



Программа для ЭВМ  
**«ЖИЛИЩНЫЙ СТАНДАРТ –  
Freemium»**

---

Функциональные характеристики

Тюмень, 2022 г.

## Оглавление

Оглавление .....	2
1. Список терминов и сокращений .....	3
2. Назначение документа.....	5
3. Общее описание и функциональные характеристики ПО .....	5
4. Архитектура системы.....	5
5. Схема информационного взаимодействия.....	7
6. Сервис аутентификации и управления пользователями «ЖС-Freemium» .....	8
7. Информация, необходимая для установки ПО.....	9
7.1. Требования к серверу .....	9
7.2. Используемое программное обеспечение.....	9
7.3. Системные требования.....	9
7.4. Установка системы .....	10
7.4.1. Подготовка сервера для размещения пользовательской копии АСУ .....	10
8. Управление учетными записями пользователей .....	18
Приложение 1 .....	19

## 1. Список терминов и сокращений

Таблица 1

№ п/п	Термин	Определение
1	Деплой сервер	Основной источник поставки программного комплекса «Жилищный Стандарт – Freemium»
2	Идентификатор инсталляции	Уникальная идентификационная строка (обычно состоит из номера региона и транслитерации сокращенного наименования организации клиента, например, 74_mur_gkh
3	Выделенный сервер	Отдельная физическая (или виртуальная изолированная) машина. Термин один, используется для упрощения
4	VPN адрес	Уникальный адрес в распределенной защищенной сети (удовлетворяющий требованиям ГОСТ или иным современным криптографически стойким стандартам)
5	Git	Распределённая система управления версиями для управления разработкой и версиями программного обеспечения (является свободным программным обеспечением, лицензия GNU GPL версии 2).
6	runner	Агент, который занимается выполнением инструкций из специального файла .gitlab-ci.yml.
7	Ansible	Система управления конфигурациями, написанная на Python, с использованием декларативного языка разметки для описания конфигураций. Используется для автоматизации настройки и развертывания программного обеспечения.
8	CI/CD	Технология и набор инструментов для непрерывной доставки и непрерывной интеграции программного обеспечения, чаще

№ п/п	Термин	Определение
		комплексов программного обеспечения
9	Nginx	Сервер для исполнения веб приложений и проксирования запросов
10	Публичный ключ	Ключ для зашифровки данных (обратный процесс выполняется с использованием закрытого ключа)
11	pg_dump	Компонент из поставки СУБД Postgres Pro, используется для изготовления резервных копий (дампов) баз данных
12	PostgreSQL	Свободная объектно-реляционная система управления базами данных
13	Сертификат	Электронный документ, подтверждающий принадлежность владельца открытого ключа или каких-либо атрибутов
14	url	Единообразный локатор (определитель местонахождения) ресурса
15	Сокет	Название программного интерфейса для обеспечения обмена данными между процессами
16	Проксирование	Механизм, позволяющий клиентам выполнять косвенные запросы к другим сетевым службам
17	Вендоры	Набор зависимых (сторонних) библиотек, распространяемых под свободными лицензиями
18	Деплой	Задача развертывания приложения на новой машине/или на той же самой, но новой его версии

## 2. Назначение документа

Настоящий документ описывает назначение, функциональные характеристики и архитектуру программного комплекса «Жилищный Стандарт – Freemium» (далее – ЖС-Freemium), а также информацию, необходимую для установки и эксплуатации.

Документ содержит описание модулей системы ЖС-Freemium и схемы взаимодействия модулей.

## 3. Общее описание и функциональные характеристики ПО

ЖС-Freemium представляет собой программный комплекс, построенный на базе свободно распространяемого программного обеспечения. ЖС-Freemium предназначен для автоматизации основных бизнес-процессов управляющих компании, ТСЖ, ресурсоснабжающих и обслуживающих организаций в сфере ЖКХ, а также органов власти:

- Работа с заявками от населения;
- Постановка задач сотрудникам;
- Документооборот;
- Ведение объектов жилищного фонда;
- Ведение реестра аварийных отключений;
- Размещение данных в ГИС ЖКХ;

## 4. Архитектура системы

Ядро ЖС-Freemium состоит из базовых сущностей (например: Пользователь, Роль, Группы доступа, Дом, Адресообразующий элемент, Справочники, Организации, и т.д.) и бизнес данных (Заявка, Задача, Аварийное отключение, и т.д.), построенных на взаимодействии базовых сущностей. Схемы и поля бизнес данных приведены в Приложении 1 «Схема данных».

Жизненные циклы процессов описываются в нотации конечного автомата с определением алфавита, состояний и переходов, влекущих за собой действия и/или пользовательский ввод. Процессы в различных состояниях могут иметь дочерние процессы или процессы следствия. Для организации удобного визуального представления организован механизм ведения регистраторов в виде журналов, объединяющих в себе совокупность родственных (по определяемым пользователям признакам) процессов.

Реализация универсального отчета построено по принципам, заложенным в системе одинарной записи в бухгалтерском учете, когда в момент изменения состояния процесса, в результате поступления символа, фиксируется: дата, состояние процесса, перечень объектов участвующих в процессе. Аналитика гибко агрегирует количественные изменения по заданному интервалу дат и интересующим базовым объектам.

Процессы в различных состояниях могут иметь шаблонные печатные формы.

Реакция на действия пользователя в ЖС-Freemium осуществляется тремя способами:

- непосредственное выполнение в браузере клиента;
- обработка в быстрой очереди (действия выполняются в фоновом режиме, время обработки минимально);
- обработка в медленной очереди (архитектурно заложена возможность горизонтального масштабирования очереди медленных событий).

## **Взаимодействие ядра ЖС-Freemium с внутренними сервисами и компонентами**

Мобильные приложения (Сотрудника Android, Сотрудника iOS, Жителя Android, Жителя iOS) авторизуются используя Bearer Token (описание стандарта OAuth 2.0 <https://tools.ietf.org/html/draft-ietf-oauth-v2-bearer-23> )

Пользователи ЖС-Freemium могут получить личный токен авторизации через сканирование QR-кода в своем профиле (подробное описание доступно в пользовательской документации [mediawiki.smart-uk.ru](http://mediawiki.smart-uk.ru))

Взаимодействие по API компонентов «Типовой сайт», Служба доставки WebSocket, АТС «ЖС-Freemium» с ядром «ЖС-Freemium» осуществляется по защищенным каналам VPN с использованием basic авторизации (стандарт http)

## 5. Схема информационного взаимодействия

### Взаимодействие с внешними сервисами

#### Адресная база ГАР

В ЖС-Freemium реализовано получение данных из адресной базы ГАР посредством API. Копия адресной базы ГАР, к которой обращается АСУ, находится на сервере ООО «Жилищный стандарт» и поддерживается в актуальном состоянии: свежие обновления базы регулярно пополняют копию адресной базы

Обмен данными с базой ГАР используется для следующих целей:

- Поиск идентификатора ГАР по адресу дома в ЖС-Freemium осуществляется со страницы Сопоставление домов с ГАР в разделе ГИС ЖКХ;
- Заполнение адреса организации в форме создания Контрагента.

Кроме того, в ЖС-Freemium реализована процедура проверки актуальности ранее внесенного адреса ГАР у Дома.

Данная процедура запускается со страницы списка Домов. Актуальность процедуры в том, что адресная база ГАР динамична, а пользователи ЖС-

Freemium не безгрешны, а потому идентификаторы ГАР, внесенные когда-либо пользователями в ЖС-Freemium, могут быть устаревшими или ошибочными. Процедура проверки сопоставляет все внесенные идентификаторы с актуальной базой адресов ГАР и базой адресов ГИС и актуализирует статусы этих идентификаторов.

### **API Государственной Информационной Системы Жилищно-Коммунального Хозяйства**

Описание API размещено на странице системы <https://dom.gosuslugi.ru/#!/regulations?userCtgrCode=1> в архиве «Регламент информационного взаимодействия внешних информационных систем с ГИС ЖКХ».

Взаимодействие по API производится через SOAP RPC tls по ГОСТ, посылки и документы подписываются ЭЦП (ГОСТ).

### **API Почты России**

Для отслеживания трек номеров исходящей корреспонденции ЖС-Freemium ведет взаимодействие с API Почты России. Документация и описание доступны по ссылке <https://www.pochta.ru/support/business/api>

## **6. Сервис аутентификации и управления пользователями «ЖС-Freemium»**

### **Система авторизации и разграничения доступа**

Разграничение доступа в ЖС-Freemium осуществляется при помощи механизмов ролевой модели (когда пользователь имеет права на просмотр, редактирование, выполнение других действий с типами данных);



## 7. Информация, необходимая для установки ПО

### 7.1. Требования к серверу

1. Процессор класса не ниже i5.
2. Оперативной памяти не меньше 8 Гб, рекомендуем 16 Гб.
3. Жесткий диск SSD мин. объема 120 Гб, 2 штуки (для надежности).
4. Жесткий диск HDD мин. объема 1 Тб, рекомендуем 2 Тб - для резервных копий.
5. Блок бесперебойного питания емкостью не меньше 700VA.
6. 2 сетевые карты.
7. Подключение к интернету, выделенный IP-адрес на точке подключения.

### 7.2. Используемое программное обеспечение

Для функционирования ЖС-Freemium, необходим следующий набор вспомогательного программного обеспечения:

1. Операционная система на базе Linux;
2. Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в среде виртуализации на уровне операционной системы. Не ниже версии 17.05.
3. OpenVPN — свободная реализация технологии виртуальной частной сети (VPN) с открытым исходным кодом для создания зашифрованных каналов типа точка-точка или сервер-клиенты между компьютерами.

### 7.3. Системные требования

ЖС-Freemium может функционировать как на операционных системах (ОС) семейства Linux - подойдет любая операционная система Linux с ядром начиная от 3.10, рекомендуется ядро => 4.9 (Например Астра Линукс).

Для функционирования системы, используются следующие TCP-порты:

1. 80 — для веб-приложения;
2. 443 — для веб-приложения по защищенному протоколу SSL;
3. 8081 - для веб-приложения;
4. 8082,8083 — для websocket по защищенному протоколу SSL;
5. 9323,2376 - системные порты для службы виртуализации.

Мониторинг и управление демоном.

## 7.4. Установка системы

### 7.4.1. Подготовка сервера для размещения пользовательской копии АСУ

#### 1. Конфигурационные файлы для Веб сервера

Создаем или правим конфигурационный файл:

```
nano <Идентификатор инсталляции>.it-uk.ru.conf
```

```
##declare php upstream
```

```
upstream php-fpm-upstream {
```

```
    server php-fpm:9000;
```

```
}
```

```
###Config asu
```

```
server {
```

```
    listen 80;
```

```
    server_name
```

```
    <доменное имя инсталляции>.it-uk.ru / smart-uk.ru
```

```
    www.<доменное имя инсталляции>.it-uk.ru / smart-uk.ru
```

```
    ;
```

```
    return 301 https://$server_name$request_uri;
```

```
}
```

```
server {
```

```

listen 443 ssl http2;
server_name
<доменное имя инсталляции>.it-uk.ru / smart-uk.ru
www.<доменное имя инсталляции>.it-uk.ru / smart-uk.ru
;
#Include ssl config
include /etc/nginx/ssl.d/файл с сертификатами;

##Data folder asu
root /home/<Идентификатор инсталляции>/release;

index index.php index.html index.htm;

location / {
    try_files $uri $uri/ =404;
}

location ~ /\.php$ {
    try_files $uri =404;
    fastcgi_pass php-fpm-upstream;
    fastcgi_read_timeout 120;
    fastcgi_send_timeout 120;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
games:/opt/node/bin:/bin;
    include fastcgi_params;

```

```
}

location ~* /api/(.*)$ {
    try_files $uri $uri/ /index.php?$query_string;
}

}

server {
    listen 80;
    server_name
    <доменное имя инсталляции сайта>
    www.<доменное имя инсталляции сайта>
    ;
    return 301 https://$server_name$request_uri;
}
```

```
### сайт
server {
    listen 443 ssl http2;
    server_name
    <доменное имя инсталляции сайта>
    www.<доменное имя инсталляции сайта>
    ;
    #Include ssl config
    include /etc/nginx/ssl.d/it-uk.ru/файл с сертификатами;
```

```

##Data folder site
root /home/<Идентификатор инсталляции>/site/web;

index index.php index.html index.htm;
location / {
# Redirect everything that isn't a real file to index.php
    try_files $uri $uri/ /index.php$is_args$args;
}

location ~ \.php$ {
    try_files $uri =404;
    fastcgi_pass php-fpm-upstream;
    fastcgi_read_timeout 120;
    fastcgi_send_timeout 120;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
games:/opt/node/bin:/bin;
    include fastcgi_params;
}
location /reformagkh {
    rewrite ^/reformagkh/(.*)$ https://<доменное имя инсталляции>.it-uk.ru /
smart-uk.ru/uploads/reformagkh/profile988/$1 redirect;
}
location /images {
    autoindex on;

```

```

    }

}

###Config websocket с ssl для доменов
server {
    listen 8081 http2 ssl;
    server_name
    <доменное имя инсталляции>.it-uk.ru / smart-uk.ru
    www.<доменное имя инсталляции>.it-uk.ru / smart-uk.ru
    ;
    #Include ssl config
    include /etc/nginx/ssl.d/it-uk.ru/файл с сертификатами;
    access_log off;
    location / {
        proxy_pass http://websocket:8082;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_read_timeout 86400;
    }
}

```

```

### proxy websocket без ssl - без ssl всегда 8082
server {
    listen 8082;

```

```
server_name
<доменное имя инсталляции>.it-uk.ru / smart-uk.ru
www.<доменное имя инсталляции>.it-uk.ru / smart-uk.ru
;
access_log off;
location / {
    proxy_pass http://websock:8082;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_read_timeout 86400;
}

}

location ~ /\.php$ {
    try_files $uri =404;
    fastcgi_pass adminer-upstream;
    fastcgi_read_timeout 120;
    fastcgi_send_timeout 120;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
```

```
    fastcgi_param  PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
games:/opt/node/bin:/bin;
    include fastcgi_params;
}

location /bup {
    alias /bup/pg_bases;
    autoindex on;
}

}
```

## 2. Настройка контейнера Базы Данных PostgreSQL

configs/postgresql.conf

Приводим к виду

```
# Add settings for extensions here
listen_addresses = '*'
shared_buffers = 6096MB
log_min_duration_statement = 400
max_connections = 500
temp_buffers = 1024MB
work_mem = 1024MB
maintenance_work_mem = 1024MB
shared_preload_libraries = 'pg_stat_statements'
```



`pg_stat_statements.max = 10000`

`pg_stat_statements.track = all`

`random_page_cost = 1.1`

`datestyle = 'iso, dmy'`

`lc_messages = 'ru_RU.UTF-8'`

`lc_monetary = 'ru_RU.UTF-8'`

`lc_numeric = 'ru_RU.UTF-8'`

`lc_time = 'ru_RU.UTF-8'`

Рассмотрим опции:

- `shared_buffers`

Рекомендуется ставить не менее 40% от доступной озу, далее прирост в производительности является несущественным

([https://wiki.postgresql.org/wiki/Tuning\\_Your\\_PostgreSQL\\_Server](https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server))

- `random_page_cost`

При уменьшении этого значения по отношению к `seq_page_cost` система начинает предпочитать сканирование по индексу; при увеличении такое сканирование становится более дорогостоящим. Оба эти значения также можно увеличить или уменьшить одновременно, чтобы изменить стоимость операций ввода/вывода по отношению к стоимости процессорных операций, которая определяется следующими параметрами.

- `log_min_duration_statement (integer)`

Записывает в журнал продолжительность выполнения всех команд, время работы которых равно или превышает указанное количество миллисекунд

- `pg_stat_statements.max (integer)`

Параметр `pg_stat_statements.max` задаёт максимальное число операторов, отслеживаемых модулем (то есть, максимальное число строк в представлении `pg_stat_statements`). Когда на обработку поступает больше,

чем заданное число различных операторов, информация о редко выполняемых операторах отбрасывается. Значение по умолчанию — 5000. Этот параметр можно задать только при запуске сервера.

## **8. Управление учетными записями пользователей**

Управление пользователями, группами и ролями осуществляется через раздел Настройки, подраздел Сотрудники. Доступ в административный раздел разрешен для пользователей с правом Администрирования.

## Схема данных

### Аварийные отключения

Отключенные подъезды model\_porch через

shutdown\_dead\_basements(shutdown\_id,porch\_id)

Таблица model\_shutdown

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- damage\_type\_id Отключенный ресурс(model\_service)
- shutdown\_type\_id Тип отключения(model\_listitem machine\_name = shutdown\_universal\_types)
- supplier\_id Поставщик ресурса(management\_companies)
- off\_time Время отключения
- on\_time\_plan Время включения (план)
- on\_time\_fact Время включения (факт)
- reason Причина
- person\_reported\_shutdown Человек, сообщивший об отключении
- person\_reported\_time\_on Человек, сообщивший о включении
- contractor\_id Производитель работ(management\_companies)
- operator\_id Оператор(tbl\_users)

### Гарантийные заявки

Таблица model\_garant\_appeal

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- description Содержание
- close\_comment Комментарий по закрытию
- flat\_id Помещение(model\_flat)
- building\_id Дом(model\_building)
- person\_id Обращающийся(model\_person)
- author\_id Автор(tbl\_users)
- phone Телефон
- qc\_type Реакция жильца(1=1, 2=2, 3=3, 4=4, 5=5)
- qc\_source Источник реакции(site=Сайт, app=Приложение, operator=Оператор, ats\_auto\_call=Автоматический обзвон)
- qc\_comment Комментарий жильца
- qc\_time Время контроля качества
- emergency Аварийная
- garant\_type\_id Тип(model\_listitem machine\_name = garant\_appeal\_types)
- garant\_doc Акт осмотра или заявление(act=Акт осмотра, petition=Заявление)
- garant\_flat\_access Доступ в квартиру(allowed=Разрешает, not\_allowed=Не разрешает, metal\_door=Металлическая дверь)
- garant\_act Акт приема-передачи(act=Акт, not=Не принята)

## Задачи

Задачи

СМС model\_sms через sms\_to\_task(task\_id, sms\_id)

Участники tbl\_users через tasks\_doers(task\_id, user\_id)

Таблица model\_task

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- id ID
- description Содержание
- completion\_comment Комментарий по закрытию
- flat\_id Помещение(model\_flat)
- building\_id Дом(model\_building)
- person\_id Обращающийся(model\_person)
- phone Телефон
- author\_id Автор(tbl\_users)
- doer\_id Исполнитель(tbl\_users)
- contractor\_id Исполнитель (контрагент)(management\_companies)
- time\_convenient Дата визита, удобная заказчику
- hour\_convenient\_from Время визита С
- hour\_convenient\_until Время визита По
- time\_accepted Время принятия задачи
- time\_begin Время начала работ
- time\_end\_guess Крайний срок исполнения

- time\_end\_fact Фактическое время выполнения
- complete Выполнено
- request\_contractor\_id Заявитель (контрагент)(management\_companies)
- was\_overdue Была просрочена

## Заявки

Заявки

СМС model\_sms через sms\_to\_appeals(appeal\_id, sms\_id)

Таблица model\_appeal

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- description Содержание
- type\_id Тип(mod\_houseacct\_enirs)
- close\_comment Комментарий по закрытию
- flat\_id Помещение(model\_flat)
- building\_id Дом(model\_building)
- person\_id Обращающийся(model\_person)
- author\_id Автор(tbl\_users)
- phone Телефон
- time\_convenient Дата визита, удобная заказчику
- hour\_convenient\_from Время визита С
- hour\_convenient\_until Время визита По
- qc\_type Реакция жильца(1=1, 2=2, 3=3, 4=4, 5=5)

- qc\_source Источник реакции(site=Сайт, app=Приложение, operator=Оператор, ats\_auto\_call=Автоматический обзвон)
- qc\_comment Комментарий жильца
- qc\_time Время контроля качества
- emergency Аварийная

### **Заявления граждан**

Заявления граждан

Таблица model\_requestperson

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- description Содержание заявления
- type\_id Тип(model\_listitem machine\_name = request\_person\_types)
- flat\_id Помещение(model\_flat)
- building\_id Дом(model\_building)
- person\_id Отправитель (физ. лицо)(model\_person)
- phone Телефон
- management\_company\_id Получатель(management\_companies)
- time\_end\_guess Крайний срок реакции на документ
- author\_id Автор(tbl\_users)

### **Исходящие гражданам**

Исходящие гражданам

Получатели копии management\_companies через  
corrdoc\_recipients\_answerperson(corr\_document\_id, organization\_id)

Таблица model\_answerperson

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- barcode Трек-номер
- barcode\_laststatus Статус отправления
- description Содержание
- author\_id Исполнитель письма(tbl\_users)
- building\_id Дом(model\_building)
- flat\_id Помещение(model\_flat)
- person\_id Получатель (физ. лицо)(model\_person)
- phone Телефон
- reply\_status\_id Статус(model\_listitem machine\_name = answer\_statuses)
- approver\_id Утвердил письмо(tbl\_users)
- management\_company\_id Отправитель(management\_companies)

### **Исходящие контрагентам**

Исходящие контрагентам

Дома model\_building через answer\_contractor\_buildings(answer\_contractor\_id,  
building\_id)

Получатели копии management\_companies через  
corrdoc\_recipients\_answercontractor(corr\_document\_id, organization\_id)



## Таблица model\_answercontractor

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- barcode Трек-номер
- barcode\_laststatus Статус отправления
- description Содержание
- author\_id Исполнитель письма(tbl\_users)
- contractor\_sender\_id Получатель контрагент(management\_companies)
- management\_company\_id Отправитель(management\_companies)
- reply\_status\_id Статус(model\_listitem machine\_name = answer\_statuses)
- approver\_id Утвердил письмо(tbl\_users)
- flat\_id Помещение(model\_flat)

### **Исходящие надзорным органам**

Исходящие надзорным органам

Дома model\_building через

answer\_supervisory\_buildings(answer\_supervisory\_id, building\_id)

Получатели копии management\_companies через

corrdoc\_recipients\_answersupervisory(corr\_document\_id, organization\_id)

Таблица model\_answersupervisory

- guid ID
- number №
- date Время создания

- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- barcode Трек-номер
- barcode\_laststatus Статус отправления
- description Содержание
- author\_id Исполнитель письма(tbl\_users)
- supervisory\_sender\_id Получатель Надзорный орган(management\_companies)
- management\_company\_id Отправитель(management\_companies)
- reply\_status\_id Статус(model\_listitem machine\_name = answer\_statuses)
- approver\_id Утвердил письмо(tbl\_users)

### **Коммерческие предложения**

Коммерческие предложения

Таблица model\_requestcommercialoffer

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- description Тема документа
- address Адрес
- sender Отправитель
- number\_external Исходящий номер отправителя

- date\_external Исходящая дата письма отправителя
- management\_company\_id Получатель(management\_companies)
- author\_id Автор(tbl\_users)

## Посылки

### Посылки

#### Таблица documents\_parcel

- guid ID
- number №
- date Время создания
- is\_deleted Удален
- status\_id Статус(model\_doc\_status)
- status\_timestamp Время изменения статуса
- priority Приоритет
- progress Прогресс
- type Тип обмена(0=import, 1=export)
- external\_system Внешняя система
- action Действие
- objects объекты обмена
- timestamp\_created Создан
- timestamp\_edited Изменен
- author\_id Автор(tbl\_users)
- tags Тэги