке “3\1.12,1991” дата обнаруживаться не должна.

### **\_C\_EMAIL (Электронная почта)**

Адрес электронной почты в соответствии с RFC 5322 с произвольным доменным именем сервера почтового ящика.

#### **Логика определения:**

- Домен определяет адреса электронной почты с максимальной длиной в 254 символа (включая знак “@” и доменное имя).
- Внутри адреса не должно находиться больше одного знака “@”.
- Наименование почтового ящика (до знака “@”) должно состоять из букв, цифр, точек и

знаков “\_”. Левая граница наименования должна определяться пробелами или началом строки.

- Доменное имя должно состоять из букв и одной точки. Правая граница доменного имени должна определяться цифрами, спец. символами или пробелами.
- Домен определяет адреса, написанные латиницей как строчными, так и заглавными буквами.

### **\_C\_IMEI (IMEI)**

15- или 16-значный международный идентификатор мобильного оборудования (IMEI/IMEISV).

#### **Логика определения:**

- Домен определяет 15- и 16-значные международные идентификаторы мобильных устройств.
- Слева и справа от идентификатора не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Первые восемь цифр идентификатора указывают на коды ТАС (Type Allocation Code). Домен определяет самые популярные коды.
- Домен определяет идентификаторы, в которых могут быть разделители “-”, “ ”, “\_” между второй и третьей, восьмой и девятой, четырнадцатой и пятнадцатой цифрами.

### **\_C\_IMSI (IMSI)**

15-значный международный идентификатор мобильного абонента (IMSI).

#### **Логика определения:**

- Домен определяет 15-значные международные идентификаторы мобильных абонентов.
- Слева и справа от идентификатора не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Первые три цифры указывают на мобильный код страны. Домен определяет следующие коды стран:
  - 250 (Россия)
  - 255 (Украина)
  - 257 (Беларусь)
  - 282 (Грузия)
  - 283 (Армения)
  - 286 (Турция)
  - 289 (Абхазия)
  - 400 (Азербайджан)
  - 401 (Казахстан)

- 434 (Узбекистан)
- 436 (Таджикистан)
- 437 (Киргизия)
- 438 (Туркменистан) Домен определяет идентификаторы, в которых могут быть разделители “-”, “ ”, “\_” между третьей и четвертой, пятой и шестой цифрами.

### **\_C\_IP (IP-адрес)**

IP-адрес версии IPv4 или IPv6.

#### **Логика определения:**

- Домен определяет IP-адрес версии IPv4 и IPv6.
- В случае IPv4 домен определяет адреса, состоящие из 4 групп, разделенных точкой. Каждая группа должна состоять из цифр в диапазоне [0-255].
- В случае IPv6 домен определяет адреса, состоящие из 8 групп, разделенных двоеточием. Каждая группа должна состоять из 4 символов, которые могут принимать значения [0-9A-Fa-f]
- Слева и справа от адреса не должно находиться цифр и букв, но могут находиться спец. символы и пробелы.

### **\_C\_LOCALITY (Населенный пункт)**

Населенный пункт, город в России.

#### **Логика определения:**

- Домен определяет наименования самых популярных городов России.
- Слева и справа от наименования не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет наименования, написанные кириллицей как строчными, так и заглавными буквами.

### **\_C\_MAC (MAC-адрес)**

MAC-адрес устройства.

#### **Логика определения:**

- Домен определяет MAC-адрес устройства, состоящие из 6 групп, разделенных дефисом или двоеточием.
- Каждая группа адреса должна состоять из 2 символов, которые могут принимать значения [0-9A-Za-z].
- Слева и справа от адреса не должно находиться цифр и букв, но могут находиться спец. символы и пробелы.

## **\_C\_PHONE (Номер мобильного телефона)**

Номер мобильного телефона мобильного оператора России, начинающийся с +7, 7 или 8.

### **Логика определения:**

- Домен определяет 11-значные телефонные номера России. Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Домен определяет номера, первые цифры которых совпадают со следующими значениями: [7, 8], а вторые цифры совпадают со следующими значениями: [4, 9].
- Домен определяет номера, в которых вторая, третья и четвертая цифры находятся в круглых скобках. Например, "+7 (999) 123-45-67".
- Домен определяет номера, в которых могут быть разделители "-", " " между первой и второй, четвертой и пятой, седьмой и восьмой, девятой и десятой цифрами.

## **\_C\_STREET (Улица)**

Популярная в населенных пунктах РФ или странах бывшего СССР улица.

### **Логика определения:**

- Домен допределяет наименования самых популярных улиц России, а также популярных улиц Москвы.
- Домен определяет наименования улиц вне зависимости от наличия типа улицы (бульвар, шоссе и пр.). Например, домен обнаруживает название как в строке "Тверская улица", так и в строке "Тверская".
- Слева и справа от наименования не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет наименования, написанные кириллицей как строчными, так и заглавными буквами.

## **\_D\_INTERNATIONAL\_PASSPORT (Заграничный паспорт)**

Номер заграничного паспорта РФ.

### **Логика определения:**

- Домен определяет 9-значные номера заграничных паспортов граждан России. Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Домен определяет номера, первые две цифры которых совпадают со следующими значениями: [10, 20, 43, 44, 49, 50, 51, 53, 60-65, 70-76].
- Домен определяет номера, в которых могут быть разделители "№", " " между второй и третьей цифрой.

## **\_D\_MEDICAL\_POLICY (Полис ОМС)**

Номер полиса ОМС.

### **Логика определения:**

- Домен определяет 16-значные номера полисов ОМС.
- Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Домен определяет номера, в которых первые две цифры совпадают со следующими значениями: [01-84], одиннадцатые и двенадцатые цифры совпадают со следующими значениями: [00-31, 50-81], а тринадцатые и четырнадцатые цифры - со значениями [00-12].
- Домен определяет номера, в которых может быть разделитель “ ” между шестой и седьмой цифрами.

## **\_D\_PASSPORT (Паспорт)**

Номер паспорта гражданина РФ.

### **Логика определения:**

- Домен определяет 10-значные номера паспортов граждан России. Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Первые две цифры номера указывают на код ОКАТО. Домен определяет номера, первые две цифры которых соответствуют кодам ОКАТО регионов России.
- Домен определяет номера, в которых могут быть разделители “№”, “ ” между четвертой и пятой цифрами.

## **\_D\_SNILS (СНИЛС)**

Номер СНИЛС с корректной контрольной суммой.

**Дополнительный метод проверки:** Алгоритм проверки контрольной суммы СНИЛС.

### **Логика определения:**

- Домен определяет 11-значные номера СНИЛС.
- Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Домен определяет номера, в которых могут быть разделители “-”, “ ”, “\_” между третьей и четвертой, шестой и седьмой, девятой и десятой цифрами.
- Домен осуществляет проверку номера на его корректность с помощью проверки контрольного числа. Алгоритм проверки указан в п. 8 Приложения 1 информационного сообщения ПФ РФ от 21.02.2013 "Особенности представления страхователями в органы ПФР отчетности, начиная с отчетности с I квартала 2013 года".

## **\_D\_TIN (ИНН физического лица)**

12-значный ИНН физического лица с корректными контрольными суммами.

**Дополнительный метод проверки:** Алгоритм проверки контрольной суммы 12-значного ИНН.

### **Логика определения:**

- Домен определяет 12-значный номер ИНН физического лица. Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Первые четыре цифры номера указывают на коды ИФНС. Домен определяет номера, первые четыре цифры которых соответствуют кодам ИФНС.
- Домен осуществляет проверку номера на его корректность с помощью утвержденного ФНС России алгоритма проверки контрольного числа.

## **\_O\_NAME (Наименование юридического лица)**

Наименование юридического лица (ИП, АО, ПАО, ООО, ЗАО, ОАО, ОДО, ГУП или ГК).

### **Логика определения:**

- Домен определяет наименования юридических лиц, записанных после вида юридических лиц, т.е. после вида юридического лица должно быть записано хотя бы одно слово.
- Домен определяет следующие виды юридических лиц (а также их сокращения, например: ИП, АО, ООО и т.д.):
  - Индивидуальный предприниматель
  - Акционерное общество
  - Публичное акционерное общество
  - Общество с ограниченной ответственностью
  - Закрытое акционерное общество
  - Открытое акционерное общество
  - Общество с дополнительной ответственностью
  - Государственное унитарное предприятие
  - Госкорпорация/государственная корпорация
- Слева от наименования не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет виды юридических лиц, написанные как строчными, так и заглавными буквами.
- Домен определяет наименования, в которых должны быть от одного до пяти пробелов и ноль или одна кавычка между видом юридического лица и наименованием. Например, ООО “Рога и копыта”.

## **\_O\_TIN (ИНН юридического лица)**

10-значный ИНН юридического лица с корректной контрольной суммой.

**Дополнительный метод проверки:** Алгоритм проверки контрольной суммы 10-значного ИНН.

### **Логика определения:**

- Домен определяет 10-значный номер ИНН юридического лица. Слева и справа от номера не должно находиться цифр, но могут находиться буквы, спец. символы и пробелы.
- Первые четыре цифры номера указывают на коды ИФНС. Домен определяет номера, первые четыре цифры которых соответствуют кодам ИФНС.
- Домен осуществляет проверку номера на его корректность с помощью проверки контрольного числа с помощью алгоритма проверки, утвержденного ФНС России.

## **\_P\_EDUCATION\_LEVEL (Уровень образования)**

Уровень образования (в том числе академическая степень).

### **Логика определения:**

- Домен определяет уровни образования в России:
  - аспирант/аспирантура
  - бакалавр/бакалавриат
  - магистр/магистратура
  - специалист/специалитет
  - высшее
  - дошкольное
  - начальное общее
  - основное общее
  - среднее общее
  - среднее профессиональное
  - кандидат наук
  - доктор наук
- Слева и справа от уровня не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет уровни, написанные как строчными, так и заглавными буквами.

## **\_P\_FIRST\_NAME (Имя)**

Популярное на территории РФ или стран бывшего СССР имя кириллицей.

### **Логика определения:**



- Домен определяет четыреста самых популярных мужских и женских имен в России, написанных кириллицей.
- Слева и справа от имен не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет имена, написанных как строчными, так и заглавными буквами.

#### \_P\_FIRST\_NAME\_LATIN (Имя (лат.))

Популярное на территории РФ или стран бывшего СССР имя латиницей.

##### **Логика определения:**

- Домен определяет четыреста самых популярных мужских и женских имен в России, написанных латиницей.
- Слева и справа от имен не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет имена, написанных как строчными, так и заглавными буквами.

#### \_P\_LAST\_NAME (Фамилия)

Популярная на территории РФ или стран бывшего СССР фамилия кириллицей.

##### **Логика определения:**

- Домен определяет пятьсот пятьдесят самых популярных мужских фамилий в России, написанных кириллицей. Женские фамилии будут определяться соответственно мужским фамилиям, т.к. они образуются путем прибавления к мужской фамилии буквы “а”.
- Слева от фамилий не должно находиться букв, но могут находиться цифры, спец. символы и пробелы. Справа от фамилий не должно находиться букв кроме одной буквы “а”, также могут находиться цифры, спец. символы и пробелы.
- Домен определяет фамилии, написанные кириллицей как строчными, так и заглавными буквами.

#### \_P\_LAST\_NAME\_LATIN (Фамилия (лат.))

Популярная на территории РФ или стран бывшего СССР фамилия латиницей.

##### **Логика определения:**

- Домен определяет пятьсот пятьдесят самых популярных мужских фамилий в России, написанных латиницей. Женские фамилии будут определяться соответственно мужским фамилиям, т.к. они образуются путем прибавления к мужской фамилии буквы “а”.
- Слева от фамилий не должно находиться букв, но могут находиться цифры, спец. символы и пробелы. Справа от фамилий не должно находиться букв кроме одной буквы “а”, также могут находиться цифры, спец. символы и пробелы.

- Домен определяет фамилии, написанные латиницей как строчными, так и заглавными буквами.

### **P\_MARITAL\_STATUS (Семейное положение)**

Семейное положение.

#### **Логика определения:**

- Домен определяет семейное положение людей:
  - в разводе
  - вдова/вдовец
  - женат/не женат
  - замужем/не замужем
  - разведен/разведён/разведена
  - холост
- Слева и справа от положения не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет положения, написанные как строчными, так и заглавными буквами.

### **P\_NATIONALITY (Национальность)**

Национальность в значении принадлежности человека какому-либо гражданству.

#### **Логика определения:**

- Домен определяет наименования самых популярных национальностей в России.
- Слева и справа от наименования не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет наименования мужских и женских национальностей.
- Домен определяет наименования, написанные как строчными, так и заглавными буквами.

### **P\_PATRONYMIC (Отчество)**

Популярное на территории РФ или стран бывшего СССР отчество кириллицей.

#### **Логика определения:**

- Домен определяет тысячу самых популярных мужских и женских отчеств в России, написанных кириллицей.
- Слева и справа от отчеств не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет отчества, написанные как строчными, так и заглавными буквами.

## P\_PATRONYMIC\_LATIN (Отчество (лат.))

Популярное на территории РФ или стран бывшего СССР отчество латиницей.

### **Логика определения:**

- Домен определяет тысячу самых популярных мужских и женских отчеств в России, написанных латиницей.
- Слева и справа от отчеств не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет отчества, написанные как строчными, так и заглавными буквами.

## P\_PHILOSOPHICAL\_VIEWS (Философские взгляды)

Популярные философские взгляды.

### **Логика определения:**

- Домен определяет наименования самых популярных философских взглядов:
  - скептицизм/скептик
  - агностицизм/агностик
  - идеализм/идеалист
  - материализм/материалист
  - марксизм/марксист
  - неомарксизм/неомарксист
  - феноменология/феноменологист
  - экзистенциализм/экзистенциалист
  - структурализм/структуралист
  - рационализм/рационалист
  - эмпиризм/эмпирист
  - прагматизм/прагматист
  - евразийство/евразист
  - западничество/западник
  - космизм/космист
  - позитивизм/позитивист
  - традиционализм/традиционалист
  - этатизм/этатист
- Слева и справа от наименований не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет наименования, написанные как строчными, так и заглавными буквами.

## P\_POLITICAL\_VIEWS (Политические взгляды)

Популярные политические взгляды.

### **Логика определения:**

- Домен определяет наименования самых популярных политических взглядов:
  - анархизм/анархист/анархистка
  - коммунизм/коммунист/коммунистка
  - либерализм/либерал/либералка
  - консерватизм/консерватор
  - либертарианство/либертарианец/либертарианка
  - монархизм/монархист/монархистка
  - национализм/националист/националистка
  - пацифизм/пацифист/пацифистка
  - социализм/социалист/социалистка
- Слева и справа от наименований не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет наименования, написанные как строчными, так и заглавными буквами.

## P\_RELIGIOUS\_VIEWS (Религиозные взгляды)

Популярные религиозные взгляды.

### **Логика определения:**

- Домен определяет наименования самых популярных религиозных взглядов:
  - буддизм/буддист/буддистка
  - ислам
  - мусульманство/мусульманин/мусульманка
  - иудаизм/иудейство/иудей/иудейка
  - католицизм/католик/католичка
  - православие/православный/православная
  - протестантство/протестант/протестантка
  - христианство/христианин/христианка
- Слева и справа от наименований не должно находиться букв, но могут находиться цифры, спец. символы и пробелы.
- Домен определяет наименования, написанные как строчными, так и заглавными буквами.

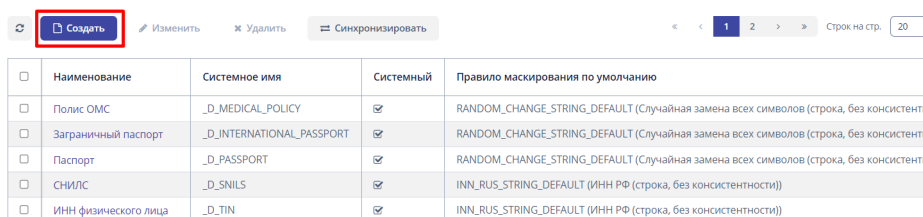
## Пользовательские домены

**Пользовательские домены** - это домены, добавленные в систему конечными пользователями, с проверками по произвольным регулярным выражениям. Пользовательские домены определяют чувствительные данные, формат которых в разных компаниях может отличаться, например: имя хоста, серийный номер оборудования, логин пользователя, наименование/код проекта и т.д. Пользовательские домены можно читать, создавать, редактировать, удалять.

Системные имена пользовательских доменов должны содержать только буквы латинского алфавита в верхнем регистре и символы нижнего подчеркивания. При этом первым символом должна быть буква.

## Создание нового пользовательского домена

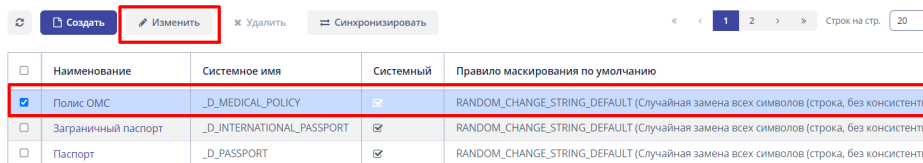
Для создания нового домена необходимо перейти в раздел **Домены** и нажать на кнопку **Создать**



<input type="checkbox"/>	Наименование	Системное имя	Системный	Правило маскирования по умолчанию
<input type="checkbox"/>	Полис ОМС	_D_MEDICAL_POLICY	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов (строка, без consistency))
<input type="checkbox"/>	Заграничный паспорт	_D_INTERNATIONAL_PASSPORT	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов (строка, без consistency))
<input type="checkbox"/>	Паспорт	_D_PASSPORT	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов (строка, без consistency))
<input type="checkbox"/>	СНИЛС	_D_SNILS	<input checked="" type="checkbox"/>	INN_RUS_STRING_DEFAULT (ИНН РФ (строка, без consistency))
<input type="checkbox"/>	ИНН физического лица	_D_TIN	<input checked="" type="checkbox"/>	INN_RUS_STRING_DEFAULT (ИНН РФ (строка, без consistency))

Далее открывается окно редактирования домена. Для создания домена необходимо заполнить обязательные поля, определённые в разделе **домены**.

Чтобы внести изменения в уже существующий домен, необходимо выбрать домен из списка и нажать на кнопку **Изменить**.



<input type="checkbox"/>	Наименование	Системное имя	Системный	Правило маскирования по умолчанию
<input checked="" type="checkbox"/>	Полис ОМС	_D_MEDICAL_POLICY	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов (строка, без consistency))
<input type="checkbox"/>	Заграничный паспорт	_D_INTERNATIONAL_PASSPORT	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов (строка, без consistency))
<input type="checkbox"/>	Паспорт	_D_PASSPORT	<input checked="" type="checkbox"/>	RANDOM_CHANGE_STRING_DEFAULT (Случайная замена всех символов (строка, без consistency))

## Комплексные домены

**Комплексный домен** — это домен, который определяет чувствительные данные более чем одного типа, содержащиеся в столбце БД. Введение понятия **Комплексного домена** необходимо для решения проблемы разнородности данных в пределах одного столбца БД.

*Пример 32. Пример данных, для которых может быть применён комплексный домен*

В таблице ниже в одном столбце БД содержатся сразу несколько доменов: фамилия, email и телефон.

*Таблица 26. Таблица с полем CONTACT\_INFORMATION, содержащей смешанные данные*

ID	CONTACT_INFORMATION
1	@ivanov.a ivanov.a@company.com +79001002030

ID	CONTACT_INFORMATION
2	@petrov.s petrov.s@company.com +79927725182
3	@samoilov.v samoilov.v@company.com +79260963271

Для комплексного домена создаются пользовательские [методы маскирования](#), содержащие индивидуальный Groovy-скрипт, который будет использоваться в маскировании таких столбцов БД. Так как в строковых полях теоретически могут быть записаны любые конфиденционные данные в свободной форме, каждая такая ситуация обрабатывается отдельно.

*Пример 33. Пример скрипта метода для обработки вышеуказанного комплексного столбца БД:*

```
import org.apache.commons.lang.*

String[] valueArray = value.split(' ')
String part1 = randomChangeMasking.randomChange(
    valueArray[0],
    saveConsistency,
    consistKey,
    startMasking,
    endMasking,
    convertEmptyValue,
    false
)
String part2 = '?'.repeat(3) + valueArray[1].substring(3, valueArray[1].length())
String part3 = valueArray[2].substring(0, valueArray[2].length() - 3) + '?'.
repeat(3)

return part1 + ' ' + part2 + ' ' + part3
```

В данном примере строка `@ivanov.a ivanov.a@company.com +79001002030` будет преобразована в `@asdcke.b ???nov.a@company.com +79001002???`. Каждая часть конфиденциальной строки маскируется отдельно, а затем конкатинируется в итоговое значение.

## Справочники

**Справочник** - это сущность, которая определяет наименование каталога данных.

Справочники и его [элементы](#) могут использоваться в определённых методах, где требуется заменить исходное значение на какое-либо другое. Например, коды справочников используются в качестве значений [параметров методов маскирования](#) в методе [Замена по справочнику](#).

Справочники по умолчанию, которые заведены в Системе:

- **Имена(Россия)** — список имен России. Все имена записаны кириллицей;
- **Фамилии(Россия)** — список фамилий России. Все фамилии записаны кириллицей;
- **Имена(Иностранные)** — список иностранных имен. Имена приведены латинскими буквами;
- **Компании (Россия)** — список несуществующих компаний;
- **IP-адреса** — список ip-адресов;
- **Несуществующие города (Россия)** — список несуществующих городов России;
- **Города (Россия)** — список городов России;
- **Проекты (Россия)** — список проектов, в которых может участвовать Организация;
- **Должности** — список должностей сотрудников;
- **Структурные подразделения (Россия)** — структурные подразделения для Организации;
- **Простые обезличенные значения (Россия)** — список простых обезличенных значений формата "Значение" + "Число";
- **ИФНС (Россия)** — список случайно сгенерированных чисел, приведенных к формату ИФНС;
- **Префиксы стран для международных номеров** — список префиксов стран, которые используются в международных банковских номерах. Применяется в методе маскирования [Замена международного номера \(IBAN\)](#)

Список справочников и их элементов может пополняться администратором системы.

## Управление справочниками

Для работы со справочником доступны следующие действия:

- **Создание.** Для создания нового справочника необходимо перейти на вкладку **Справочники** и нажать на кнопку **Создать**:

<input type="checkbox"/>	Наименование	Код	Комментарий	Активный
<input type="checkbox"/>	Компании (Россия)	RUS_COMPANY		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Имена (Иностранные)	ENG_FIRST_NAME		<input checked="" type="checkbox"/>

После этого открывается окно по созданию справочника:

Справочники > Редактирование справочника

Наименование  \*

Код  \*

Комментарий

Активный

После заполнения всех обязательных полей необходимо нажать на кнопку **OK** для сохранения данных. Для отмены создания нового справочника необходимо нажать на кнопку **Закрыть**.

- **Изменение.** При выборе справочника из списка становится доступной кнопка **Изменить**. После ее нажатия открывается страница редактирования выбранного справочника. На странице редактирования возможно изменить значения атрибутов для выбранного справочника. Если сделать справочник неактивным, то он будет недоступен для выбора при маскировании данных.
- **Удаление.** При выборе справочника из списка становится доступной кнопка **Удалить**. После нажатия на кнопку открывается окно подтверждения удаления справочника. При нажатии на кнопку подтверждения справочник будет удалён.

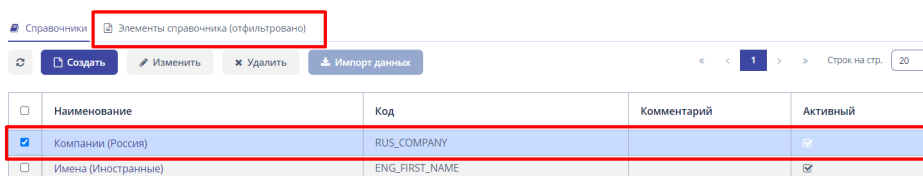
## Элементы справочника

**Элементы справочника** - это сущности, которые определяют наименования, содержащиеся в указанном [справочнике](#).



<input type="checkbox"/>	Наименование	Код	Комментарий	Активный
<input type="checkbox"/>	Компании (Россия)	RUS_COMPANY		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Имена (Иностранные)	ENG_FIRST_NAME		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Фамилии (Россия)	RUS_LAST_NAME		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Имена (Россия)	RUS_FIRST_NAME		<input checked="" type="checkbox"/>

Для того, чтобы открыть элементы конкретного справочника, необходимо предварительно выбрать необходимый справочник из списка на вкладке **Справочники** (отметить его галочкой) и перейти на вкладку **Элементы справочника (отфильтровано)**. Если предварительно не выбрать справочник, то на вкладке **Элементы справочника** будут присутствовать элементы всех справочников. **Отфильтровано** в список элементов добавляется только в том случае, если был выбран справочник, по которому необходимо отобразить элементы.



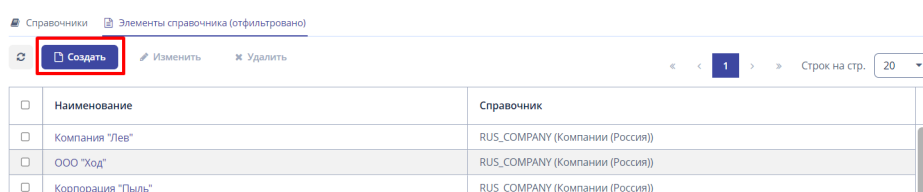
<input type="checkbox"/>	Наименование	Код	Комментарий	Активный
<input checked="" type="checkbox"/>	Компании (Россия)	RUS_COMPANY		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Имена (Иностранные)	ENG_FIRST_NAME		<input checked="" type="checkbox"/>

В открывшейся таблице на вкладке **Элементы справочника** можно посмотреть из каких элементов состоит выбранный справочник.

## Действия с элементами справочника

Для работы с элементами справочника доступны следующие действия:

- **Создание.** В каждом справочнике реализована возможность добавить новый элемент. Для этого нужно нажать на кнопку **Создать** на вкладке **Элементы справочника**.



<input type="checkbox"/>	Наименование	Справочник
<input type="checkbox"/>	Компания "Лев"	RUS_COMPANY (Компании (Россия))
<input type="checkbox"/>	ООО "Ход"	RUS_COMPANY (Компании (Россия))
<input type="checkbox"/>	Корпорация "Пыль"	RUS_COMPANY (Компании (Россия))

Открывается окно по добавлению нового элемента в справочнике:



Справочники > Редактирование элемента справочника

Наименование

Справочник

Для создания нового элемента заполняются следующие поля:

- **Наименование** - наименование элемента справочника;
- **Справочник** - наименование справочника, для которого создается элемент. Из выпадающего списка можно выбрать любой предложенный справочник.

После заполнения наименования необходимо нажать на кнопку **OK** для сохранения данных. Для отмены создания нового элемента необходимо нажать на кнопку **Заккрыть**.

- **Изменение.** При выборе одного элемента справочника из списка становится доступной кнопка **Изменить**.

Справочники | Элементы справочника (отфильтровано)

« < 1 > » Строк на стр. 20

<input type="checkbox"/>	Наименование	Справочник
<input checked="" type="checkbox"/>	Компания "Лев"	RUS_COMPANY (Компании (Россия))
<input type="checkbox"/>	ООО "Ход"	RUS_COMPANY (Компании (Россия))

После ее нажатия открывается страница редактирования элемента справочника. \* **Удаление.** При выборе одного элемента справочника из списка становится доступной кнопка **Удалить**.

Справочники | Элементы справочника (отфильтровано)

« < 1 > » Строк на стр. 20

<input type="checkbox"/>	Наименование	Справочник
<input checked="" type="checkbox"/>	Компания "Лев"	RUS_COMPANY (Компании (Россия))
<input type="checkbox"/>	ООО "Ход"	RUS_COMPANY (Компании (Россия))

После нажатия на кнопку открывается окно подтверждения удаления элемента справочника.

## Массовая загрузка (импорт) элементов справочника

В системе реализован механизм, позволяющий пользователю загружать элементы справочника в систему через форму массовой загрузки. Таким образом, у пользователя есть возможность загружать элементы в справочник сразу массивом данных, а не создавать их через интерфейс по одному.

В разделе **Справочники** при выборе конкретного справочника присутствует кнопка **Импорт данных**.

Справочники | Элементы справочника (отфильтровано)

« < 1 > » Строк на стр. 20

<input type="checkbox"/>	Наименование	Код	Комментарий	Активный
<input checked="" type="checkbox"/>	Компании (Россия)	RUS_COMPANY		<input type="checkbox"/>
<input type="checkbox"/>	Имена (Иностранные)	ENG_FIRST_NAME		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Фамилии (Россия)	RUS_LAST_NAME		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Имена (Россия)	RUS_FIRST_NAME		<input checked="" type="checkbox"/>

Пользователь выбирает нужный справочник из списка, кнопка **Импорт данных** активируется и пользователь может нажать на нее для загрузки элементов. Если выбрано более 1 справочника одновременно, кнопка не активна.

<input type="checkbox"/>	Наименование	Код	Комментарий	Активный
<input checked="" type="checkbox"/>	Компании (Россия)	RUS_COMPANY		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Имена (Иностранные)	ENG_FIRST_NAME		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Фамилии (Россия)	RUS_LAST_NAME		<input checked="" type="checkbox"/>

После нажатия на кнопку **Импорт данных** появляется всплывающее окно, в которое пользователь копирует элементы справочника. Пользователю выводится подсказка, что в данное поле необходимо добавить значения из файла или ввести значения вручную. Каждый элемент справочника должен начинаться с новой строки, это будет считаться разделителем между элементами справочника.

### Импорт данных ×

Справочник: Компании (Россия)

Введите список наименований элементов справочника. Каждый элемент справочника должен быть указан на отдельной строке.

**Импорт данных**

Отмена

Для начала импорта данных необходимо нажать на кнопку **Импорт данных** во всплывающем окне. После завершения загрузки появляется уведомление с количеством загруженных данных в справочник и с информацией о том, сколько записей было добавлено в справочник, а также сколько было обнаружено задублированных значений.

## Импорт данных

×

Справочник: Компании (Россия)

ОАО "Новая"  
ООО "ТДМ"  
ОАО "Новая"

### Импорт данных справочника

Справочник: Компании (Россия)

Загружены 2 записи

Найден 1 дубликат (не будет добавлен)

Импорт данных

Отмена

### Дополнительные проверки

1. Проверка на дубли в существующих справочниках. Если происходит добавление элемента в справочник, в котором уже есть такой элемент, то он не добавляется;
2. Проверка на дубли в загружаемом массиве элементов. Если в загружаемом массиве данных есть дубли, то они считаются одним элементом. Значения в разных регистрах, но с одинаковым набором символов не считаются дублиями;
3. За один раз пользователь может загрузить не более 10 тыс. записей. Если лимит превышен, то система уведомляет, что количество записей превысило допустимое. Необходимо разбить список элементов на несколько массивов данных.

# Администрирование

В данном разделе приведены функции по администрированию системы.

## Пользователи

В разделе **Пользователи** содержится информация о всех учётных записях, которые зарегистрированы в системе. В данном разделе можно управлять параметрами учетной записи.

Список атрибутов карточки пользователя:

- **Логин** — логин пользователя, используемый для входа;
- **Пароль** — пароль для входа
- **Подтвердите пароль** — подтверждение пароля;
- **Имя** — имя пользователя;
- **Фамилия** — фамилия пользователя;
- **E-mail** — электронная почта пользователя;
- **Часовой пояс** — часовой пояс пользователя;
- **Роли** — список ролей пользователя;
- **LDAP** — является ли учётная запись пользователя синхронизированной с LDAP (при установленном признаке) или локальной (при неустановленном признаке);
- **Запись активна** - признак, который делает учетную запись пользователя активной или деактивирует её.

Пользователю по умолчанию после создания не будет назначена никакая **роль**, поэтому при первом входе в систему пользователь увидит сообщение **Доступ запрещён**. После создания записи пользователя необходимо назначить ему необходимые роли, после чего под вновь созданной учётной записью станет возможно работать в системе.

## LDAP-пользователи

LDAP-пользователи могут загружаться в систему из Active Directory (см. раздел **LDAP**). При попытке входа через LDAP учётную запись, аутентификация в системе будет происходить средствами Active Directory.

LDAP учётной записи нельзя сбросить или сменить пароль, сбросить токен **Запомнить меня**, а также деактивировать её - это контролируется Active Directory. При деактивации в Active Directory учётная запись будет также деактивирована в системе.

У LDAP-пользователей всегда установлен признак **ldap**.

### Создание нового LDAP-пользователя

При первом входе под доменными учётными данными в базе данных системы будет создан

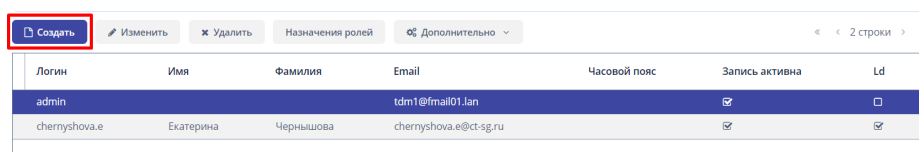
соответствующий пользователь (если ранее он не был создан) и произведена его синхронизация с LDAP (все атрибуты пользователя будут приведены в соответствие с Active Directory).

## Локальные пользователи

Учётные записи могут создаваться в самой системе, в этом случае в карточке пользователя признак **ldap** не будет установлен. У локальной учётной записи корректность введённого логина и пароля проверяется самой системой. Локальной учётной записи можно сбросить или сменить пароль, сбросить токен **Запомнить меня**, а также деактивировать её.

### Создание нового локального пользователя

Для создания нового пользователя необходимо перейти в раздел **Пользователи** и нажать на кнопку **Создать**. Создание пользователя возможно только из под учётной записи администратора.



Логин	Имя	Фамилия	Email	Часовой пояс	Запись активна	Ld
admin			tdm1@mail01.lan		<input checked="" type="checkbox"/>	<input type="checkbox"/>
cherryshova.e	Екатерина	Чернышова	cherryshova.e@ct-sg.ru		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Открывается страница создания нового пользователя.

Пользователи > Пользователь

Логин  \*

Пароль  \*

Подтвердите пароль  \*

Имя

Фамилия

Email

Часовой пояс

Запись активна

Ldap

Для создания пользователя необходимо заполнить обязательные атрибуты на карточке.

Для сохранения изменений и возвращения на экран **Пользователи** необходимо нажать кнопку **OK**.

При ручном создании пользователя учётная запись всегда будет являться **локальной**.

После создания записи пользователя необходимо назначить ему необходимые роли, после чего под вновь созданной учётной записью станет возможно работать в системе.

## Ролевая модель управления доступом

В системе заложена преднастроенная ролевая модель управления доступом. Существуют **Ресурсные роли** (ограничивающие права доступа на сами сущности, экраны, атрибуты и т.д.) и **Row-level роли** (ограничивающие видимость конкретных записей в системе).

Каждый пользователь может иметь несколько ролей. В данном случае пользователь получает права по объединению прав всех имеющихся у него ролей.

В системе настроены 4 ресурсные роли:

- **Администратор Системы** - имеет неограниченный доступ ко всем функциям системы.
- **Администратор Безопасности** – имеет доступ к настройке ролей и полномочий, к списку пользователей и логам/журналу аудита;
- **Разработчик правил маскирования** – имеет доступ только к чтению, созданию, изменению и удалению [правил/методов/параметров методов](#) маскирования. Остальные разделы доступны только на чтение;
- **Инженер по маскированию** - имеет доступ только к чтению, созданию, изменению и удалению [статических](#) и [динамических](#) планов обезличивания. Остальные разделы доступны только на чтение;

И 1 Row-level роль:

- **Отображать только свои Планы** - определяет ограничение доступа к планам обезличивания, созданным другими пользователями. В случае наличия данной Row-level роли пользователь видит только свои планы обезличивания;

По ресурсным ролям действует правило: если роль не назначена - значит нет и прав на эти ресурсы. По row-level ролям наоборот: если роль не назначена, значит доступны все записи.

## Матрица ролей и полномочий

Таблица 27. Матрица ролей и полномочий

Роль / Функции	Администратор системы	Администратор безопасности	Разработчик правил маскирования	Инженер по маскированию
<b>Сущности системы</b>				
<a href="#">Пользователи</a> и Роли	CRUD	CRUD	R	R
<a href="#">Правила маскирования</a> (+ параметры)	CRUD		CRUD	R
<a href="#">Методы маскирования</a> (+ параметры)	CRUD		CRUD	R

Роль / Функции	Администратор системы	Администратор безопасности	Разработчик правил маскирования	Инженер по маскированию
Домены	CRUD		R	R
Политики данных	CRUD		R	R
Справочники (+ Элементы справочника)	CRUD		R	R
Типы данных	CRUD		R	R
Планы обезличивания	CRUD	R	CRU	CRU
Подключения к БД	CRUD		CRU	CRU
<b>Другие разделы</b>				
Логи системы	CRUD	CRUD	C	C
Анализатор БД	R		R	R
Планировщик	CR		CR	CR
Дашборды	CRUD		R	R
Информация о лицензии	R	R		



**CRUD** - создание (CREATE), чтение (READ), изменение (UPDATE), удаление (DELETE), (пустая ячейка) – отсутствие доступа.

## Сопоставление ролей и групп Active Directory

В системе реализован механизм автоматического присвоения пользователям **ролей** в системе, исходя из его принадлежности какой-либо группе Active Directory. Для функционирования данного механизма необходимо, чтобы система была подключена к **LDAP**.

Сопоставление ролей и групп Active Directory задаётся в файле `application.properties` модуля **JMIX**.

*Пример 34. Пример сопоставления ролей и групп AD в `application.properties` модуля JMIX:*

```
jmix.ldap.synchronize-role-assignments = true
jmix.ldap.roleMapper.mask-engineer = Engineers
jmix.ldap.roleMapper.mask-rule-developer = Developers
jmix.ldap.roleMapper.secure-admin = SecurityAdmins
jmix.ldap.roleMapper.system-admin = Admins
jmix.ldap.roleMapper.own-plans = Engineers,Developers,SecurityAdmins
```

Так как в системе всего 5 ролей (включая ресурсные и row-level роли), то весь конфиг состоит из 5 записей.

Свойство `jmix.ldap.synchronize-role-assignments` включает функционал сопоставления ролей и групп Active Directory.

Далее для каждой роли в системе после знака `=` через запятую перечислены соответствующие роли в Active Directory (1 запись в свойствах = 1 роль в системе), где `jmix.ldap.roleMapper` - префикс роли, а `secure-admin` - конкретный код роли в системе.

#### Таким образом:

Если пользователь окажется в группе `Engineers` в Active Directory, то он получит роли `mask-engineer` и `own-plans` в системе.

Если пользователь окажется в группе `Admins` в Active Directory, то он получит одну роль `system-admin` в системе.

Если пользователь окажется в группах `Admins` и `Developers` в Active Directory, то он получит 3 роли в системе: `system-admin`, `mask-rule-developer` и `own-plans`, т.е. у него будут полные права администратора (из-за наличия у него роли `system-admin`), но он будет видеть только свои [планы обезличивания](#).

## LDAP

В системе присутствует встроенный компонент (аддон) **LDAP**, который позволяет использовать возможности службы каталогов, например, Microsoft Active Directory. LDAP обеспечивает централизованное хранилище логинов и паролей, что позволяет приложениям подключаться к LDAP-серверу для подтверждения доступа.

С помощью интеграции с LDAP также происходит синхронизация пользователей. При первой попытке входа пользователя под своими данными учётная запись пользователя автоматически создаётся в системе с соответствующим логином. **Роль** соответственно при этом никакая не назначается и при попытке входа система уведомит пользователя об отсутствии доступа. Для предоставления доступа администратор должен назначить роль соответственно [матрице ролей и полномочий](#). После назначения роли при входе под своими учётными данными пользователь беспрепятственно работает под своими правами.

## Конфигурация

Параметры подключения к LDAP задаются в `application.properties`-файле модуля JMX.

*Пример 35. Параметры подключения к LDAP в `application.properties` модуля JMX:*

```
jmix.ldap.enabled = true
jmix.ldap.urls = ldap://192.168.1.100:389/
jmix.ldap.baseDn = dc=company,dc=local
jmix.ldap.managerDn = CN=my_ldap_user,OU=Service,OU=Company,DC=company,DC=local
jmix.ldap.managerPassword = password
jmix.ldap.userSearchFilter = (uid={0})
jmix.ldap.useActiveDirectoryConfiguration = true
```



```
jmix.ldap.activeDirectoryDomain = company.local
jmix.ldap.synchronizeUserOnLogin = true
```

## Email

Email-уведомления приходят пользователю по событиям, связанным с его текущими [статическими планами обезличивания](#). Если пользователь запустил профилирование, то ему приходит уведомление на почту автоматически.

Список событий, о которых происходит оповещение:

- Завершение или ошибка профилирования;
- Завершение или ошибка маскирования;
- Завершение или ошибка проверки.

### Шаблон сообщения для пользователя

Заголовок: JayData: <тип события> (например, JayData: Завершен процесс маскирования)

Тело сообщения: Уважаемый пользователь <имя\_пользователя>, Процесс профилирования/маскирования/проверки плана <название\_плана> завершен/завершен с ошибкой в <дата\_и\_время\_события>

С Уважением, команда Jay Data.

## Конфигурация

Параметры отправки Email задаются в application.properties-файле модуля DATAPROCESSOR.

*Пример 36. Параметры отправки Email в application.properties модуля DATAPROCESSOR:*

```
#####
#                               Mail                               #
#####
spring.mail.host=fmail01.lan
spring.mail.port=25
spring.mail.protocol=smtp
#spring.mail.username=
#spring.mail.password=
jmix.email.from-address=notifications@jaydata
# имя пользователя, от которого будет отправляться email (оставить пустым, чтобы
запрос шёл от системного пользователя)
jmix.email.async-sending-username=
# локаль загружаемых шаблонов
tdm.email.messages.locale=ru
# периодичность отправки сообщений (каждую минуту)
jmix.email.email-sending-cron=0 * * * * ?
```

```
# периодичность очистки сообщений (каждый день в 00:00)
jmix.email.email-cleaning-cron=0 0 0 * * ?
# срок жизни сообщений
jmix.email.max-age-of-non-important-messages=1
jmix.email.max-age-of-important-messages=14
# при зачистке удалять также вложения
jmix.email.clean-file-storage=true
```

## Аудит действий пользователя

Вся информация по логируемым действиям доступна в лог файлах. В записи аудита действий пользователя в лог-файле содержится следующая информация:

- **Тип действия** - какое именно действие из журналируемых событий было совершено;
- **Дата и время** - дата когда было совершено действие;
- **Описание действия** - подробная информация о действии;
- **Пользователь** - логин пользователя, который совершил действие.

### Список журналируемых событий:

- вход в систему;
- выход из системы;
- неудачная попытка входа;
- создание нового пользователя;
- удаление пользователя;
- изменение роли пользователя;
- запуск маскирования;
- завершение маскирования;
- запуск проверки;
- завершение проверки;
- создание бизнес-сущности;
- изменение бизнес-сущности;
- удаление бизнес-сущности;
- истек срок лицензии;
- лицензия истекает через 14 дней;
- лицензия истекает через 3 дня;
- заканчивается число БД, доступных для подключения;
- достигнуто максимальное число БД, доступных для подключения.

События аудита в лог-файлах начинаются с ключевого слова

c.c.t.c.l.AuditEventListener.AUDIT.