

1. Руководство администратора	2
1.1 Установка компонентов системы	3
1.1.1 Требования к конфигурации серверов для установки Системы	4
1.1.2 Используемые порты и протоколы	5
1.1.3 Порядок действий по установке серверной части Vitro-CAD (Linux)	6
1.1.4 Порядок действий по установке серверной части Vitro-CAD (Windows)	18
1.1.4.1 Настройка обратного прокси в IIS	37
1.1.4.1.1 Создание веб-сайта IIS для обратного прокси	41
1.1.5 Описание процесса обновления серверной части Vitro-CAD и модуля Vitro Workflow (Linux)	50
1.1.6 Описание процесса обновления серверной части Vitro-CAD и модуля Vitro Workflow (Windows)	54
1.1.7 Процесс установки и настройки фермы конвертации Vitro Forge	57
1.1.7.1 Требования к ферме конвертации моделей для Vitro Bim Viewer.	58
1.1.7.2 Установка и запуск фермы конвертации моделей для Vitro Bim Viewer.	60
1.1.7.3 Установка RabbitMQ на Windows	63
1.1.7.4 Установка Vitro.Forge.Consumer на Linux	67
1.1.8 Установка и настройка клиентской части	72
1.1.9 Удаленная установка клиентской части	77
1.2 Настройка объектов хранения данных	79
1.2.1 Пространства	80
1.2.1.1 Пространство - создание	81
1.2.1.2 Сайдбар - настройка	82
1.2.2 Списки - создание и настройка	83
1.2.3 Типы элементов - создание и настройка	85
1.2.4 Атрибуты - создание и настройка	86
1.2.5 Сервис Вычисления	88
1.2.5.1 Вычисляемые поля - создание и настройка	89
1.2.5.2 Счетчики - создание и настройка	91
1.2.5.3 Синхронизация атрибутов - настройка	93
1.2.5.4 Функции для формул вычисления	95
1.2.6 Пример описания настройки структуры хранения документов	100
1.3 Пользователи и доступ	101
1.3.1 Пользователи и их группировка	102
1.3.2 Настройка прав доступа	105
1.3.3 Настройка протокола Kerberos для использования доменной аутентификации в Linux	108
1.3.4 Синхронизация списка пользователей с LDAP-каталогом (AD)	112
1.3.5 Включение LDAP аутентификации на портале (кнопка "Вход через Windows")	115
1.3.6 Настройка протокола Kerberos для использования доменной аутентификации в Windows развертывании	117
1.4 REST API	120
1.4.1 Описание - версия 24.1.37	121
1.4.2 Описание - версия 24.1.23	170
1.4.3 Пример работы с API на Python	174
1.4.4 Настройка системы с использованием командной строки (CLI)	175
1.4.5 Создание действия с вызовом REST сервиса	180
1.4.6 Вызов REST сервиса из обработчика действия	183
1.4.7 Поиск элементов по структуре - версия 25.1.39.16	185
1.5 Резервное копирование и восстановление данных	188
1.5.1 Резервное копирование данных	189
1.5.2 Восстановление баз данных	192
1.6 Прочие настройки	193
1.6.1 Настройка расписания в ОС Linux на примере задания на запуск утилиты синхронизации ОШС с AD/LDAP	194
1.6.2 Настройка функционала проверки QR-кодов	196
1.6.3 Настройки безопасности учетных записей	198
1.7 Отчеты	199
1.7.1 Размещение шаблонов отчетов в систему	200
1.8 Автоматизация	205
1.8.1 Вид Канбан для настройки правил автоматизации	207
1.8.1.1 Типы автоматических действий	209
1.8.2 Настройка процесса согласования с использованием сервиса Автоматизации	210
1.9 Валидаторы	216
1.10 Рекомендации по обеспечению информационной безопасности	218
1.10.1 Журналы регистрации событий в системе	219
1.10.2 Список выявленных уязвимостей	221

Руководство администратора

- Установка компонентов системы
- Настройка объектов хранения данных
- Пользователи и доступ
- Процессы
- REST API
- Описание таблиц БД
- Резервное копирование и восстановление данных
- Прочие настройки
- Отчеты
- Автоматизация
- Валидаторы
- Ошибки и вопросы
- Рекомендации по обеспечению информационной безопасности

Установка компонентов системы

Используемые при установке компонентов системы порты и протоколы

№ п/п	Сервис	Протокол	Порт
1	postgres	tcp	5432
2	Vitro.Server.Core.Web	tcp	4615
3	nginx	http	80
4	nginx	https	443
5	vitro-workflow	tcp	4616

- Требования к конфигурации серверов для установки Системы
- Используемые порты и протоколы
- Порядок действий по установке серверной части Vitro-CAD (Linux)
- Порядок действий по установке серверной части Vitro-CAD (Windows)
- Описание процесса обновления серверной части Vitro-CAD и модуля Vitro Workflow (Linux)
- Описание процесса обновления серверной части Vitro-CAD и модуля Vitro Workflow (Windows)
- Процесс установки и настройки фермы конвертации Vitro Forge
- Установка и настройка клиентской части
- Удаленная установка клиентской части
- Порядок действий по установке модуля Vitro Workflow
- Порядок действий по установке серверной части Vitro-CAD (Linux) (до 25.1.39.19)
- Описание процесса обновления серверной части Vitro-CAD и модуля Vitro Workflow (Linux) (до 25.1.39.19)

Требования к конфигурации серверов для установки Системы

1. Требования к производственной среде при работе в системе менее 50 пользователей

[Vitro_CAD_Инфраструктура_продуктива_Требования для старта менее 50 пользователей_к_VM.xlsx](#)

2. Требования к производственной среде при работе в системе до 200 пользователей (конфигурация при большем числе пользователей проговаривается отдельно)

[Vitro_CAD_Инфраструктура_продуктива_Требования для 100-200 пользователей_к_VM.xlsx](#)

3. Требования к тестовой среде

[Vitro-CAD_Инфраструктура_тестовой_среды_Требования_к_VM.xlsx](#)

Используемые порты и протоколы

№п/п	Сервис	Протокол	Порт
1	PostgreSQL	tcp	5432
2	nginx	http	80
3	nginx	https	443
4	vitro-workflow	tcp	4616

Порядок действий по установке серверной части Vitro-CAD (Linux)

Сервер БД

1. Установка необходимых пакетов ПО из репозитория:

```
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install unzip curl postgresql postgresql-client postgresql-contrib
```

2. Отредактировать конфигурационные файлы postgresql:

Проверяем версию postgresql

```
psql --version
```

Открываем конфигурационный файл postgresql.conf с учетом установленной версии

```
nano /etc/postgresql/14/main/postgresql.conf
```

Изменить в секции # - Connection Settings - значение параметра listen_addresses с localhost на * должно получиться:

```
listen_addresses = '*'
```

Открываем конфигурационный файл pg_hba.conf с учетом установленной версии

```
nano /etc/postgresql/14/main/pg_hba.conf
```

Изменить в секции # IPv4 Local connections: значение параметра ADDRESS с 127.0.0.1/32 на 0.0.0.0/0

```
# IPv4 local connections:
host      all       all       0.0.0.0/0      scram-sha-256
```

3. Перезапустить службу:

```
sudo systemctl restart postgresql
```

4. Проверить выполняется ли прослушивание на порту 5432 для всех активных адресов:

```
netstat -ltnp
tcp      0      0 0.0.0.0:5432      0.0.0.0:*        LISTEN   14569/postgres
```

5. Добавить пользователя postgres в группу владельца директории, из которой выполняются действия:

Проверка владельца текущей папки (обозначена ".")

```
ls -la
```

Добавление пользователя postgres в группу владельца

```
sudo usermod -aG dir_owner_group postgres
```

6. Изменить права группе владельца директории (дать права на изменение). В данном случае путь до директории - . (текущая директория):

```
chmod -R 775 .
```

7. Войти в cli postgresql от имени пользователя postgres:

```
sudo -u postgres psql
```

8. Создать пользователя с ролью суперпользователя:

```
postgres-#CREATE USER vitrodbuser WITH PASSWORD '@pwd4vitro!' SUPERUSER;
```

9. Создать новую базу данных с именем vitrodb:

```
postgres-#CREATE DATABASE vitrodb with owner = vitrodbuser encoding = 'UTF8' connection limit = -1;
```

10. Создать новую базу данных с именем workflow:

```
postgres-#CREATE DATABASE workflow with owner = vitrodbuser encoding = 'UTF8' connection limit = -1;
```

11. Отключиться от cli postgresql:

```
postgres-# \q
```

12. Скачать и распаковать дистрибутив:

```
curl "https://linkTo/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx.zip" --output 'Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx.zip'
```

```
sudo unzip Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx.zip
```

13. Инициализация базы данных

- a. Версия дистрибутива 24.1.38 или выше:

Скопировать в доступное для пользователей расположение (к примеру tmp) файлы инициализации базы данных:

```
sudo cp ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx/DB/Install/db_func_create.sql /tmp
sudo cp ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx/DB/Install/db_table_create.sql /tmp
sudo cp ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx/DB/Install/db_data_init.sql /tmp
```

Запустить инициализацию базы из скриптов:

```
sudo -u postgres psql -U postgres -d vitrodb -f /tmp/db_func_create.sql -f /tmp/db_table_create.sql -f /tmp/db_data_init.sql
```

- b. Версия дистрибутива ниже 24.1.38:

Скопировать в доступное для пользователей расположение (к примеру tmp) файл резервной копии базы данных:

```
sudo cp ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx/DB/vitro.bak /tmp
```

Запустить восстановление БД из бэкапа:

```
sudo -u postgres pg_restore -v -U vitrodbuser -h localhost -d vitrodb /tmp/vitro.bak
```

Запустить восстановление БД из бекапа (для файлов с расширением sql):

```
sudo -u postgres psql -U vitrodbuser -h localhost -d vitrodb < /tmp/vitro.bak.sql
```

Сервер приложений/фронт энд:

1. Установка необходимых пакетов ПО из репозитория:

```
sudo apt-get update && sudo apt-get upgrade  
sudo apt-get install unzip curl nginx nginx-extras libgdiplus
```

2. Скачать и распаковать дистрибутив:

```
curl "https://linkTo/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx.zip" --output 'Vitro.Server.  
MultiPlatform.Linux-xx.x.xx.xx.zip'  
  
sudo unzip Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx.zip
```

3. Скопировать папку файлового хранилища в расположение, в которое смонтирован раздел для хранения:

```
sudo cp -r ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx/DB/VitroFileStorage /mnt/storage
```

4. Сменить владельца для папки файлового хранилища:

```
sudo chown -R www-data:www-data /mnt/storage/VitroFileStorage
```

5. Скопировать папку Vitro в расположение /etc/:

```
sudo cp -r ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx/Vitro/ /etc/
```

6. Настроить уровень доступа на папке:

```
sudo chmod 500 /etc/Vitro/
```

7. Внести изменения в конфигурационные файлы:

- a. Файл db.json:

```
sudo nano /etc/Vitro/db.json
```

Изменить строку соединения, указав в качестве значения для атрибута "server" имя или IP сервера БД, а в качестве значения для атрибута "database" название созданной в п.9 базы данных:

```
"ConnectionString": "server=vitrodbserver;database=vitrodb;user id=vitrodbuser;  
password=@pwd4vitro!",
```

Изменить параметр "FileStoragePath", указав корректный путь к папке файлового хранилища:

```
"FileStoragePath": "/mnt/storage/VitroFileStorage",
```

изменить параметр "TempPath", указав корректный путь к папке временных файлов:

```
"TempPath": "/tmp",
```

b. Файл workflow.json:

```
sudo nano /etc/Vitro/workflow.json
```

Изменить строку соединения, указав в качестве значения для атрибута "server" имя или IP сервера БД, а в качестве значения для атрибута "database" название созданной в п.10 базы данных:

```
"ConnectionString": "server=vitrodbserver;database=workflow;user id=vitrodbuser;  
password=@pwd4vitro!;Include Error Detail=true",
```

Изменить параметр "BaseUrl", указав корректный Url сайта:

```
"BaseUrl": "http://www.example.com",
```

Изменить параметр "PortalUrl", указав корректный Url сайта:

```
"PortalUrl": "http://www.example.com",
```

Изменить параметр "BasePath", указав корректный относительный путь для Url:

```
"BasePath": "/workflow/manager",
```

Указать логин/пароль учетной записи администратора:

```
"Login": "admin",  
"Password": "admin"
```

c. Файл mail.json:

Указать адрес и порт SMTP сервера:

```
"Host": "smtp.example.ru",  
"Port": 587,
```

Указать данные УЗ для подключения к SMTP серверу:

```
"Login": "mail@example.ru",  
"Password": "123456",
```

Указать адрес отправителя:

```
"FromAddress": "mail@example.ru",  
"FromName": "mail@example.ru",
```

Указать используется ли SSL для подключения к SMTP серверу:

```
"EnableSsl": true,
```

Убедиться, что в файле есть параметр SecureSocketOptions:

```
"SecureSocketOptions": "StartTls"
```

Остальные параметры оставить по умолчанию.

8. Скопировать папку приложения Vitro.Server.Core.Web в желаемое расположение:

```
sudo cp -r ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx/Vitro.Server.Core.Web /usr/local
```

9. Сменить владельца для папки приложения:

```
sudo chown -R www-data:www-data /usr/local/Vitro.Server.Core.Web
```

10. Установить атрибут "Исполняемый" для файла Vitro.Server.Core.Web:

```
sudo chmod +x /usr/local/Vitro.Server.Core.Web/Vitro.Server.Core.Web
```

11. Создать и отредактировать файл модуля systemd для запуска службы:

```
sudo nano /lib/systemd/system/vitro-server.service
```

Содержимое файла модуля:

```
[Unit]
Description=Vitro .NET Web Application

[Service]
WorkingDirectory=/usr/local/Vitro.Server.Core.Web
ExecStart=/usr/local/Vitro.Server.Core.Web/Vitro.Server.Core.Web --urls http://localhost:4615
Restart=always
# Restart service after 10 seconds if the dotnet service crashes:
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=dotnet-example
User=www-data
SyslogIdentifier=Vitro-server

[Install]
WantedBy=multi-user.target
```

12. Применить изменения и запустить службу:

```
sudo systemctl enable vitro-server

sudo systemctl start vitro-server
```

13. Проверить выполняется ли прослушивание на порту, заданном в файле модуля (здесь 4615) для адреса Lo интерфейса:

```
netstat -ltnp

Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
...
tcp        0      0 0.0.0.0:*                  0.0.0.0:*               LISTEN      -
...
```

14. Скопировать папку приложения в желаемое расположение:

```
sudo cp -r ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx.xx/Vitro.Server.Workflow.Manager.Web /usr/local
```

15. Сменить владельца для папки приложения:

```
sudo chown -R www-data:www-data /usr/local/Vitro.Server.Workflow.Manager.Web
```

16. Установить атрибут "Исполняемый" для файла Workflow.Manager.Web:

```
sudo chmod +x /usr/local/Vitro.Server.Workflow.Manager.Web/Vitro.Server.Workflow.Manager.Web
```

17. Создать и отредактировать файл модуля systemd для запуска службы:

```
sudo nano /lib/systemd/system/vitro-workflow.service
```

Содержимое файла модуля:

```
[Unit]
Description=Vitro Workflow Manager .NET Web Application

[Service]
WorkingDirectory=/usr/local/Vitro.Server.Workflow.Manager.Web
ExecStart=/usr/local/Vitro.Server.Workflow.Manager.Web/Vitro.Server.Workflow.Manager.Web --urls
http://localhost:4616
Restart=always
# Restart service after 10 seconds if the dotnet service crashes:
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=dotnet-example
User=www-data
SyslogIdentifier=Vitro-workflow

[Install]
WantedBy=multi-user.target
```

18. Применить изменения и запустить службу:

```
sudo systemctl enable vitro-workflow

sudo systemctl start vitro-workflow
```

19. Проверить выполняется ли прослушивание на порту, заданном в файле модуля (здесь 4616) для адреса Lo интерфейса:

```
netstat -ltnp

Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
...
tcp        0      0 0.0.0.0:4616             0.0.0.0:*                LISTEN      -
...
```

Конвертация комплекта сертификата '*.pfx' в crt/pem средствами OpenSSL

1. Скачать сертификат в формате *.pfx

```
curl "https://linkTo/example.ru.pfx" --output 'example.ru.pfx'
```

2. Для получения из *.pfx сертификата сервера в формате pem выполнить:

```
openssl pkcs12 -in example.ru.pfx -clcerts -nokeys -out example.ru.pem
```

В процессе будет запрошен пароль.

3. для получения из *.pfx приватного ключа в формате pem выполнить:

```
openssl pkcs12 -in example.ru.pfx -nocerts -out example.ru.enc.key
```

В процессе будет запрошен пароль. Далее будет запрошен новый пароль для нового файла ключа (в этом случае в качестве пароля вводится НЕ пустое значение, любой пароль).

4. Для получения незащищенного паролем приватного ключа (из защищенного паролем файла ключа полученного на предыдущем шаге) выполнить:

```
openssl rsa -in example.ru.enc.key -out example.ru.key
```

5. Скопировать полученные файлы сертификата и незащищенного паролем приватного ключа в /etc/ssl/private (либо иное удобное расположение)

```
sudo cp example.ru.* /etc/ssl/private
```

Настройка обратного проксирования

1. Настроить обратное проксирование при помощи nginx, отредактировав конфигурационный файл:

```
sudo nano /etc/nginx/sites-available/default
```

Должно получиться:

```

# Default server configuration
#

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # Only allow access if the host is correct
    if ( $host != "vitro.example.ru" ){
        return 444; #CONNECTION CLOSED WITHOUT RESPONSE
    }

    #permanent redirect from http to https
    return 301 https://$host$request_uri;
}

server {
    # SSL configuration
    #
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    server_name vitro.example.ru;

    ssl_certificate      /etc/ssl/private/example.ru.pem;
    ssl_certificate_key  /etc/ssl/private/example.ru.key;

    # Only allow access if the host is correct
    if ( $host != "vitro.example.ru" ){
        return 444; #CONNECTION CLOSED WITHOUT RESPONSE
    }

    proxy_buffer_size    1M;
    proxy_buffers        4 1M;
    proxy_read_timeout   1h;
    proxy_send_timeout   1h;

    location / {
        #proxying traffic to upstream application (Vitro.Server.Core.Web)
        proxy_pass http://127.0.0.1:4615;
    }

    location /workflow/manager {
        proxy_pass http://localhost:4616/workflow/manager;
    }
}

```

Необходимые настройки:

- a. Для обеспечения загрузки больших файлов добавить в конфигурационный файл сайта Nginx в явном виде значение параметра `client_max_body_size`

```
client_max_body_size 100000M;
```

- b. Для исключения переполнения корневого раздела диска, необходимо включить в Nginx кеширование для обратного проксирования. Ключи для помещения в секцию `server` (или `location`):

```
proxy_buffering    on;
```

2. Также для исключения переполнения корневого раздела диска, необходимо включить в Nginx кеширование для обратного проксирования в файле `/etc/nginx/nginx.conf`

```
sudo nano /etc/nginx/nginx.conf
```

Ключ для размещения в конфигурационном файле:

```
proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=STATIC:10m inactive=24h max_size=1g;
```

3. Итоговые конфигурационные файлы:

/etc/nginx/nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
events {
    worker_connections 768;
    # multi_accept on;
}
http {
    ##
    # Basic Settings
    ##
    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;
    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    ##
    # SSL Settings
    ##
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;
    ##
    # Logging Settings
    ##
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
    ##
    # Gzip Settings
    ##
    gzip on;

    proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=STATIC:10m inactive=24h max_size=1g;

    # gzip_vary on;
    # gzip_proxied any;
    # gzip_comp_level 6;
    # gzip_buffers 16 8k;
    # gzip_http_version 1.1;
    # gzip_types text/plain text/css application/json application/javascript text/xml application/xml
application/xml+rss text/javascript;
    ##
    # Virtual Host Configs
    ##
    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}
```

```
/etc/nginx/sites-available/default
```

```
# Default server configuration
#

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # Only allow access if the host is correct
    if ( $host != "vitro.example.ru" ){
        return 444; #CONNECTION CLOSED WITHOUT RESPONSE
    }

    #permanent redirect from http to https
    return 301 https://$host$request_uri;
}

server {
    # SSL configuration
    #
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    server_name vitro.example.ru;

    client_max_body_size 100000M;

    ssl_certificate      /etc/ssl/private/example.ru.pem;
    ssl_certificate_key  /etc/ssl/private/example.ru.key;

    # Only allow access if the host is correct
    if ( $host != "vitro.example.ru" ){
        return 444; #CONNECTION CLOSED WITHOUT RESPONSE
    }

    proxy_buffer_size    1M;
    proxy_buffers        4 1M;
    proxy_read_timeout   1h;
    proxy_send_timeout   1h;

    location / {
        #proxying traffic to upstream application (Vitro.Server.Core.Web)
        proxy_pass http://127.0.0.1:4615;
        proxy_buffering    on;
    }

    location /workflow/manager {
        proxy_pass http://localhost:4616/workflow/manager;
    }
}
```

4. Выполнить перезагрузку службы nginx:

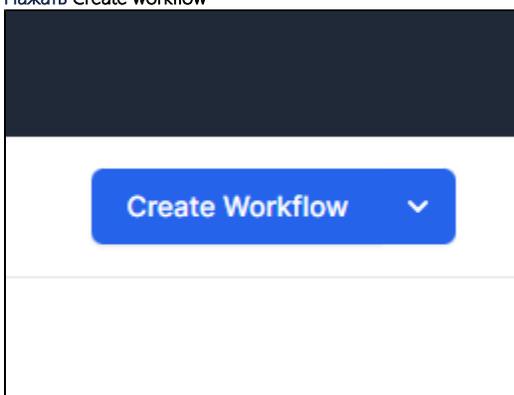
```
sudo systemctl reload nginx
```

Добавление шаблонов рабочих процессов

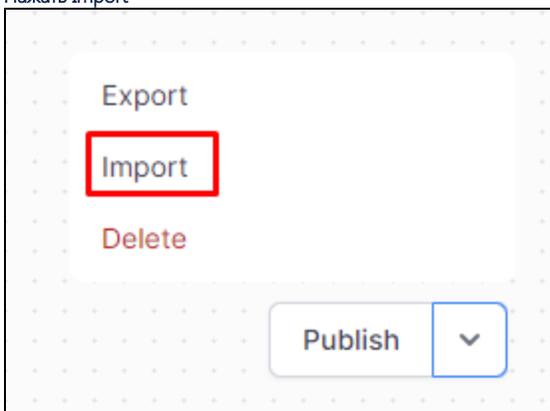
1. Перейти на вкладку Workflow Defenitions в Elsa

```
http://vitro.example.ru/workflow/manager/workflow-definitions
```

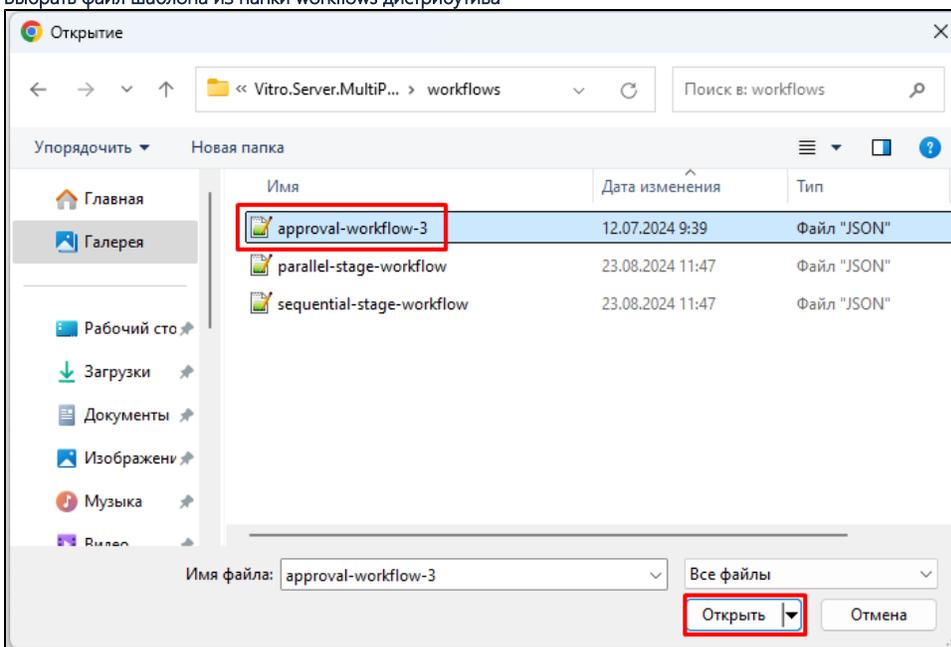
2. Нажать Create workflow



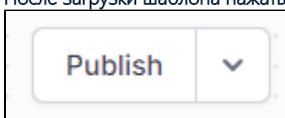
3. Нажать Import



4. Выбрать файл шаблона из папки workflows дистрибутива



5. После загрузки шаблона нажать Publish



6. Перейти обратно на вкладку Workflow definitions и убедиться, что шаблон создан

7. Таким образом добавить все шаблоны из папки workflows дистрибутива

Настройка системы

Необходимо указать адрес портала в ключе `Site.PortalUrl`.

Для этого нужно:

1. Перейти в список Конфигурация пространства Центр Администрирования.
2. Открыть папку Общие настройки
3. Для ключа Адрес портала указать в поле Значение (строка) адрес портала в виде <https://example.com>

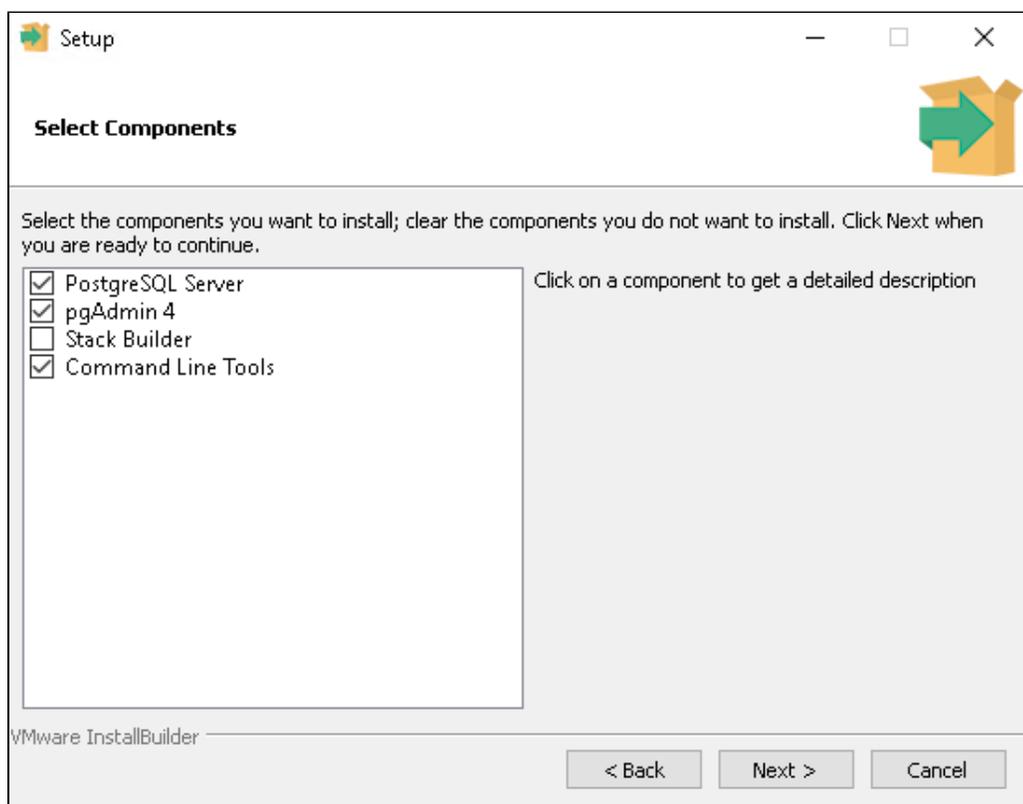
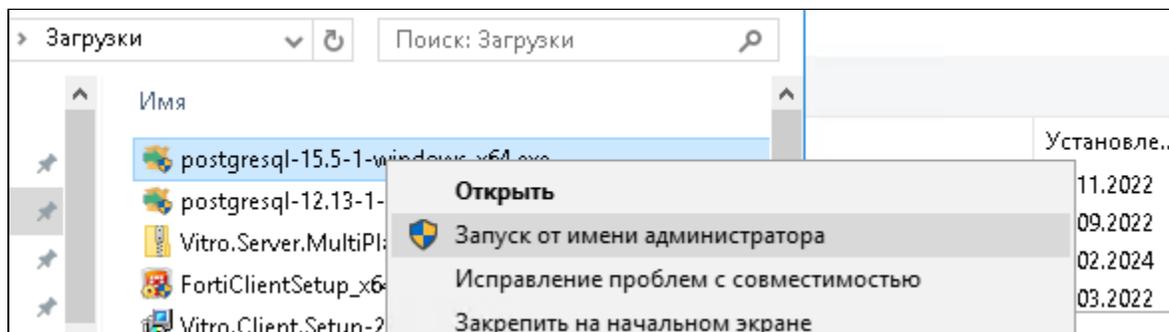


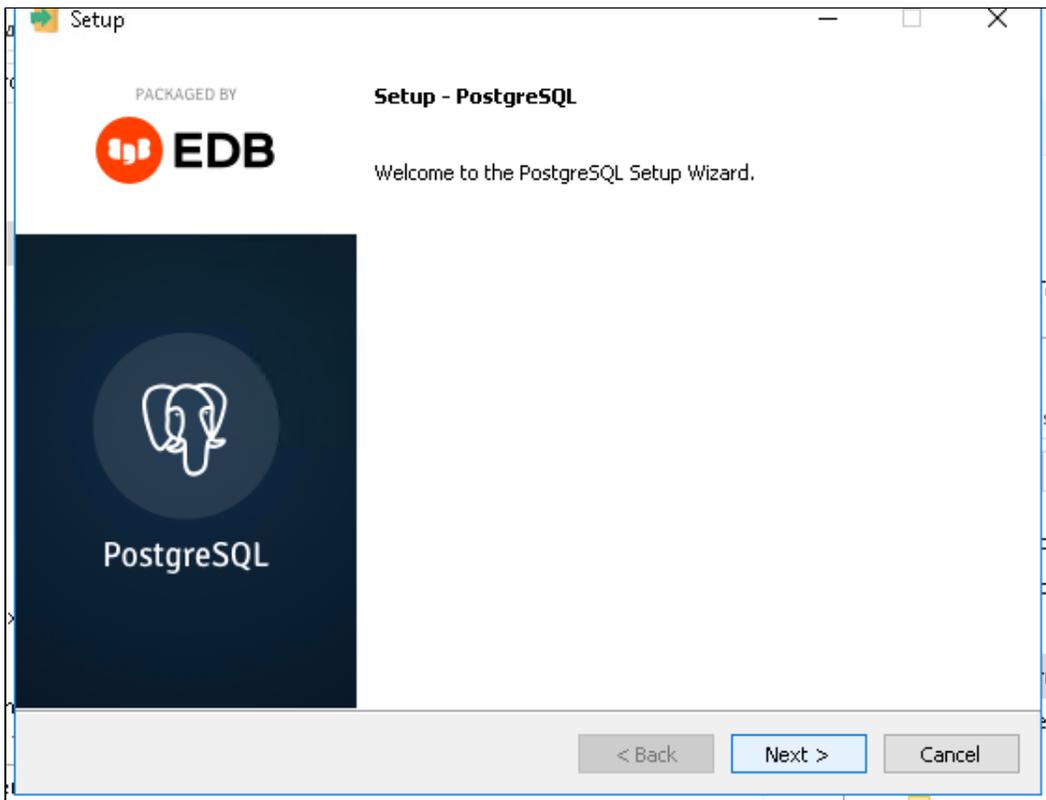
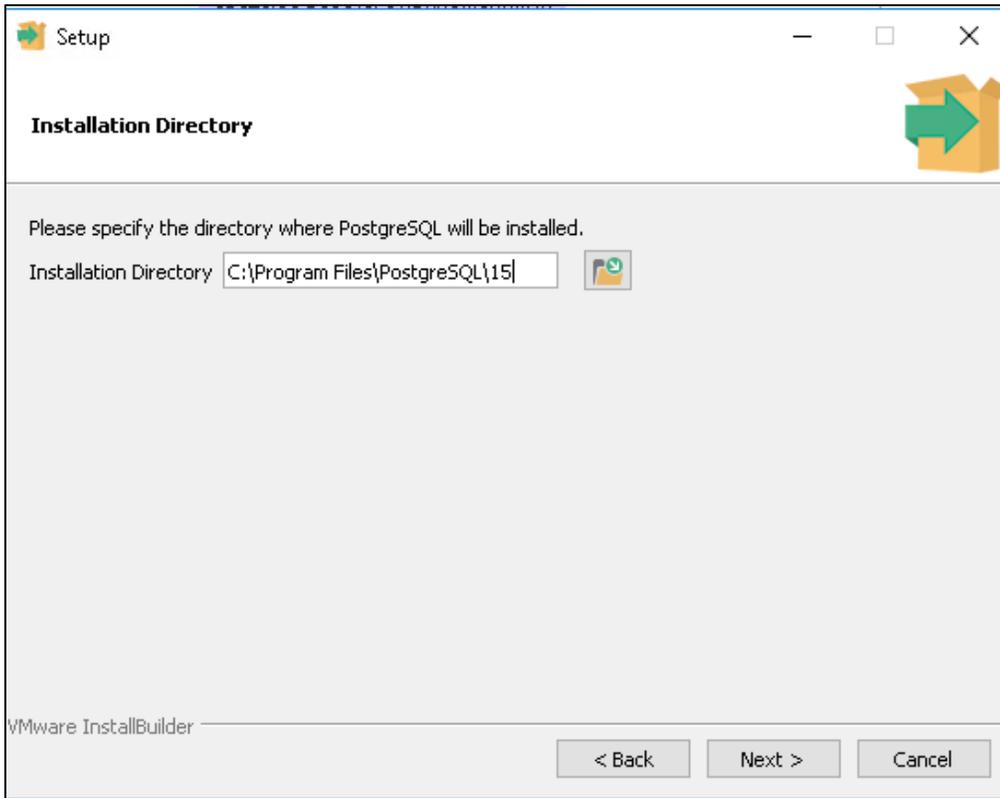
Важно!

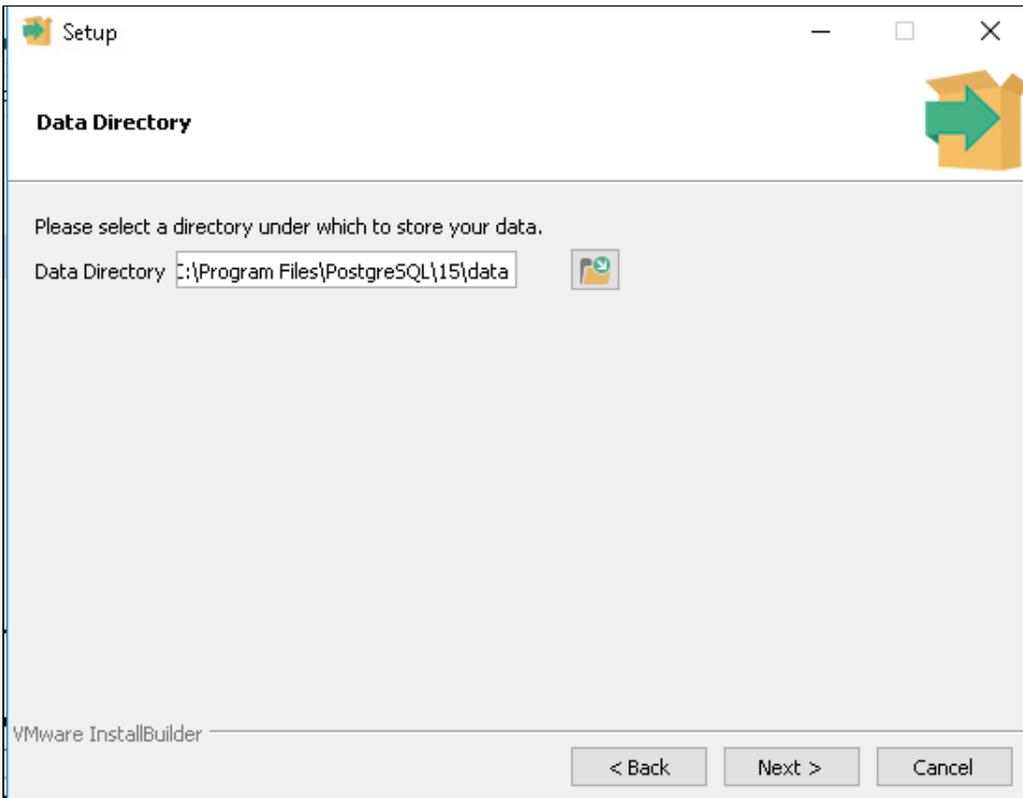
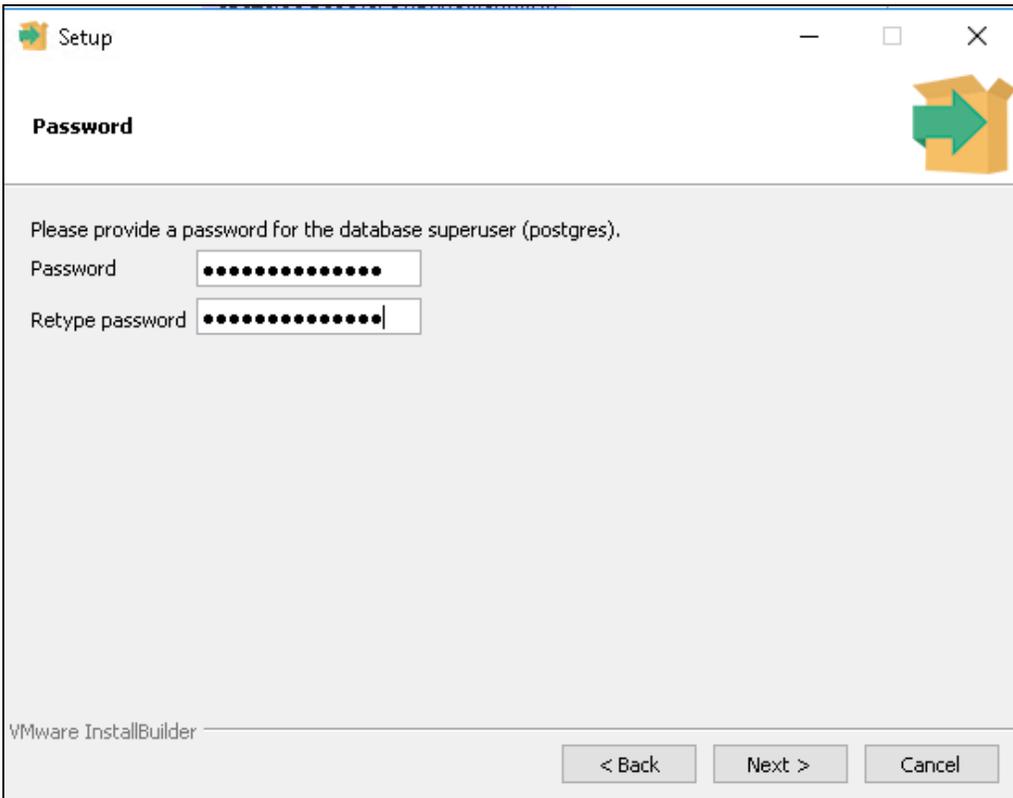
Адрес портала не должен заканчиваться символом слэша.

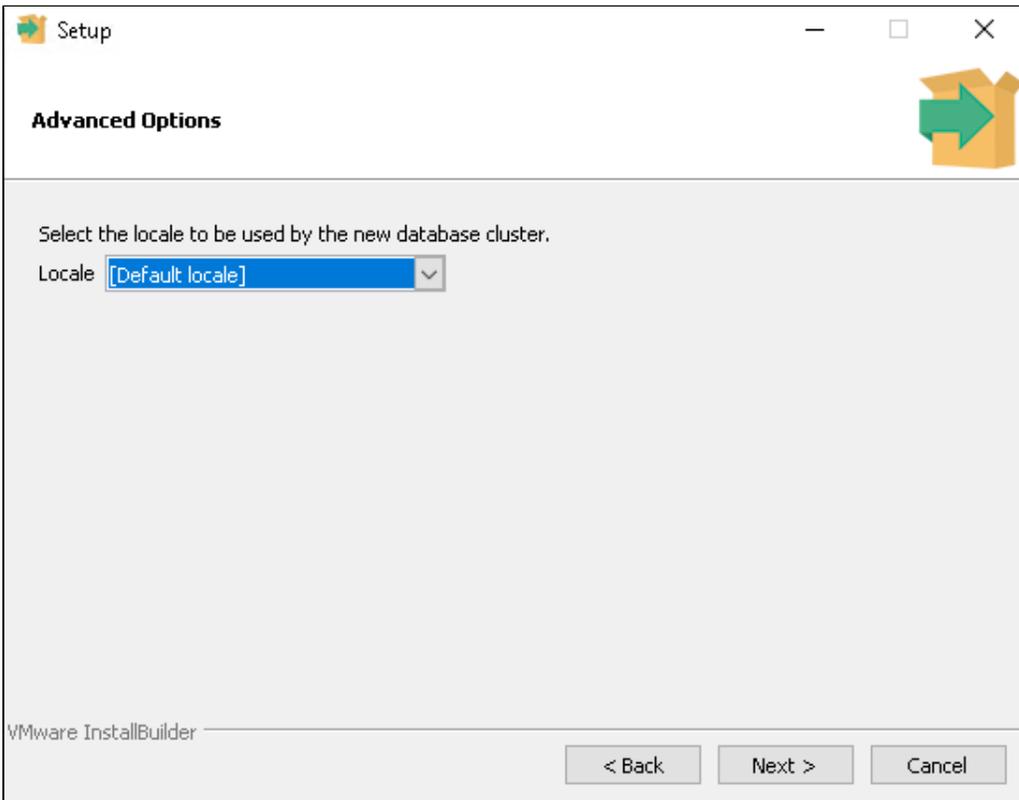
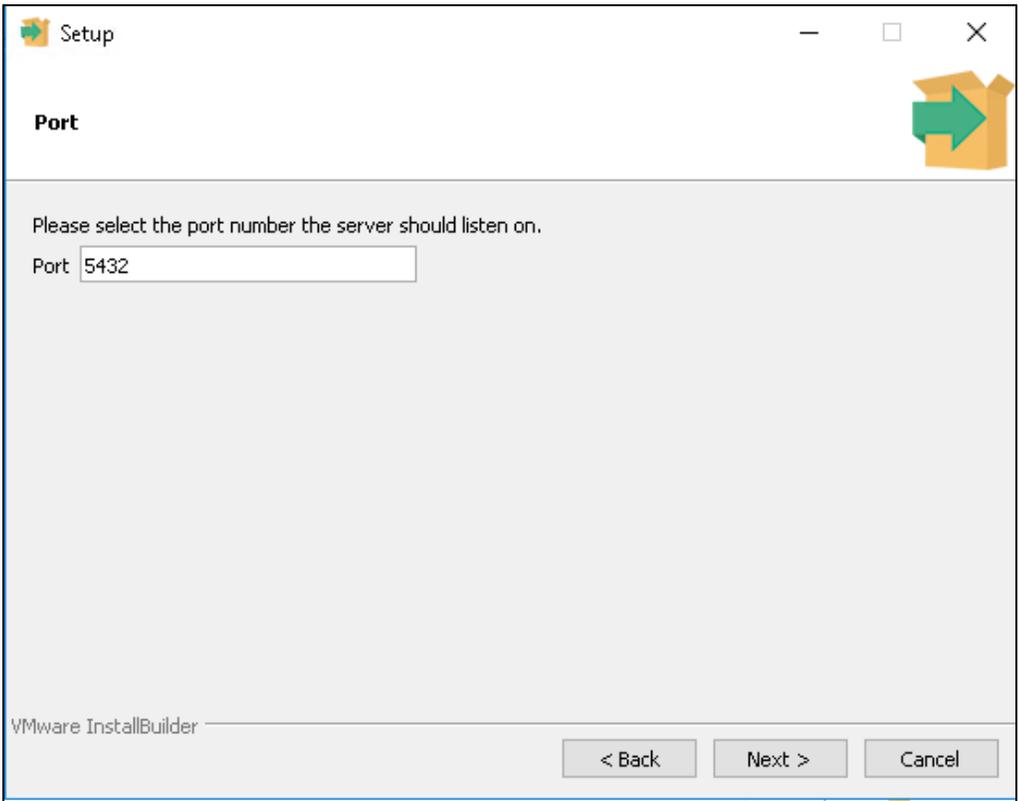
Порядок действий по установке серверной части Vitro-CAD (Windows)

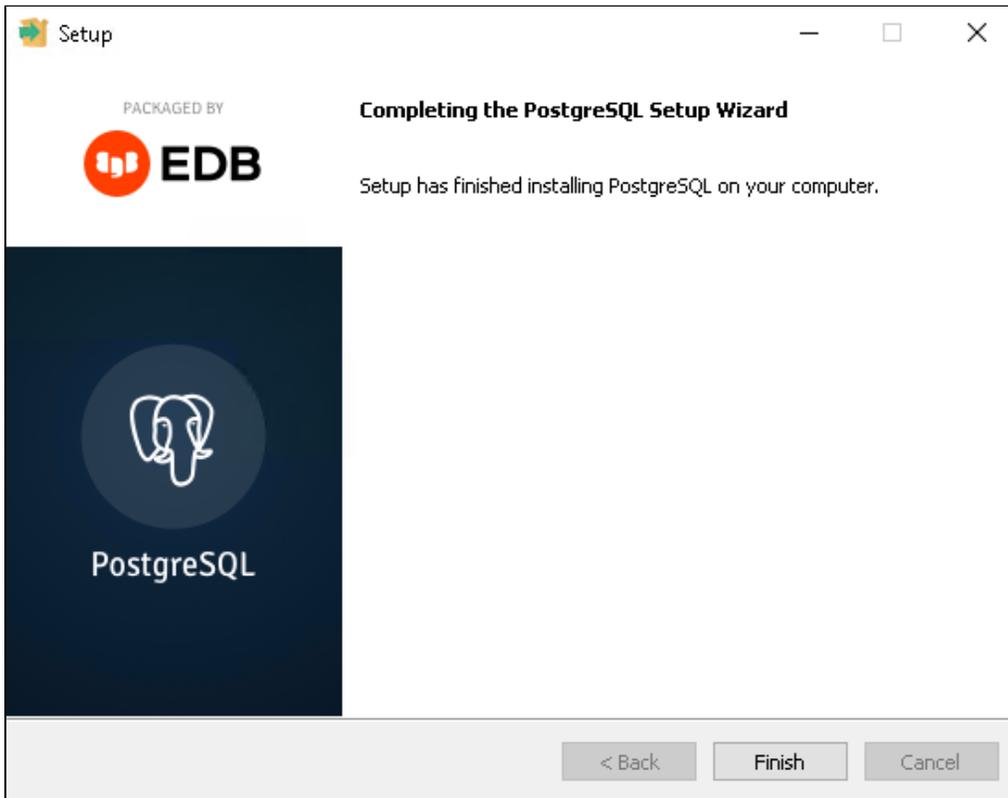
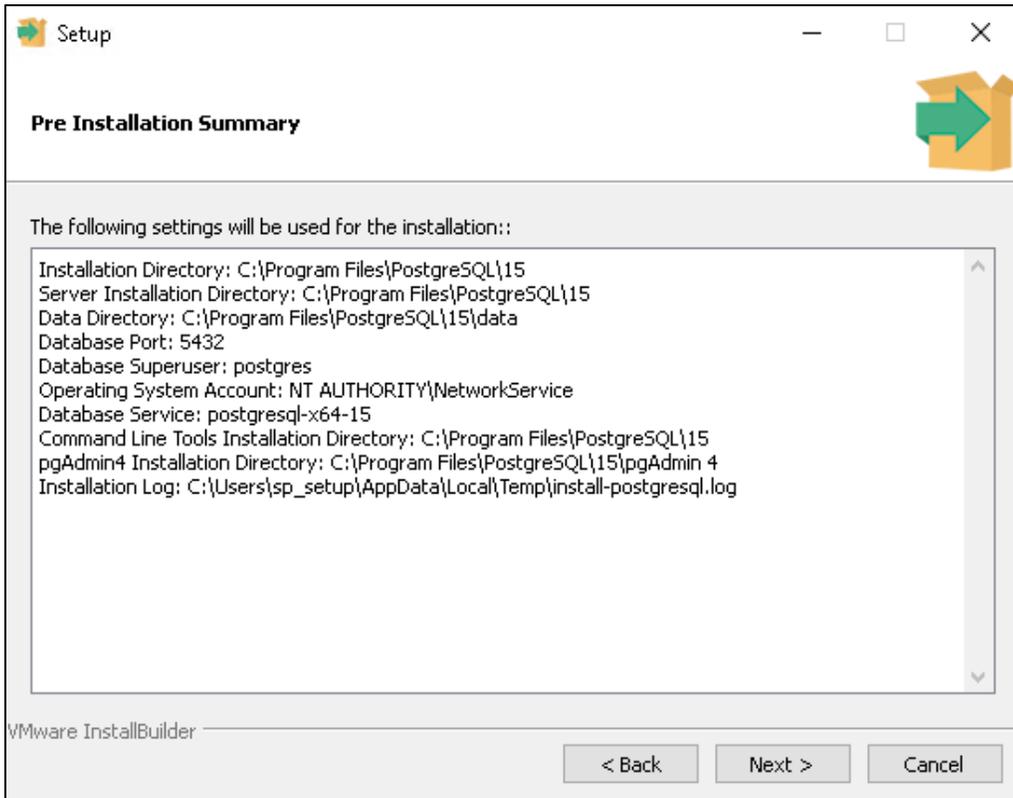
1. Установка PostgreSQL инсталлятором











2. Проверить конфигурационный файл C:\Program Files\PostgreSQL\15\data\pg_hba.conf:

```
67 # This file is read on server startup and when the server receives a
68 # SIGHUP signal.  If you edit the file on a running system, you have to
69 # SIGHUP the server for the changes to take effect, run "pg_ctl reload",
70 # or execute "SELECT pg_reload_conf()".
71 #
72 # Put your actual configuration here
73 # -----
74 #
75 # If you want to allow non-local connections, you need to add more
76 # "host" records.  In that case you will also need to make PostgreSQL
77 # listen on a non-local interface via the listen_addresses
78 # configuration parameter, or via the -i or -h command line switches.
79
80
81
82 # TYPE      DATABASE     USER        ADDRESS            METHOD
83
84 # "local" is for Unix domain socket connections only
85 local      all         all         scram-sha-256
86 # IPv4 local connections:
87 host       all         all         127.0.0.1/32     scram-sha-256
88 # IPv6 local connections:
89 host       all         all         ::1/128          scram-sha-256
90 # Allow replication connections from localhost, by a user with the
91 # replication privilege.
92 local      replication all         scram-sha-256
93 host       replication all         127.0.0.1/32     scram-sha-256
94 host       replication all         ::1/128          scram-sha-256
95
```

Раздел "#IPv4 local connections" должен содержать запись:

```
host      all      all      127.0.0.1/32      scram-sha-256
```

При разнесенной (выделенный сервер SQL) установке, или необходимости обеспечить внешний доступ к БД, следует добавить раздел "IPv4 network connections":

```

78 # configuration parameter, or via the -i or -h command line switches.
79
80
81
82 # TYPE DATABASE USER ADDRESS METHOD
83
84 # "local" is for Unix domain socket connections only
85 local all all scram-sha-256
86 # IPv4 local connections:
87 host all all 127.0.0.1/32 scram-sha-256
88 # IPv4 network connections:
89 host all all 0.0.0.0/0 scram-sha-256
90 # IPv6 local connections:
91 host all all ::1/128 scram-sha-256
92 # Allow replication connections from localhost, by a user with the
93 # replication privilege.
94 local replication all scram-sha-256
95 host replication all 127.0.0.1/32 scram-sha-256
96 host replication all ::1/128 scram-sha-256
97

```

и строку:

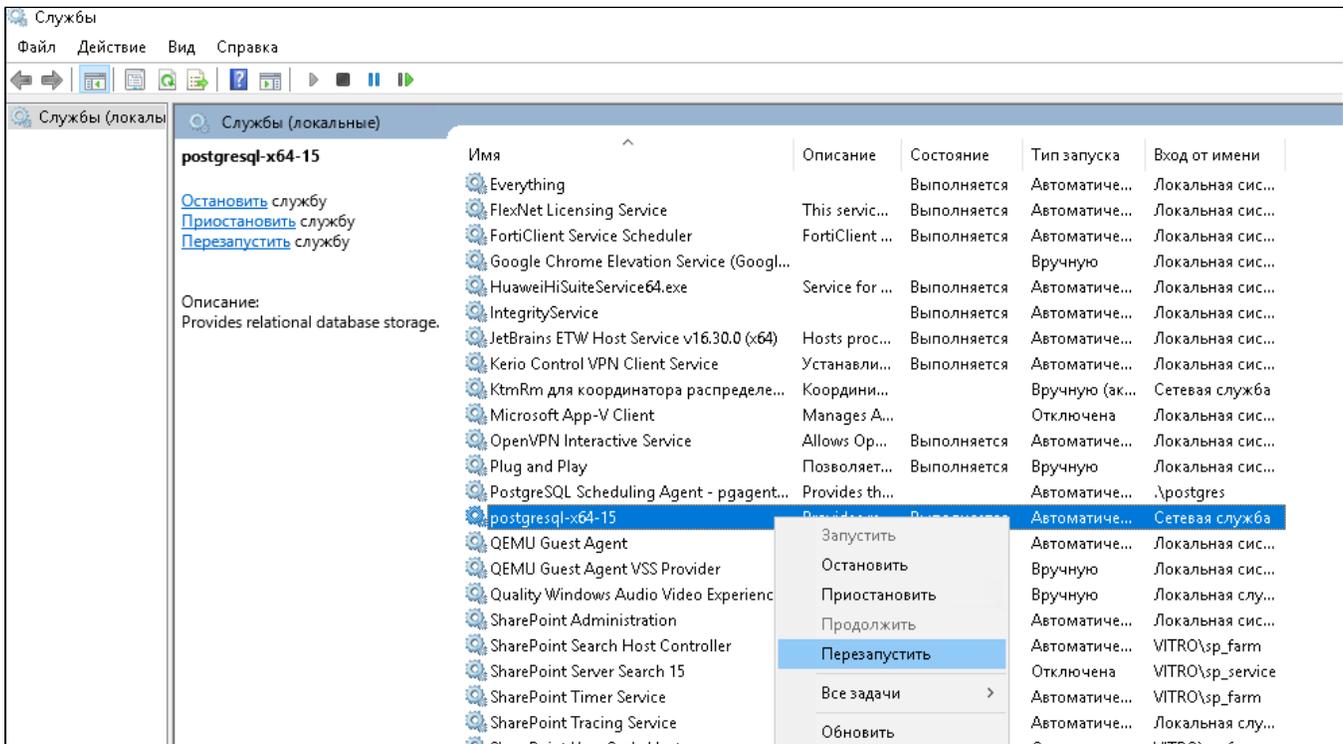
```

# IPv4 local connections:

host all all 0.0.0.0/0 scram-sha-256

```

3. Перезапустить службу:



4. Проверить выполняется ли прослушивание на порту 5432 для всех активных адресов:

```
C:\>netstat -a

...
TCP    0.0.0.0:5432          Vitrow2k16:0      LISTENING
```

5. Войти в cli postgresql от имени пользователя postgres:

```
c:\>cd "C:\Program Files\PostgreSQL\15\bin"
C:\Program Files\PostgreSQL\15\bin>psql -U postgres -h 127.0.0.1
postgres: < "postgres", Postgresql>
```

6. Создать пользователя с ролью суперпользователя:

```
postgres-#CREATE USER vitrodbuser WITH PASSWORD '@pwd4vitro!' SUPERUSER;
```

7. Создать новую базу данных с именем vitrodb:

```
postgres-#CREATE DATABASE vitrodb with owner = vitrodbuser encoding = 'UTF8' connection limit = -1;
```

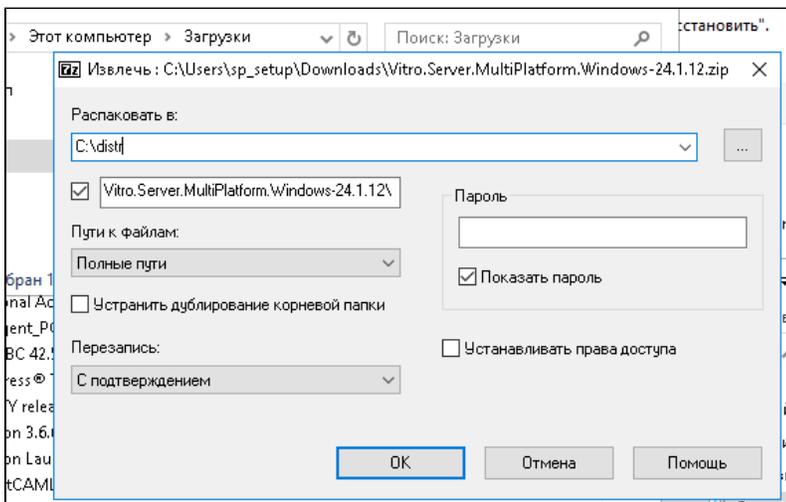
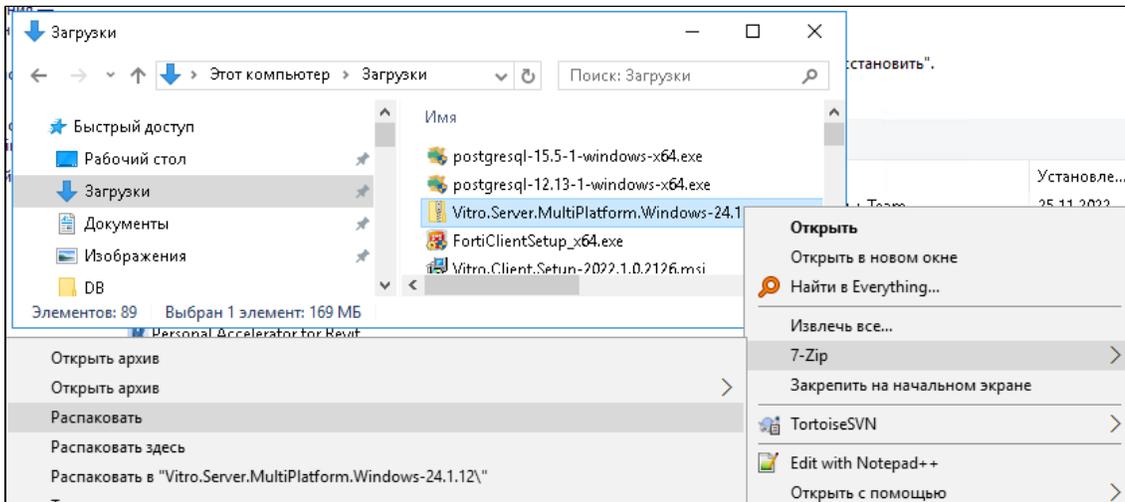
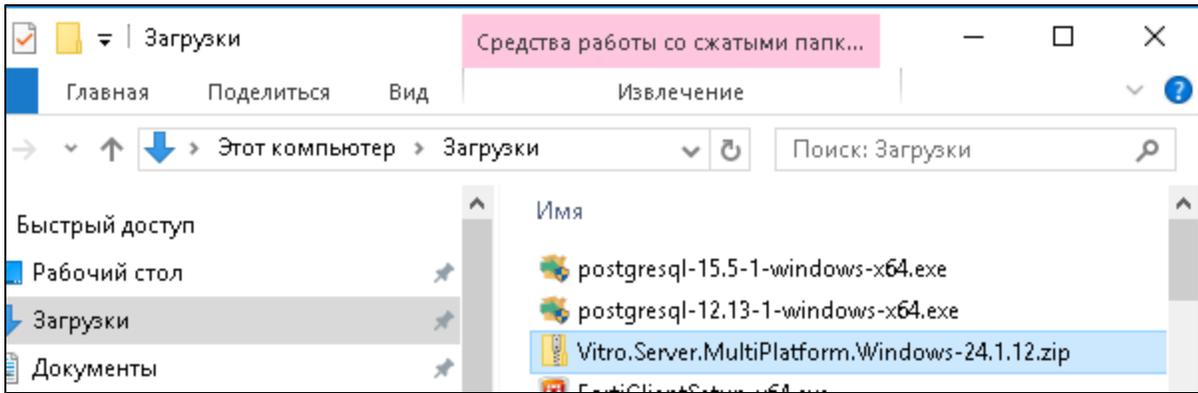
8. Создать новую базу данных с именем workflow:

```
postgres-#CREATE DATABASE workflow with owner = vitrodbuser encoding = 'UTF8' connection limit = -1;
```

9. Отключиться от cli postgresql:

```
postgres-# \q
```

10. Скачать и распаковать дистрибутив:



Инициализация базы данных

Версия дистрибутива 24.1.38 или выше:

11. Файлы инициализации базы данных лежат в папке DB\Install (внутри распакованного дистрибутива):

```
C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.38.7\DB\Install\db_func_create.sql
C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.38.7\DB\Install\db_table_create.sql
C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.38.7\DB\Install\db_data_init.sql
```

12. Запустить инициализацию базы из скриптов:

```
C:\Program Files\PostgreSQL\15\bin>psql -U vitrodbuser -h 127.0.0.1 -d vitrodb -f "C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.38.7\DB\Install\db_func_create.sql" -f "C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.38.7\DB\Install\db_table_create.sql" -f "C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.38.7\DB\Install\db_data_init.sql"
```

Версия дистрибутива ниже 24.1.38:

11.1 Скопировать путь к файлу резервной копии базы данных (внутри распакованного дистрибутива):

```
C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.12\DB\vitrodb.bak
```

12.1 Запустить восстановление БД из бэкапа:

```
C:\Program Files\PostgreSQL\15\bin>psql -U vitrodbuser -h 127.0.0.1 vitrodb < "C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.12\DB\vitrodb.bak"
```

Файлы конфигурации

13. Скопировать следующие папки из дистрибутива в расположение установки в "C:\Program Files\Vitro" :

```
C:\Windows\system32>xcopy C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.12\vitro "C:\Program Files\Vitro\" /e
C:\Windows\system32>xcopy C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.12\Vitro.Server.Core.Web "C:\Program Files\Vitro\Server\Vitro.Server.Core.Web\" /e
C:\Windows\system32>xcopy C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.12\Vitro.Server.Workflow.Manager.Web "C:\Program Files\Vitro\Server\Vitro.Server.Workflow.Manager.Web\" /e
```

14. Скопировать папку файлового хранилища в расположение, в которое смонтирован раздел для хранения:

```
C:\Windows\system32>xcopy C:\distr\Vitro.Server.MultiPlatform.Windows-24.1.12\DB\VitroFileStorage "D:\Data\VitroFileStorage\" /e
```

15. Внести изменения в конфигурационный файл db.json:

```
C:\Program Files\Vitro Software\Vitro\Server\Conf\db.json
```

- изменить строку соединения, указав в качестве значения для атрибута "server" имя или IP сервера БД, а в качестве значения для атрибута "database" название созданной в п.7 базы данных:

Json

```
"ConnectionString": "server=localhost;database=vitrodb;user id=vitrodbuser;password=@pwd4vitro!;Include Error Detail=true",
```

- изменить параметр "FileStoragePath", указав корректный путь к папке файлового хранилища:

Json

```
"FileStoragePath": "D:\Data\VitroFileStorage\",
```

- изменить параметр "TempPath", указав корректный путь к папке временных файлов:

```
"TempPath": "/tmp",
```

16. Внести изменения в конфигурационный файл workflow.json:

```
C:\Program Files\Vitro Software\Vitro\Server\Conf\workflow.json
```

- изменить строку соединения, указав в качестве значения для атрибута "server" имя или IP сервера БД, а в качестве значения для атрибута "database" название созданной в п.7 базы данных:

Json

```
"ConnectionString": "server=localhost;database=workflow;user id=vitrodbuser;password=@pwd4vitro!;Include Error  
Detail=true",
```

- изменить параметр "BaseUrl", указав корректный Url сайта:

Json

```
"BaseUrl": "http://vitro.example.ru",
```

- изменить параметр "PortalUrl", указав корректный Url сайта:

Json

```
"PortalUrl": "http://www.example.com",
```

- изменить параметр "BasePath", указав корректный относительный путь для Url:

Json

```
"BasePath": "/workflow/manager",
```

- Указать логин/пароль учетной записи администратора:

Json

```
"Login": "admin",  
"Password": "admin",
```

17. Внести изменения в конфигурационный файл mail.json:

- Указать адрес и порт SMTP сервера:

```
"Host": "smtp.example.ru",  
"Port": 587,
```

- Указать данные УЗ для подключения к SMTP серверу:

```
"Login": "mail@example.ru",  
"Password": "123456",
```

- Указать адрес отправителя:

```
"FromAddress": "mail@example.ru",  
"FromName": "mail@example.ru",
```

- Указать используется ли SSL для подключения к SMTP серверу:

```
"EnableSsl": true,
```

- Убедиться, что есть параметр SecureSocketOptions:

```
"SecureSocketOptions": "StartTls"
```

Остальные параметры оставить по умолчанию.

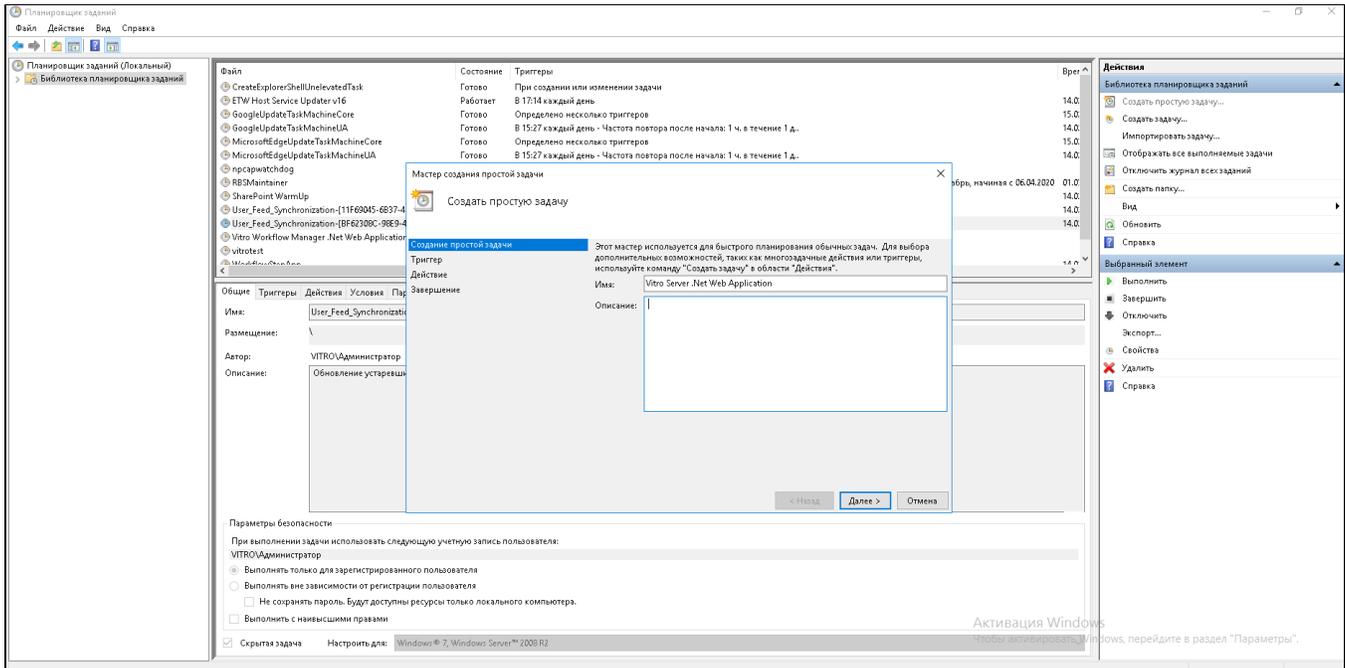
Для автоматического запуска Веб-приложений создать и настроить задания для планировщика Windows:

18. Нажать Win+s и в строке поиска ввести "планировщик заданий". Запустить найденное приложение.

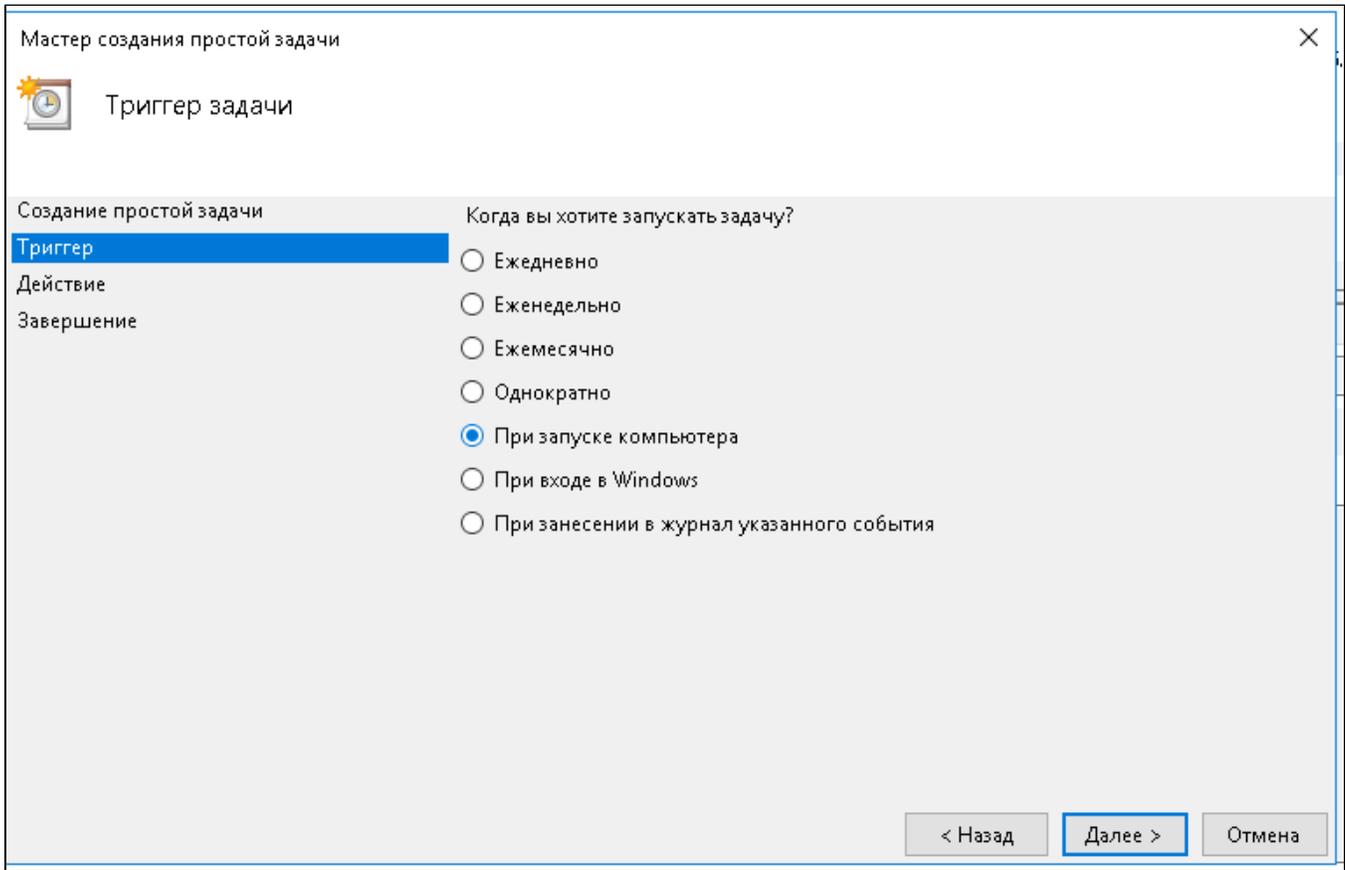


19. В открывшемся окне планировщика на левой панели открыть библиотеку планировщика заданий. На панели "Действия" вызвать функцию "Создать простую задачу". В открывшемся окне ввести "Имя" задачи "Vitro Server .Net Web Application".

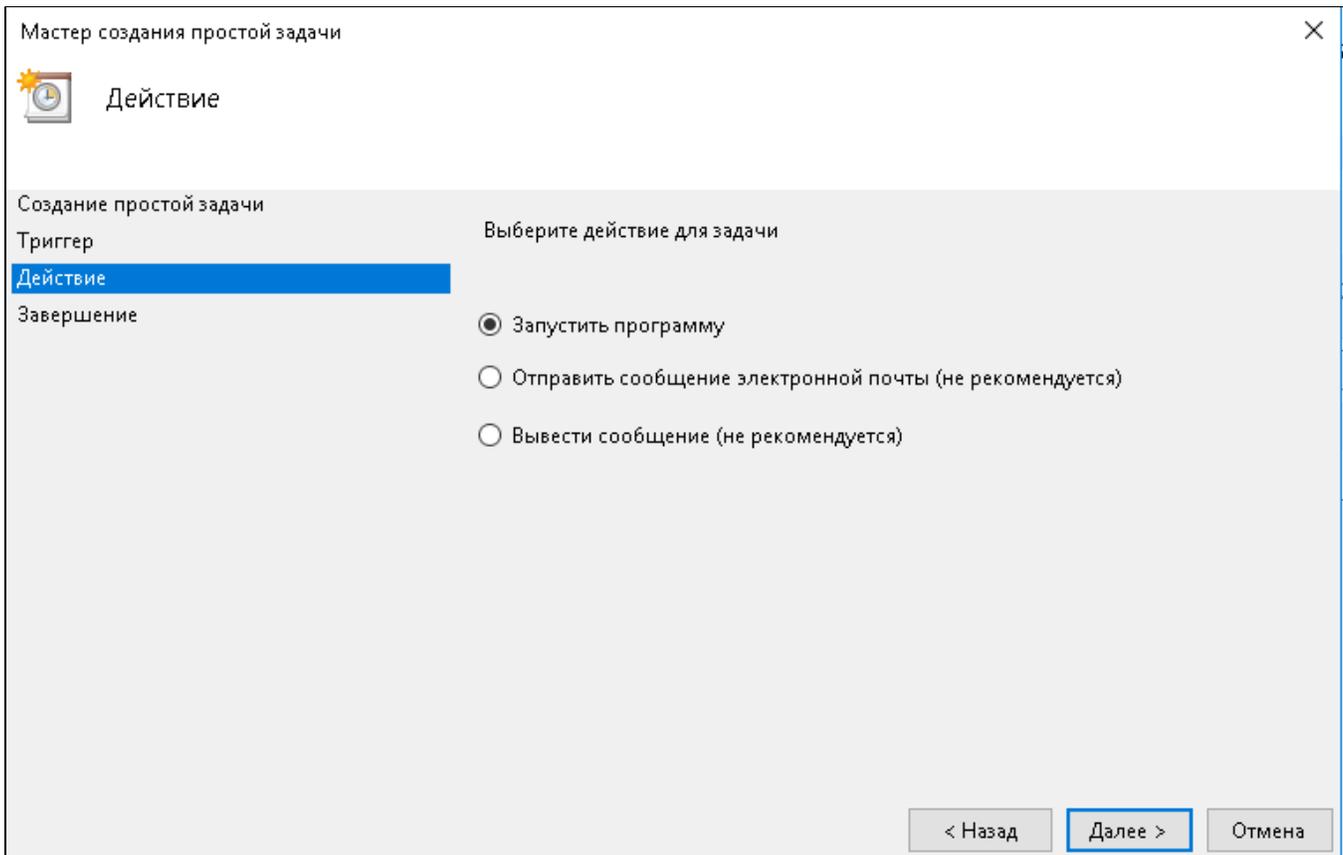
Нажать "Далее".



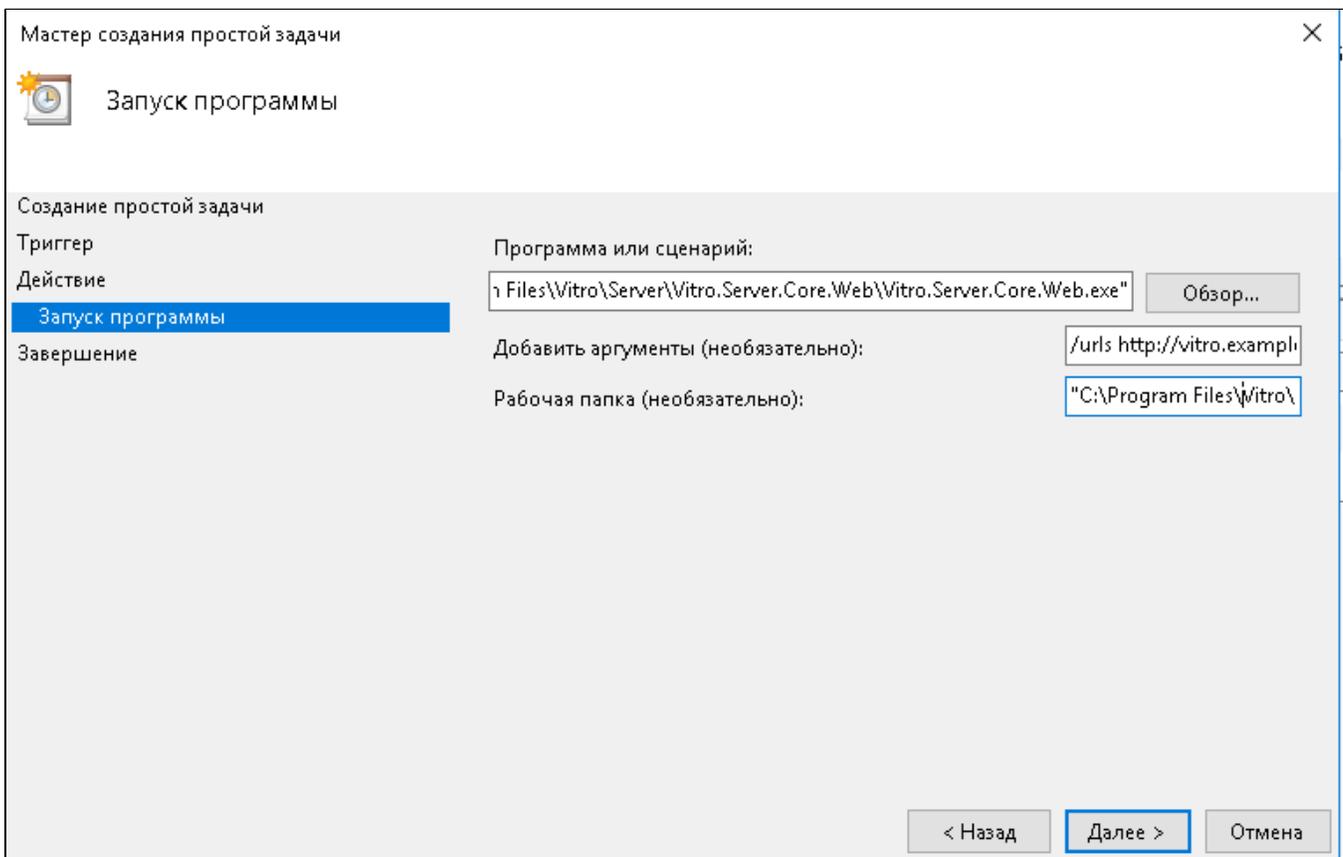
20. На шаге "Триггер" мастера создания выбрать пункт "При запуске компьютера". Нажать "Далее".



21. На шаге "Действие" выбрать пункт "Запустить программу". Нажать "Далее".



22. На шаге "Запуск программы" :



- в строке "Программа или сценарий" указать полный путь к исполняемому файлу приложения:

```
"C:\Program Files\Vitro\Server\Vitro.Server.Core.Web\Vitro.Server.Core.Web.exe"
```

- в строке аргументов указать:

```
/urls http://localhost:4615
```

- в строке "рабочая папка" указать (без кавычек) путь к папке приложения:

```
C:\Program Files\Vitro Software\Vitro\Server\Vitro.Server.Core.Web
```

23. На шаге "Завершение" отметить флаг "Открыть окно "Свойства" для..." и нажать "Готово" :

Мастер создания простой задачи

Сводка

Создание простой задачи

Триггер

Действие

Запуск программы

Завершение

Имя: Vitro Server .Net Web Application

Описание:

Триггер: При запуске; При включении компьютера

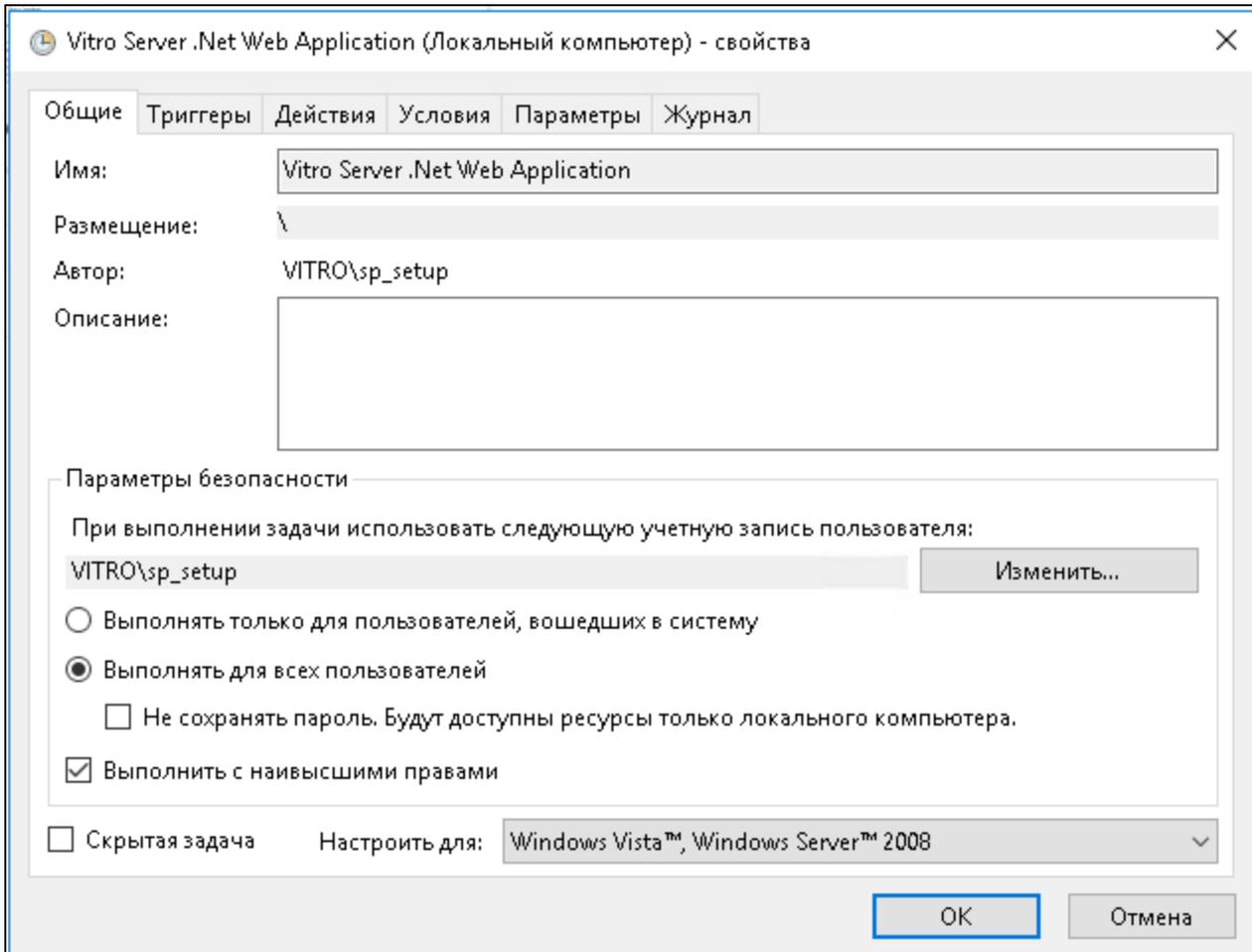
Действие: Запуск программы; "C:\Program Files\Vitro\Server\Vitro.Server.Core.Web\

Открыть окно "Свойства" для этой задачи после нажатия кнопки "Готово"

После нажатия кнопки "Готово" новая задача создается и добавляется в расписание Windows.

< Назад Готово Отмена

24. в открывшемся окне свойств установить переключатель в положение "Выполнять для всех пользователей", и отметить флаг "Выполнить с наивысшими правами". Также можно сменить пользователя, от которого будет выполняться задание. После подтверждения ввести пароль пользователя.



25. Повторить шаги 18 - 23 для приложения Workflow Manager задав следующие параметры:

```
" " : "Vitro Workflow Manager .Net Web Application"
" " : "C:\Program Files\Vitro\Server\Vitro.Server.Workflow.Manager.Web\Workflow.Manager.Web.exe"
: /urls http://localhost:4616
" " C:\Program Files\Vitro\Server\Vitro.Server.Workflow.Manager.Web
```

26. запустить задания вручную и проверить выполняется ли прослушивание на портах, заданных в аргументах команд (здесь 4615 и 4616) для адреса Lo интерфейса:

```
netstat -a

...
TCP    127.0.0.1:4615          Vitrow2k16:0          LISTENING  ...
TCP    127.0.0.1:4616          Vitrow2k16:0          LISTENING  ...
```

27. Настроить обратное проксирование.

Если планируется использование авторизации через Kerberos, то необходимо [настроить обратное проксирование через IIS](#).

Иначе возможно использовать NGINX в качестве обратного прокси.

Пример конфигов для обратного проксирования для NGINX (for Windows):

```
C:\nginx\conf\nginx.conf
```

```

#user nobody;
worker_processes 1;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;

#pid logs/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;

    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
    # '$status $body_bytes_sent "$http_referer" '
    # '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log logs/access.log main;

    sendfile on;
    #tcp_nopush on;

    #keepalive_timeout 0;
    keepalive_timeout 65;

    #gzip on;

server {
    listen 80 default_server;
    server_name vitro.example.ru;

    proxy_http_version 1.1;
    proxy_pass_request_headers on;
    proxy_pass_request_body on;

    proxy_buffer_size 1M;
    proxy_buffers 4 1M;
    proxy_read_timeout 1h;
    proxy_send_timeout 1h;

    location / {
        #proxying traffic to upstream application (Vitro.Server.Core.Web)
        proxy_pass http://127.0.0.1:4615;
    }

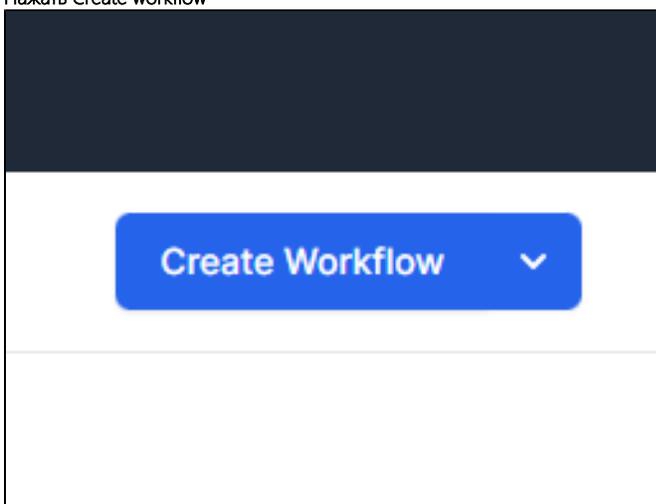
    location /workflow/manager {
        proxy_pass http://localhost:4616/workflow/manager;
    }
}
}

```

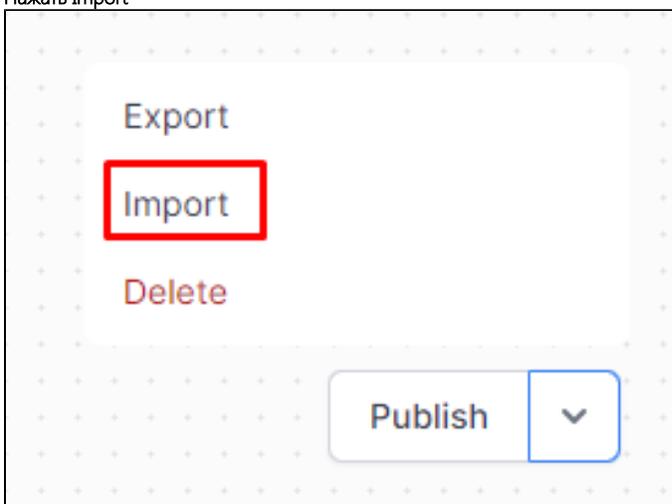
Добавление шаблонов рабочих процессов

1. Перейти на вкладку Workflow Defenitions в Elsa
<http://vitro.example.ru/workflow/manager/workflow-definitions>
 где vitro.example.ru - адрес сайта

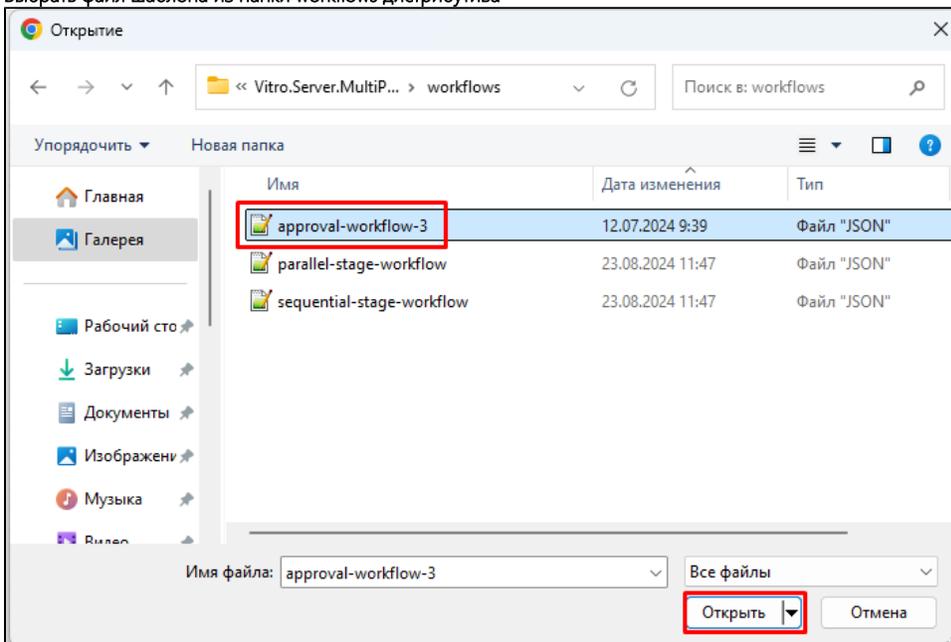
2. Нажать Create workflow



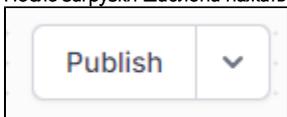
3. Нажать Import



4. Выбрать файл шаблона из папки workflows дистрибутива



5. После загрузки шаблона нажать Publish



6. Перейти обратно на вкладку Workflow defenitions и убедиться, что шаблон создан
7. Таким образом добавить все шаблоны из папки workflows дистрибутива

Настройка системы

Необходимо указать адрес портала в ключе `Site.PortalUrl`.

Для этого нужно:

1. Перейти в список Конфигурация пространства Центр Администрирования.
2. Открыть папку Общие настройки
3. Для ключа Адрес портала указать в поле Значение (строка) адрес портала в виде <https://example.com>



Важно!

Адрес портала не должен заканчиваться символом слэша.

Настройка обратного прокси в IIS

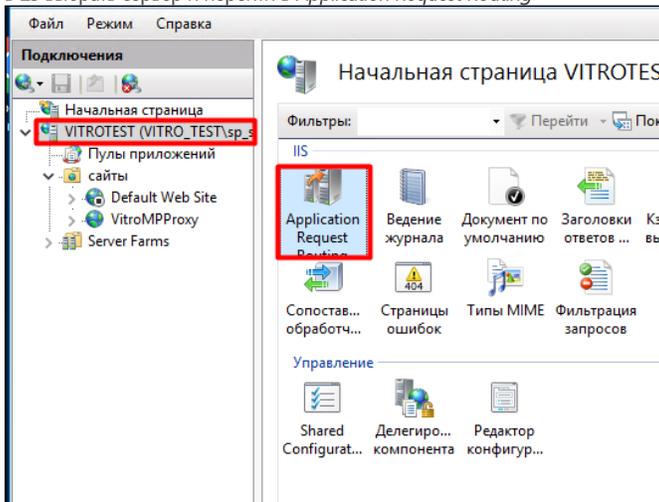
Если планируется использовать авторизацию через Kerberos, то необходимо настраивать reverse proxy на базе IIS.

❗ Если ранее в качестве обратного прокси использовался другой веб-сервер (например, nginx), то его необходимо отключить.

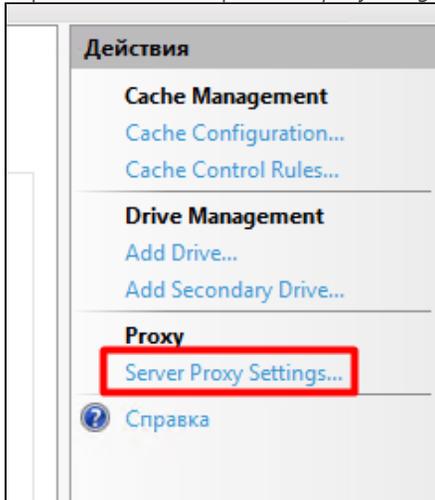
1. Создать пустой сайт на необходимом порту (80 для http или 443 для https) без дополнительных настроек. Для https так же необходимо добавить SSL/TLS сертификат.
Подробнее в [статье](#).
2. Скачать аддоны для IIS:
 - a. [ARR](#)
 - b. [URLRewrite](#)
3. Установить *ARR*:
 - a. Остановить службы *WAS* и *WMSVC* (службы *WMSVC* может не быть)

```
net stop was /y  
net stop wmsvc /y
```
 - b. Установить *AAR* с помощью скачанного дистрибутива
 - c. Включить службы *WAS* и *WMSVC*

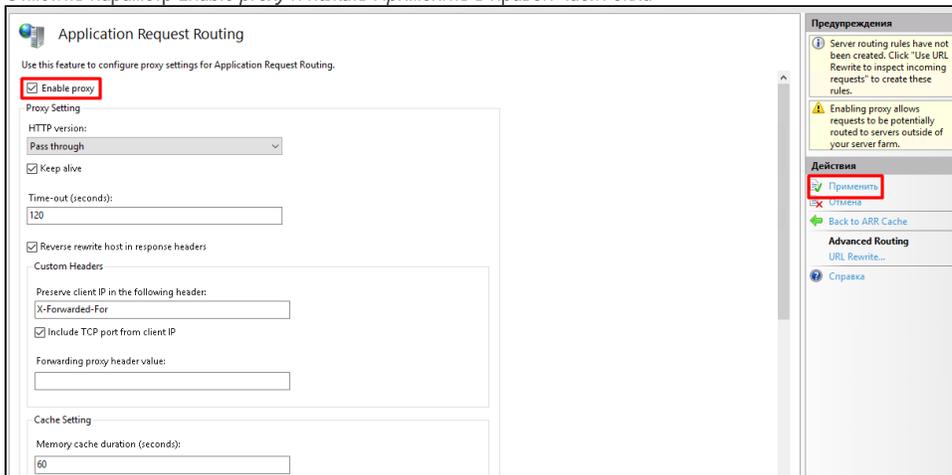
```
net start was  
net start wmsvc
```
4. Установить *URLRewrite*:
 - a. Установить *URLRewrite* с помощью скачанного дистрибутива
 - b. Перезапустить IIS командой `iisreset`
5. Включить проксирование:
 - a. В IIS выбрать сервер и перейти в *Application Request Routing*



- b. В правой части окна выбрать *Server proxy settings*



- с. Отметить параметр *Enable proxy* и нажать *Применить* в правой части окна



6. Разместить в папке сайта *web.config* ([ссылка для скачивания](#))

7. Исправить в *web.config* адрес сайта в указанных местах (подставить адрес сайта заказчика вместо *dev-vs*)

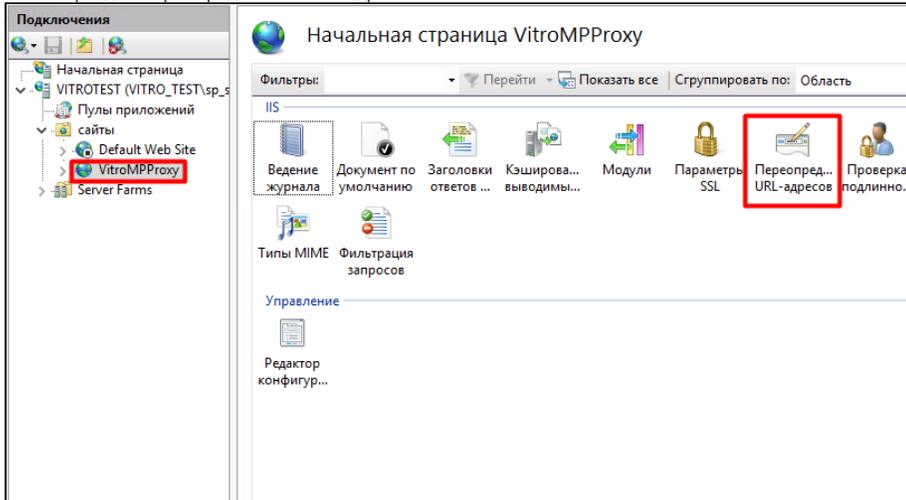
```

<system.webServer>
  <rewrite>
    <outboundRules>
      <rule name="ReverseProxyOutboundRule2" preCondition="ResponseIsHtml1">
        <match filterByTags="A, Form, Img" pattern="http(s)?://localhost:4616/(.*)" />
        <action type="Rewrite" value="http(R:1)://dev-vs/workflow/manager/(R:2)" />
      </rule>
      <rule name="ReverseProxyOutboundRule1" preCondition="ResponseIsHtml1">
        <match filterByTags="A, Form, Img" pattern="http(s)?://localhost:4615/(.*)" />
        <action type="Rewrite" value="http(R:1)://dev-vs/(R:2)" />
      </rule>
      <rule name="RestoreAcceptEncoding" preCondition="NeedsRestoringAcceptEncoding">
        <match serverVariable="HTTP_ACCEPT_ENCODING" pattern="*/.*" />
      </rule>
    </outboundRules>
  </rewrite>
</system.webServer>

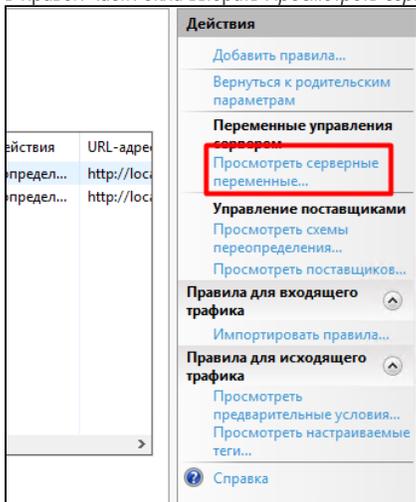
```

8. Создать переменные для *URLRewrite*.
 а. В IIS выбрать сайт, созданный в п.1

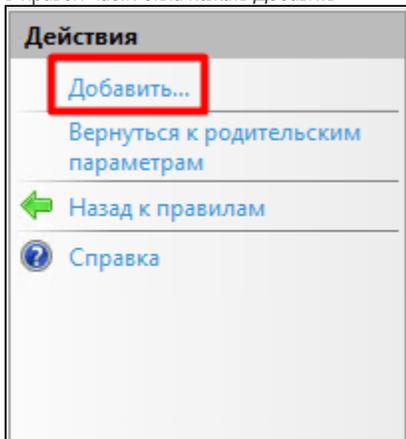
b. Зайти в раздел *Переопределение URL-адресов*



c. В правой части окна выбрать *Просмотреть серверные переменные*



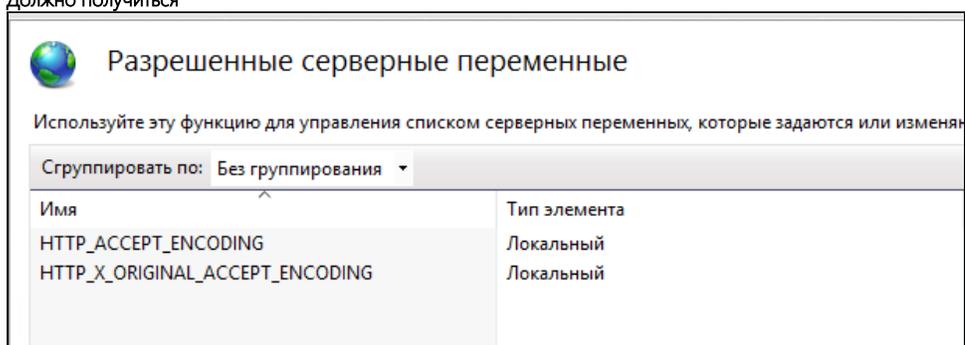
d. В правой части окна нажать *Добавить*



e. Указать имя серверной переменной `HTTP_ACCEPT_ENCODING` и нажать *OK*

f. Повторить действия из п.д-е указав имя переменной `HTTP_X_ORIGINAL_ACCEPT_ENCODING`

g. Должно получиться



9. Перезапустить IIS командой `iisreset`

Дополнительные статьи по теме:

[Setting up IIS as Reverse Proxy for the .Net Core Kestrel Web Server \[pdf\]](#)

[IIS with URL Rewrite as a reverse proxy - part 2 – dealing with 500.52 status codes \[pdf\]](#)

[Настройка ограниченного делегирования Kerberos для прокси-страниц веб-регистрации \[pdf\]](#)

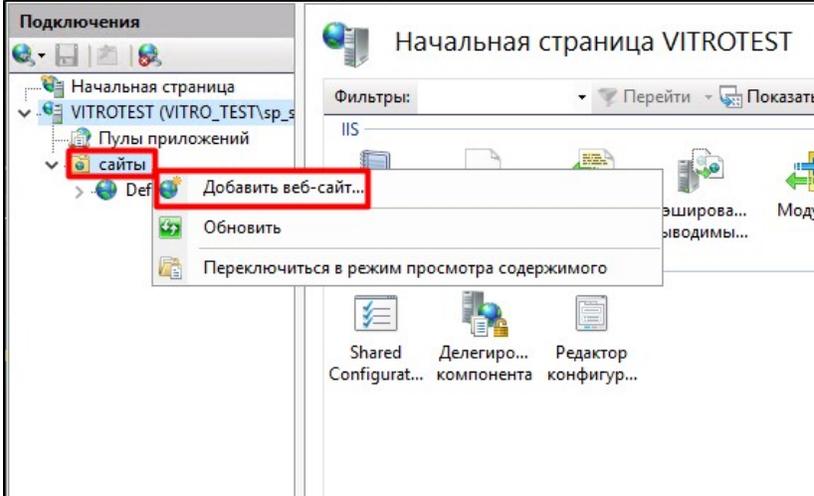
[Настройка Kerberos авторизации на сайте IIS \[pdf\]](#)

Создание веб-сайта IIS для обратного прокси

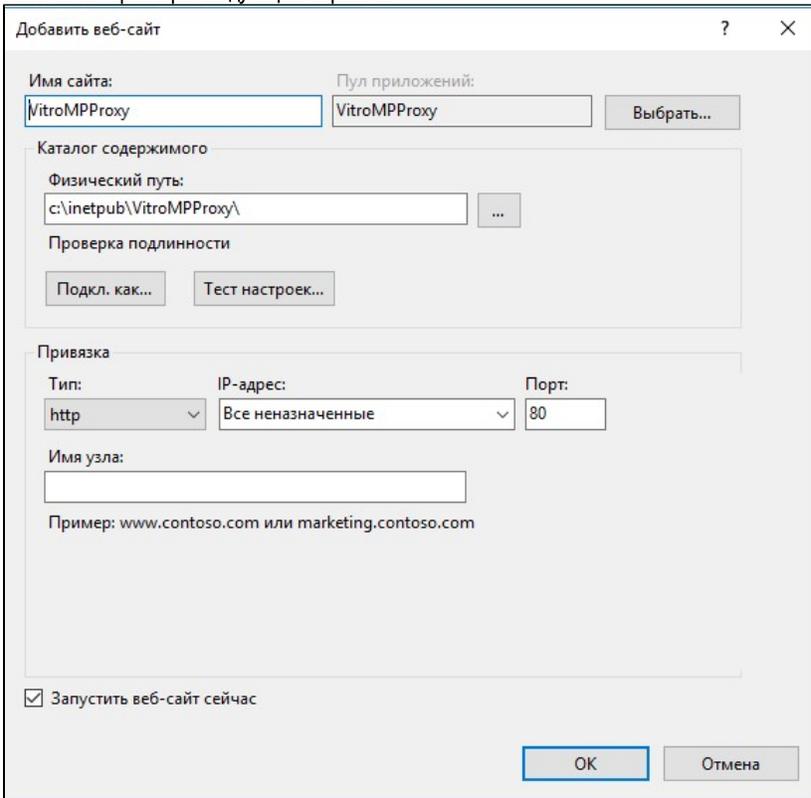
Создание веб-сайта

Для подключения по http

1. В контекстном меню папки Сайты выбрать Добавить веб-сайт



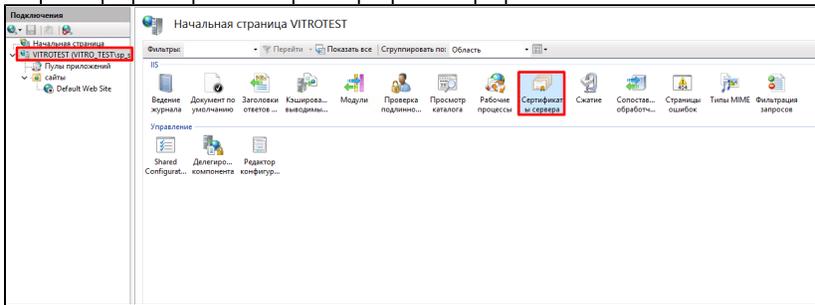
2. Заполнить параметры следующим образом:



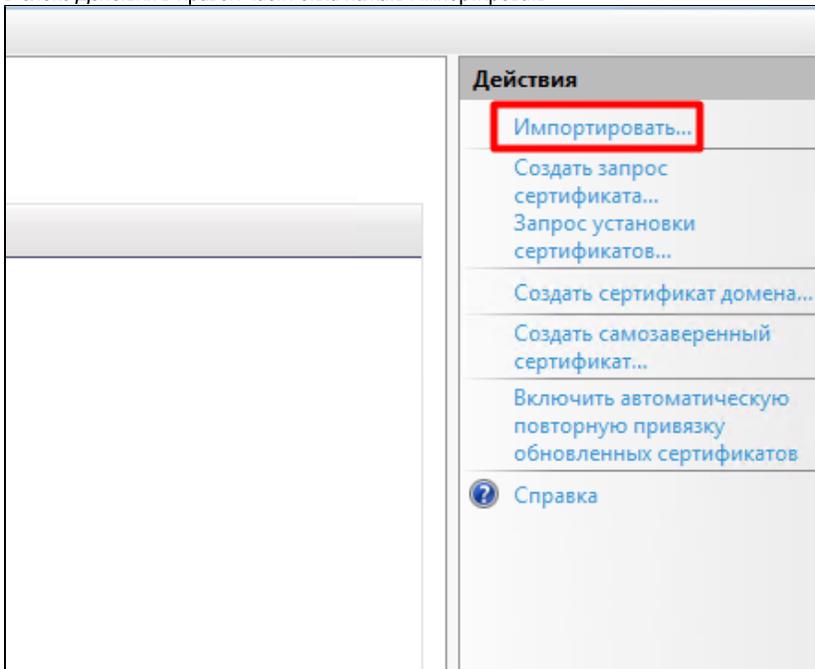
- а. Имя сайта - VitroMPPроху
- б. Физический путь - c:\inetpub\VitroMPPроху\ (папку нужно создать заранее)
- в. Тип - http
- г. Порт - 80
- д. Остальные параметры по умолчанию

Для подключения по https

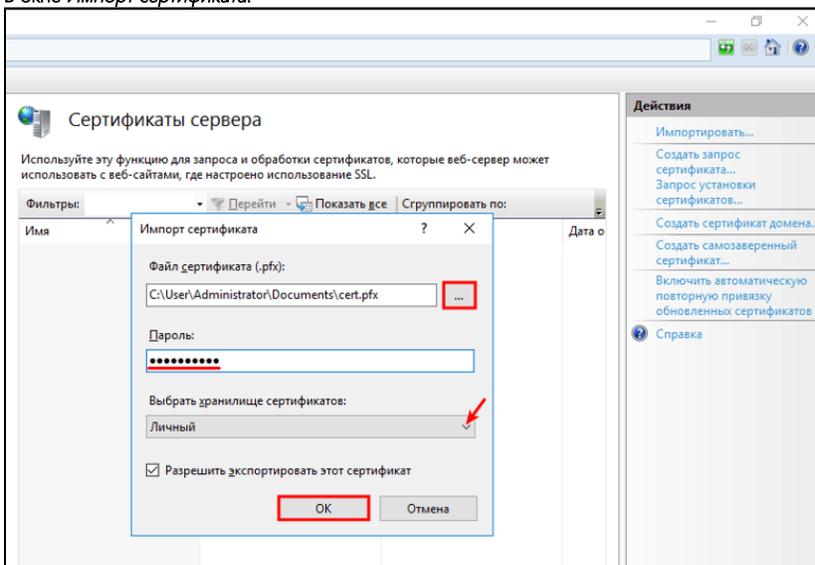
1. Выбрать сервер и открыть категорию Сертификаты сервера



2. В блоке Действия в правой части окна нажать Импортировать

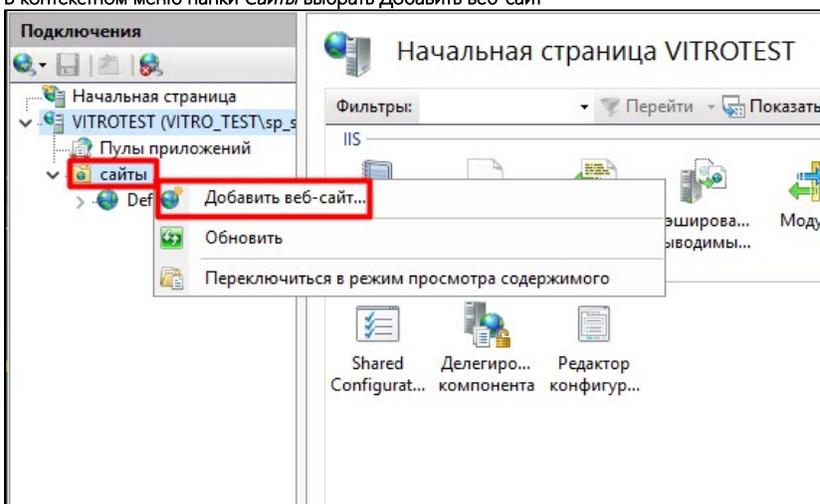


3. В окне Импорт сертификата:

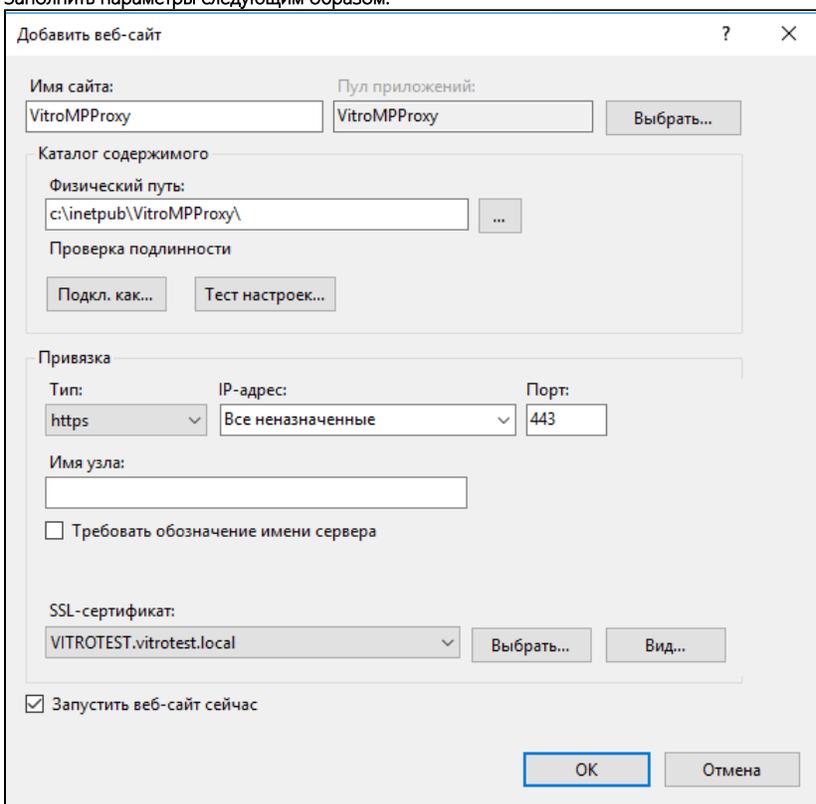


- Выбрать PFX файл сертификата
- Ввести пароль для PFX файла сертификата
- Выбрать хранилище сертификатов Личный
- Нажать OK

4. В контекстном меню папки Сайты выбрать Добавить веб-сайт



5. Заполнить параметры следующим образом:



- a. Имя сайта - VitroMPPроху
- b. Физический путь - c:\inetpub\VitroMPPроху\ (папку нужно создать заранее)
- c. Тип - https
- d. Порт - 443
- e. SSL-сертификат - выбрать добавленный в п.3 сертификат
- f. Остальные параметры по умолчанию

Создание и настройка учетной записи для веб-сайта

1. Создать учетную запись в AD. Пример УЗ *vitro_service_app*:

Свойства: vitro_service_app

Организация	Опубликованные сертификаты	Член групп			
Репликация паролей	Входящие звонки	Объект	Безопасность	Среда	
Сеансы	Удаленное управление				
Профиль служб удаленных рабочих столов	COM+	Редактор атрибутов			
Общие	Адрес	Учетная запись	Профиль	Телефоны	Делегирование

 vitro_service_app

Имя: vitro_service_app Инициалы:

Фамилия:

Выводимое имя: vitro_service_app

Описание:

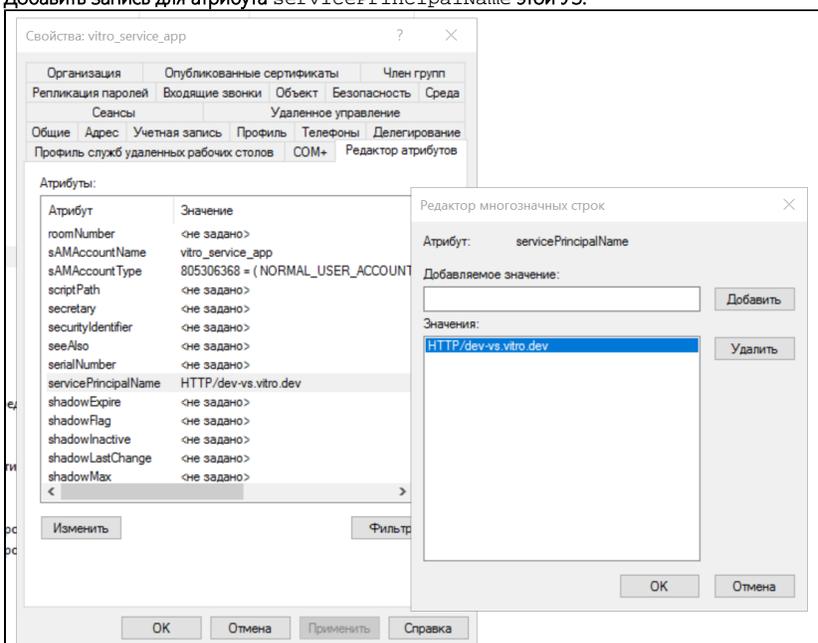
Комната:

Номер телефона:

Эл. почта:

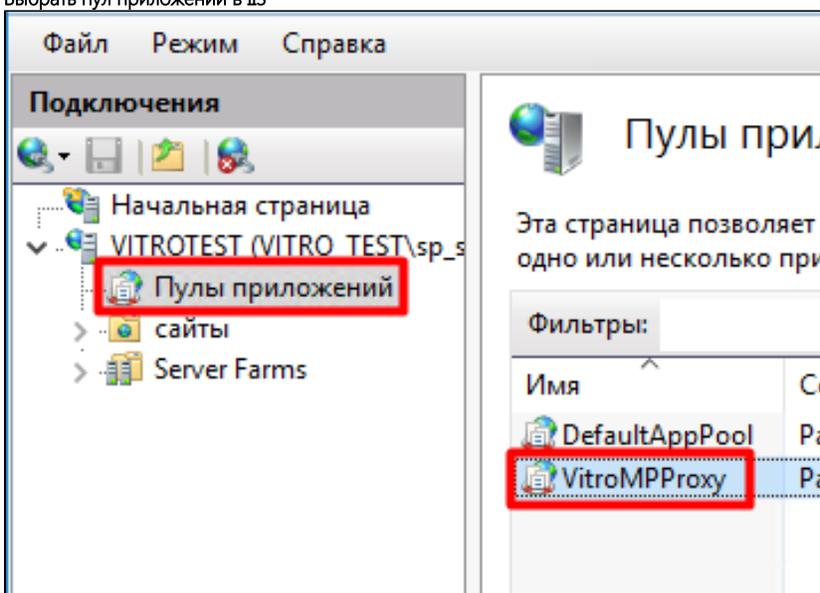
Веб-страница:

2. Добавить запись для атрибута servicePrincipalName этой УЗ:



Где dev-vs.vitro.dev - полное доменное имя веб-сервера, на котором производится настройка веб-сайта. Если для веб-сервера в сети настроен алиас, то его так же необходимо добавить.

3. Выбрать пул приложений в IIS



Действия

 **Добавить пул приложений...**
Определить значения по умолчанию для пула приложений...

Задачи пула приложений

 Запустить

 Остановить

 Перезапуск...

Изменить пул приложений

 Основные настройки...

 Перезапуск...

Дополнительные параметры...

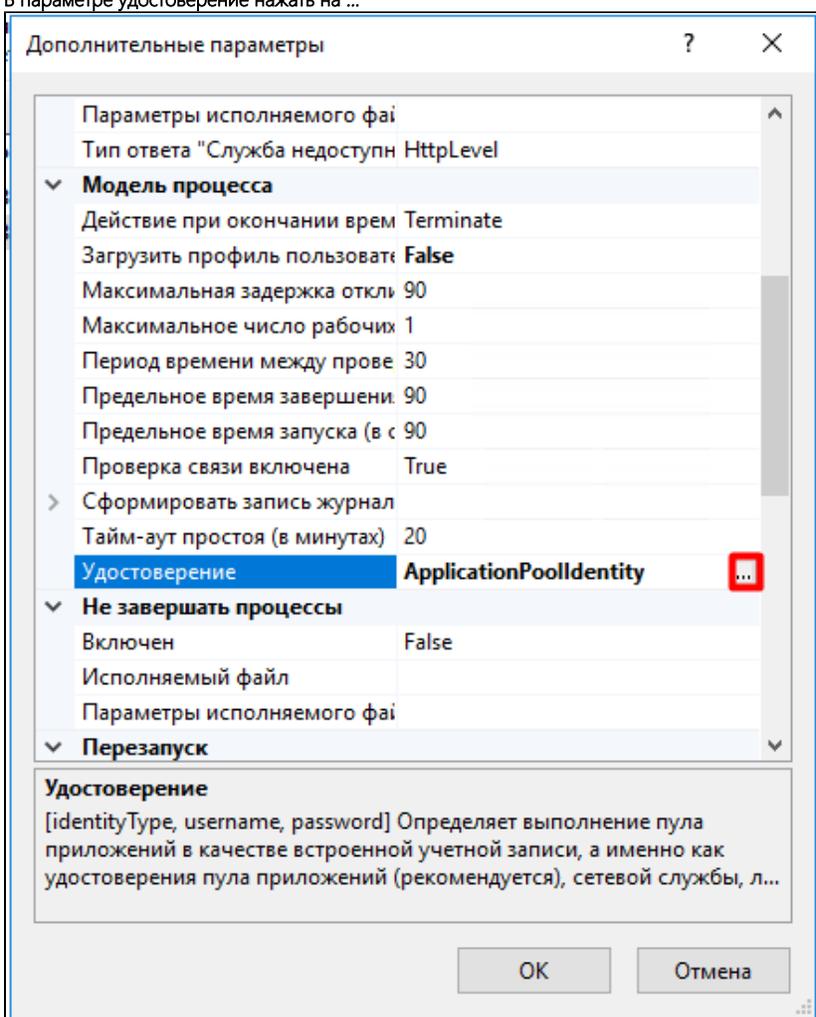
 Переименовать

 Удалить

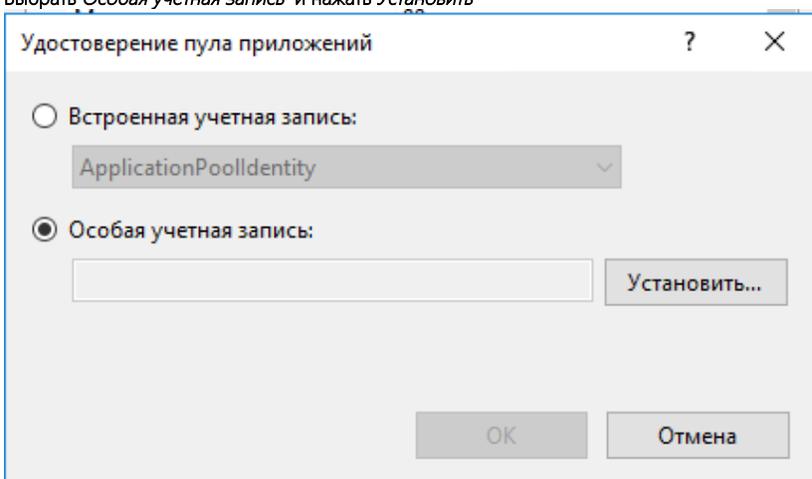
Просмотреть приложения

 Справка

5. В параметре удостоверение нажать на ...



6. Выбрать *Особая учетная запись* и нажать *Установить*



7. Ввести данные УЗ, созданной в п.1 и нажать ОК

Задание учетных данных

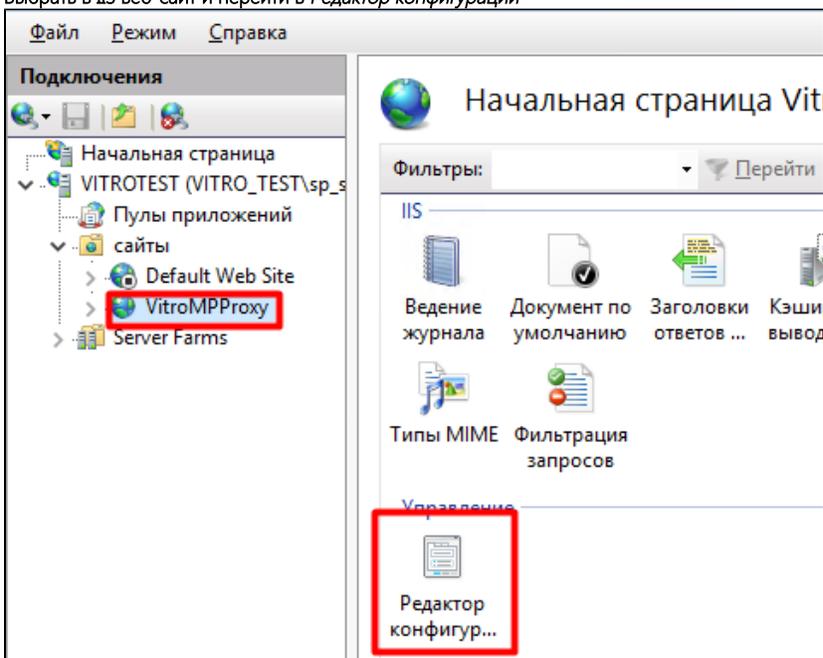
Имя пользователя:
vitro_service_app

Пароль:
●●●●●●●●

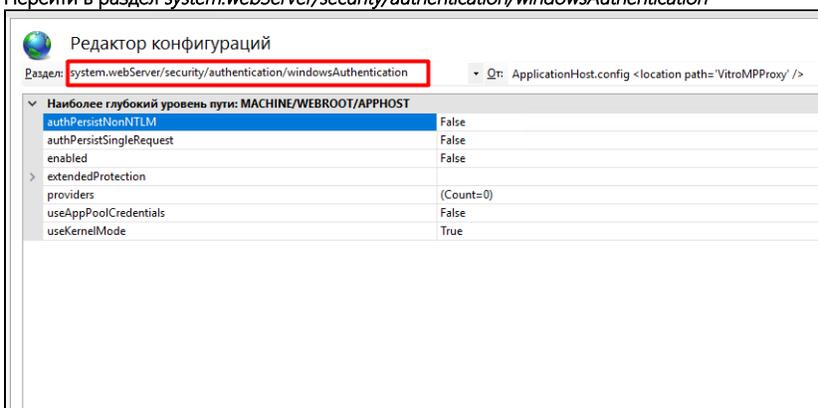
Подтвердите пароль:
●●●●●●●●

ОК Отмена

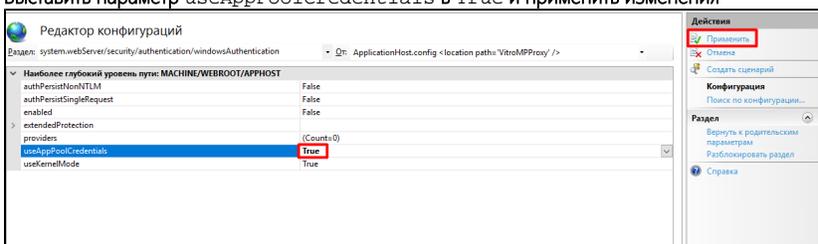
8. Выбрать в IIS веб-сайт и перейти в Редактор конфигураций



9. Перейти в раздел `system.webServer/security/authentication/windowsAuthentication`



10. Выставить параметр `useAppPoolCredentials` в `True` и применить изменения



11. Открыть cmd от имени администратора и выполнить команду `klist purge`

12. Выполнить перезагрузку сервера (не IIS, а полностью перезапустить машину)

Описание процесса обновления серверной части Vitro-CAD и модуля Vitro Workflow (Linux)

1. Остановить службы

```
sudo systemctl stop vitro-server  
sudo systemctl stop vitro-workflow
```

Обновление базы данных:

2. Скачать и распаковать дистрибутив:

```
curl "https://linkTo/Vitro.Server.MultiPlatform.Linux-xx.x.xx.zip" --output 'Vitro.Server.MultiPlatform.Linux-xx.x.xx.zip'  
sudo unzip Vitro.Server.MultiPlatform.Linux-xx.x.xx.zip
```

3. Для обновления БД перейти в распакованную на сервере БД папку дистрибутива:

```
cd ~/Vitro.Server.MultiPlatform.Linux-xx.x.xx/DB/Update
```

4. Установить атрибут "Исполняемый" для файла update.sh:

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

5. Запустить скрипт update.sh с необходим набором параметров



Описание параметров запуска

- d <dbname> – имя обновляемой базы данных
- H <hostname> – имя хоста или IP адрес сервера БД
- g <logon user group> – имя группы пользователя, от которого выполнен вход в сессию консоли
- u <dbusername> – имя пользователя PostgreSQL – владельца БД
- p <password> – пароль пользователя PostgreSQL – владельца БД

Примеры запуска скрипта:

запуск с минимальным набором параметров, в случае если на сервере БД для пользователя «postgres» разрешено локальное подключение СУБД без пароля.

```
./update.sh -d vitrodb -g vitroadmin
```

запуск с полным набором параметров, в случае если на сервере БД выполнена настройка безопасности, подразумевающая необходимость сетевого подключения для пользователя – владельца БД (в примере «vitrodbuser») с паролем

```
./update.sh -d vitrodb -g vitroadmin -H vitro-db-srv -u vitrodbuser -p @pwd4vitro!
```

Обновление программных компонентов на сервере Веб-приложения:

6. Создать директорию для резервного копирования конфигурационных файлов:

```
mkdir ./bak
```

7. Скопировать конфигурационные файлы из директории с установленным приложением Vitro.Server.Core.Web в директорию резервного копирования:

```
sudo cp -r /etc/Vitro/Server/Site/TableView/ ./bak/  
sudo cp -r /etc/Vitro/Server/Conf/ ./bak/
```



Важно!

Если установлен дистрибутив версии 25.1.39.12 и ранее, то необходимо создать резервные копии конфигурационных файлов следующим образом:

Копирование конфигурационных файлов

```
mkdir -p bak/core bak/workflow  
sudo cp -r /usr/local/Vitro.Server.Core.Web/Vitro/Server/Site/TableView/ bak/core/  
sudo cp -r /usr/local/Vitro.Server.Core.Web/Vitro/Server/Conf bak/core/  
sudo cp -r /usr/local/Vitro.Server.Workflow.Manager.Web/Vitro/Server/Site/TableView/ bak  
/workflow/  
sudo cp -r /usr/local/Vitro.Server.Workflow.Manager.Web/Vitro/Server/Conf bak/workflow/
```

После нужно скопировать папку Vitro в расположение /etc:

Копирование папки Vitro в /etc/

```
sudo cp -r /usr/local/Vitro.Server.Core.Web/Vitro/ /etc/
```

Выставить уровень прав на папке:

```
sudo chmod 500 /etc/Vitro/
```

8. Удалить папки с установленными приложениями Vitro.Server.Core.Web и Vitro.Server.Workflow.Manager.Web:

```
sudo rm -rf /usr/local/Vitro.Server.Core.Web  
sudo rm -rf /usr/local/Vitro.Server.Workflow.Manager.Web
```

9. Скопировать рекурсивно директории приложений Vitro.Server.Core.Web и Vitro.Server.Workflow.Manager.Web в целевое расположение:

```
sudo cp -r Vitro.Server.MultiPlatform.Linux-xx.x.xx/Vitro.Server.Core.Web/ /usr/local/  
sudo cp -r Vitro.Server.MultiPlatform.Linux-xx.x.xx/Vitro.Server.Workflow.Manager.Web/ /usr/local/
```

10. Сменить владельца директорий приложений рекурсивно:

```
chown -R www-data:www-data /usr/local/Vitro.Server.Core.Web  
chown -R www-data:www-data /usr/local/Vitro.Server.Workflow.Manager.Web
```

11. Установить атрибут разрешающий запуск на основные файлы приложений:

```
chmod +x /usr/local/Vitro.Server.Core.Web/Vitro.Server.Core.Web  
chmod +x /usr/local/Vitro.Server.Workflow.Manager.Web/Vitro.Server.Workflow.Manager.Web
```

 При обновлении с версии 25.1.39.22 и ниже необходимо указать адрес портала в ключе `Site.PortalUrl`.

Для этого нужно:

- a. Перейти в список Конфигурация пространства Центр Администрирования.
- b. Открыть папку Общие настройки
- c. Для ключа Адрес портала указать в поле Значение (строка) адрес портала в виде <https://example.com>

 **Важно!**
Адрес портала не должен заканчиваться символом слэша.

При обновлении с версий 24.1.37 и ниже до версий 24.1.38 и выше необходимо учесть, что изменилось имя исполняемого файла модуля рабочих процессов на `Vitro.Server.Workflow.Manager.Web`.

В связи с этим команда для предоставления прав на выполнение файлу выглядит так:

```
chmod +x /usr/local/Vitro.Server.Workflow.Manager.Web/Vitro.Server.Workflow.Manager.Web
```

Так же:

при обновлении в файле `/lib/systemd/system/vitro-workflow.service` нужно изменить путь до исполняемого файла:

```
ExecStart=/usr/local/Vitro.Server.Workflow.Manager.Web/Workflow.Manager.Web
```

на

```
ExecStart=/usr/local/Vitro.Server.Workflow.Manager.Web/Vitro.Server.Workflow.Manager.Web
```

и выполнить команду:

```
systemctl daemon-reload
```

В файлах конфигурации `/usr/local/Vitro.Server.Workflow.Manager.Web/Vitro/Server/Conf/workflow.json` и `/usr/local/Vitro.Server.Core.Web/Vitro/Server/Conf/workflow.json` нужно прописать логин/пароль учетной записи администратора:

```
"Login": "admin",
```

```
"Password": "admin",
```

```
:
```

```
"ConnectionString": "server=vm-vs-2022;database=workflow;user",
"BaseUrl": "http://localhost:4616",
"PortalUrl": "http://localhost:4616",
"BasePath": "/workflow/manager",
"LoginPagePath": "",
"SystemUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"Login": "admin",
"Password": "admin",
```

В ELSA импортировать новые планшеты рабочих процессов. Планшеты лежат в папке с дистрибутивом:

```
workflows\parallel-stage-workflow.json
```

```
workflows\sequential-stage-workflow.json
```

12. Удалить папку `/etc/Vitro/Server/Solutions`:

Удаление папки `Solutions`

```
sudo rm -rf /etc/Vitro/Server/Solutions/
```

13. Скопировать из дистрибутива папки в /etc/Vitro:

Копирование папки Vitro

```
sudo cp -r Vitro.Server.MultiPlatform.Linux-xx.x.xx/Vitro/Server/Solutions/ /etc/Vitro/Server/  
sudo cp -rf Vitro.Server.MultiPlatform.Linux-xx.x.xx/Vitro/Server/Site/TableView/. /etc/Vitro/Server/Site/  
/TableView  
sudo cp -rn Vitro.Server.MultiPlatform.Linux-xx.x.xx/Vitro/Server/Conf/. /etc/Vitro/Server/Conf
```

14. Сменить владельца директории Vitro рекурсивно:

Смена владельца директории Vitro

```
sudo chown -R www-data:www-data /etc/Vitro/
```

15. Заменить системные файлы в файловом хранилище:

```
sudo cp -rf Vitro.Server.MultiPlatform.Linux-xx.x.xx/DB/VitroFileStorage/. /mnt/VitroFileStorage
```

16. Сменить владельца директории файлового хранилища рекурсивно:

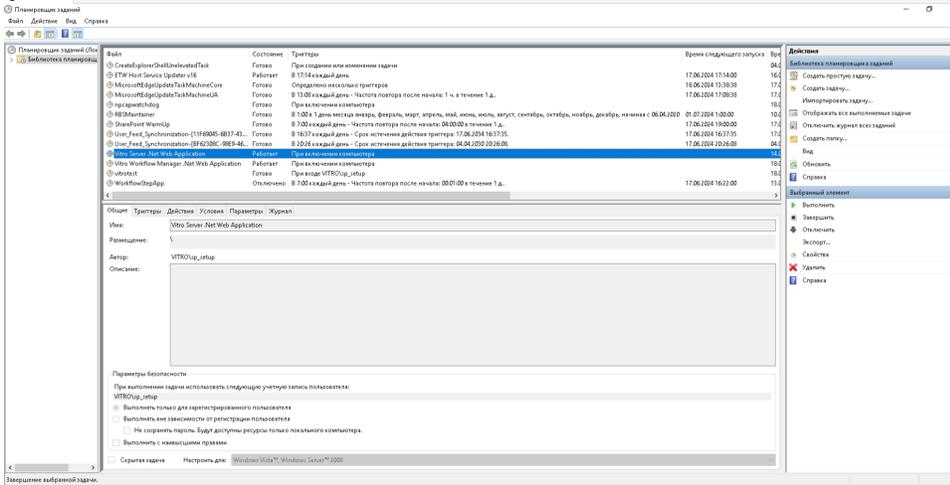
```
sudo chown -R www-data:www-data /mnt/VitroFileStorage/
```

17. Запустить службы приложений:

```
sudo systemctl start vitro-server  
sudo systemctl start vitro-workflow
```

Описание процесса обновления серверной части Vitro-CAD и модуля Vitro Workflow (Windows)

1. Остановить задания автоматического запуска Веб-приложений vitro-server и vitro-workflow



Обновление базы данных:

2. Скачать и распаковать дистрибутив актуальной версии Vitro.Server.MultiPlatform.Windows-xx.x.xx.zip.
3. Для обновления БД запустить от имени администратора сеанс PowerShell и сменить директорию на распакованную на сервере БД папку дистрибутива:

```
cd C:\< >\Vitro.Server.MultiPlatform.Windows-xx.x.xx\DB\Update
```

4. Снять (рекурсивно) блокирующий атрибут с файлов внутри папки с обновлениями, выполнив в PowerShell :

```
Get-Childitem -Recurse | Unblock-File
```

5. Запустить скрипт update.ps1 с необходим набором параметров

i Описание параметров запуска

- PgBinPath <PathToPostgresql> - путь к директории с исполняемыми файлами Postgresql
- DBServerName <hostname> – IP адрес сервера БД
- DBName <DB name> – имя базы данных
- VitroDBLogin <dbusername> – имя пользователя Postgresql - владельца БД
- VitroDBPassword <password> – пароль пользователя Postgresql - владельца БД

Пример запуска скрипта:

```
.\Update.ps1 -PgBinPath 'C:\Program Files\PostgreSQL\15\bin' -DBServerName '192.168.0.214' -DBName 'vitrodb1' -VitroDBLogin 'vitrodbuser' -VitroDBPassword '@pwd4vitro!'
```

Обновление программных компонентов на сервере Веб-приложения:

6. Создать директорию для резервного копирования конфигурационных файлов:

```
mkdir "c:\temp\bak"
```

7. Скопировать конфигурационные файлы из директории с установленным приложением Vitro.Server.Core.Web в директорию резервного копирования:

```
C:\Windows\system32>xcopy "C:\Program Files\Vitro Software\Vitro\Server\Site" "C:\temp\Site" /e  
C:\Windows\system32>xcopy "C:\Program Files\Vitro Software\Vitro\Server\Conf" "C:\temp\Conf" /e
```

8. Удалить папку с установленными серверными компонентами:

```
del "C:\Program Files\Vitro Software\Vitro\Server"
```

9. Скопировать рекурсивно директории серверных компонентов в целевое расположение:

```
C:\Windows\system32>xcopy "C:\< >\Vitro.Server.MultiPlatform.Windows-xx.x.xx\Vitro\Server" "C:\Program  
Files\Vitro Software\Vitro\Server" /e  
C:\Windows\system32>xcopy "C:\< >\Vitro.Server.MultiPlatform.Windows-xx.x.xx\Vitro.Server.Core.Web" "C:  
\Program Files\Vitro Software\Vitro\Server\Vitro.Server.Core.Web" /e  
C:\Windows\system32>xcopy "C:\< >\Vitro.Server.MultiPlatform.Windows-xx.x.xx\Vitro.Server.Workflow.  
Manager.Web" "C:\Program Files\Vitro Software\Vitro\Server\Vitro.Server.Workflow.Manager.Web" /e
```

10. Восстановить из директории резервного копирования директории с конфигурационными файлами, скопировав их рекурсивно с заменой в необходимые целевые расположения:

```
C:\Windows\system32>xcopy "C:\temp\Site" "C:\Program Files\Vitro Software\Vitro\Server\Site" /e  
C:\Windows\system32>xcopy "C:\temp\Conf" "C:\Program Files\Vitro Software\Vitro\Server\Conf" /e
```

11. Очистить содержимое директории резервного копирования:

```
del "c:\temp\bak"
```

12. Заменить системные файлы в файловом хранилище:

```
C:\Windows\system32>xcopy "C:\< >\Vitro.Server.MultiPlatform.Windows-xx.x.xx\DB\VitroFileStorage" "C:  
\Data\DB\VitroFileStorage" /e
```

13. Запустить задания автоматического запуска Веб-приложений vitro-server и vitro-workflow



При обновлении с версии 25.1.39.22 и ниже необходимо указать адрес портала в ключе `Site.PortalUrl`.

Для этого нужно:

- Перейти в список Конфигурация пространства Центр Администрирования.
- Открыть папку Общие настройки
- Для ключа Адрес портала указать в поле Значение (строка) адрес портала в виде <https://example.com>



Важно!
Адрес портала не должен заканчиваться символом слэша.

При обновлении с версии 24.1.37 и ниже до версий 24.1.38 и выше необходимо обновить путь до исполняемого файла модуля рабочих процессов, т.к. изменилось его имя на `Vitro.Server.Workflow.Manager.Web`.

Параметры обновленной задачи планировщика

```
" " : "Vitro Workflow Manager .Net Web Application"  
" " : "C:\Program Files\Vitro\Server\Vitro.Server.Workflow.Manager.Web\Vitro.Server.Workflow.  
Manager.Web.exe"  
: /urls http://localhost:4616  
" " C:\Program Files\Vitro\Server\Vitro.Server.Workflow.Manager.Web
```

В файле конфигурации *C:\Program Files\Vitro Software\Vitro\Server\Conf\workflow.json* нужно прописать логин /пароль учетной записи администратора:

```
"Login": "admin",  
"Password": "admin",
```

Пример:

```
"ConnectionString": "server=vm-vs-2022;database=workflow;user  
"BaseUrl": "http://localhost:4616",  
"PortalUrl": "http://localhost:4616",  
"BasePath": "/workflow/manager",  
"LoginPagePath": "",  
"SystemUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",  
"Login": "admin",  
"Password": "admin",
```

В ELSA импортировать новые планшеты рабочих процессов. Планшеты лежат в папке с дистрибутивом:
workflows\parallel-stage-workflow.json
workflows\sequential-stage-workflow.json

Процесс установки и настройки фермы конвертации Vitro Forge

Модуль Vitro BIM Viewer позволяет работать с информационными моделями. При размещении моделей в Vitro-CAD, на ферме конвертации Vitro Forge происходит конвертация файла в формат просмотра Vitro BIM Viewer.

В данном разделе описываются шаги по установке и настройке фермы конвертации:

1. Подготовить среду для фермы конвертации согласно требований к конфигурации. По требованию установить Revit и Navisworks.
 2. Установить RabbitMQ.
 3. Установить и настроить ферму конвертации.
 4. Добавить параметры конфигурации на сервере BimViewer.
- [Требования к ферме конвертации моделей для Vitro Bim Viewer.](#)
 - [Установка и запуск фермы конвертации моделей для Vitro Bim Viewer.](#)
 - [Установка RabbitMQ на Windows](#)
 - [Установка Vitro.Forge.Consumer на Linux](#)

Требования к ферме конвертации моделей для Vitro Bim Viewer.

1. Установить Navisworks Manage и Revit.

Примечание 1

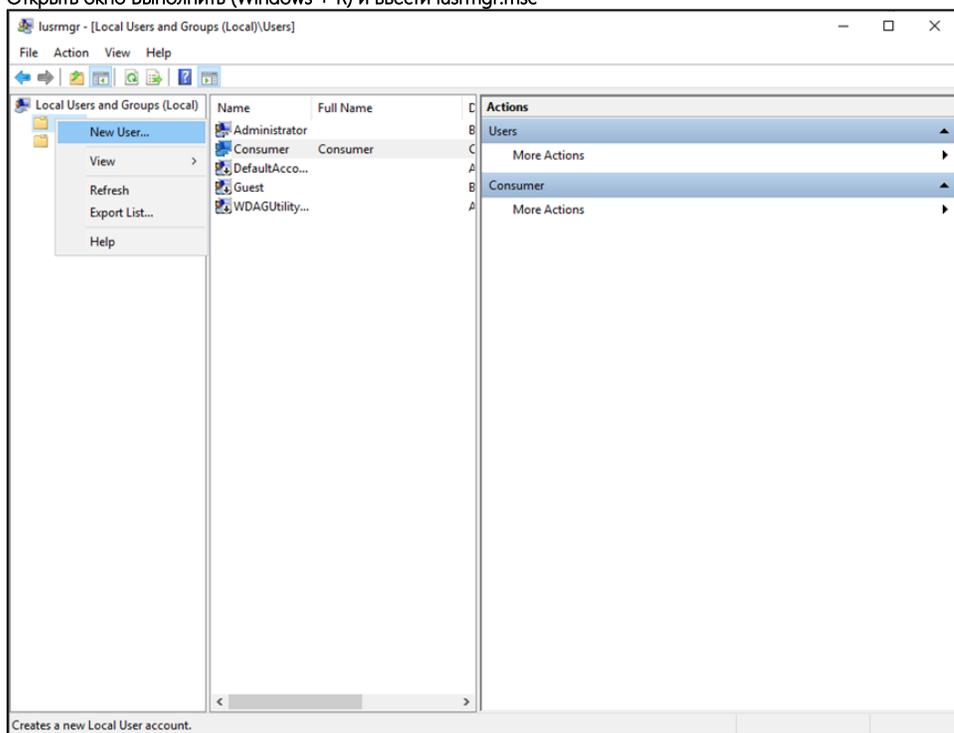
Необходимо выполнить установку всех используемых в компании версии Revit и Navisworks Manage.

Примечание 2

В текущей версии не поддерживается работа фермы с версией 2025. Поддержка будет добавлена в одном из следующих обновлений.

2. Добавить пользователя Windows.

- a. Открыть окно Выполнить (Windows + R) и ввести lusrmgr.msc



- b. В открывшемся окне в папке «Пользователи» выполнить «Новый пользователь...»

с. Ввести данные пользователя.

The image shows a 'New User' dialog box with the following fields and options:

- User name: Consumer
- Full name: Consumer
- Description: Consumer
- Password: [masked]
- Confirm password: [masked]
- User must change password at next logon
- User cannot change password
- Password never expires
- Account is disabled

Buttons: Help, Create, Close

3. Выполнить запуск под добавленной учетной записью для Revit и NavisWorks и выполнить активацию лицензии Autodesk продуктов для данного пользователя Windows.
4. Сделать чтобы сессия добавленного пользователя должна быть активна всегда, для этого необходимо внести соответствующие настройки в групповых политиках и настроить на машине конвертации:
 - a. Нажать Win + R, ввести gpedit.msc и нажать Enter, чтобы открыть редактор локальной групповой политики.
 - b. Перейти в следующий раздел:
 - Computer Configuration → Administrative Templates → Windows Components → Remote Desktop Services → Remote Desktop Session Host → Session Time Limits.
 - c. Изменить следующие параметры:
 - Set time limit for disconnected sessions – установить Never
 - Set time limit for active but idle Remote Desktop Services sessions – установить Never
 - End session when time limits are reached – установить Disabled.
5. Выполнить перезагрузку системы и выполнить вход от добавленного пользователя п.2

i Примечание 3

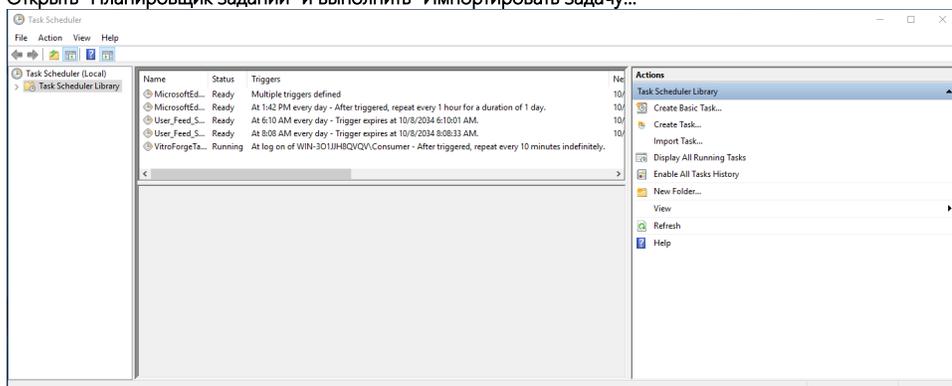
Опционально можно настроить автоматический вход в систему для пользователя. Для этого надо поправить реестр «HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon». Установите необходимые значения для параметров:

- AutoAdminLogon - 1
- DefaultUserName - имя пользователя
- DefaultPassword - пароль пользователя
- DefaultDomainName - имя домена, данное поле можно оставить пустым

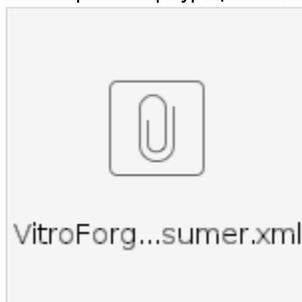
Установка и запуск фермы конвертации моделей для Vitro Vim Viewer.

1. Выполнить установку Vitro.Forge.MultiPlatform.Windows.Consumer.msi или Vitro.Forge.SharePoint.Windows.Consumer.msi
2. Добавить задание в планировщик заданий:

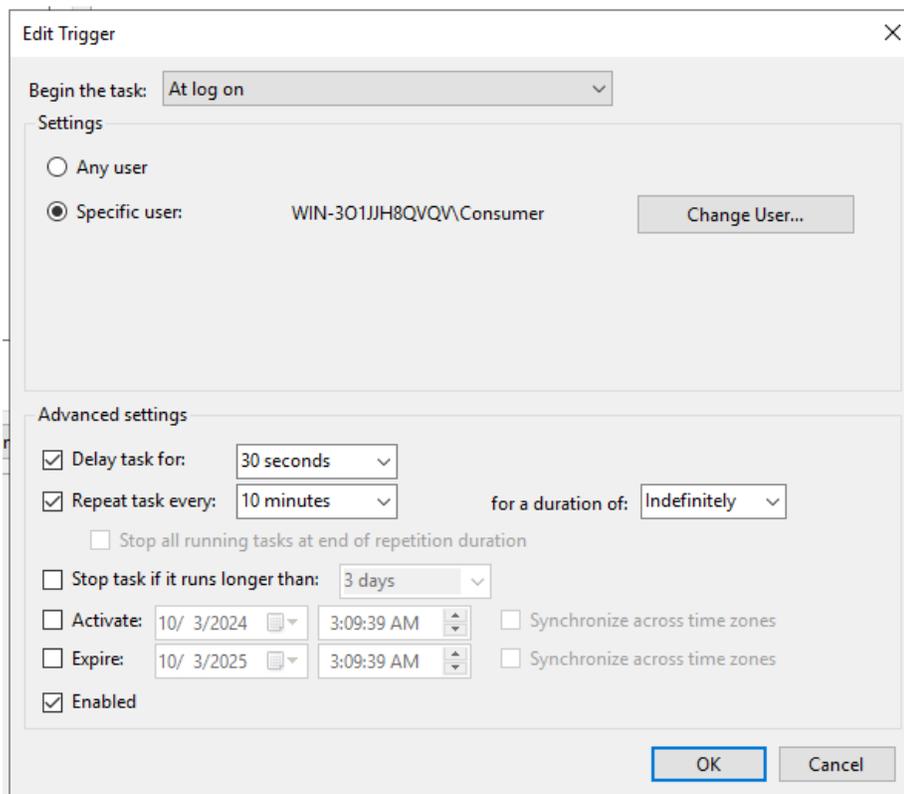
а. Открыть "Планировщик заданий" и выполнить "Импортировать задачу..."



б. Указать файл конфигурации VitroForgeMultiPlatformConsumer.xml.



- с. Перейти во вкладку "Триггеры", выбрать триггер и нажать кнопку "Изменить..."
- д. Заменить пользователя кнопкой "Сменить пользователя..." на добавленного пользователя. Если на машине конвертации у Revit и NavisWorks сетевая лицензия то изменить поле "Отложить задачу на:" чтобы задание запускалось после запуска сервера лицензии Autodesk.



- Открыть файл по пути установки Consumer (по умолчанию: «C:\Program Files\Vitro Software\Vitro\Forge») файл «Consumer\Conf\Consumer.json» и выполнить настройку файла.

```
{
  "WorkFolder": "C:\\VitroTemp", //путь кэша для работы Consumer.
  "ConsumerList": [
    /* Обработчик "ifc", "ifxml", "ifczip" */
    {
      "RabbitMQ": {
        "ConnectionString": "amqp://login:password@79.143.70.139:5672", //строка подключения к RabbitMQ
        "Queue": "IfcToXkt", //очередь которую будет слушать обработчик
        "Exchange": "Vitro",
        "ErrorQueue": "Error",
        "ErrorQueueKey": "Error"
      },
      "VitroSitePath": "http://79.143.70.139",
      "VitroUserName": "login",
      "VitroPassword": "password",
      "WorkingDirectory": "C:\\VitroTemp",
      "PathExe": "C:\\Program Files\\Vitro Software\\Vitro\\Forge\\MultiPlatform\\Consumer\\Bin\\Plugin\\IfcToXktConverter\\IfcToXktConverter.exe",
      "Arg": "{}",
      "Prefix": "",
      "ExtensionList": [ "xkt", "db", "lxkt", "xdb" ] // расширения файлов которое загрузится в систему Vitro
    },
    /* Обработчик ".nwd", "nwc", "nwf" */
    {
      "RabbitMQ": {
        "ConnectionString": "amqp://login:password@79.143.70.139:5672",
        "Queue": "NavisToXkt", //очередь которую будет слушать обработчик
        "Exchange": "Vitro",
        "ErrorQueue": "Error",
        "ErrorQueueKey": "Error"
      },
      "VitroSitePath": "http://79.143.70.139",
      "VitroUserName": "login",
      "VitroPassword": "password",
      "WorkingDirectory": "C:\\VitroTemp",
    }
  ]
}
```

```

"PathExe": "C:\\Program Files\\Vitro Software\\Vitro\\Forge\\MultiPlatform\\Consumer\\Bin\\Plugin\\AppStartup\\AppStartup.exe",
"Arg": "\\{0}\\",
"Prefix": "",
"ExtensionList": [ "xkt", "db", "lxt", "xdb" ]// расширения файлов которое загрузится в систему Vitro
},
/* Обработчик ".rvt", ".rvt", ".rte" */
{
  "RabbitMQ": {
    "ConnectionString": "amqp://login:password@79.143.70.139:5672",
    "Queue": "RevitToXkt",//очередь которую будет слушать обработчик
    "Exchange": "Vitro",
    "ErrorQueue": "Error",
    "ErrorQueueKey": "Error"
  },
  "VitroSitePath": "http://79.143.70.139",
  "VitroUserName": "login",
  "VitroPassword": "password",
  "WorkingDirectory": "C:\\VitroTemp",
  "PathExe": "C:\\Program Files\\Vitro Software\\Vitro\\Forge\\MultiPlatform\\Consumer\\Bin\\Plugin\\AppStartup\\AppStartup.exe",
  "Arg": "\\{0}\\",
  "Prefix": "",
  "ExtensionList": [ "xkt", "db", "lxt", "xdb" ]// расширения файлов которое загрузится в систему Vitro
},
{
  "RabbitMQ": {
    "ConnectionString": "amqp://login:password@79.143.70.139:5672",
    "Queue": "DwgToDxf",
    "Exchange": "Vitro",
    "ErrorQueue": "Error",
    "ErrorQueueKey": "Error"
  },
  "VitroSitePath": "http://79.143.70.139",
  "VitroUserName": "login",
  "VitroPassword": "password",
  "WorkingDirectory": "C:\\VitroTemp",
  "PathExe": "C:\\Program Files\\Vitro Software\\Vitro\\Forge\\MultiPlatform\\Consumer\\Bin\\Plugin\\DwgConverter\\DwgConverter.exe",
  "Arg": "\\{0}\\",
  "Prefix": "",
  "ExtensionList": [ "dxf" ]
}
]
}

```

- для Sharepoint версии Web сервера Vitro поменять "MultiPlatform" на "Sharepoint" в путях файлов.

Установка RabbitMQ на Windows

Установка RabbitMQ

1. На странице <https://www.rabbitmq.com/docs/which-erlang> посмотреть рекомендуемый Erlang и установить.
2. Скачать и установить RabbitMQ с <https://www.rabbitmq.com/install-windows.html>

Direct Downloads		
Description	Download	Signature
Installer for Windows systems (from GitHub)	rabbitmq-server-3.13.3.exe	Signature

3. Открыть RabbitMQ Command Prompt и выполнить команды:
 - a. Выполняем команду для установки Web-плагина:
rabbitmq-plugins.bat enable rabbitmq_management
 - b. Выполняем команду для включения трассировки:
rabbitmq-plugins.bat enable rabbitmq_tracing
4. Зайти на <http://localhost:15672/> логин guest пароль guest.
Если страница не доступна то вставить services.msc в диалоговом окне «Выполнить» и нажать Enter и перезапустить службу RabbitMQ.
5. Отключить consumer_timeout:
В файле C:\Users\{UserName}\AppData\Roaming\RabbitMQ\advanced.config вставить:

```
[  
  {rabbit, [  
    {consumer_timeout, undefined}  
  ]}  
].
```

Настройка RabbitMQ с помощью Web приложения

1. Добавить обменник Vitro:

The screenshot shows the RabbitMQ web interface with the 'Exchanges' tab selected. The 'All exchanges (9)' section displays a table of existing exchanges. Below this, the 'Add a new exchange' form is visible, showing the following configuration:

- Virtual host: /
- Name: Vitro
- Type: direct
- Durability: Durable
- Auto delete: No
- Internal: No
- Arguments: (empty)

2. Добавить очереди IfcToXkt, NavisToXkt, RevitToXkt и Error.

Overview Connections Channels Exchanges **Queues and Streams** Admin

Queues

▼ All queues (0)

Pagination

Page of 0 - Filter: Regex ?

... no queues ...

▼ Add a new queue

Virtual host:

Type:

Name: *

Durability:

Arguments: =

Add [Auto expire ?](#) | [Message TTL ?](#) | [Overflow behaviour ?](#)
[Single active consumer ?](#) | [Dead letter exchange ?](#) | [Dead letter routing key ?](#)
[Max length ?](#) | [Max length bytes ?](#)
[Leader locator ?](#)

a.

Overview Connections Channels Exchanges **Queues and Streams** Admin

Queues

▼ All queues (0)

Pagination

Page of 0 - Filter: Regex ?

... no queues ...

▼ Add a new queue

Virtual host:

Type:

Name: *

Durability:

Arguments: =

Add [Auto expire ?](#) | [Message TTL ?](#) | [Overflow behaviour ?](#)
[Single active consumer ?](#) | [Dead letter exchange ?](#) | [Dead letter routing key ?](#)
[Max length ?](#) | [Max length bytes ?](#)
[Leader locator ?](#)

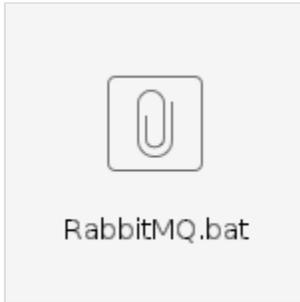
b.

- Открыть обменник Vitro и настроить привязки для очередей IfcToXkt, NavisToXkt, RevitToXkt, Error в поле "Routing key" задать IfcToXkt, NavisToXkt, RevitToXkt, Error соответственно

- Добавить пользователя обработчика очереди и настраиваем его права. Этот же логин и пароль нужно будет прописать в файле appsettings.json на сервере обработки.

- Если требуется включить трассировщика.

Настройка RabbitMQ с помощью bat файла



1. Для быстрой настройки скачать и установить python с сайта <https://www.python.org/downloads/>
2. Скачать RabbitMQ.bat и задать:
 - a. Путь до python.exe в строке `SET PYTHON_PATH = "C:\Users\Developer\AppData\Local\Programs\Python\Python312\python.exe"`
 - b. Путь до папки sbin в RabbitMQ Server в строке `SET RABBITMQ_SBIN_PATH="C:\Program Files\RabbitMQ Server\rabbitmq_server-3.12.8\sbin"`
 - c. Пользователя обработчика очереди. В строках `SET USER_NAME="admin"` и `SET PASSWORD="admin"`
Этот же логин и пароль нужно будет прописать в файле `C:\Program Files\Vitro Software\Vitro\Forge\SharePoint\Consumer\Conf\Consumer.json` на сервере обработки.
3. Выполнить RabbitMQ.bat.
4. Добавить пользователя обработчика очереди с помощью Web приложения.

Установка Vitro.Forge.Consumer на Linux

Для работы программы требуется Ubuntu с графической оболочкой

Добавление ifc конвертера

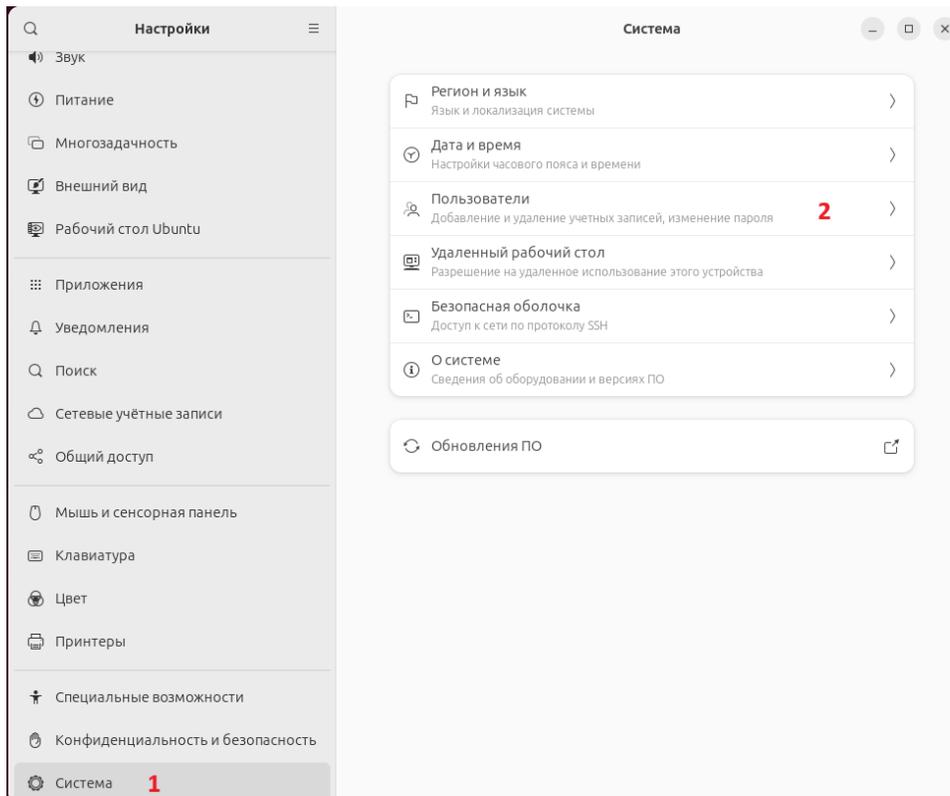
1. Установить wine от администратора через терминал

```
dpkg --add-architecture i386 \  
&& apt-get update \  
#  
&& apt-get install -qfy --install-recommends \  
software-properties-common \  
gnupg2 \  
wget \  
xvfb \  
cabextract \  
# Wine  
&& wget -nv https://dl.winehq.org/wine-builds/winehq.key  
\  
&& apt-key add winehq.key \  
&& apt-add-repository 'deb https://dl.winehq.org/wine-  
builds/ubuntu/ bionic main' \  
# Wine  
&& add-apt-repository ppa:cybermax-dexter/sdl2-backport  
\  
# Wine  
&& apt-get install -qfy --install-recommends \  
winehq-staging \  
winbind \  
#  
&& apt-get -y clean \  
&& rm -rf \  
/var/lib/apt/lists/* \  
/usr/share/doc \  
/usr/share/doc-base \  
/usr/share/man \  
/usr/share/locale \  
/usr/share/zoneinfo
```

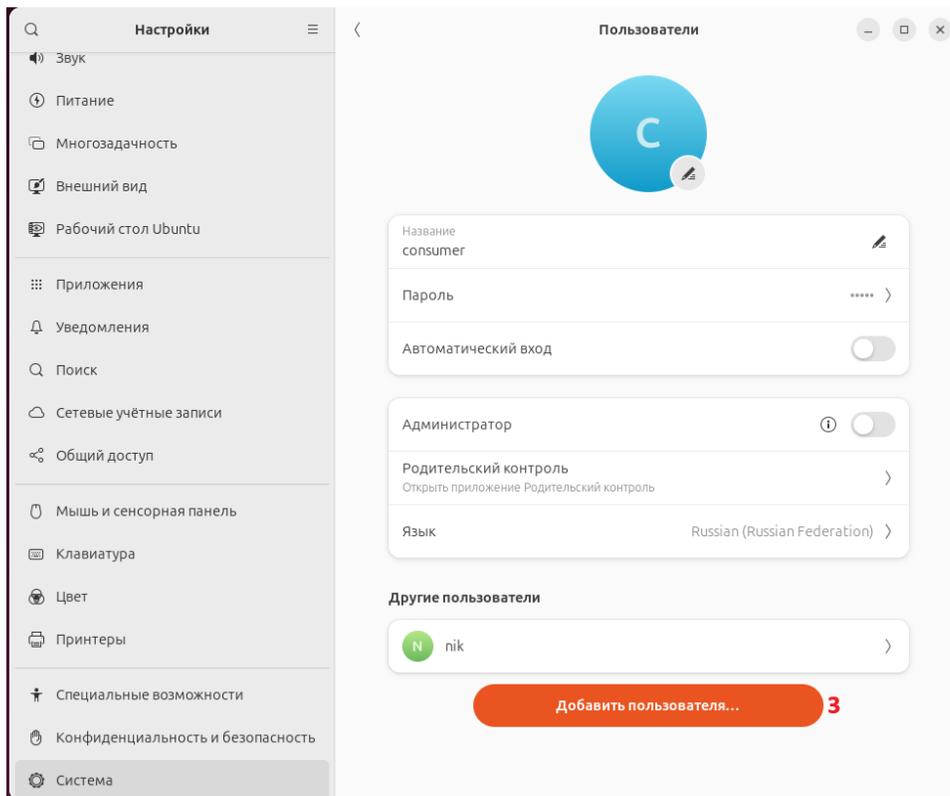
2. Добавить пользователя consumer

2.1. Выбрать В окне настройки "Система".

2.2. Кликнуть на поле "Пользователи"



2.3 Кликнуть на поле "Добавить пользователя"



2.4 Заполнить поля и нажать кнопку "Добавить"

Отменить Добавить пользователя Добавить

Название: consumer ✓

Имя пользователя: consumer | Данное имя пользователя недоступно. Попробуйте ввести другое. ⚠

Администратор:

Пароль

Пользователь устанавливает пароль при первом входе в систему

Установить пароль сейчас

Пароль: ✓

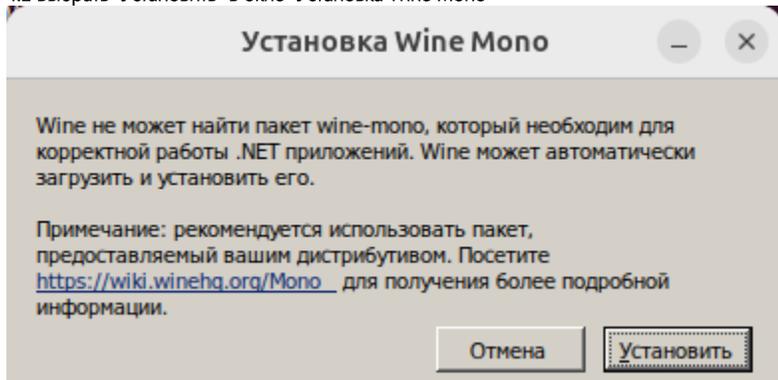
Подтвердить: ✓

Добавление большего количества букв, цифр и знаков препинания сделает пароль надежнее

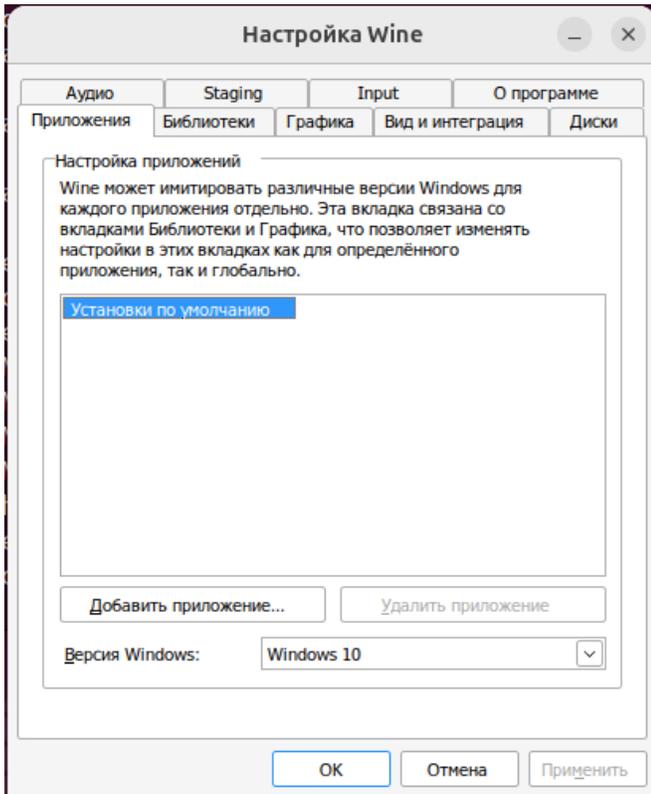
3. Войти в ubuntu под добавленным пользователем обязательно с графической оболочкой.
4. Выполнить конфигурацию wine

```
winecfg
```

- 4.1 Выбрать "Установить" в окне "Установка Wine Mono"



- 4.2. Кликнуть "OK" в окне "Настройка Wine"



5. Скачать winetricks с помощью терминал

```
wget https://raw.githubusercontent.com/Winetricks/winetricks/master/src/winetricks -O /home/consumer/winetricks
```

6. Установить необходимые компоненты

```
wineboot -u && winetricks -q dotnet472 && xvfb-run winetricks -q vcrun2015
```

7. Скачать Vitro.Forge.IfctoXktConverter.Linux

```
curl "https://linkTo/Vitro.Forge.IfctoXktConverter.Linux.zip" --output 'Vitro.Forge.IfctoXktConverter.Linux.zip'
unzip Vitro.Forge.IfctoXktConverter.Linux.zip
```

8. Скопировать папку приложения в желаемое расположение:

```
cp -r ~/Vitro.Forge.IfctoXktConverter.Linux /home/consumer/Vitro.Forge.IfctoXktConverter
```

Настройка Vitro.Forge.Consumer

Производить настройку от администратора системы:

1. Скачать и распаковать дистрибутив:

```
curl "https://linkTo/Vitro.Forge.Consumer.Linux.zip" --output 'Vitro.Forge.Consumer.Linux.zip'
sudo unzip Vitro.Forge.MultiPlatform.Consumer.Linux.zip
```

2. Скопировать папку приложения в желаемое расположение:

```
sudo cp -r ~/Vitro.Forge.Consumer.Linux /usr/local/bin/Vitro.Forge.Consumer
```

3. Установить атрибут "Исполняемый" для файла Consumer:

```
sudo chmod +x /usr/local/bin/Vitro.Forge.Consumer
```

```
sudo chmod +x /usr/local/bin/Vitro.Forge.Consumer/Consumer
```

4. Внести изменения в конфигурационный файл Consumer.json:

```
sudo nano /usr/local/bin/Vitro.Forge.Consumer/Consumer.json
```

- пример конфигурационного файла Consumer.json:

Json

```
{
  "WorkFolder": "/home/consumer/VitroTemp",
  "ConsumerList": [
    {
      "RabbitMQ": {
        "ConnectionString": "amqp://login:password@localhost:5672",
        "Queue": "IfcToXkt",
        "Exchange": "Vitro",
        "ErrorQueue": "Error",
        "ErrorQueueKey": "Error"
      },
      "VitroSitePath": "http://localhost",
      "VitroUserName": "login",
      "VitroPassword": "password",
      "WorkingDirectory": "/home/consumer/VitroTemp",
      "PathExe": "wine",
      "Arg": "/home/consumer/Vitro.Forge.IfctoXktConverter/IfctoXktConverter.exe \"%0\"",
      "Prefix": "",
      "ExtensionList": [ "xkt", "db" ]
    }
  ]
}
```

5. Создать и отредактировать файл модуля systemd для запуска службы:

```
sudo nano /lib/systemd/system/vitro_forge_consumer.service
```

- содержимое файла модуля:

```
[Unit]
Description=Consumer Service

[Service]
WorkingDirectory=/usr/local/bin/Vitro.Forge.Consumer
ExecStart=/usr/local/bin/Vitro.Forge.Consumer --urls http://localhost:4006
Restart=always
# Restart service after 10 seconds if the dotnet service crashes:
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=dotnet-example
User=consumer
SyslogIdentifier=Vitro-Forge-Consumer

[Install]
WantedBy=multi-user.target
```

6. Применить изменения и запустить службу:

```
sudo systemctl enable vitro_forge_consumer

sudo systemctl start vitro_forge_consumer
```

Установка и настройка клиентской части

Установка клиентской части

Клиентская часть системы Vitro Client устанавливается на локальные машины пользователей с ОС Windows.



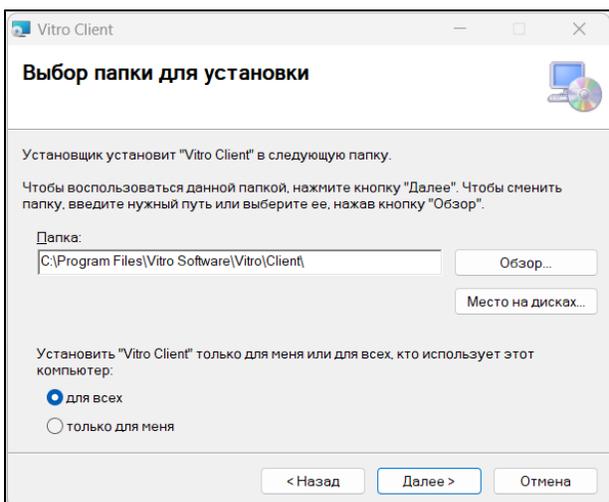
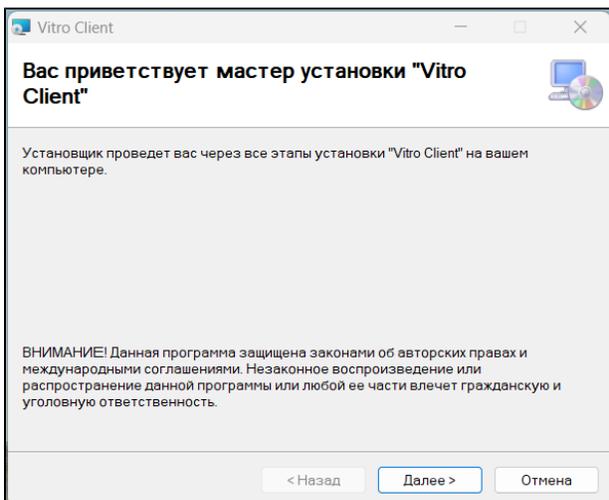
Важно

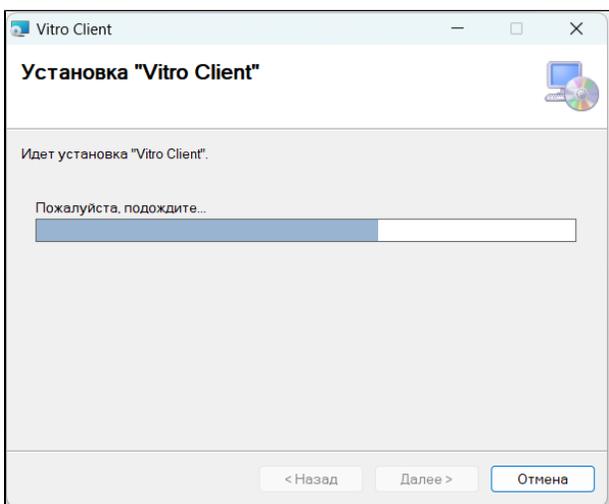
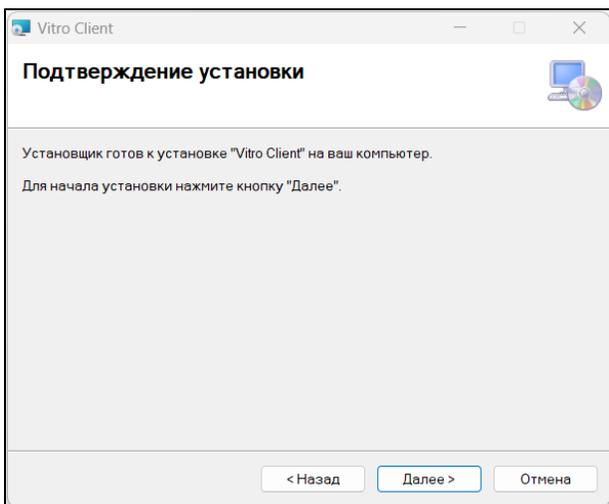
Перед выполнением установки Vitro Client проверить, что установлены компоненты:

- .NET Framework v4 (или выше)
- Visual Studio 2010 Tools for Office Runtime

Для установки Vitro Client необходимо выполнить следующие действия:

1. Скачать дистрибутив клиентской части. Запросить актуальный дистрибутив можно путем обращения в службу технической поддержки.
2. Произвести запуск пакета установки Vitro Client, щелкнув по файлу *Vitro.Client.Setup-XXX.X.X.XXXX.msi*, где [XXX.X.X.XXXX] - номер версии клиентской части
3. Следовать инструкциям мастера установки

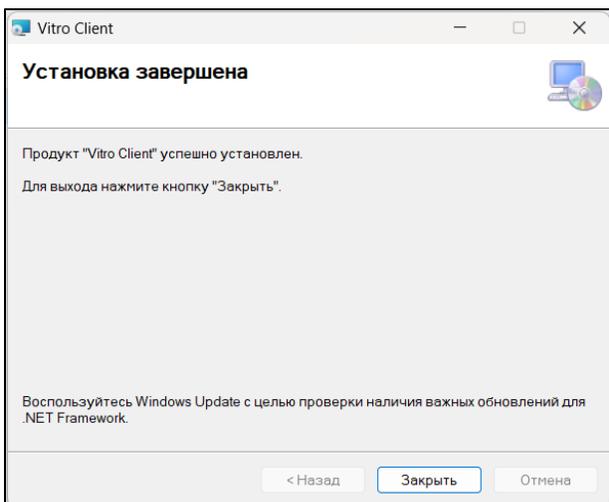




4. Заполнить и нажать клавишу ОК на форме настройки Vitro Client, которое откроется в ходе установки (как заполнить - см. раздел [Настройка клиентской части](#))



5. Закрыть мастер установки после завершения процедуры



Настройка клиентской части

Окно настройки Vitro Client

- открывается в процессе установки
- вызывается в любое время после завершения установки корректировок в настройку

В окне настроек следует провести следующую настройку:

1. Язык - установить язык интерфейса форм клиентской части
2. Сервер - указать адрес сервера системы, к которому будет происходить подключение
3. Локальное хранилище - задать расположение локального хранилища файлов системы



Важно

Директория локального хранилища должна располагаться на локальном диске ПК.

Всем пользователям компании настоятельно рекомендуется указывать один путь (например, D:\VitroLocal\).

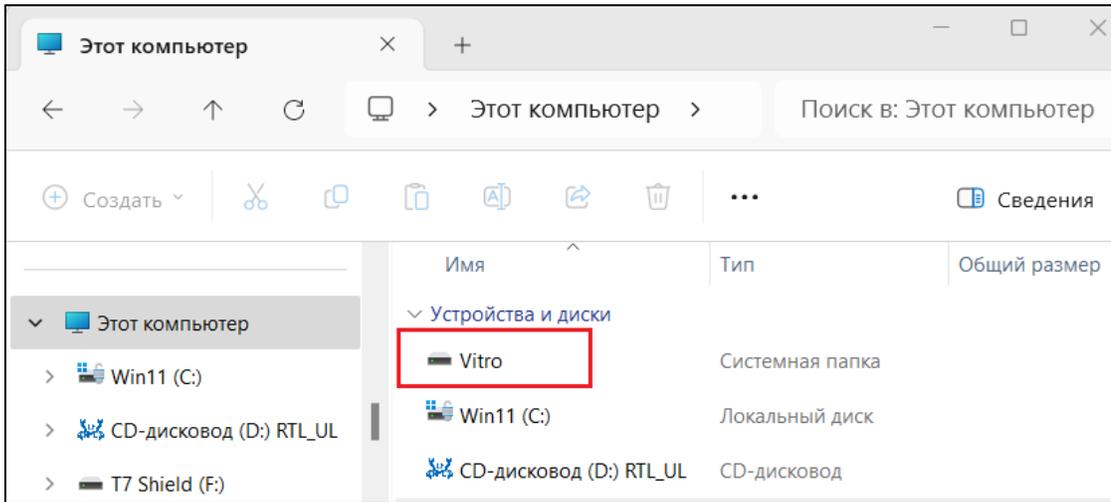
Путь не должен быть расположен внутри другого облачного хранилища (например, Яндекс.Диск).

Функциональные кнопки формы настройки:

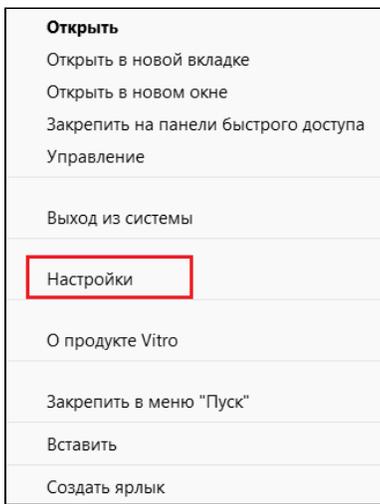
- Очистить локальное хранилище - позволяет очистить локальный кэш по указанному выше в настройках пути
- Установить плагины - показывает список CAD систем, из числа поддерживаемых, в которые можно установить плагины Vitro, если плагин еще не установлен по умолчанию

Для вызова окна настройки необходимо:

1. Открыть контекстное меню на диске Vitro на локальном компьютере правой кнопкой мыши



2. Открыть в контекстном меню Настройки



Обновление клиентской части

Для установки очередной версии приложения Vitro Client, необходимо произвести запуск пакета установки *Vitro.Client.Setup-XXX.X.X.YYYY.msi* из дистрибутива новой версии.

Обновление можно производить поверх предыдущей установки

Установка из командной строки в silent режиме

Для установки Vitro Client из командной строки в silent режиме нужно выполнить следующие действия:

1. В зависимости от разрядности операционной системы скачать клиент.
2. Запустить *Vitro.Client.Setup-XXX.X.X.YYYY.msi* со следующими параметрами:
 - /quiet - установка Vitro Client в silent режиме;
 - /forcerestart - по окончании установки будет выполнена перезагрузка;
 - SILENT=true - не будет показана форма настроек Vitro Client;
 - CAD-REGISTRY-ALL=true - будут установлены плагины для всех CAD систем;
 - URL=http://адрес_сервера- адрес сервера системы;
 - LICENSE=http://адрес_сервера_лицензии:порт- адрес сервера лицензии (при наличии);
 - LOCALE=ru-RU - язык. Возможные варианты: ru-RU, en-US;
 - CREDENTIAL-TYPE=fba - включить авторизацию на основе форм
 - LOCALSTORAGE=путь_до_локального_хранилища - путь до локального хранилища файлов системы. По-умолчанию C:\Users\<пользователь>\Documents\Vitro;

- TARGETDIR=путь_установки - куда устанавливать. По-умолчанию: *c:\Program Files\Vitro Software\Vitro\Client*;

Пример запуска установки в silent режиме:

`.\Vitro.Client.Setup-XXX.X.X.YYYY.msi /quiet SILENT=true URL=http://vitro LICENSE=http://vitro:1234 LOCALE=ru-RU`

Удаленная установка клиентской части

В данной статье рассмотрена удаленная установка и обновление Vitro Client с помощью групповой политики домена.

Установка Vitro Client

1. Для распространения клиентского ПО Vitro-CAD на компьютерах, входящих в домен Windows используются установочные пакеты (файлы с расширением .msi), входящие в комплект инсталлятора Vitro-клиент.
2. В процессе стандартной установки требуется ввод некоторых параметров, необходимых для корректной настройки клиента. Поскольку установка с помощью групповой политики не предусматривает передачу параметров инсталлятору, требуется предварительное добавление необходимых ключей и их значений в реестр на клиентской машине. Добавляемые ключи соответствуют параметрам описанным в статье [Установка клиентской части](#) в разделе Установка из командной строки в silent режиме.

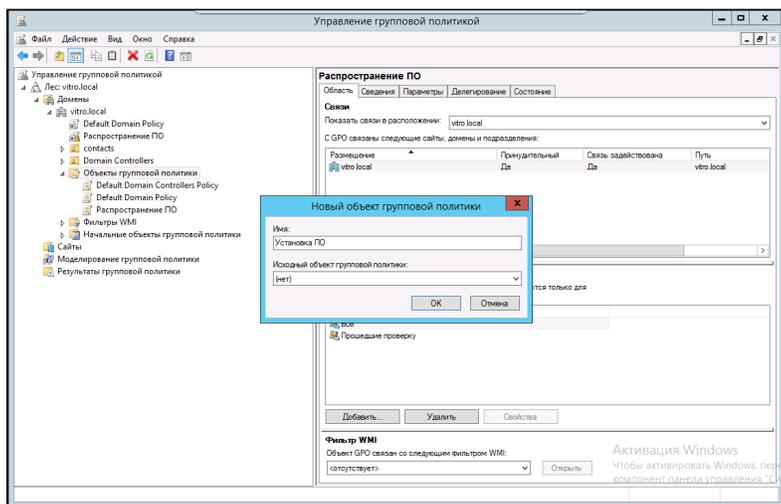
Пример настройки объекта групповой политики



Важно

Рекомендуется предварительно выполнить на клиентских ПК установку требуемых компонентов, указанных в статье [Установка клиентской части](#).

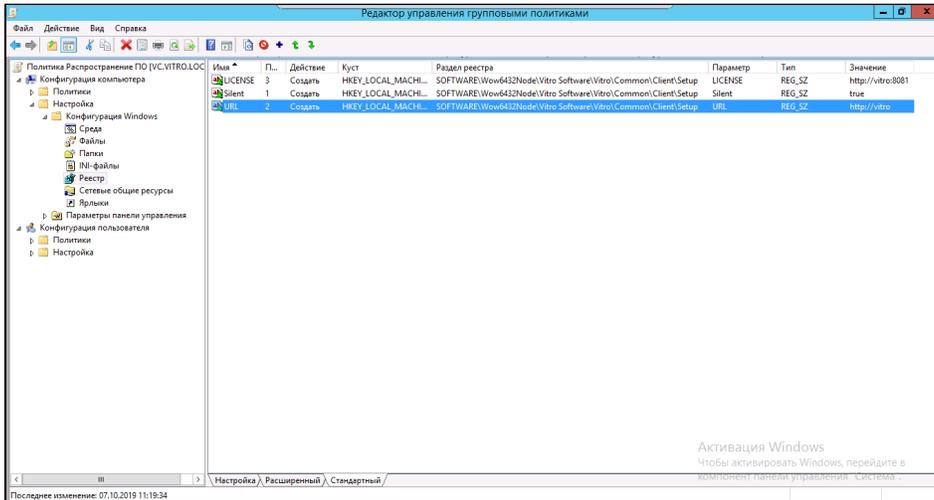
1. Создать новый объект групповой политики



2. Открыть объект групповой политики в редакторе
3. Перейти к конфигурации компьютера в раздел *Настройки - конфигурация Windows - Реестр*
4. Создать необходимые ключи в следующей ветви реестра:

```
reg
```

```
HKLM\SOFTWARE\Wow6432Node\Vitro Software\Vitro\Common\Client\Setup
```



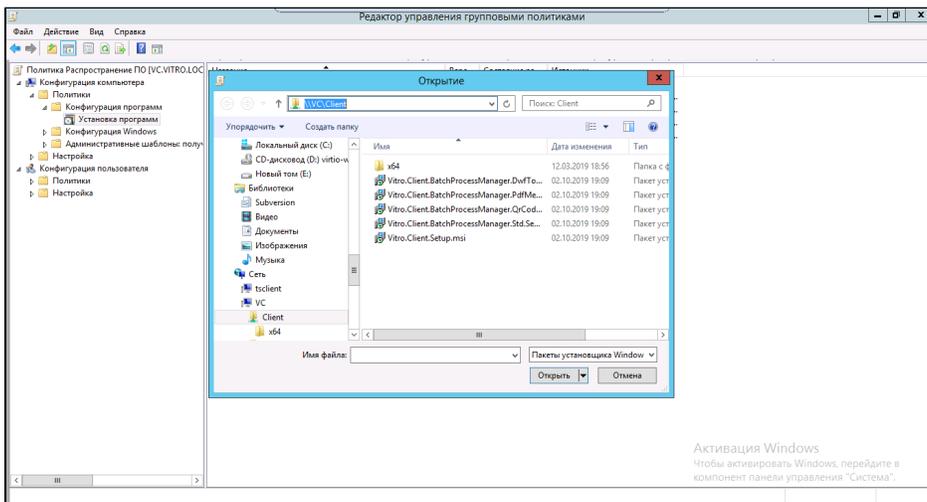
Минимальный набор ключей реестра для установки (пример на скриншоте):

- имя: "URL", тип: строка (string), значение: http://[FQDN_адрес_сервера_системы];
- имя: "LICENSE" (если используется), тип: строка (string), значение: http://[FQDN_адрес_лицензионного_сервиса]:[порт];
- имя: "Silent", тип: строка (string), значение: "True" (для "тихой" установки).

Полный список параметров см. [Установка клиентской части](#) в разделе Установка из командной строки в silent режиме

5. В редакторе групповой политики, в конфигурации компьютера перейти к разделу *Политики - Конфигурация программ - установка программ*

6. Добавить инсталляционные пакеты указав в обязательном порядке сетевое размещение



6. Сохранить объект политики, связать с необходимыми разделами AD

7. Указать, для каких групп безопасности применяется

При первой (после создания объекта политики) перезагрузке клиентского ПК будут добавлены параметры реестра, после следующей - будет произведена установка и настройка клиента Vitro.

Настройка объектов хранения данных

Элементы настройки объектов хранения

Структуру объектов хранения в Системе обеспечивают:

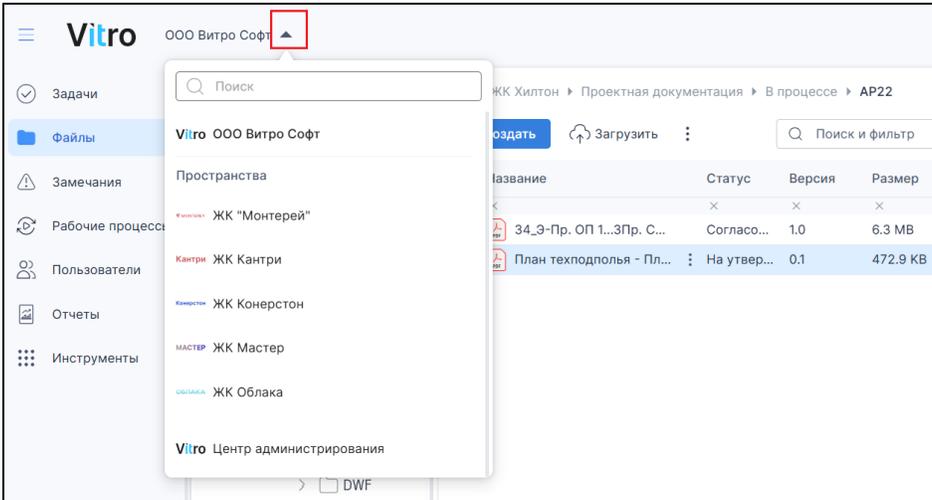
- Пространство/Пространства - изолированные рабочие зоны
- Списки
 - элементы списков строчного типа
 - элементы списков типа "папка" (для группировки элементов списка)
 - свойства списков
- Типы элементов (наборы атрибутов для элементов списков)
- Атрибуты (именованные базовые свойства, характеризующиеся типом данных)

В системе создаются Атрибуты, они объединяются в Типы элементов. Типы элементов присоединяются к Списку и формируют его облик

- [Пространства](#)
- [Списки - создание и настройка](#)
- [Типы элементов - создание и настройка](#)
- [Атрибуты - создание и настройка](#)
- [Сервис Вычисления](#)
- [Представления - создание и настройка](#)
- [Пример описания настройки структуры хранения документов](#)

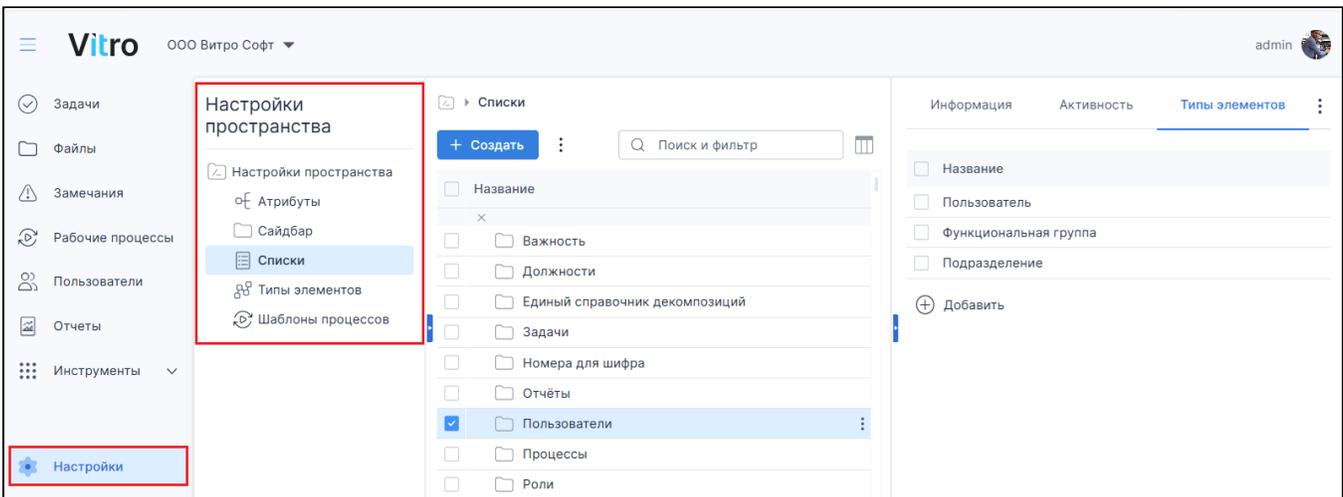
Пространства

1. Пространство - изолированная рабочая зона, в которой можно настроить свои списки и индивидуальный сайдбар
2. По умолчанию в Системе есть Основное пространство компании и пространство Центра администрирования, где можно создать дочерние пространства компании
3. Выбор пространств при работе осуществляется через выпадающий список - треугольничек рядом с названием текущего пространства в верхней части экрана



4. Настройки данных внутри пространства можно вызвать по ссылке "Настройки" из нижней части сайдбара

ВНИМАНИЕ! Атрибуты и Типы элементов настраиваются только на Основном пространстве



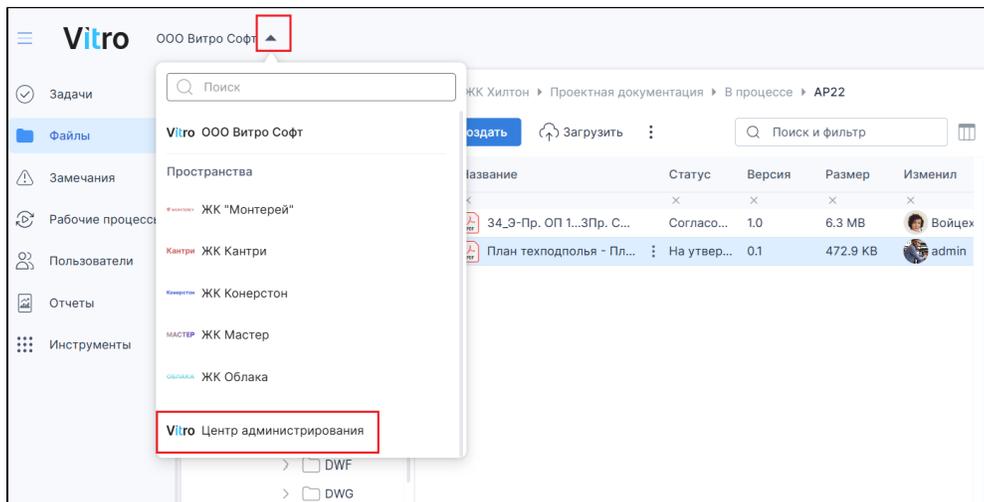
[Пространство - создание](#)

[Сайдбар - настройка](#)

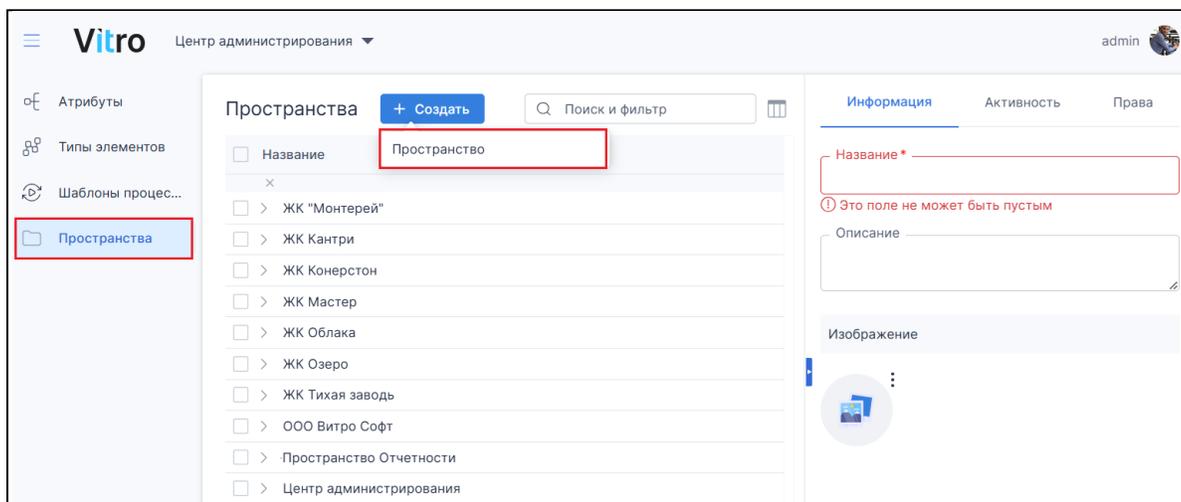
Пространство - создание

Для создания нового Пространства в Системе необходимо выполнить следующие действия:

1. Через выпадающий список пространств перейти в пространство Центр администрирования.



2. В открывшемся пространстве в разделе Пространства в перечне пространств нажать кнопку "Создать"



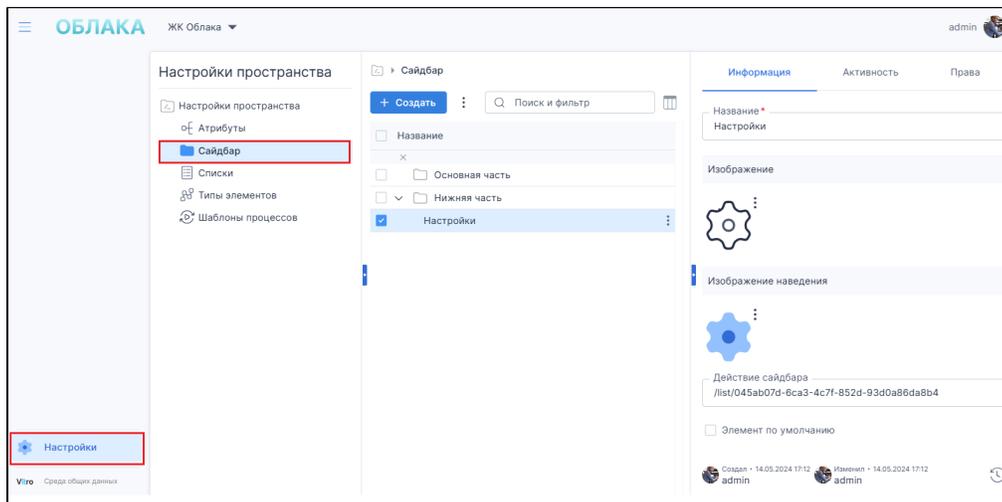
ВНИМАНИЕ! Название пространства не должно содержать кавычек

3. Заполнить справа свойства для нового пространства. Обязательный атрибут - название

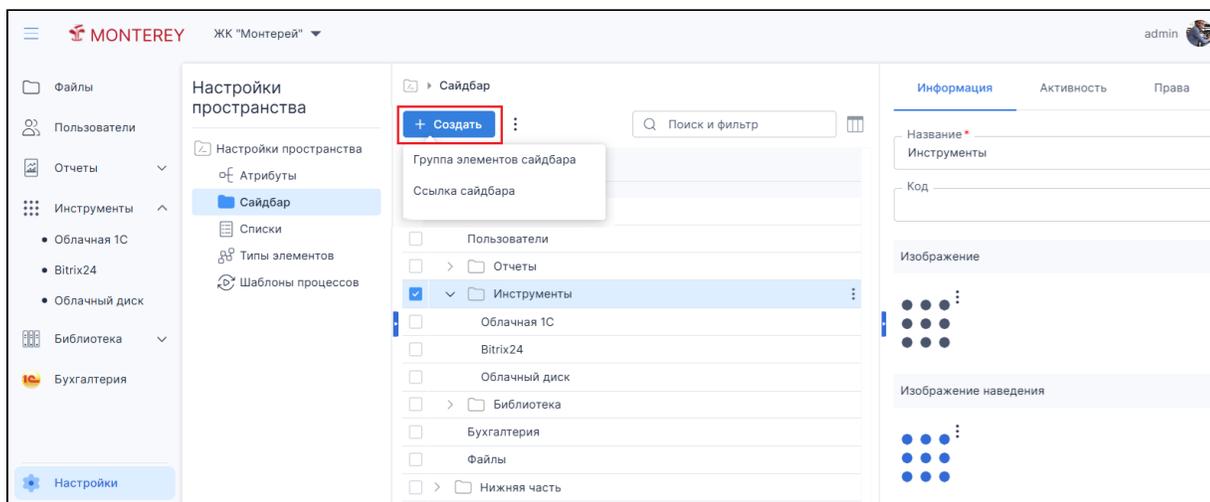
Сайдбар - настройка

Для любого пространства в Системе можно настроить сайдбар (меню быстрых ссылок, располагающееся с левой стороны экрана)

1. При первом входе в новое пространство в Системе открывается раздел настроек - единственный в сайдбаре нового пространства



2. Система позволяет делать иерархию быстрых ссылок. Для создания нового элемента сайдбара требуется нажать кнопку "Создать"



3. Для создания подзаголовка для группы ссылок требуется при создании:

- выбрать в выпадающем списке позицию "Группа элементов сайдбара"
- Дать название
- При желании подгрузить иконки

4. Для создания ссылки в левой части экрана требуется при создании:

- выбрать в выпадающем списке позицию "Ссылка сайдбара"
- Дать название
- При желании подгрузить иконки
- Указать адрес ссылки в поле "Действие сайдбара". Ссылка может быть как на страницу данных в Системе (берется относительный путь), так и на любой внешний ресурс

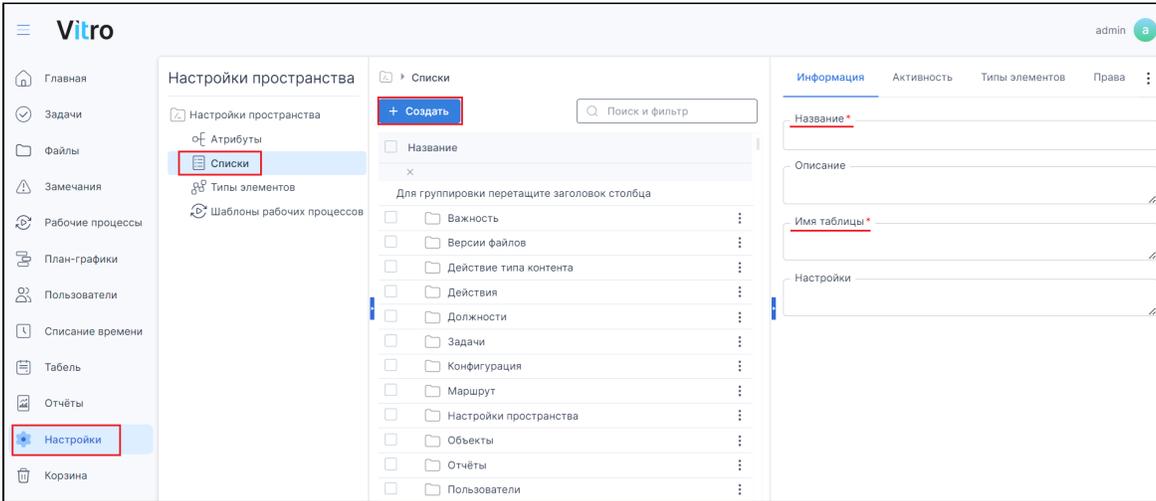
Списки - создание и настройка

Списки системы находятся в блоке Настройки разделе Списки. При двойном клике мышкой на название списка в центральной части можно попасть к содержимому списка.

Для создания в Системе нового Списка необходимо выполнить следующие действия.:

1. В настройках открыть раздел Списки и нажать кнопку Создать.
2. В открывшейся справа карточке настроек/свойств списка заполнить обязательные поля Название и Имя таблицы.

ВНИМАНИЕ! Имя таблицы надо заполнять по правилам Snake case: латинскими буквами с маленькой буквы, части названия разделяются символом подчеркивания (_)

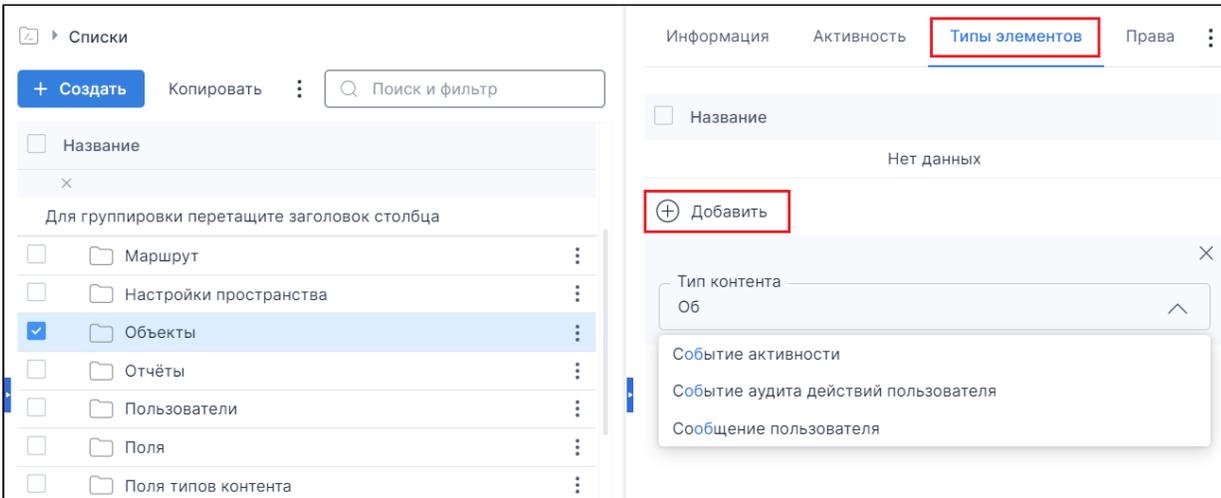


ВНИМАНИЕ! Рекомендуемые правила создания списков:

- Основная часть списков должна создаваться в основном пространстве
- В созданных пространствах могут быть созданы собственные списки, если они сильно уникальные – они автоматически привязываются к пространству
- Пользовательские названия списков, принадлежащих разным пространствам, могут быть неуникальными
- Если назначение списков по смыслу совпадают (например, список зданий/сооружений), то можно указать одно системное имя таблицы для разных - записи попадут в одну таблицу БД, и по ним потом проще строить запросы для отчетов
- Не надо объединять в таблицу с одним именем списки с потенциально большим (десятки тысяч записей) количеством записей

3. В карточке настроек на закладке Типы элементов добавить требуемые для списка типы элементов из системы:

- нажать кнопку Добавить
- выбрать требуемый тип элемента, используя контекстную подсказку
- повторить действия добавление нужное количество раз



4. В карточке настроек на закладке Права настроить права доступа к данному списку. Настройка прав описана [в отдельной статье](#)

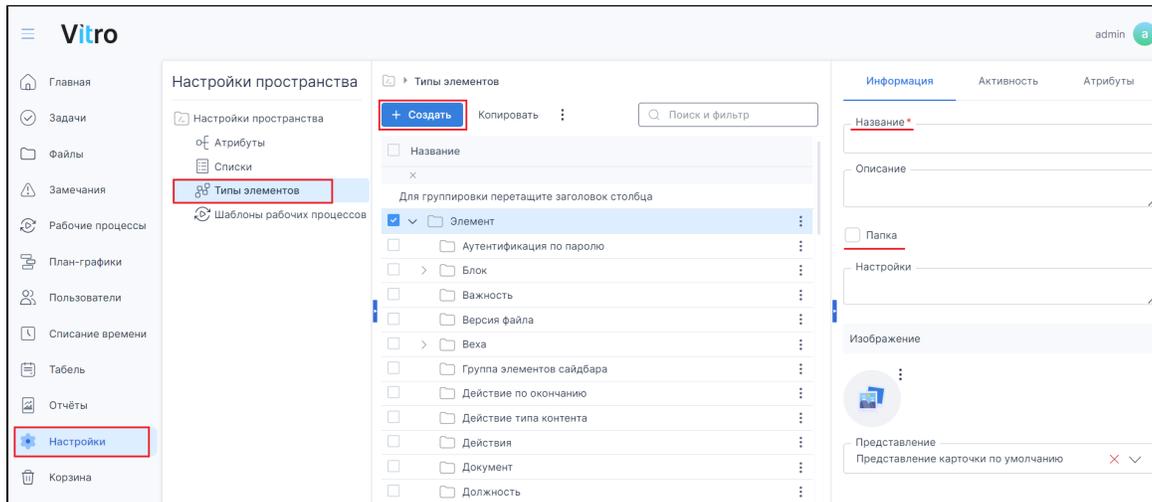
Типы элементов - создание и настройка

Для создания нового типа элемента в Системе необходимо выполнить следующие действия:

1. В настройках открыть раздел Типы элементов и нажать кнопку Создать.

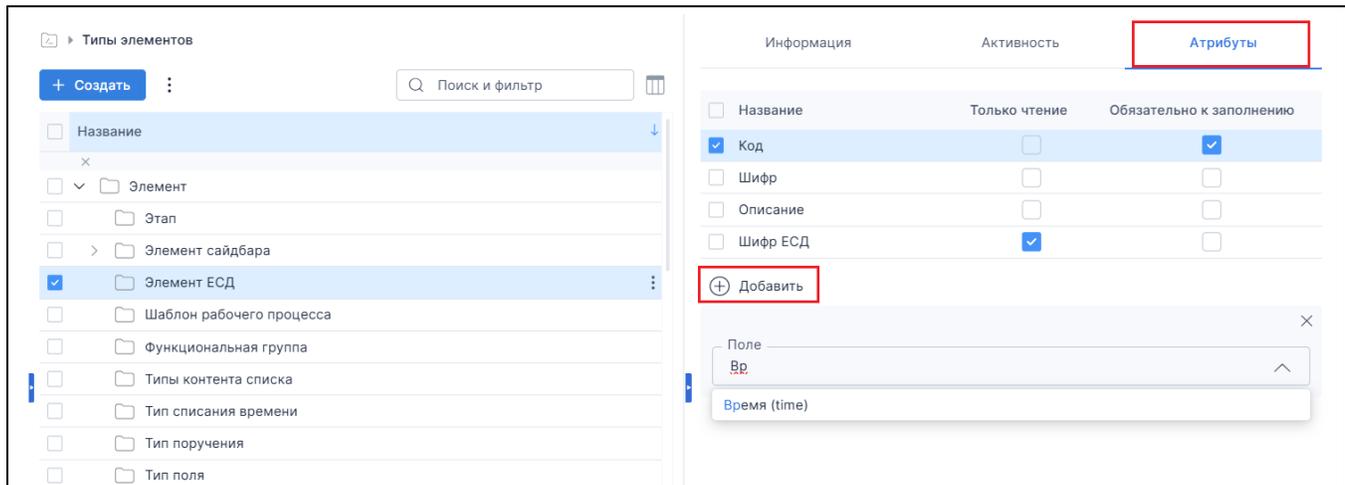
ВНИМАНИЕ! Рекомендуем выделить в табличной части тип элемента Элемент и потом нажимать кнопку создания. В этом случае создается дочерний тип элемента, который наследует атрибуты родительского.

2. В открывшейся справа карточке настроек/свойств типа элемента заполнить обязательное поле Название. Если тип элементов создается для папки, то отметить флаг Папка. При необходимости - задать иконку для отметки данного типа элемента в системе в разделе карточки "Изображение"



3. В карточке настроек на закладке Атрибуты добавить требуемые атрибуты из системы:

- нажать кнопку Добавить
- выбрать требуемый атрибут, используя контекстную подсказку
- отметить при необходимости режим "Только для чтения" (режим блокирует исправление значения атрибута пользователем вручную)
- отметить обязательность заполнения атрибута для данного типа элемента



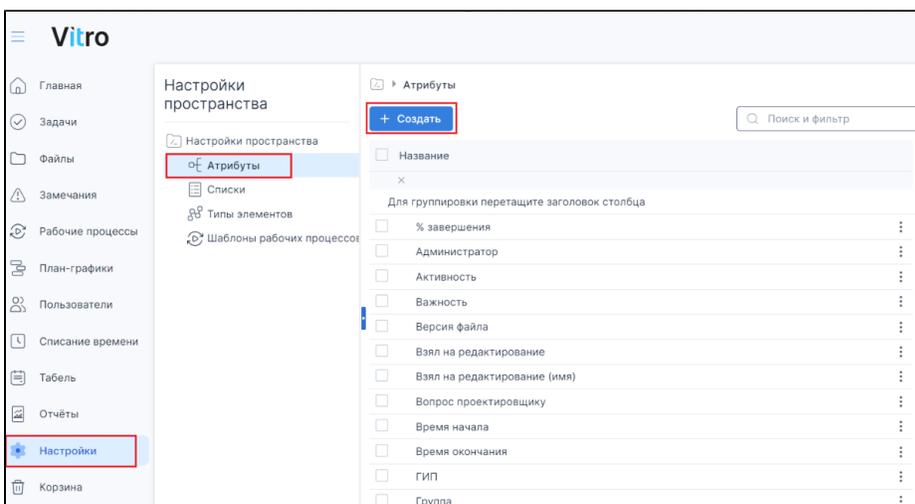
Атрибуты - создание и настройка

Атрибуты в Системе создаются в системе с использованием следующих типов данных:

1. Многострочный текст - для хранения многострочного текста. Тип столбца в базе данных: TEXT
2. Однострочный текст - для хранения однострочного текста. Максимальная длина: 255 символов. Тип столбца в базе данных: VARCHAR(255)
3. Число - для хранения целых чисел. Тип столбца в базе данных: INTEGER
4. Дата - для хранения даты. Тип столбца в базе данных: TIMESTAMP WITHOUT TIME ZONE
5. Время - для хранения времени. Тип столбца в базе данных: TIMESTAMP WITHOUT TIME ZONE
6. Флаг - для хранения булевского значения (да/нет). Тип столбца в базе данных: BOOLEAN
7. Ссылка на элемент списка - для хранения элементов, которые заданы в другом списке. Тип столбца в базе данных: UUID
8. Десятичное дробное число - для хранения дробных чисел. Используется для хранения точных значений. При сложении таких чисел не возникает ошибок, связанных с округлением. Точность: 4 знака после запятой. Тип столбца в базе данных: DECIMAL(18,4)
9. Число с плавающей точкой - для хранения дробных чисел. Точность: 15 знаков после запятой. Тип столбца в базе данных: DOUBLE PRECISION
10. Уникальный идентификатор - для хранения GUID'ов. Тип столбца в базе данных: UUID
11. Данные в json формате - для хранения данных в формате JSON. Тип столбца в базе данных: JSONB
12. Изображение - ссылка на картинку. Тип столбца в базе данных: UUID
13. Ссылка - для хранения URL. Тип столбца в базе данных: JSONB в формате {"name":"текст ссылки", "url":"ссылка"}
14. Вложения - ссылки на тела файлов. Тип столбца в базе данных: UUID[]

Для создания нового атрибута в Системе необходимо выполнить следующие действия:

1. В настройках открыть раздел Атрибуты и нажать кнопку Создать



2. Из выпадающего перечня выбрать тип данных для атрибута

3. В открывшейся справа карточке настроек атрибута заполнить обязательные поля Название и Системное имя - придумать самим

ВНИМАНИЕ! Системное имя надо заполнять по правилам Snake case: латинскими буквами с маленькой буквы, части названия разделяются символом подчеркивания (_). Системное имя не может быть в дальнейшем изменено.

Атрибуты

+ Создать

Поиск и фильтр

Название

Для группировки перетащите заголовок столбца

- % завершения
- Администратор
- Активность
- Важность
- Версия файла
- Взял на редактирование
- Взял на редактирование (имя)
- Вопрос проектировщику
- Время начала
- Время окончания
- ГИП

Информация

Активность

Название *

Системное имя *

Описание

Компонент

Отображаемые поля

Список

Мульти lookup

Минимальное количество символов для пои...

Рекомендуемые правила создания атрибутов:

- Пользовательские названия атрибутов желательно делать короткими
- Пользовательские названия атрибутов могут быть неуникальными
- Системные названия разных атрибутов должны быть уникальными
- Учитывать, что один и тот же атрибут при использовании в разных типах элементов ведет себя одинаково в соответствии со своими настройками

4. Для атрибутов с типом данных Ссылка на элемент списка (Lookup)

- в поле Список указать название связанного с атрибутом справочника
- отметить флаг Мульти Lookup, если надо выбирать несколько значений из справочника
- проставить число в поле Минимальное количество символов для поиска, для быстрого контекстного поиска по нескольким символам.

Сервис Вычисления

Сервис Вычисления - это универсальный механизм, который поддерживает различные операции с данными по заданной формуле для определенного элемента Vitro-CAD. В общем случае для выполнения вычислений применяются следующие данные:

1. Текущий элемент – это объект, который подлежит обработке. Обычно для идентификации сервис получает GUID элемента.
2. Формула – текстовое представление формулы, которая должна быть применена к элементу.
Описание синтаксиса формул: [Функции для формул вычисления_old](#)

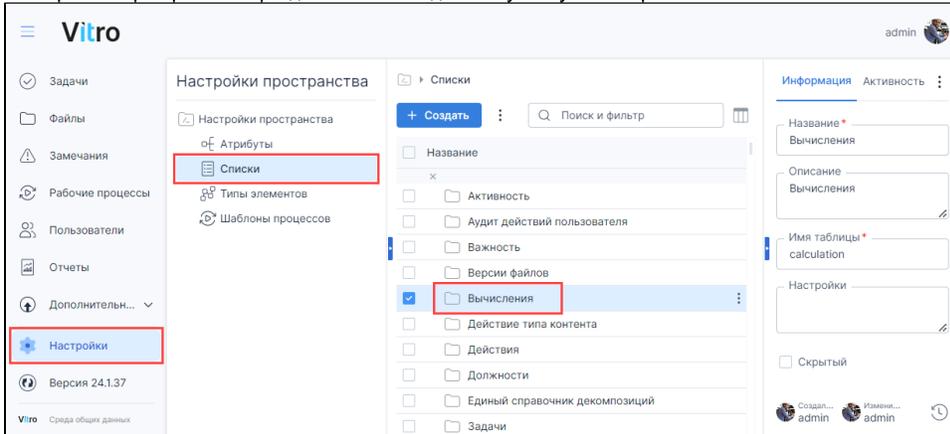
С помощью данного сервиса реализованы следующие функциональные возможности:

1. Автоматически заполняемые атрибуты (Вычисляемые поля) - позволяет автоматически заполнять атрибуты элементов в системе, используя заданные формулы.
Пример: вычисление сборного шифра с разных уровней структуры хранения и заполнение в определенный атрибут элемента.
[Вычисляемые поля - создание и настройка.](#)
2. Счетчики - особый случай вычисляемых полей, где при создании нового элемента происходит инкрементное изменение значения.
Пример: автоматическое присвоение порядкового номера проекта при создании нового элемента по правилам компании.
[Счётчики - создание и настройка.](#)
3. Синхронизация атрибутов - функция автоматической подстановки значений атрибутов из элементов системы в атрибуты файлов.
Пример: заполнение основных надписей чертежей DWG значениями из структуры хранения.
[Синхронизация атрибутов - настройка](#)

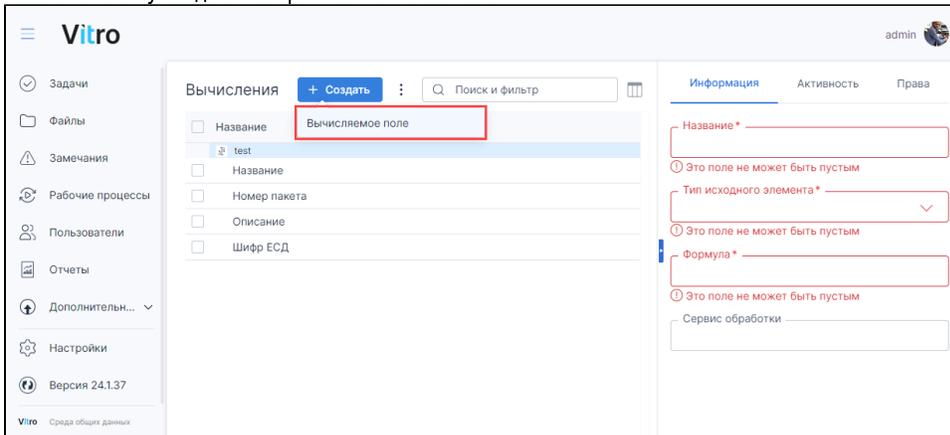
Вычисляемые поля - создание и настройка

Порядок создания вычисляемого поля

1. Определить тип элемента и его атрибут, для которого будет **выполняться вычисление** и заполнение значения. При необходимости **создать новый тип элемента** и **создать атрибут** для заполнения в результате вычисления.
2. Определить логику вычисления относительно обрабатываемого элемента. Какие атрибуты вычислить с самого элемента? Какие атрибуты вычислить с родительских элементов? Какие логические функции или функции преобразования данных понадобятся для вычисления?
3. В настройках пространства в разделе Списки по двойному клику ЛКМ перейти в список Вычисления.



4. Нажать на кнопку Создать и выбрать Вычисляемое поле.



5. В зоне свойств справа на закладке Информация заполняем поля:

- Название - обязательное поле - указать Название или Системное имя атрибута, в который должно быть записано вычисленное значение.
ВНИМАНИЕ! Указанный атрибут должен быть добавлен в тип элемента заранее. (см. [Атрибуты - создание и настройка](#)).
- Тип исходного элемента - обязательное поле - выбрать Тип элемента, для которого будет производиться вычисление.
- Формула - обязательное поле - указать формулу в соответствии с определенной логикой вычисления (см. [Функции для формул вычисления_old](#)).
- Сервис обработки - резерв места под развитие функциональных возможностей с вызовом REST сервисов. *Продолжение следует..*

6. После сохранения элемента проверить работу вычисления.

Пример настроек Вычисляемого поля

При добавлении или изменении элемента с типом Документ происходит вычисление по указанной формуле и полученное значение возвращается в атрибут Шифр документа.

Vitro admin

Задачи

Файлы

Замечания

Процессы

Пользователи

Отчеты

Дополнительно...

Отчет по проекту

Настройки

Справка об учетных данных

ЖК Монтерей 03_РД 03_АР В процессе AP11

Создать Загрузить Открыть Поиск и фильтр

Название	Статус	Версия	Размер	Изменил
417-AP-01.dwg	Отклоне...	0.3	476.6 KB	Левитин А
417-AP-01.pdf	Отклоне...	0.2	236.4 KB	Левитин А
417-AP-02.dwg	Согласо...	1.0	180.0 KB	Левитин А

Информация Активность Версии

Название *
417-AP-01.dwg

Статус документа
Отклонено

Шифр документа
ЖК Монтерей-03_РД-03_АР/417-AP-01.dwg

Пример настройки Вычисляемого поля

Название *
Шифр документа

Тип исходного элемента *
Документ

Формула *
Parent(Проект;[Название])+'-'+Parent(Стадия;[Название])+'-'+Parent(Комплект;[Название])+'/'+'Тис([Название])

Сервис обработки

Для быстрой загрузки перенесите файлы на экран

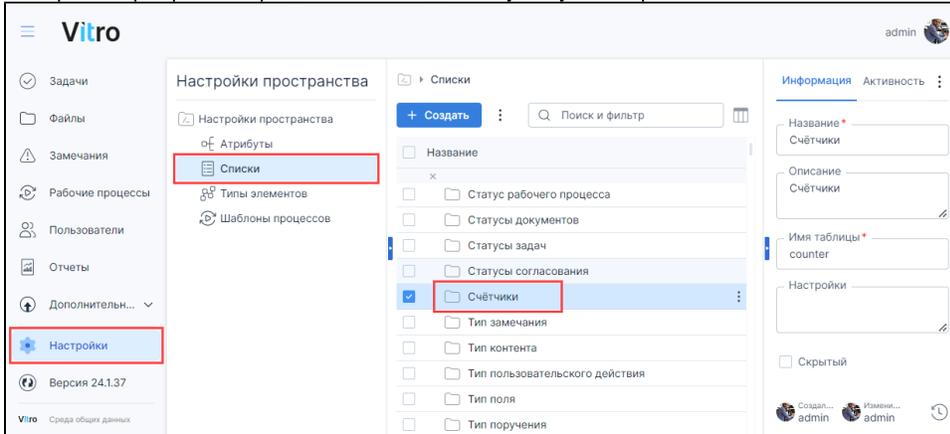
Создан - 11.04.2024 09:37 Левитин А. Изменен - 26.04.2024 13:17 Левитин А.

Счётчики - создание и настройка

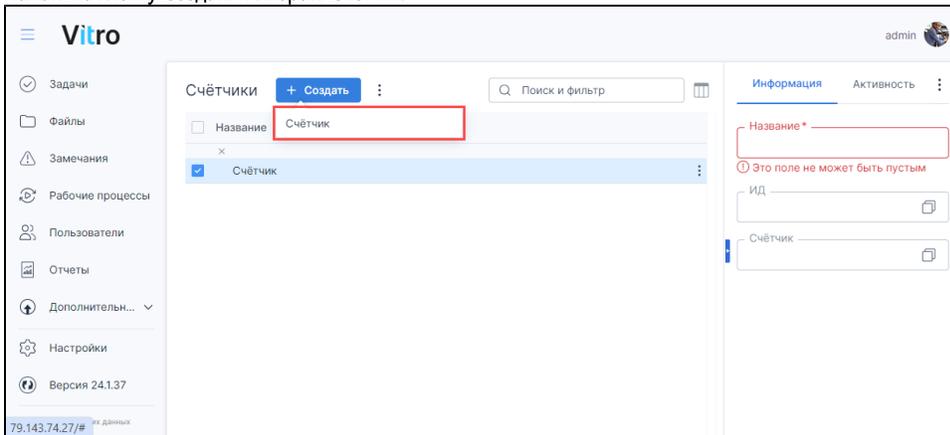
Счётчики используются для автоматической нумерации элементов с помощью сервиса Вычисления.

Порядок создания счётчика

1. Определить тип элемента и его атрибут, для которого будет выполняться вычисление и заполнение значения. При необходимости [создать новый тип элемента](#) и [создать атрибут](#) для заполнения в результате вычисления.
2. Определить логику вычисления Счётчика.
3. В настройках пространства в разделе Списки по двойному клику ЛКМ перейти в список Счётчики.



4. Нажать на кнопку Создать и выбрать Счётчик.



5. В зоне свойств справа на закладке Информация заполняем поля:

- Название - обязательное поле - указать название Счётчика, это описание назначения данного Счётчика.
- ИД - только для чтения - отображается порядковый номер Счётчика, на который будет ссылаться формула Вычисления для функции Inc().
ВНИМАНИЕ! Запомнить этот ИД для настройки на следующем шаге.
- Счётчик - только для чтения - хранится текущее значение Счётчика, которое будет получено при следующем вычислении.

6. В настройках пространства в разделе Списки по двойному клику ЛКМ перейти в список Вычисления и создать Вычисляемое поле с использованием функции Inc() для Счётчика.

Формат функции для Счётчика: Inc('Счётчики.Счётчик.ИД'), где ИД - значение атрибута ИД элемента списка Счётчики, с которым ведется работа при вычислении. Возможно использование вместе с другими функциями вычислений (см. [Вычисляемые поля - создание и настройка](#)).

7. После сохранения элемента проверить работу вычисления.

Пример настроек Счётчика

В примере на изображении ниже при создании папки с типом элемента Проект вычисляется следующее по порядку значение Счётчика в 4-значном формате и возвращается в атрибут Шифр проекта (авто).

Vitro admin

Задачи
Файлы
Замечания
Рабочие процессы
Пользователи
Отчеты
Дополнительн...
Настройки
Версия 24.1.37
Vitro Среда общих данных

Вычисления + Создать Поиск и фильтр

- Название
- test
- Название
- Номер пакета
- Описание
- Шифр ЕСД
- Шифр проекта (авто)

Информация Активность Права

Название *
Шифр проекта (авто)

Тип исходного элемента *
Проект

Формула *
Format('0:0000');Inc("Счётчики.Счётчик.1")

Сервис обработки

Создал • 27.04.202... admin
Изменил • 27.04.20... admin

Синхронизация атрибутов - настройка

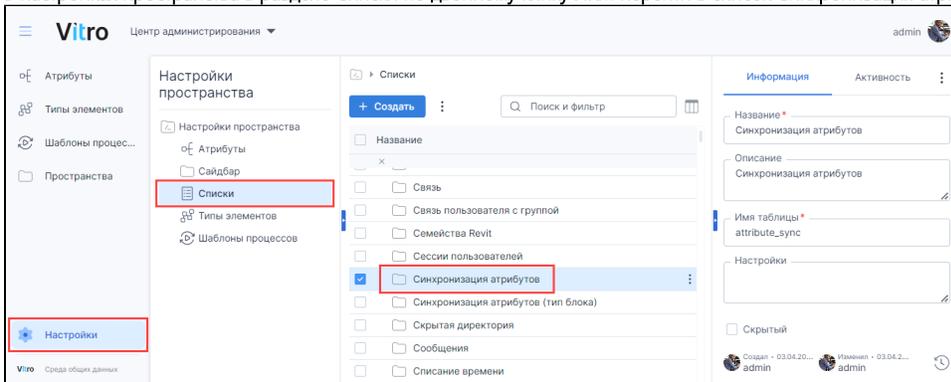
Под синхронизацией атрибутов понимается обновление свойств листов (Sheet), подшивок (SheetSet), атрибутов вхождений блоков (BlockReference) у файлов чертежей, текстовых полей Word и Excel, а так же параметров сведений о проекте Revit (Property), со значениями определенных полей (атрибутов) структуры хранения Vitro-CAD.

Для Vitro-CAD на платформе Vitro MP поддержка работы с Синхронизацией атрибутов доступна в Vitro Client версии 23.1.0.2309

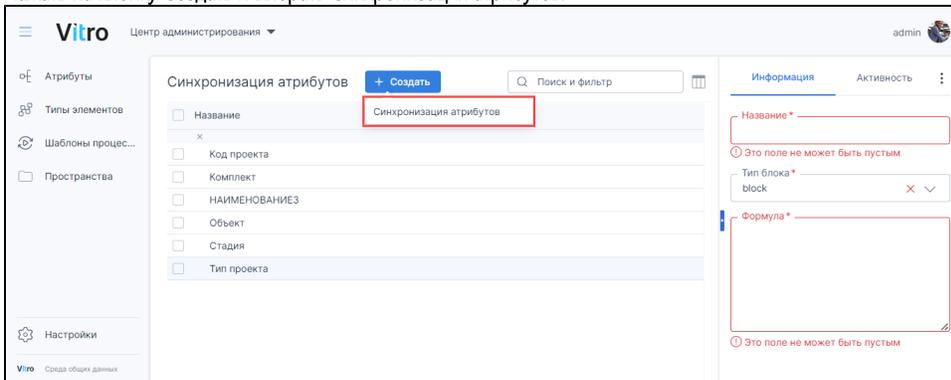
Правила синхронизации атрибутов настраиваются в списке Синхронизация атрибутов.

Порядок создания правила синхронизации атрибутов

1. Определить тип элемента и его атрибут, для которого будет **выполняться вычисление** для подстановки в свойства файла при синхронизации атрибутов. При необходимости **создать новый тип элемента** и **создать атрибут** для заполнения в результате вычисления.
2. Определить логику вычисления относительно обрабатываемого элемента. Какие атрибуты вычислить с самого элемента? Какие атрибуты вычислить с родительских элементов? Какие логические функции или функции преобразования данных понадобятся для вычисления?
3. Подготовить шаблон файла (DWG, DOCx, XLSx, RVT) для которого будет настраиваться синхронизация атрибутов. Как правило, пользуются существующими шаблонами, которые есть в организации.
4. Определить тип элемента и его наименование (тег) внутри шаблона файла, в который будет возвращаться вычисленное значение.
5. В настройках пространства в разделе Списки по двойному клику ЛКМ перейти в список Синхронизация атрибутов.



6. Нажать на кнопку Создать и выбрать Синхронизация атрибутов.



7. В зоне свойств справа на закладке Информация заполняем поля:

- Название * - обязательное поле - указать тег атрибута (при отсутствии тега используется наименование атрибута) из шаблона файла, в который должно быть записано вычисленное значение.
- Тип блока - обязательное поле - выбрать Тип блока атрибута, в который должно быть записано вычисленное значение.

block - значение будет вставлено в атрибут блока (DWG, DOCx, XLSx)

sheet - значение будет вставлено в свойство листа подшивки, которому соответствует открытый документ

sheetset - значение будет вставлено в свойство подшивки, которой принадлежит открытый документ

- **property** - значение будет вставлено в свойство проекта или семейства Revit, которому соответствует открытый документ

• Формула - обязательное поле - указать формулу в соответствии с определенной логикой вычисления (см. [Функции для формул вычисления_old](#)).

8. После сохранения элемента выполнить перезапуск процесса проводника на рабочем месте пользователя и проверить работу синхронизации атрибутов.



* Блоки DWG для синхронизации в настройках и в шаблонах **должны быть помечены** при помощи строк с зарезервированными служебными значениями **Block.Name**, **Block.Attr** или **Block.Layer**.

Использование этих значений играет роль фильтров при обработке атрибутов блоков файла чертежа. Особенно это актуально при использовании групповой синхронизации атрибутов.

При использовании значения **Block.Name**, в поле Формула необходимо указать имя блока. Синхронизация атрибутов будет выполняться только для блоков с указанным именем.

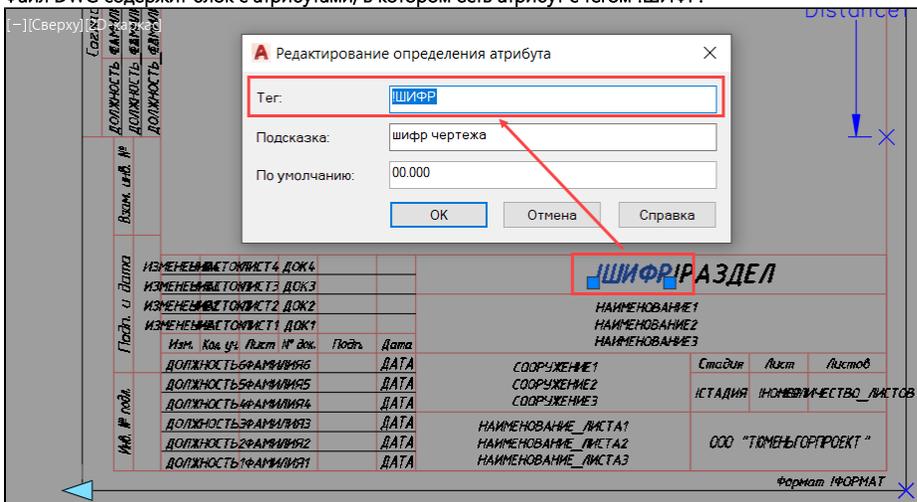
При использовании значения **Block.Attr**, в поле Формула необходимо указать имя атрибута блока. Синхронизация атрибутов будет выполняться только для блоков, у которых имеется атрибут с указанным именем.

При использовании значения **Block.Layer**, в поле Формула необходимо указать имя слоя. Синхронизация атрибутов будет выполняться только для блоков, которые находятся на слое с указанным именем.

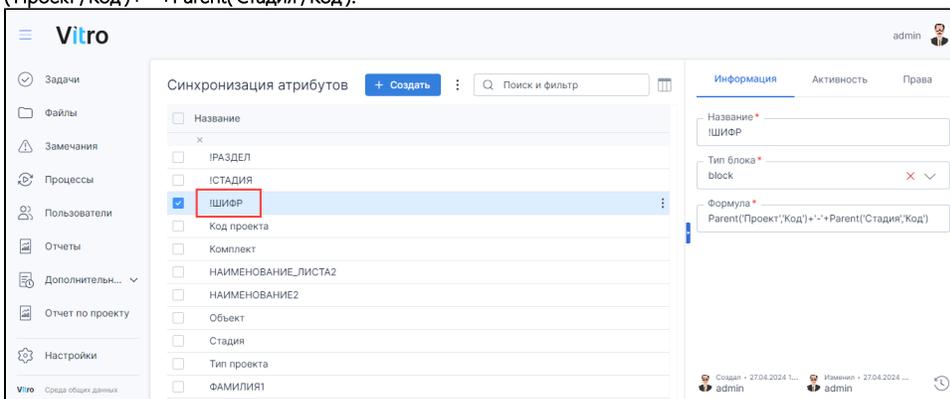
Если эти значения не указать, то пользователю нужно будет задавать имя блока в AutoCAD / nanoCAD при выполнении Синхронизации атрибутов.

Пример настроек Синхронизации атрибутов

1. Файл DWG содержит блок с атрибутами, в котором есть атрибут с тегом !ШИФР.



2. Настроено правило для заполнения атрибута блока DWG типа block с тегом !ШИФР значением, которое вычисляется по формуле Parent('Проект','Код')+'-'+Parent('Стадия','Код').



3. В результате при вызове в AutoCAD / nanoCAD в плагине Vitro действия Синхронизировать атрибуты вычисляется значение из родительских папок с типами Проект (значение атрибута Код) и Стадия (значение атрибута Код) и это значение возвращается в атрибут блока этого DWG с тегом !ШИФР.

Функции для формул вычисления

Функции и примеры вызова

В таблице ниже представлен список доступных функций, пример их вызова и описание полученного результата.

Примечание: необязательные параметры указаны в квадратных скобках - []

Тип функции	Функция	Описание	Входные параметры	Выходные параметры	Пример	Результат вычисления примера
Конфигурация	Cfg(key)	Возвращает значение элемента из списка "Конфигурация" по ключу	key: значение атрибута "Системное имя" элемента из списка "Конфигурация"	Возвращаемое значение соответствует типу найденного по ключу элемента (строка, число, флаг, картинка)	Cfg('Site.PortalUrl')	Вернет значение элемента конфигурации по пути: "Общие настройки" → "Адрес портала"
Структура хранения данных	Field(element, fieldName, checkParent, includeCurrent)	Возвращает значение заданного атрибута у заданного элемента. При отсутствии заданного элемента, значение атрибута будет считано с текущего элемента	element: строка (необязательный) - Название или тип элемента fieldName: строка (обязательный) - Название атрибута checkParent: bool (необязательный) - Проверять родительский элемент includeCurrent: bool (необязательный, используется только при checkParent=True) - Проверять текущий элемент	Значение поля	Field(Parent('В процессе'), 'Название')	Вернет значение атрибута "Название" родительской папки с типом элемента или названием "В процессе"
	Lookup(listName, elementName)	Возвращает заданный элемент, заданного списка.	listName: строка (обязательный) - Название списка elementName: строка (обязательный) - Название элемента	Элемент списка	Lookup('Статусы документов.Размещено')	Вернет элемент с названием "Размещено" списка "Статусы документов"
	Parent(element, name, includeCurrent)	Вернет первый объект удовлетворяющий заданным условиям, являющийся родительским по отношению к текущему или заданному.	element: объект (необязательный) - элемент, для которого будет выполнен поиск родительского элемента. name: строка (необязательный) - Название типа искомого элемента или название самого элемента includeCurrent: bool (необязательный) - проверять ли сам текущий элемент.	Элемент или значение заданного поля элемента	<ol style="list-style-type: none"> Parent() Parent('В процессе') Parent(Ref('Связь с замечанием'), 'Проект') Parent('Стадия', true) 	<ol style="list-style-type: none"> Вернет первый родительский элемент текущего элемента Вернет первый родительский элемент с названием или типом элемента "В процессе" Вернет первый родительский элемент с типом элемента "Проект" для документа связанного с текущим замечанием. Вернет элемент первый элемент с типом элемента "Стадия" начиная с самого текущего элемента.
Связи	Ref(contentType)	Возвращает первый элемент из списка связей элемента с заданным типом элемента связи	contentType: строка (обязательный) - Тип элемента связи	Элемент	Ref('Связь с замечанием')	Вернет первый элемент списка связей с типом элемента 'Связь с замечанием'
	Refs(contentType)	Возвращает список связей элемента с заданным типом элемента связи	contentType: строка (обязательный) - Тип элемента связи	Список элементов	Refs('Связь с замечанием')	Вернет список связей с типом элемента 'Связь с замечанием'
Логические	if(statement, 'value is true', 'value is false')	Проверяет результат заданного логического выражения и в зависимости от него возвращает одно из двух значений	statement: bool - логическое выражение, результатом которого является True или False	Значение	if(This('Шифр') = '', This('Статус'), This('Шифр'))	Проверит заполнение атрибута "Шифр" и, если этот атрибут пустой, то вернет значение атрибута "Статус", а если "Шифр" заполнен, то вернет его значение

		<p>value is true (обязательное) - возвращаемое значение, если значение выражения True</p> <p>value is false (обязательное) - возвращаемое значение, если значение выражения False</p>			
statement_t1 statement_2	Логическое "ИЛИ" - проверит значение обоих выражений и вернет True, если хотя-бы одно из них является верным.	<p>statement_1: bool - первое логическое выражение, результатом которого является True или False</p> <p>statement_1: bool - второе логическое выражение, результатом которого является True или False</p>	bool	Field(Ref("Вложение процесса"), 'Статус документа.Название') == 'Размещено' Field(Ref("Вложение процесса"), 'Статус документа.Название') == 'На корректировке'	Вернет True если значением атрибута "Статус документа" вложения процесса является "Размещено" или "На корректировке" и False, если нет.
statement_t1 && statement_2	Логическое "И" - проверит значение обоих выражений и вернет True, если оба из них являются верными и False, если хотя-бы одно неверно.	<p>statement_1: bool - первое логическое выражение, результатом которого является True или False</p> <p>statement_1: bool - второе логическое выражение, результатом которого является True или False</p>	bool	This('Статус.Название') == 'Выполняется' && IsNotNull(This('Ответ'))	Вернет True, если значение атрибута "Статус" текущего элемента - "Выполняется" и значение атрибута "Ответ" не является пустым и False если какое-то из этих условий не выполнено.
IsEndWith(str, subStr)	Проверяет, заканчивается ли указанная строка заданной подстрокой и возвращает True или False	<p>str: строка (обязательное) - Исходная строка</p> <p>subStr: строка - искомая подстрока</p>	bool	IsEndWith(This('Имя'), '.docx')	Вернет True если в значении This('Имя') строка заканчивается на .docx
IsNull(value)	Функция возвращает true, если у атрибута нет значения (пустая строка или null), и false, если атрибут содержит значение	value (обязательный) - Атрибут или результат вычисления другой формулы	bool	IsNull(Field(Parent(), 'Ответственный'))	Вернет False, если поле "Ответственный" родительской папки текущего элемента заполнено и True, если нет.
IsNotNull(value)	Функция возвращает true, если у атрибута есть значение, и false, если атрибут не содержит значения (пустая строка или null)	value (обязательный) - Атрибут или результат вычисления другой формулы	bool	IsNotNull(Field(Parent(), 'Ответственный'))	Вернет True, если поле "Ответственный" родительской папки текущего элемента заполнено и False, если нет.
Switch(checkKey, key1, value1, key2, value2, ..., keyN, valueN)	Проверит совпадение заданного значения с ключом из списка "ключ, значение" и вернет соответствующее этому ключу значение. Как ключ, так и значение могут быть различных типов - строка, число, объект, bool	<p>checkKey (обязательно) - искомый ключ</p> <p>список пар key, value (обязательно) - список пар "ключ, значение", разделенных запятой. Пары разделяются запятыми.</p>	Значение, соответствующее заданному ключу.	Switch(Field(Ref("Вложение процесса"), 'Статус документа.Название'), 'Размещено', Lookup('Статус процесса.Входной контроль'), 'Проверено', Lookup('Статус процесса.Согласующие'), 'Согласовано', Lookup('Статус процесса.Утверждающие'))	<p>Здесь:</p> <p>Field(Ref("Вложение процесса"), 'Статус документа.Название') - искомый ключ: название статуса документа являющегося вложением в процесс.</p> <p>'Размещено', Lookup('Статус процесса. Входной контроль') - первая пара "ключ, значение"</p> <p>'Проверено', Lookup('Статус процесса. Согласующие') - вторая пара "ключ, значение"</p> <p>'Согласовано', Lookup('Статус процесса. Утверждающие') - третья пара "ключ, значение"</p> <p>В зависимости от статуса документа вернет одно из возможных значение:</p> <ol style="list-style-type: none"> объект Lookup('Статус процесса. Входной контроль') объект Lookup('Статус процесса. Утверждающие')

						3. объект Lookup('Статус процесса. Утверждающие')
Текстовые	FileExt(file)	Возвращает расширение файла заданного элемента с атрибутом "Файл". При отсутствии заданного элемента, будет вычислено расширение файла текущего элемента	file: объект (необязательный) - Элемент с атрибутом "Файл"	Строка	FileExt(Ref('Связь с замечанием'))	Будет вычислено расширение файла у элемента, к которому оставлено замечание.
	Format(format, value)	Возвращает строку, сформированную по заданному шаблону	format: строка - Шаблон, по которому надо сформировать заданные значения value: число (обязательное) - число, которое надо привести к заданному формату	Строка	Format('{0:0000};12)	Вернет строку '0012' Полный набор форматов соответствует работе функции String.Format() https://msdn.microsoft.com/ru-ru/library/0c899ak8(v=vs.110).aspx
	Substring(str, startIndex, subStringLength)	Возвращает подстроку указанной длины от начала исходной строки или от указанного номера символа	str: строка (обязательный) - Исходная строка startIndex: число (необязательный) - номер символа, с которого надо начинать отсчет подстроки subStringLength: число (необязательный) - длина подстроки	Строка	1. Substring('Анкор-МПРТ.03', 5) 2. Substring('Анкор-МПРТ.03', 6, 4)	1. Вернет строку 'Анкор' 2. Вернет строку 'МПРТ'
	Translit(str)	Функция заменяет русские буквы на английские и возвращает результат*	str: строка (обязательный) - исходная строка	Строка	Translit('В12345/555Д')	Вернет строку "V12345_555D"
	TrimEnd(str, subStr)	Функция обрезает строку с конца, начиная с указанной подстроки и возвращает полученный результат. Подстрока ищется с конца исходной строки. Можно указать несколько вариантов подстрок, разделив их символом #	str: строка (обязательный) - исходная строка subStr: строка (обязательный) - искомая подстрока.	Строка	TrimEnd('тест_ПД_ver01','_ver#_rev')	Вернет строку "тест_ПД"
	+	Возвращает результат конкатенации двух строк	value1: строка (обязательное) - первое слагаемое value2: строка (обязательное) - второе слагаемое	строка	Field(Parent('Проект'), 'Шифр') + '-' + Field(Parent(), 'Марка')	Возвращает разделенные дефисом значение атрибута "Шифр" у первого родительского элемента с типом элемента "Проект", и значение атрибута "Шифр" родительского элемента.
Математические	математические операторы: + - * /	Возвращает результат выполнения соответствующей математической функции	value1: число (обязательное) - первое слагаемое value2: число (обязательное) - второе слагаемое	число	(This('Поле А') + This('Поле В') + This('Поле С')) * Field(Parent('Комплект'), 'Множитель') / 3	Вернет результат сложения значений атрибутов "Поле А", "Поле В" и "Поле С" текущего элемента, умноженный на значение атрибута "Множитель" родительского элемента с типом элемента "Комплект" и поделенного на 3
	Count(listObject)	Возвращает количество элементов заданного списка	listObject: список (обязательный) - Список объектов	число	Count(Refs('Вложение рабочего процесса'))	Вернет кол-во связей с типом элемента 'Вложение рабочего процесса' рабочего процесса
	Inc(listFieldName, id, step)	Увеличивает значение указанного атрибута на заданную величину и возвращает итоговое значение	list: строка (обязательный) - наименование списка с элемента которого должно быть взято исходное значение. fieldName: строка (обязательный)-название атрибута id: число (обязательный) - порядковый номер элемента из списка "Счетчики"	число	Inc('Счетчики.Счетчик2', 3)	Возвращает значение атрибута "Счетчик" элемента с ИД=2 из списка "Счетчики" увеличенного на 3. Данная функция служит для реализации счетчика, например, для автоматической нумерации каждого добавляемого в папку документа.

			step: число (необязательный) - значение, на которое нужно увеличить текущее значение.			
	Pow(base, factor)	Возводит в заданное число степень	base: число (обязательный) - основание степени factor: число (обязательный) - показатель степени	число	Pow(2, 3)	Вернет $2^3 = 8$.
	Sqrt(value)	Возвращает квадратный корень заданного числа	value: число (обязательный)	число	Sqrt(4)	Вернет 2
Дата время	AddDay([date], [incStepValue], [format])	Возвращает дату [date], увеличенную на заданное количество дней [incStepValue]. Если указан параметр [format], то возвращает дату в виде строки	date type: Дата-время/строка. default: Текущая дата-время (UTC) incStepValue type: число. default: 0 (может быть отрицательным) format type: строка. default: ""	Дата-время /строка	AddDay(This('Created'), 1, 'dd.MM.yyyy') AddDay(-10, 'dd.MM.yyyy')	Дата создания данного элемента, увеличенная на один день, в формате dd.MM.yyyy Текущая дата-время минус 10 дней, в формате dd.MM.yyyy
	AddMonth([date], [incStepValue], [format])	Возвращает дату [date], увеличенную на заданное количество месяцев [incStepValue]. Если указан параметр [format], то возвращает дату в виде строки	date type: Дата-время/строка. default: Текущая дата-время (UTC) incStepValue type: число. default: 0 (может быть отрицательным) format type: строка. default: ""	Дата-время /строка	AddMonth(Field(Parent('Раздел'), 'Created'), 3) AddMonth(3, 'MMMM')	Дата создания родительского элемента с типом контента 'Раздел', увеличенная на 3 месяца Название месяца через 3 месяца от текущей даты
	AddYear([date], [incStepValue], [format])	Возвращает дату [date], увеличенную на заданное количество месяцев [incStepValue]. Если указан параметр [format], то возвращает дату в виде строки	date type: Дата-время/строка. default: Текущая дата-время (UTC) incStepValue type: число. default: 0 (может быть отрицательным) format type: строка. default: ""	Дата-время /строка	AddYear(Field(Parent(), 'Created'), 2, 'yyyy')	Дата создания родительской папки, увеличенная на 2 года, в формате год
	DateTime(dateTime, format)	Возвращает дату-время в заданном формате. Если не задано значение даты-времени, будет возвращено текущее значение	dateTime: dateTime (необязательный) - Дата-время format: строка (необязательный) - Формат даты-времени	dateTime	DateTime(Field(Parent(), 'Время начала'), 'dd.MM.yyyy')	Вернет значение атрибута "Время начала" родительской папки элемента в указанном формате.
	ParseDateTime(dateTime)	Преобразует строку с указанием даты и времени в формат DateTime. Если указать только дату без времени, то значение времени будет установлено в 00:00:00. Если указать только время, без даты, то будет установлена текущая дата	dateTime: строка (обязательный) - Строка в формате 'dd.mm.yy hh:mm:ss'	dateTime	ParseDateTime('20.05.2024 10:50')	Вернет 20.05.2024 10:50 в формате DateTime
Преобразование типов данных	ParseFloat(value)	Преобразует строку в число с плавающей запятой	value: строка (обязательный) - Строка в которой целая и дробная часть отделены точкой	Число с плавающей запятой	ParseFloat('12,05')	Вернет число 12.05
	ParseBool(value)	Преобразует строки 'true', 'false', '0' или '1' в булево значение	value (обязательный) - строка или число	bool	1. ParseBool('true') 2. ParseBool('false') 3. ParseBool('0')	1. Вернет True 2. Вернет False 3. Вернет False
Работа со списками	Contains(listObject, checkObject)	Проверяет, находится ли заданный объект в списке объектов и возвращает True или False	listObject: список (обязательный) - Список объектов checkObject: объект (обязательный) - искомый объект	bool	Contains(Field(Parent(Ref('Вложение рабочего процесса')), 'Разрешен запуск процесса согласования'), This('Инициатор'))	Вернет True если перечень пользователей указанных в атрибуте 'Разрешен запуск процесса согласования' родительской папки вложения процесса содержит инициатора процесса, и False, если не содержит.

При вычислении сложной формулы система сама определяет, какого типа данные возвращают функции This() Parent(). Но могут возникнуть ситуации, когда нужно привести значение к типу "строка" или "число". Для этого необходимо сделать следующее:

- чтобы результат вычисления функции сделать числом нужно в формуле добавить к нему 0: (This('FieldName') + 0)
- чтобы сделать строкой - прибавить пустой символ: (This('FieldName') + '')

*Таблица соответствия букв для функции Translit

а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
a	b	v	g	d	e	e	zh	z	i	i	k	l	m	n	o	p	r	s	t	u	f	kh	tc	ch	sh	shch	_	y	_	e	iu	ia

Пример описания настройки структуры хранения документов

Статья содержит набор настроенных элементов из примера, входящего в поставку версии 24.1.37, для формирования структуры данных файлового хранилища

Тип элемента	Вид	Имя атрибута	Системное имя атрибута	Тип данных	Примечание
Папка	Папка	Название	name	Однострочный текст	Обязательный
Проект	Папка	Название	name	Однострочный текст	Обязательный
		Полное название	full_name	Однострочный текст	
		Код	code	Однострочный текст	Обязательный
		Ответственный	assign_to_project	Ссылка на элемент списка	Обязательный Выбор из списка: Пользователи
		Время начала	start_date	Дата	
		Время конца	end_date	Дата	
		Тип проекта	project_type	Однострочный текст	
		Статус	task_status	Ссылка на элемент списка	Выбор из списка: Статусы задач
Комплект	Папка	Название	name	Однострочный текст	Обязательный
		Ответственный	assign_to_project	Ссылка на элемент списка	Обязательный Выбор из списка: Пользователи
		Описание	description	Многострочный текст	
		Шифр	cipher	Многострочный текст	
		Согласующие	approval_user_list	Ссылка на элемент списка	Выбор из списка: Пользователи
Стадия	Папка	Название	name	Однострочный текст	Обязательный
		Описание	description	Многострочный текст	
		Код	code	Однострочный текст	Обязательный
		Входной контроль	approval_user_list_1	Ссылка на элемент списка	Выбор из списка: Пользователи
		Согласующие	approval_user_list_2	Ссылка на элемент списка	Выбор из списка: Пользователи
		Утверждающие	approval_user_list_final	Ссылка на элемент списка	Выбор из списка: Пользователи
		Разрешен запуск процесса согласования	approval_workflow_initiator	Ссылка на элемент списка	Выбор из списка: Пользователи
		Наблюдатели	observers	Ссылка на элемент списка	Выбор из списка: Пользователи
		Трудозатраты этап 1	duration_plan_stage_1	Число	
		Трудозатраты этап 2	duration_plan_stage_2	Число	
Трудозатраты этап 3	duration_plan_stage_final	Число			
Документ	Не папка (Представление карточки файла)	Название	name	Однострочный текст	Обязательный
		Статус документа	document_status	Ссылка на элемент списка	Обязательный Выбор из списка: Статусы документов
		Цикл согласования	approval_count	Число	

Пользователи и доступ

[Пользователи и их группировка](#)

[Настройка прав доступа](#)

Пользователи и их группировка

Элементы настройки пользователей

Структуру списка пользователей в Системе составляют:

- Пользователь - элемент списка, регистрирующий пользователя системы, его логин и пароль для аутентификации
- Подразделение - элемент группировки пользователей, может иметь внутри себя другие подразделения и пользователей, а также функциональные группы. может иметь руководителя (Начальника) подразделения
- Функциональная группа - элемент группировки пользователей, который включает в себя ссылки на пользователей, подразделения и функциональные группы, относящиеся к данной группе

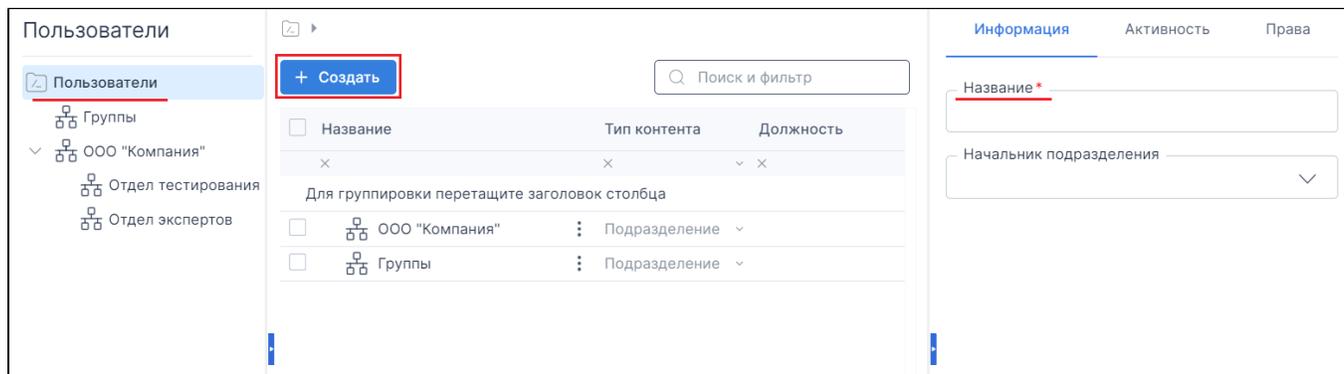
Чаще всего верхним уровнем иерархического списка пользователей является подразделение уровня всей компании

Права доступа на элементы списка пользователей управляются общими правилами настройки прав - на эту тему есть [отдельная статья](#)

Создание подразделения

Для создания нового подразделения в Системе необходимо выполнить следующие действия:

1. В разделе Пользователи в левой части экрана встать на требуемый уровень списка пользователей, нажать кнопку Создать и выбрать из выпадающего списка значение Подразделение
2. В открывшейся справа карточке настроек/свойств подразделения заполнить обязательное поле Название

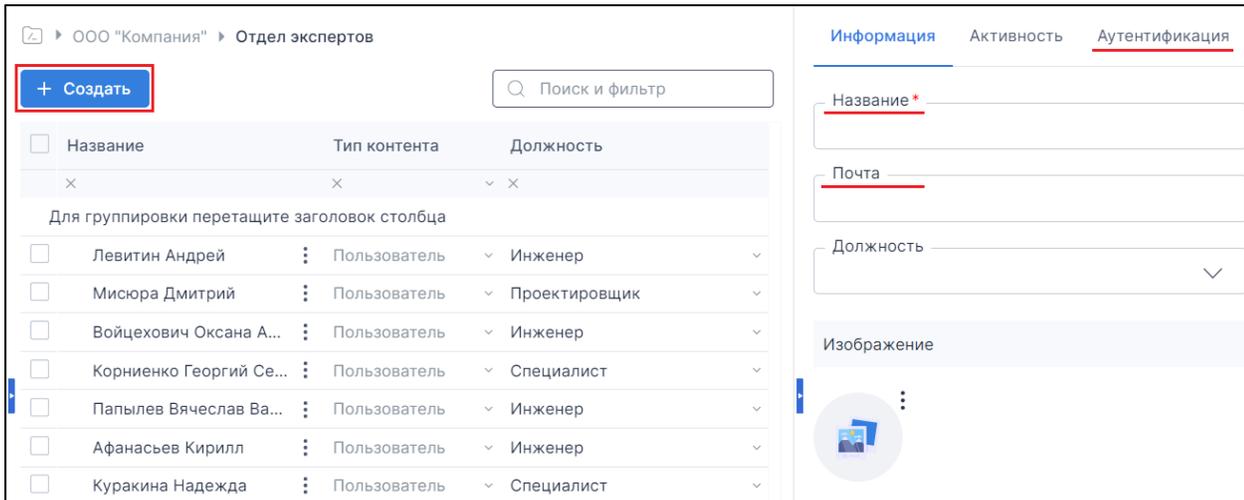


После создания в списке элементов типа Пользователь на карточке настроек/свойств подразделения можно заполнить поле Начальник подразделения, используя контекстный поиск

Создание Пользователя

Для создания нового пользователя в Системе необходимо выполнить следующие действия:

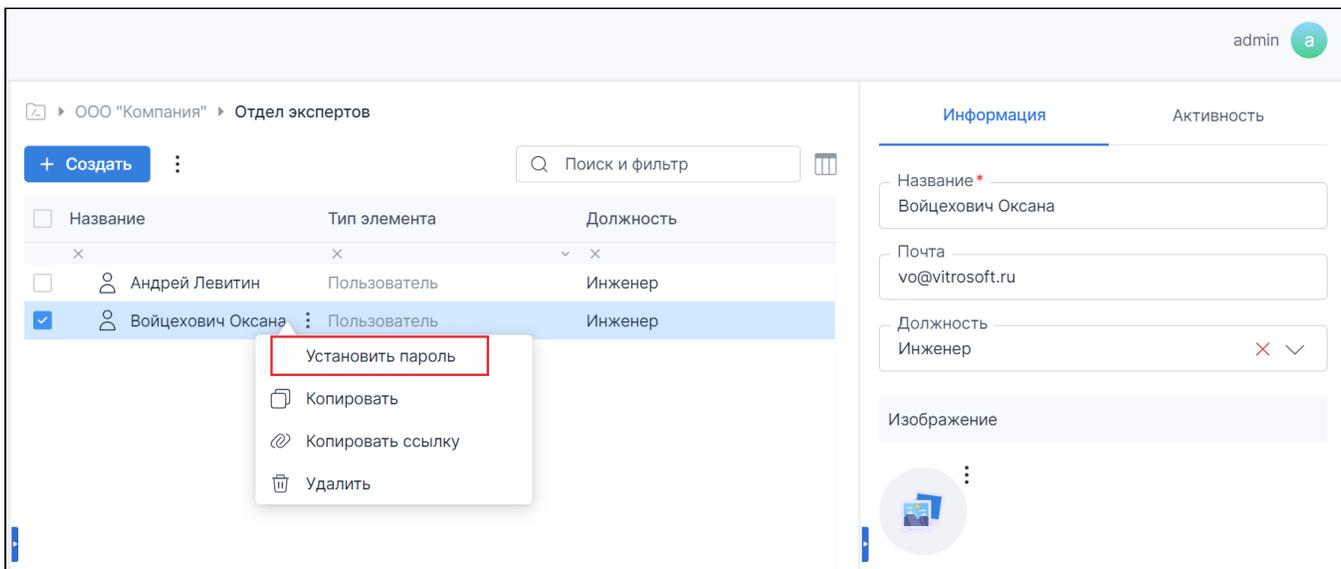
1. В разделе Пользователи в левой части экрана встать на требуемый уровень списка пользователей, нажать кнопку Создать и выбрать из выпадающего списка значение Пользователь
2. В открывшейся справа карточке настроек/свойств пользователя заполнить обязательное поле Название, дополнительные поля Должность и Изображение



Для обеспечения подключения пользователя к системе требуется

- заполнить поле Почта - это значение станет логином для входа пользователя в систему
- перейти на закладку Аутентификация и задать пароль для входа

Для создания пароля на элементе пользователя надо вызвать контекстное меню (три точки или правая клавиша мышки) и нажать на кнопку Установить пароль



На форме установки\смены пароля при необходимости можно дать возможность самому пользователю задать пароль для своей учетной записи в системе - отметить галочкой поле Сменить пароль при следующем входе

Сменить пароль

Новый пароль *

••••••

Сменить пароль при следующем входе

Отмена **OK**

Создание Функциональной группы

Для создания функциональной группы в списке пользователей необходимо выполнить следующие действия::

1. В разделе Пользователи в левой части экрана встать на требуемый уровень списка пользователей, нажать кнопку Создать и выбрать из выпадающего списка значение Функциональная группа
2. В открывшейся справа карточке настроек/свойств функциональной группы заполнить обязательное поле Название

Группы

+ Создать Поиск и фильтр

Пользователь

Функциональная группа

Подразделение

Бюро ГИПов

Администраторы системы

Тип контента

Д...

Название *

Описание

3. На карточке настроек/свойств функциональной группы открыть закладку Состав, нажать кнопку Добавить и выбрать Пользователя / Группу пользователя, используя контекстную подсказку. Повторить добавление повторно нужное количество раз

Группы

+ Создать Копировать

Поиск и фильтр

Название

Тип контента

Д...

Для группировки перетащите заголовок столбца

Администраторы системы

Бюро ГИПов

Новая группа пользователей

Информация **Состав** Активность

Пользователь

Нет данных

+ Добавить

Пользователь/группа *

Настройка прав доступа

Элементы прав доступа

Систему управления правами доступа в Vitro-CAD обеспечивают:

- Право на элементарные действия. Перечень элементарных прав: Просмотр, Изменить, Создать, Скачать, Удалить, Взять на редактирование, Управление версиями, Управление правами
- Уровень доступа - набор элементарных прав. Например, уровень доступа "Изменение" = Просмотр+Создать+Изменить+Удалить
- Назначение прав - действия по присвоению группе пользователей/пользователю определенного уровня доступа к элементу структуры хранения данных системы
- Механизм наследования прав - автоматическое наследование назначенных прав от родительского элемента на дочерние, а также возможность прерывания наследования и назначения уникальных прав на любом уровне хранения информации
- Механизм сложения и проверки прав - права конкретного пользователя складываются из уровней прав всех прямых назначений и всех групп, в которые пользователь входит

Последовательность начальных действий при настройке прав:

- Создать / проверить уровни доступа
- Обеспечить группировку пользователей для настройки прав - в подразделения или функциональные группы (работа с пользователями описана в [отдельной статье](#))
- Назначить для всего проектного пространства для всех пользователей системы единый уровень доступа, например "Просмотр"
- Назначить права на списки и уровни файлового хранилища в соответствии с требуемой матрицей прав

Создание уровня доступа

Для создания нового уровня доступа необходимо выполнить следующие действия::

1. В настройках открыть раздел Списки, найти список Уровни доступа и двойным кликом открыть его.
2. Внутри списка - нажать кнопку Создать. Заполнить обязательные поля Название и Права, используя контекстную подсказку для выбора нескольких значений.

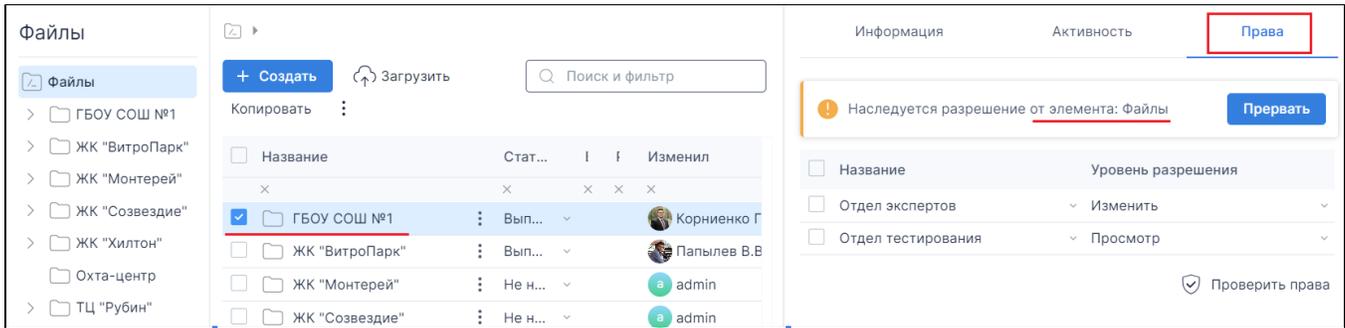
The screenshot shows the 'Create' button highlighted in red in the top left. Below it is a table with columns 'Name' and 'Content Type'. The table lists several roles: 'No access', 'Change', 'View', and 'Collaborative work', each with a 'Role' dropdown. To the right, the 'Rights' configuration panel is open, showing fields for 'Name' and 'Rights'. The 'Rights' field is populated with 'уп' and a list of permissions: 'Управление версиями (version.manage)' and 'Управление правами (permission.manage)'.

Назначение прав

Назначать и просматривать права в системе может администратор системы и пользователи, имеющие в составе своих прав элементарное право "Управление правами"

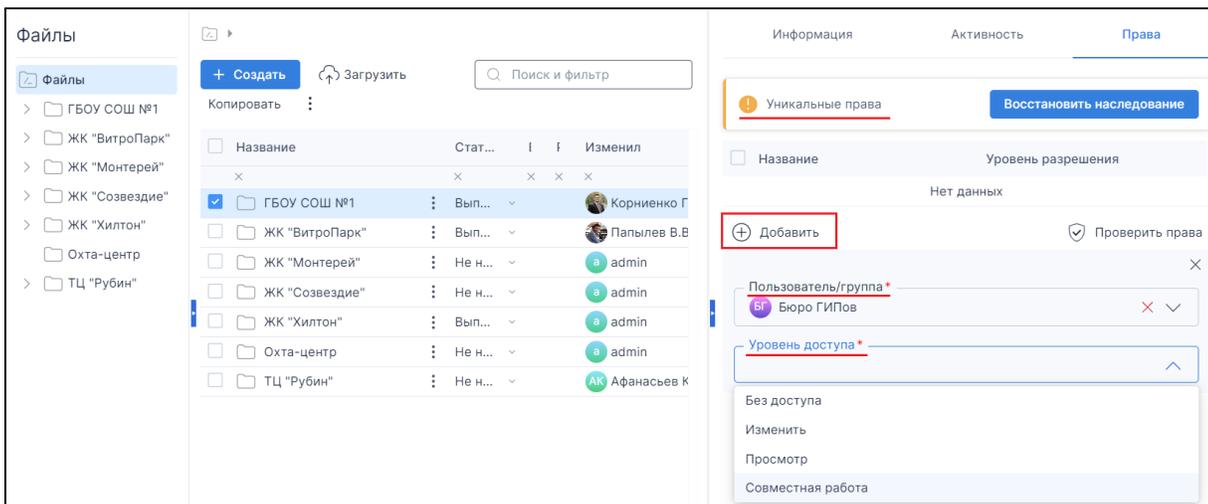
Для назначения прав на элемент системы любого уровня (для списков, для хранилища файлов) необходимо выполнить следующие действия:

1. В карточке свойств выделенного элемента системы открыть закладку Права. На закладке будет показана информация какие группы пользователей / пользователи какие права имеют на элемент и от какого родительского элемента они были унаследованы



2. Для назначения уникальных прав на элемент необходимо:

- прервать наследование, нажав на кнопку Прервать
- нажать кнопку Добавить
- выбрать Пользователя / Группу пользователей, используя контекстную подсказку
- выбрать Уровень доступа

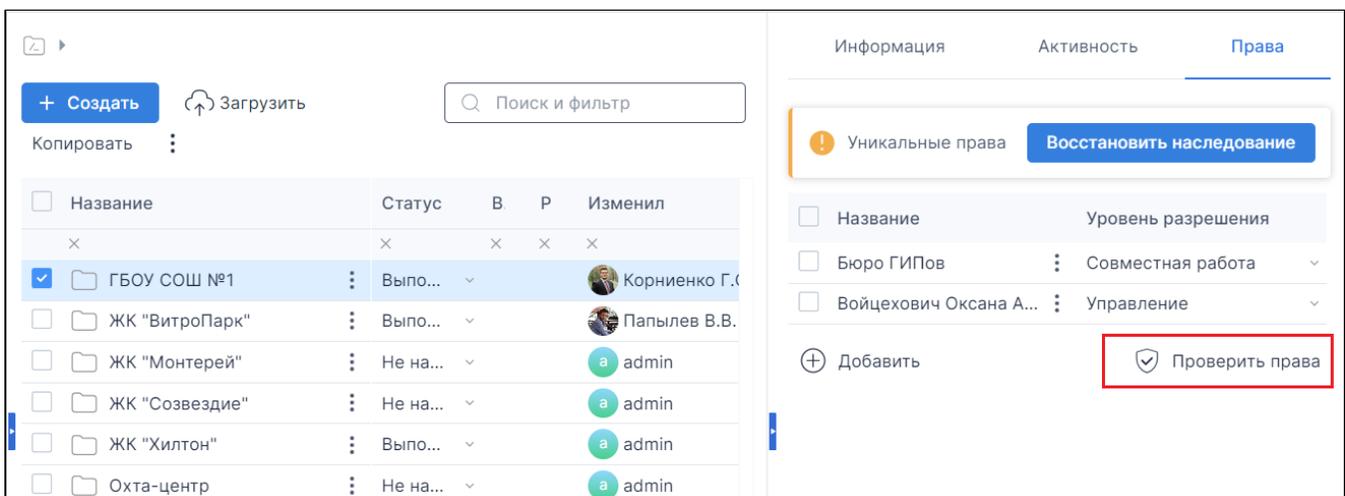


3. Для восстановления прав доступа, наследуемых от родительского уровня структуры хранения данных, надо нажать кнопку Восстановить наследование

Проверка прав

Для получения полного перечня элементарных прав пользователя / группы пользователей на конкретном элементе системы необходимо выполнить следующие действия:

1. В карточке свойств выделенного элемента системы открыть закладку Права. На закладке нажать кнопку Проверить права



2. На форме проверки прав выбрать Пользователя / Группу пользователей, используя контекстную подсказку

 Проверка прав пользователя/группы
Проверка прав объекта: ГБОУ СОШ №1

Пользователь/группа
Войцехович Оксана Александровна

Права	Предоставлено от
Управление правами	 Войцехович О.А.
Управление версиями	 Войцехович О.А.
Просмотр	 Бюро ГИПов
Создать	 Бюро ГИПов
Изменить	 Бюро ГИПов
Удалить	 Бюро ГИПов
Взять на редактирование	 Бюро ГИПов

После выбора на форме проверки прав будет выведен список элементарных прав выбранного пользователя/группы и комментариев, принадлежность к какой группе пользователей это право предоставила

Первичная настройка прав



Примечание

Рекомендуется выполнять первичную настройку прав через назначение доступа на Подразделения, находящиеся в корне списка Пользователи.

1. Для пространств Портал и Центр администрирования назначить всем пользователям уровень доступа Просмотр.
2. В пространстве Портал назначить уровень доступа Изменить на следующие списки
 - a. Задачи - тем пользователям (Подразделениям), которые потенциально могут работать процессами, задачами, замечаниями.
 - b. Процессы - тем пользователям (Подразделениям), которые потенциально могут работать с процессами.
3. В пространстве Центр администрирования назначить всем пользователям уровень доступа Изменить на следующие системные (скрытые) списки.
 - a. Версии файлов
 - b. Связь
 - c. Скрытая директория
 - d. Сообщения
 - e. Маршрут
 - f. Файловое хранилище



Примечание

Предоставление уровня доступа Изменить для данных системных списков необходимо для корректной работы функций системы и не предоставляет пользователям дополнительных возможностей при работе.

Настройка протокола Kerberos для использования доменной аутентификации в Linux

Статья посвящена настройке интеграции системы, развернутой на сервере под управлением ОС Linux, с глобальным каталогом LDAP (Active Directory) с использованием механизма Kerberos.

Связанные статьи:

[Включение LDAP аутентификации на портале \(кнопка "Вход через Windows"\)](#)

[Синхронизация списка пользователей с LDAP-каталогом \(AD\)](#)

Для интеграции Linux с корпоративным LDAP (в данном примере, на основе Active Directory) требуется установить необходимые пакеты и настроить сам Kerberos, для этого необходимо:

1. Установить необходимые пакеты
2. Задать имя серверу
3. Настроить синхронизацию времени
4. Создание учетной записи в AD и файла keytab
5. Создание keytab-файла
6. Настройка kerberos на Linux сервере
7. Настройка Веб-приложения Vitro для авторизации в каталоге LDAP/AD при помощи сервис принципала

1. Установка пакетов

Установить необходимые пакеты из репозитория.

а) на Ubuntu/Debian:

```
apt install heimdal-clients
```

б) на Centos:

```
yum install krb5-workstation
```

2. Имя сервера

В процессе настройки аутентификации через LDAP необходимо задать в качестве принципала имя нашего сервера:

```
hostnamectl set-hostname vitroweb.domain.local
```

** Для примера настройки в качестве FQDN имени сервера будет использовано vitroweb.domain.local (домен domain.local).*

3. Настройка времени

Для работы с LDAP необходимо, чтобы время на контроллере совпадало с временем на нашем веб-сервере. Для этого необходимо настроить синхронизацию.

1. Для начала настраиваем временную зону:

```
timedatectl set-timezone Europe/Moscow
```

** в данном примере задается зона соответствующая московскому времени. Список все доступных зон можно посмотреть командой:*

```
timedatectl list-timezones
```

2. Устанавливаем утилиту для синхронизации времени, настраиваем ее, разрешаем запуск демона и запускаем его. Набор команд будет зависеть от дистрибутива Linux.

а) если на системе Ubuntu / Debian:

```
apt install chrony
```

```
nano /etc/chrony/chrony.conf
```

б) если на системе CentOS / Red Hat:

```
yum install chrony
```

```
nano /etc/chrony.conf
```

```
chrony.conf
```

```
...
server dcl.domain.local iburst
# server ...
# server ...
# server ...
...
```

4. Создание учетной записи в AD и файла keytab

Для подключения к контроллеру домена нам необходимо подтверждать подлинность. Это выполняется с помощью учетной записи в LDAP и ее отпечатка в файле keytab.

На контроллере домена в оснастке "Пользователи и компьютеры" добавить нового пользователя со стандартными правами. От этой учетной записи будут выполняться запросы к AD DS.

В текущем примере создаем пользователя с именем входа vitro_krb5.



Учетная запись должна быть размещена по пути, в котором присутствуют названия только на латинице. Подразделения и контейнеры не должны быть на русском. В противном случае, при выполнении команды ниже мы получим ошибку «Password set failed! 0x00000020».

5. Создание keytab-файла

Данный файл позволяет пройти идентификацию в Kerberos без запроса пароля. Он содержит пары имен субъектов Kerberos и зашифрованные ключи, полученные из пароля Kerberos.

Надо создать данный файл на контроллере домена и скопировать на сервер Веб-приложения Vitro. Для этого на контроллере домена и от имени администратора надо запустить Powershell или обычную командную строку. Ввести:

```
ktpass /princ HTTP/vitroweb.public.com@DOMAIN.LOCAL /mapuser vitro_krb5@DOMAIN.LOCAL /crypto ALL /ptype KRB5_NT_PRINCIPAL /out C:\temp\krb5.keytab /pass < >
```



vitroweb.public.com — FQDN имя сервера Vitro. Указывается имя используемое конечными пользователями (публичное или внутрисетевое) для подключения к portalу, зарегистрированное в корпоративном DNS (запись типа A, или CNAME);
DOMAIN.LOCAL — домен предприятия (realm kerberos);
vitro_krb5@DOMAIN.LOCAL * — учетная запись в AD для выполнения запросов (создана на шаге выше);
* регистр важен.

В нашем примере, после выполнения команды на контроллере домена в директории C:\temp появится файл krb5.keytab. Его следует скопировать на Linux-сервер, например, при помощи WinSCP.



Важно!

Имя файла должно быть krb5.keytab.

На Linux-сервере файл необходимо разместить в каталоге /etc.

Необходимо выдать разрешения на файл krb5.keytab командой `sudo chmod 777 /etc/krb5.keytab`

6. Настройка kerberos на Linux сервере

1. Открыть на редактирование файл:

```
nano /etc/krb5.conf
```

2. Привести файл к виду:

```
/etc/krb5.conf
```

```
[libdefaults]
default_realm = DOMAIN.LOCAL

# The following krb5.conf variables are only for MIT Kerberos.
kdc_timesync = 1
ccache_type = 4
forwardable = true
proxiable = true

# The following libdefaults parameters are only for Heimdal Kerberos.
fcc-mit-ticketflags = true

[realms]
  DOMAIN.LOCAL = {
    kdc = 10.0.0.1
    kdc = 10.0.0.5
    admin_server = domain.local
    default_domain = domain.local
  }

[domain_realm]
  .domain.local = DOMAIN.LOCAL
  domain.local = DOMAIN.LOCAL
```



DOMAIN.LOCAL — домен предприятия (realm kerberos);

kdc — перечень контроллеров домена;

admin_server — первичный контроллер (в данном примере будет использоваться случайный).

3. Перейти в каталог с файлом keytab и выполнить команду:

```
kinit -S HTTP/vitroweb.domain.local@DOMAIN.LOCAL vitro_krb5
```

При выполнении команды будет запрошен пароль пользователя vitro_krb5. В результате выполнения команды будет создан токен авторизации и сохранен в системную БД.

4. Выполнить:

```
klist
```

Результат выполнения:

```
Credentials cache: FILE:/tmp/krb5cc_0
Principal: vitro_krb5@DOMAIN.LOCAL

Issued                Expires                Principal
Jul 30 21:09:34 2024  Jul 31 07:09:34 2024  HTTP/vitroweb.domain.local@DOMAIN.LOCAL
```

7. Настройка Веб-приложения Vitro для авторизации в каталоге LDAP/AD при помощи сервис принцепала

Отредактировать (а при отсутствии предварительно создать) файл `/usr/local/Vitro.Server.Core.Web/Vitro/Server/Conf/auth.json*`

auth.json

```
{
  "DomainName": "dcl.domain.local",
  "Container": "DC=vitro,DC=intra",
  "UserName": "vitro_krb5",
  "Password": "@pwd4krb!",
  "AuthType": "Basic"
}
```

* путь `/usr/local/Vitro.Server.Core.Web/Vitro/Server/Conf` - это путь к директории с конфигурационными файлами Vitro. В случае размещения рабочей директории `"Vitro.Server.Core.Web"` по иному префикс-пути, отличного от `"/usr/local"` следует изменить значение пути на соответствующее собственное.



Описание параметров

"DomainName" - FQDN имя контроллера домена

"Container" - объект структуры каталога LDAP/AD в котором осуществляется поиск пользователей

"UserName" - имя пользователя, от имени которого осуществляется подключение к каталогу LDAP/AD

"Password" - пароль пользователя, от имени которого осуществляется подключение к каталогу LDAP/AD

"AuthType" - тип аутентификации (оставить "Basic")

Синхронизация списка пользователей с LDAP-каталогом (AD)

Связанные статьи:

[Настройка протокола Kerberos для использования доменной аутентификации в Linux](#)

[Включение LDAP аутентификации на портале \(кнопка "Вход через Windows"\)](#)

Настройка утилиты выполняется в следующей последовательности:

1. Настройка параметров запуска утилиты.
2. Настройка расписания в планировщике заданий.
 - а. [Настройка расписания в ОС Linux на примере задания на запуск утилиты синхронизации ОШС с AD/LDAP](#)

Общая информация

1. Синхронизация сотрудников и подразделений из LDAP каталога со списками "Пользователи" и "Должности" в Vitro-CAD (версия MP) выполняется с помощью утилиты "Vitro.Server.OrgStructureLdap".
2. Корневой каталог утилиты - каталог, где располагается исполняемый файл утилиты. Для примера, на системах с Windows это каталог: "c:\Program Files\Vitro Software\Vitro\Server\Tools\OrgStructureLdap\".
3. Настройки располагаются в файле "config.json" корневого каталога утилиты.
4. Утилита записывает результаты работы в лог. Файлы логов размещаются в папке \logs корневого каталога утилиты. Имя файла лога формируется по маске "Sync-%date(yyyyMMdd-HHmm).log", где date - текущая дата.

Настройка параметров запуска утилиты

Mode
<pre>. - (). "Mode": "Org" - . " " () " " () " " AD. "Mode": "Group" - , AD. , , () . "Mode": "List" - , . , , "ListRootDepartment" (.) .</pre>
Action
<pre>. : "Action": "sync.position" - . "Action": "sync.orgStructure" - () .</pre>
DomainName
<pre>. "DomainName": "vitro"</pre>
Container
<pre>AD, . "Container": "CN=Users,DC=vitro,DC=dev" : , , , . . .</pre>

LdapUserName

LDAP .

LdapPassword

LDAP .

AuthType

LDAP .

"AuthType": "Negotiate" - Windows

"AuthType": "Basic" - Linux

WebUrl

URL- Vitro, "" "".
"WebUrl": "http://vitro"

Login

- Vitro.

Password

- Vitro.

TitleField

, . ;
sn + " " + givenName + " " + middleName. - , "displayName".

"TitleField": "displayName"

ExcludeOrgUnitList

OU (Organizational Unit), .
: []
ExcludeOrgUnitList: "[excludeOU1;excludeOU2]"

ExcludePrincipallist

/ ("SamAccountName"), .
: []
ExcludePrincipallist: "[testGroup1;testUser1]"

AddPrincipalToDepartmentHeader

```
LDAP = , , .  
: "false".  
"AddPrincipalToDepartmentHeader": false - .  
"AddPrincipalToDepartmentHeader": true - .
```

DefaultDepartment

```
, . , .  
"DefaultDepartment": "__"
```

ListRootDepartment

```
"Mode": "List"!  
( ) . , .  
, "DefaultDepartment", , "".
```

IsTest

```
( / ). "false".  
"IsTest": false - .  
"IsTest": true - SP, .
```

История изменений

13.11.2024

25.1.39.8 - Добавлен новый параметр "TitleField".

Включение LDAP аутентификации на портале (кнопка "Вход через Windows")

Связанные статьи:

[Настройка протокола Kerberos для использования доменной аутентификации в Linux](#)

[Синхронизация списка пользователей с LDAP-каталогом \(AD\)](#)

Для активации функции NTLM аутентификации необходимо добавить ключ в реестр конфигурации. Ключ определяет, включена ли NTLM аутентификация на сервере.

Ключ типа boolean
Security.Authentication.NTLM.Enabled

В списке Конфигурация создается следующая структура:

Список Конфигурация
<pre>- (- , - Security) -- (- , - Authentication) --- NTLM (- , - NTLM)</pre>

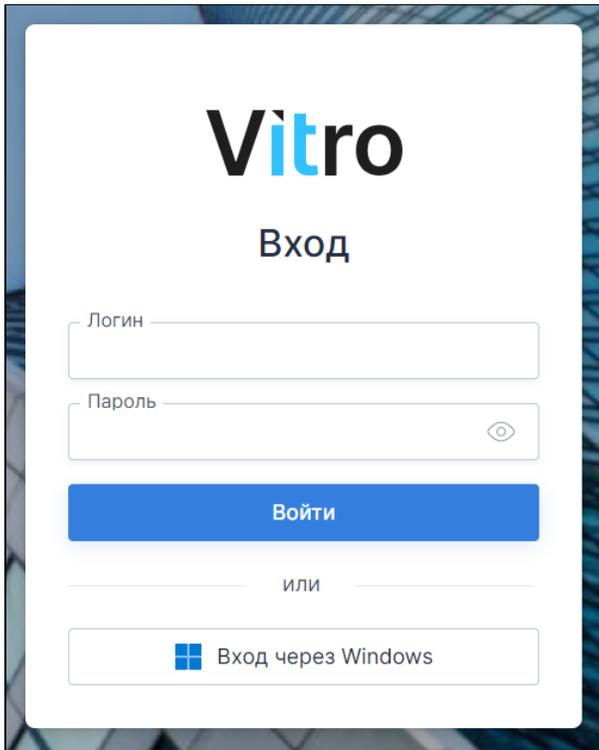
После создания структуры в папку конфигурации "NTLM" добавляется ключ с указанием значения (см. таблица ключей NTML аутентификации)

Таблица ключей NTML аутентификации

Название	Тип элемента	Код	Значение по умолчанию	Ед. изм.	Примечание
Включен	Значение конфигурации Флаг	Enabled	True		

Изменения, произведенные в списке Конфигурация вступают в силу в течении 2-х минут.

В результате выполненных действий на странице авторизации появится дополнительная кнопка "Вход через Windows".



The image shows a login interface for 'Vitro'. At the top, the 'Vitro' logo is displayed in a large, bold font, with the 'i' in blue. Below the logo, the word 'Вход' (Login) is centered. There are two input fields: the first is labeled 'Логин' (Login) and the second is labeled 'Пароль' (Password) with an eye icon for toggling visibility. A blue button labeled 'Войти' (Login) is positioned below the password field. Underneath the button, the word 'или' (or) is centered. At the bottom, there is a button with the Windows logo and the text 'Вход через Windows' (Login through Windows).

ВНИМАНИЕ! Надо учитывать, что синхронизация учетной записи Windows и учетной записи пользователя в Vitro происходит через адрес почты (в карточке пользователя в списке "Пользователи" должна быть указана та же почта, что и в AD).

Настройка протокола Kerberos для использования доменной аутентификации в Windows развертывании

Статья посвящена настройке интеграции системы, развернутой на сервере под управлением ОС Linux, с глобальным каталогом LDAP (Active Directory) с использованием механизма Kerberos.

Связанные статьи:

[Включение LDAP аутентификации на портале \(кнопка "Вход через Windows"\)](#)

[Синхронизация списка пользователей с LDAP-каталогом \(AD\)](#)

1. Создание учетной записи в AD

Для подключения к контроллеру домена нам необходимо подтверждать подлинность. Это выполняется с помощью учетной записи в LDAP.

На контроллере домена в оснастке "Пользователи и компьютеры" добавить нового пользователя со стандартными правами. От имени этой учетной записи будут выполняться запросы к AD DS.

В текущем примере создаем пользователя с именем входа vitro_krb5.

2. Настройка связанной SPN

Настроить субъект-службу для сервисной учетной записи. Для этого на контроллере домена и от имени администратора надо запустить Powershell или обычную командную строку. Ввести:

```
Setspn -s HTTP/vitroweb.public.com vitro_krb5@domain.local
```

i vitroweb.public.com — FQDN имя сервера Vitro. Указывается имя используемое конечными пользователями (публичное или внутрисетевое) для подключения к portalу, зарегистрированное в корпоративном DNS (запись типа A, или CNAME);
DOMAIN.LOCAL — домен предприятия (внутренний);
vitro_krb5@DOMAIN.LOCAL — учетная запись в AD для выполнения запросов (создана на шаге выше);

3. Настройка Веб-приложения Vitro для авторизации в каталоге LDAP/AD при помощи сервис принцепала

Отредактировать (а при отсутствии предварительно создать) файл C:\Program Files\Vitro Software\Vitro\Server\Conf\auth.json

auth.json

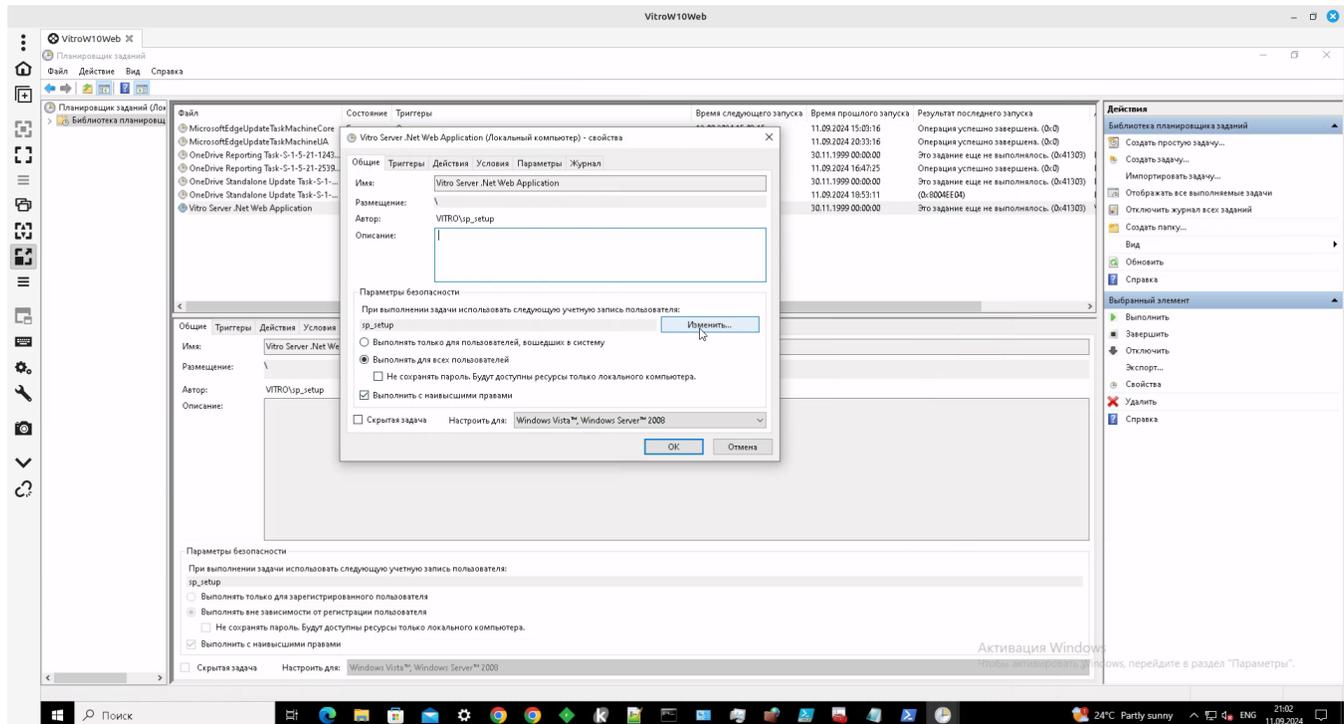
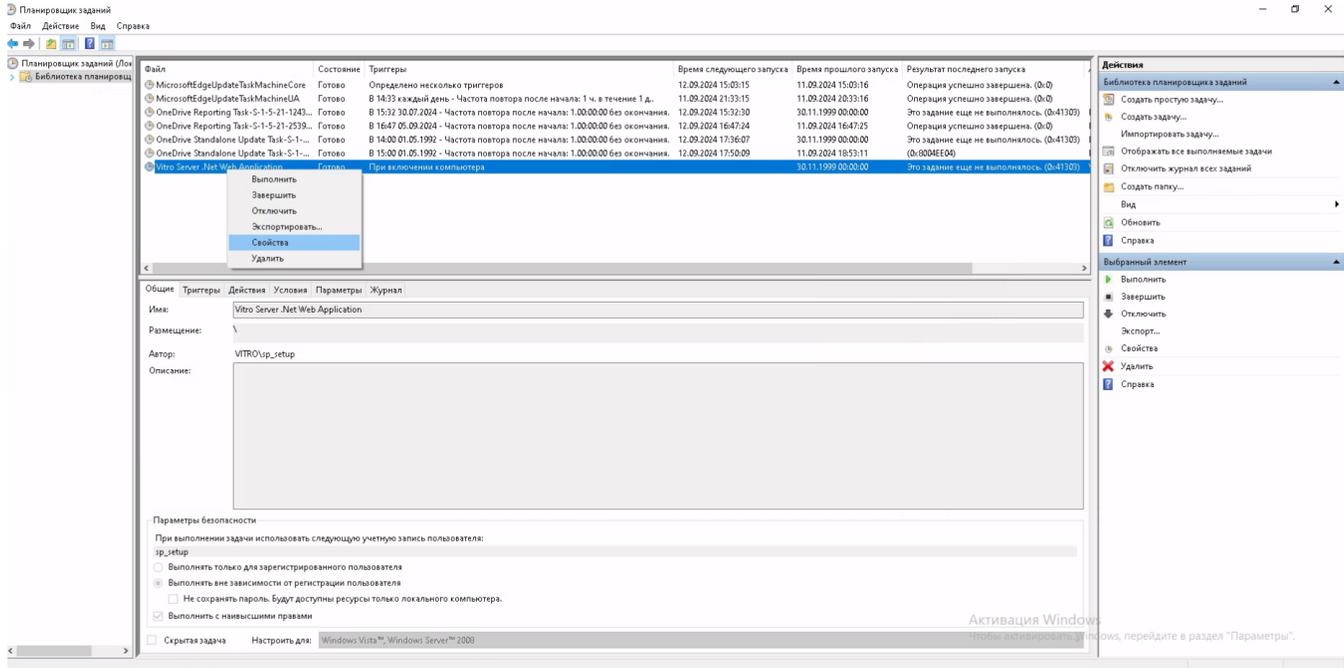
```
{  
  "DomainName": "dcl.domain.local",  
  "Container": "DC=vitro,DC=intra",  
  "UserName": "vitro_krb5",  
  "Password": "@pwd4krb!",  
  "AuthType": "Basic"  
}
```

i Описание параметров

- "DomainName" - FQDN имя контроллера домена
- "Container" - объект структуры каталога LDAP/AD в котором осуществляется поиск пользователей
- "UserName" - имя пользователя, от имени которого осуществляется подключение к каталогу LDAP/AD
- "Password" - пароль пользователя, от имени которого осуществляется подключение к каталогу LDAP/AD
- "AuthType" - тип аутентификации (оставить "Basic")

4. Настройка запуска Веб-приложения Vitro

Настроить в расписании Windows запуск службы Vitro Server .Net Web Application от созданной учетной записи. Если запуск Веб-приложения Vitro **настраивался с помощью планировщика Windows**, то меняем учетную запись для запуска, как указано в п.23 и перезапускаем службу.



VitroW10Web

Планировщик заданий

Файл Действие Вид Справка

Планировщик заданий (Локальный компьютер)

Библиотека планировщика заданий

Общие Триггеры Действия Условия

Имя: Vitro Server .Net Web Application (Локальный компьютер) - свойства

Размещение: \

Автор: VITRO\vip_setup

Описание: Введите имена выбираемых объектов (полезно):
Имя: Имя HTTP://локаль:web.vitro.kz@VITRO.INTRA

Выбор: "Пользователь", "Учетная запись службы" или "Группа"

Выберите тип объекта:

Размещение: "Пользователь", "Учетная запись службы" или "Встроенный суб-тип объектов..."

Введите имена выбираемых объектов (полезно):
Имя: Имя HTTP://локаль:web.vitro.kz@VITRO.INTRA

Дополнительно...

OK Отмена

Состояние Триггеры

Имя	Время следующего запуска	Время прошлого запуска	Результат последнего запуска
Vitro Server .Net Web Application	11.09.2024 15:03:16	11.09.2024 15:03:16	Операция успешно завершена. (0x0)
Vitro Server .Net Web Application	11.09.2024 20:33:16	11.09.2024 20:33:16	Операция успешно завершена. (0x0)
Vitro Server .Net Web Application	30.11.1999 00:00:00	30.11.1999 00:00:00	Это задание еще не выполнялось. (0x41307)
Vitro Server .Net Web Application	11.09.2024 16:47:25	11.09.2024 16:47:25	Операция успешно завершена. (0x0)
Vitro Server .Net Web Application	30.11.1999 00:00:00	30.11.1999 00:00:00	Это задание еще не выполнялось. (0x41307)
Vitro Server .Net Web Application	11.09.2024 18:53:11	11.09.2024 18:53:11	Операция успешно завершена. (0x0)
Vitro Server .Net Web Application	30.11.1999 00:00:00	30.11.1999 00:00:00	Это задание еще не выполнялось. (0x41307)

Действия

Библиотека планировщика заданий

- Создать простую задачу...
- Создать задачу...
- Импортировать задачу...
- Отображать все выполняемые задачи
- Отключить журнал всех заданий
- Создать папку...
- Вид
- Обновить
- Справка

Выбранный элемент

- Выполнить
- Завершить
- Отключить
- Экспорт...
- Удалить
- Справка

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

24°C Partly sunny 21:10 11.09.2024

REST API

- [Описание - версия 24.1.37](#)
- [Описание - версия 24.1.23](#)
- [Пример работы с API на Python](#)
- [Настройка системы с использованием командной строки \(CLI\)](#)
- [Создание действия с вызовом REST сервиса](#)
- [Вызов REST сервиса из обработчика действия](#)
- [Поиск элементов по структуре - версия 25.1.39.16](#)

Описание - версия 24.1.37

Работа с API начинается со входа в систему:

Сущность	Операции доступные через API	REST API
Вход в систему	Получение токена	<p>URL: api/security/login Метод: POST Описание: Авторизация Входные параметры: json объект</p> <pre>{ "login": "логин", "password": "пароль" }</pre> <p>Ответ: токен пользователя</p> <p>(Токен пользователя передается в заголовке Authorization)</p> <p>Пример:</p> <p>Получить токен авторизации</p> <p>URL: https://my-site.ru/api/security/login</p> <p>Тело запроса:</p> <pre>{ "login": "admin", "password": "admin" }</pre> <pre>{ "id": "c2d06dbc-e251-403c-b7d8-bdf21e6ec5cb", "user": { "id": "aab39600-92b4-4c38-a55c-53efac8db9cc", "name": "admin", "email": "admin@email.test", "item": { "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b", "listId": "e3a94bde-0ca9-456f-b338-4465d40389ee", "parentId": "c0857f66-fbfa-448f-a35d-afac00a3a9cb", "itemPath": { "path": ["3064dc08-2e02-8de4-aa70-1b2ae9eb890b", "e3a94bde-0ca9-456f-b338-4465d40389ee", "c0857f66-fbfa-448f-a35d-afac00a3a9cb"] } } } }</pre>

```
"scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
"childCount": 0
},
"contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
"permissionList": null,
"id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"name": null,
"isNew": false,
"insertUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin"
  }
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",
"insertDate": "2024-08-01T11:30:46.202188Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2024-08-01T11:30:46.202188Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "name": "admin",
  "sequence": 60,
  "email": "admin@email.test",
  "login": "admin"
```

```

    }
  },
  "isActive": true,
  "isAdmin": true,
  "groupList": [
    "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
    "e3a94bde-0ca9-456f-b338-4465d40389ee",
    "c0857f66-fbfa-448f-a35d-afac00a3a9cb",
    "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "6ea6149b-5eac-4636-adf1-b01b00b7de4e"
  ]
},
"token": "1TPeDt3ytwMvbfanCKXwipGIJrZ990V10JYwAWqndE_",
"expires": "2024-10-07T06:34:55.3626915Z",
"status": 1,
"loginTypeList": [
  {
    "id": "05ef8c8e-f104-4dc0-8887-b97c1475dd7a",
    "hidden": true
  },
  {
    "id": "17a6ceda-cc38-416c-9f72-379d978a142f",
    "hidden": false,
    "is_two_factor_authentication_enabled": false
  }
],
"requireLoginInactivityTimeoutMinuteCount": 0
}

```

Все
элементы

Чтение метаданных

URL: api/item/get/{itemId}
 Метод: POST
 Описание: получить информацию об элементе по ID элемента
 Входные параметры:
 itemId: Guid - ID элемента списка
 Выходные параметры:
 json объект:

```

{
  id: Guid - ID элемента
  siteId: Guid - ID сайта
  listId: Guid - ID списка
  parentId: Guid - ID родительского элемента
  itemPath: ItemPath - путь до элемента + количество дочерних элементов
  contentTypeId - ID типа контента элемента
  fieldValueMap - объект ключ-значение. Ключ - internalName поля, значение - значение этого поля в элементе
}

```

Пример:

Получить метаданные элемента с id 6ecc311f-7df7-441d-8618-534392e9d31b

URL:

<http://my-site.ru/api/item/get/6ecc311f-7df7-441d-8618-534392e9d31b>

```
{
  "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
  "listId": "966e62c5-a803-49a0-a1be-e680d130c481",
  "parentId": "966e62c5-a803-49a0-a1be-e680d130c481",
  "itemPath": {
    "path": [
      "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
      "966e62c5-a803-49a0-a1be-e680d130c481"
    ],
    "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
    "childCount": 0
  },
  "contentType": "6065b2ae-63e3-44aa-a92d-2d414c30f808",
  "permissionList": [
    "read",
    "create",
    "download",
    "update",
    "delete"
  ],
  "id": "6ecc311f-7df7-441d-8618-534392e9d31b",
  "name": null,
  "isNew": false,
  "insertUser": {
    "id": "12fadb08-5758-20d4-b460-0563cfdaa4bf",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "Administrator",
      "image": "ee17dc08-1297-b510-b79b-a530e5439e6c"
    }
  },
  "insertUserId": "12fadb08-5758-20d4-b460-0563cfdaa4bf",
  "insertUserName": "Administrator",
  "insertDate": "2024-09-02T07:18:27.587948Z",
  "updateUser": {
```

```

    "id": "12fadb08-5758-20d4-b460-0563cfdaa4bf",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "Administrator",
      "image": "ee17dc08-1297-b510-b79b-a530e5439e6c"
    }
  },
  "updateUserId": "12fadb08-5758-20d4-b460-0563cfdaa4bf",
  "updateUserName": "Administrator",
  "updateDate": "2024-09-02T07:18:32.140205Z",
  "deleteUser": null,
  "deleteUserId": "00000000-0000-0000-0000-000000000000",
  "deleteUserName": null,
  "deleteDate": null,
  "status": 1,
  "fieldValueMap": {
    "name": "rst_basic_sample_project.rvt",
    "project_item_revit_server_url": "RSN://79.143.70.137/EK/test1/rst_basic_sample_project.rvt",
    "document_status": {
      "id": "b9d061b1-7ce7-4756-8230-e502cfe3d8d8",
      "contentType": "2067cdf9-2683-4658-9eef-ae6800802a2f",
      "fieldValueMap": {
        "name": "Размещено"
      }
    }
  },
  "approval_count": 0,
  "calc_field_test1": "--/rst_basic_sample_project.rvt",
  "parent_date": " - - ",
  "file": {
    "id": "a524fba8-409a-4b13-b12e-3924440d2eb8",
    "contentType": "587f682a-fbf2-4647-8d91-e2f6460f59bf",
    "fieldValueMap": {
      "file_size": 6467584,
      "upd_date": "2024-09-02T07:18:27.582827Z"
    }
  },
  "version": {
    "id": "7c4f3449-5896-45d8-b71e-9cf7b8e94c84",

```

```

"contentType": "61895e0d-ba2b-48f3-8864-e0d0836c2e53",
"fieldValueMap": {
  "name": "0.1",
  "version_minor": 1,
  "version_major": 0
}
}
}
}
}

```

Создание/изменение

URL: `api/item/update`
 Метод: POST
 Описание: Создать/обновить элемент списка (или массив элементов)
 Входные параметры:
 multipart/form-data

itemListJson - строка, сериализованный в json массив элементов ItemUpdateRequest

```

{
  list_id: Guid - ID списка, элемент которого нужно создать (обязательное)
  content_type_id: Guid - ID типа контента элемента (обязательное при создании элемента).
  parent_id: Guid - ID родительского элемента.
  ...поля типа контента элемента, которые нужно изменить, например:
  name: "новое имя"
}

```

Правила передачи значений

1. Если поле является лупаком, то нужно передать строку с уникальным идентификатором нового значения. Пример:
`"document_status": "583ba97b-4688-41cc-a21a-9e4d3f6794b1"`
2. Если поле является множественным лупаком, то нужно передать массив строк - уникальных идентификаторов. Пример:
`"approval_workflow_initiator": ["aab39600-92b4-4c38-a55c-53efac8db9cc", "3a9a648b-98a0-43e2-b5bf-b06d0084303a"]`
3. Если поле является датой, то нужно передать строку в формате ISO 8601 в UTC. Пример:
`"end_date_plan": "2024-01-28T08:40:26.168Z"`
4. Если поле является числом, то нужно передать число:
`"duration_plan": 3`
5. Если поле является строкой, то нужно передать строку. Пример:
`"name": ".pdf"`

Пример:

Изменить статус элемента (документа) с id `6ecc311f-7df7-441d-8618-534392e9d31b`

URL:

<http://my-site.ru/api/item/update>

ItemListJson:

```

[
  {
    "list_id": "966e62c5-a803-49a0-a1be-e680d130c481",
    "id": "6ecc311f-7df7-441d-8618-534392e9d31b",
    "content_type_id": "6065b2ae-63e3-44aa-a92d-2d414c30f808",
    "parent_id": "966e62c5-a803-49a0-a1be-e680d130c481",
    "document_status": "22a8669a-9674-4aac-950b-3379df1059bd"
  }
]

```

```

    },
  ]
  [
    {
      "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
      "listId": "966e62c5-a803-49a0-a1be-e680d130c481",
      "parentId": "966e62c5-a803-49a0-a1be-e680d130c481",
      "itemPath": {
        "path": [
          "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
          "966e62c5-a803-49a0-a1be-e680d130c481"
        ],
        "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
        "childCount": 0
      },
      "contentType": "6065b2ae-63e3-44aa-a92d-2d414c30f808",
      "permissionList": [
        "read",
        "create",
        "download",
        "update",
        "delete"
      ],
      "id": "6ecc311f-7df7-441d-8618-534392e9d31b",
      "name": null,
      "isNew": false,
      "insertUser": {
        "id": "12fadb08-5758-20d4-b460-0563cfdaa4bf",
        "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
        "fieldValueMap": {
          "name": "Administrator",
          "image": "ee17dc08-1297-b510-b79b-a530e5439e6c"
        }
      },
      "insertUserId": "12fadb08-5758-20d4-b460-0563cfdaa4bf",
      "insertUserName": "Administrator",
      "insertDate": "2024-09-02T07:18:27.587948Z",
      "updateUser": {
        "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",

```

```
"contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
"fieldValueMap": {
  "name": "admin",
  "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2024-10-04T06:37:14.025188Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "name": "rst_basic_sample_project.rvt",
  "project_item_revit_server_url": "RSN://79.143.70.137/EK/test1/rst_basic_sample_project.rvt",
  "document_status": {
    "id": "22a8669a-9674-4aac-950b-3379df1059bd",
    "contentType": "2067cdf9-2683-4658-9eef-ae6800802a2f",
    "fieldValueMap": {
      "name": "На корректировке"
    }
  },
  "approval_count": 0,
  "calc_field_test1": "--/rst_basic_sample_project.rvt",
  "parent_date": " - - ",
  "file": {
    "id": "a524fba8-409a-4b13-b12e-3924440d2eb8",
    "contentType": "587f682a-fbf2-4647-8d91-e2f6460f59bf",
    "fieldValueMap": {
      "file_size": 6467584,
      "upd_date": "2024-09-02T07:18:27.582827Z"
    }
  },
  "version": {
    "id": "7c4f3449-5896-45d8-b71e-9cf7b8e94c84",
    "contentType": "61895e0d-ba2b-48f3-8864-e0d0836c2e53",
```

```

        "fieldValueMap": {
            "name": "0.1",
            "version_minor": 1,
            "version_major": 0
        }
    }
}
]

```

Удаление

URL: `api/item/delete`
 Метод: POST
 Описание: Удалить элемент (массив элементов)
 Входные параметры:
 itemListJson - строка, сериализованный в json массив элементов

```

[
  {
    id : Guid - ID первого удаляемого элемента
  },
  {
    id : Guid - ID второго удаляемого элемента
  },
  ...
]

```

Примечание: при вызове данного метода элемент перемещается в корзину и может быть восстановлен методом описанным ниже. Для удаления элемента без возможности восстановления требуется вызвать данный метод повторно.

Пример:

Удалить элемент с id `b128f641-7169-419f-b84f-99fc34b51aff`

URL:

<http://my-site.ru/api/item/delete>

ItemJson:

```

[
  {
    "id": "b128f641-7169-419f-b84f-99fc34b51aff"
  }
]
["b128f641-7169-419f-b84f-99fc34b51aff"]

```

Восстановление из корзины

URL: `api/item/restore/{itemId}`
 Метод: POST
 Описание: восстановить ранее удаленный в корзину элемент
 Входные параметры:
 itemId: Guid - ID элемента списка

Пример:

Восстановить элемент с id `b128f641-7169-419f-b84f-99fc34b51aff`

	<p>URL:</p> <p>http://my-site.ru/api/item/restore/4c095482-c768-439a-87bd-ed1886441</p> <p>-</p>
Установить уникальные права доступа к элементу	<p>URL: <code>api/security/setItemUniquePermission/</code> Метод: POST</p> <p>Описание: Прервать наследование прав/ установить уникальные права доступа к элементу</p> <p>Входные параметры:</p> <p>Входные параметры:</p> <pre>{ id : Guid - ID элемента, copy_permission: bool (требуется ли скопировать и установить прежние разрешения на элемент) }</pre> <p>Пример:</p> <p>Установить уникальные права доступа для элемента с id <code>b128f641-7169-419f-b84f-99fc34b51aff</code></p> <p>URL:</p> <p>http://my-site.ru/api/security/setItemUniquePermission/</p> <p>Тело запроса:</p> <pre>{ "id": "b128f641-7169-419f-b84f-99fc34b51aff", "copy_permission": true }</pre> <p>-</p>
Удалить уникальные права доступа к элементу	<p>URL: <code>api/security/removeItemUniquePermission/{itemId}</code> Метод: POST</p> <p>Описание: Восстановить наследование прав/ Удалить уникальные права доступа к элементу</p> <p>Входные параметры:</p> <p><code>itemId</code> - ID элемента</p> <p>Пример:</p> <p>Удалить уникальные права доступа для элемента с id <code>b128f641-7169-419f-b84f-99fc34b51aff</code></p> <p>URL:</p> <p>http://my-site.ru/api/security/removeItemUniquePermission/b128f641-7169-419f-b84f-99fc34b51aff</p> <p>-</p>
Назначение прав доступа к элементу пользователю или группе	<p>URL: <code>api/item/update</code> Метод: POST</p> <p>Описание: Задать права пользователю на элемент с уникальными правами. Задание прав происходит путем создания элемента «Разрешение пользователя» в списке «Разрывы прав».</p> <p>Входные параметры:</p> <p><code>multipart/form-data</code></p> <p><code>itemListJson</code> - строка, сериализованный в json массив элементов <code>ItemUpdateRequest</code></p> <pre>{ list_id: Guid - ID списка «Разрывы прав» (обязательное) , content_type_id: Guid – ID типа элемента «Разрешение пользователя» (обязательное), source: Guid - ID элемента, для которого задаются права (обязательное), principal: Guid - ID пользователя или группы (обязательное),</pre>

permission_level: Guid - ID уровня доступа (обязательное)

}

Пример:

Установить уровень доступа "Изменить" (id 9d29f390-70d9-4598-816a-68035a0d6d9f) для пользователя с id ba2bdc08-f633-5666-94f7-a91901fdf3da к папке с id b128f641-7169-419f-b84f-99fc34b51aff

URL:

<http://my-site.ru/api/item/update>

ItemListJson:

```
[
  {
    "list_id": "ecc16787-6b93-40b9-b563-704d9b090fe8",
    "content_type_id": "30a13e40-f715-438d-af7e-cc47cf3a5891",
    "parent_id": "ecc16787-6b93-40b9-b563-704d9b090fe8",
    "name": "Разрывы прав - Test",
    "source": "b128f641-7169-419f-b84f-99fc34b51aff",
    "principal": "ba2bdc08-f633-5666-94f7-a91901fdf3da",
    "permission_level": "9d29f390-70d9-4598-816a-68035a0d6d9f"
  }
]
[
  {
    "siteId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
    "listId": "ecc16787-6b93-40b9-b563-704d9b090fe8",
    "parentId": "ecc16787-6b93-40b9-b563-704d9b090fe8",
    "itemPath": {
      "path": [
        "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
        "ecc16787-6b93-40b9-b563-704d9b090fe8"
      ],
      "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
      "childCount": 0
    },
    "contentType": "30a13e40-f715-438d-af7e-cc47cf3a5891",
    "permissionList": [
      "read",
      "create",
      "download",
      "update",
      "delete"
    ]
  }
]
```

```
  ],
  "id": "a67826fc-0cda-45af-b362-0809e02d38ca",
  "name": null,
  "isNew": false,
  "insertUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "insertUserName": "admin",
  "insertDate": "2024-10-04T07:40:37.065934Z",
  "updateUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "updateUserName": "admin",
  "updateDate": "2024-10-04T07:40:37.065934Z",
  "deleteUser": null,
  "deleteUserId": "00000000-0000-0000-0000-000000000000",
  "deleteUserName": null,
  "deleteDate": null,
  "status": 1,
  "fieldValueMap": {
    "name": "Разрывы прав - Test",
    "principal": {
      "id": "ba2bdc08-f633-5666-94f7-a91901fdf3da",
      "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
      "fieldValueMap": {
        "name": "Мисюра Дмитрий"
      }
    }
  }
}
```

		<pre> } }, "source": "b128f641-7169-419f-b84f-99fc34b51aff", "permission_level": { "id": "9d29f390-70d9-4598-816a-68035a0d6d9f", "contentTypeid": "363859fc-2bea-47ff-958d-9949779191f5", "fieldValueMap": { "name": "Изменить" } } } } } }] </pre>
	<p>Получить список прав текущего пользователя на элемент</p>	<p>URL: <code>api/security/getItemPermissionList/{itemId}</code> Метод: POST</p> <p>Описание: Получить список элементарных прав пользователя на заданный элемент</p> <p>Входные параметры:</p> <p><code>itemId</code> - ID элемента</p> <p>Пример:</p> <p>Получить список прав текущего пользователя на элемент с id <code>2bf2db08-c99d-716f-8cee-7028ebb8afe9</code></p> <p>URL:</p> <p>http://my-site.ru/api/security/getItemPermissionList/2bf2db08-c99d-716f-8cee-7028ebb8afe9 ["read","create","download","update","delete"]</p>
<p>Списки</p>	<p>Чтение метаданных</p>	<p>URL: <code>api/list/get/{listId}</code></p> <p>Метод: POST</p> <p>Описание: получить информацию о списке по ID списка</p> <p>Входные параметры:</p> <p><code>listId</code>: Guid - ID списка</p> <p>Выходные параметры:</p> <p>json объект</p> <p>List</p> <pre> { id: Guid - ID списка name: string - Название списка description: string - описание списка } </pre> <p>Пример:</p> <p>Получить метаданные списка "Пользователи" (id <code>e3a94bde-0ca9-456f-b338-4465d40389ee</code>)</p> <p>URL:</p>

```
http://my-site.ru/api/list/get/e3a94bde-0ca9-456f-b338-4465d40389ee
{
  "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
  "databaseTableName": "principal",
  "description": "Пользователи",
  "data": "{\n\t\"isTreeEnabled\": true,\n\t\"refreshMode\": \"none\"\n}",
  "isHidden": false,
  "isDocumentArchive": false,
  "id": "e3a94bde-0ca9-456f-b338-4465d40389ee",
  "name": "Пользователи",
  "isNew": false,
  "insertUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentTypeid": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "insertUserName": "admin",
  "insertDate": "2023-11-30T07:02:44.776582Z",
  "updateUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentTypeid": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "updateUserName": "admin",
  "updateDate": "2024-09-02T11:25:14.758643Z",
  "deleteUser": null,
  "deleteUserId": "00000000-0000-0000-0000-000000000000",
  "deleteUserName": null,
  "deleteDate": null,
  "status": 1,
  "fieldValueMap": {
```

```

    "isTreeEnabled": true,
    "refreshMode": "none"
  }
}

```

Элементы списка

URL: `api/item/getList/{parentId}`
 Метод: POST
 Описание: получить информацию о всех дочерних элементах по ID родительского элемента
 Входные параметры:
 parentId - ID родительского элемента (ID списка или ID папки внутри списка)
 Выходные параметры:
 json объект

Пример:

Получить данные об элементах списка "Пользователи" (id e3a94bde-0ca9-456f-b338-4465d40389ee)

URL:
<http://my-site.ru/api/item/getList/e3a94bde-0ca9-456f-b338-4465d40389ee>

```

[
  {
    "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
    "listId": "e3a94bde-0ca9-456f-b338-4465d40389ee",
    "parentId": "e3a94bde-0ca9-456f-b338-4465d40389ee",
    "itemPath": {
      "path": [
        "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
        "e3a94bde-0ca9-456f-b338-4465d40389ee"
      ],
      "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
      "childCount": 3
    },
    "contentType": "733af6e2-e187-4e75-8a2b-ae4072f402e0",
    "permissionList": [
      "read",
      "create",
      "download",
      "update",
      "delete"
    ],
    "id": "e873dc08-6b36-669f-98bc-54161b14b08a",
    "name": null,
    "isNew": false,
    "insertUser": {
      "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",

```

```

"contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
"fieldValueMap": {
  "name": "admin",
  "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
}
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",
"insertDate": "2024-05-14T07:33:56.65254Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2024-05-14T07:33:56.65254Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "sequence": 560,
  "name": "ООО \"Проектировщики\""
}
},
{
  "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
  "listId": "e3a94bde-0ca9-456f-b338-4465d40389ee",
  "parentId": "e3a94bde-0ca9-456f-b338-4465d40389ee",
  "itemPath": {
    "path": [
      "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
      "e3a94bde-0ca9-456f-b338-4465d40389ee"
    ]
  }
}

```

```
    ],
    "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
    "childCount": 1
  },
  "contentType": "733af6e2-e187-4e75-8a2b-ae4072f402e0",
  "permissionList": [
    "read",
    "create",
    "download",
    "update",
    "delete"
  ],
  "id": "d8059186-c988-4633-90ee-864073d8252b",
  "name": null,
  "isNew": false,
  "insertUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "insertUserName": "admin",
  "insertDate": "2024-08-01T07:32:17.006529Z",
  "updateUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "updateUserName": "admin",
  "updateDate": "2024-08-01T07:32:17.006529Z",
  "deleteUser": null,
```

```
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "sequence": 820,
  "name": "ПИР-1.08.24"
}
},
]
```

URL: `api/item/getRecursive/{parentId}`

Метод: POST

Описание: рекурсивно получить информацию о всех дочерних элементах по ID родительского элемента (дочерние элементы + их дочерние элементы и т.д.)

Входные параметры:

`parentId` - ID родительского элемента (ID списка или ID папки внутри списка)

Выходные параметры:

json объект

Пример:

Получить список всех дочерних элементов для папки с id `4a862423-20fe-4798-b013-1ea0471d6552` и их дочерних элементов

URL:

<http://my-site.ru/api/item/getRecursive/4a862423-20fe-4798-b013-1ea0471d6552>

```
[
  {
    "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
    "listId": "966e62c5-a803-49a0-a1be-e680d130c481",
    "parentId": "966e62c5-a803-49a0-a1be-e680d130c481",
    "itemPath": {
      "path": [
        "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
        "966e62c5-a803-49a0-a1be-e680d130c481"
      ],
      "scopeId": "4a862423-20fe-4798-b013-1ea0471d6552",
      "childCount": 2
    },
    "contentType": "09ad2c16-6047-4dfb-8274-6c4a5f1edbe5",
    "permissionList": [
      "read",
      "create",
      "download",
      "update",

```

```

    "delete"
  ],
  "id": "4a862423-20fe-4798-b013-1ea0471d6552",
  "name": null,
  "isNew": false,
  "insertUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "insertUserName": "admin",
  "insertDate": "2024-07-03T10:53:22.960485Z",
  "updateUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "updateUserName": "admin",
  "updateDate": "2024-07-03T10:55:04.151855Z",
  "deleteUser": null,
  "deleteUserId": "00000000-0000-0000-0000-000000000000",
  "deleteUserName": null,
  "deleteDate": null,
  "status": 1,
  "fieldValueMap": {
    "name": "00_проверка прав на фр"
  }
},
{
  "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",

```

```
"listId": "966e62c5-a803-49a0-a1be-e680d130c481",
"parentId": "4a862423-20fe-4798-b013-1ea0471d6552",
"itemPath": {
  "path": [
    "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
    "966e62c5-a803-49a0-a1be-e680d130c481",
    "4a862423-20fe-4798-b013-1ea0471d6552"
  ],
  "scopeId": "4a862423-20fe-4798-b013-1ea0471d6552",
  "childCount": 1
},
"contentType": "09ad2c16-6047-4dfb-8274-6c4a5f1edbe5",
"permissionList": [
  "read",
  "create",
  "download",
  "update",
  "delete"
],
"id": "52c15625-baf1-40ef-83c6-c47a5f520cea",
"name": null,
"isNew": false,
"insertUser": {
  "id": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "Левитин Андрей"
  }
},
"insertUserId": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
"insertUserName": "Левитин Андрей",
"insertDate": "2024-07-03T10:56:19.905178Z",
"updateUser": {
  "id": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "Левитин Андрей"
  }
}
```

```
    },
    "updateUserId": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
    "updateUserName": "Левитин Андрей",
    "updateDate": "2024-07-03T10:56:19.905178Z",
    "deleteUser": null,
    "deleteUserId": "00000000-0000-0000-0000-000000000000",
    "deleteUserName": null,
    "deleteDate": null,
    "status": 1,
    "fieldValueMap": {
      "name": "123"
    }
  },
  {
    "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
    "listId": "966e62c5-a803-49a0-a1be-e680d130c481",
    "parentId": "52c15625-baf1-40ef-83c6-c47a5f520cea",
    "itemPath": {
      "path": [
        "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
        "966e62c5-a803-49a0-a1be-e680d130c481",
        "4a862423-20fe-4798-b013-1ea0471d6552",
        "52c15625-baf1-40ef-83c6-c47a5f520cea"
      ],
      "scopeId": "4a862423-20fe-4798-b013-1ea0471d6552",
      "childCount": 0
    },
    "contentType": "09ad2c16-6047-4dfb-8274-6c4a5f1edbe5",
    "permissionList": [
      "read",
      "create",
      "download",
      "update",
      "delete"
    ],
    "id": "b128f641-7169-419f-b84f-99fc34b51aff",
    "name": null,
    "isNew": false,
```

```

"insertUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",
"insertDate": "2024-10-04T06:43:30.960319Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2024-10-04T06:46:35.554054Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "name": "Test",
  "sequence": 1050
}
}
]

```

URL: `api/lookup/GetList/{fieldId}`

Метод: POST

Описание: получить информацию об элементах списка на который ссылается lookup-атрибут по ID этого атрибута

Входные параметры:

`fieldId` - ID lookup-атрибута

Выходные параметры:

json объект

Пример:

Получить перечень элементов списка на который ссылается атрибут "Важность" (id атрибута 8abf69f6-52fd-46b3-b6e3-f2b3af46d151)

URL:

<http://my-site.ru/api/lookup/GetList/8abf69f6-52fd-46b3-b6e3-f2b3af46d151>

```
{
  "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
  "listId": "eafdc20-d5b1-4893-8a2a-b94520d36a77",
  "parentId": "eafdc20-d5b1-4893-8a2a-b94520d36a77",
  "itemPath": {
    "path": [
      "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
      "eafdc20-d5b1-4893-8a2a-b94520d36a77"
    ],
    "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
    "childCount": 0
  },
  "contentType": "55231344-ec21-46e5-ac17-a4d00dc21a79",
  "permissionList": [
    "read",
    "create",
    "download",
    "update",
    "delete"
  ],
  "id": "9d77cfc4-930b-4e29-b041-15e72c146788",
  "name": null,
  "isNew": false,
  "insertUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "insertUserName": "admin",
  "insertDate": "2023-11-30T07:02:44.776582Z",
  "updateUser": {
```

```
"id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
"fieldValueMap": {
  "name": "admin",
  "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2023-11-30T07:02:44.776582Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "name": "Быковая",
  "sequence": 10,
  "item_id": 1
},
{
  "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
  "listId": "eafdc20-d5b1-4893-8a2a-b94520d36a77",
  "parentId": "eafdc20-d5b1-4893-8a2a-b94520d36a77",
  "itemPath": {
    "path": [
      "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
      "eafdc20-d5b1-4893-8a2a-b94520d36a77"
    ],
    "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
    "childCount": 0
  },
  "contentType": "55231344-ec21-46e5-ac17-a4d00dc21a79",
  "permissionList": [
    "read",
    "create",
    "download",
```

```
"update",
"delete"
],
"id": "14c58a82-372f-4869-b4a0-43bc0a677882",
"name": null,
"isNew": false,
"insertUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentTypeid": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",
"insertDate": "2023-11-30T07:02:44.776582Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentTypeid": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2023-11-30T07:02:44.776582Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "name": "Низкая",
  "sequence": 30,
  "item_id": 3
}
}
```

```

    },
    {
      "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
      "listId": "eafdc20-d5b1-4893-8a2a-b94520d36a77",
      "parentId": "eafdc20-d5b1-4893-8a2a-b94520d36a77",
      "itemPath": {
        "path": [
          "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
          "eafdc20-d5b1-4893-8a2a-b94520d36a77"
        ],
        "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
        "childCount": 0
      },
      "contentType": "55231344-ec21-46e5-ac17-a4d00dc21a79",
      "permissionList": [
        "read",
        "create",
        "download",
        "update",
        "delete"
      ],
      "id": "34ba677c-bd32-4322-a47e-28add7e09a85",
      "name": null,
      "isNew": false,
      "insertUser": {
        "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
        "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
        "fieldValueMap": {
          "name": "admin",
          "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
        }
      },
      "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
      "insertUserName": "admin",
      "insertDate": "2023-11-30T07:02:44.776582Z",
      "updateUser": {
        "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
        "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",

```

```

    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "updateUserName": "admin",
  "updateDate": "2023-11-30T07:02:44.776582Z",
  "deleteUser": null,
  "deleteUserId": "00000000-0000-0000-0000-000000000000",
  "deleteUserName": null,
  "deleteDate": null,
  "status": 1,
  "fieldValueMap": {
    "name": "Обычная",
    "sequence": 20,
    "item_id": 2
  }
}

```

Типы элементов списка

URL: api/contenttype/getByList/{listId}

Метод: POST

Описание: получить информацию о типах контента списка по ID списка

Входные параметры:

listId: Guid - ID списка

Выходные параметры:

json объект

ContentType[] - массив типов контента списка

ContentType

{

id: Guid - ID типа контента

name: string - Название типа контента

description: string - описание типа контента

isFolder: boolean - могут ли элементы этого типа контента содержать в себе дочерние элементы

fieldList: Field[] - массив полей, которые привязаны к типу контента

}

Пример:

Получить типы элементов списка "Пользователи" (id e3a94bde-0ca9-456f-b338-4465d40389ee)

URL:

<http://my-site.ru/api/contentType/getByList/e3a94bde-0ca9-456f-b338-4465d40389ee>

```
[
{
  "description": "Функциональная группа",
  "parentId": "e1a0f60e-cafb-459f-aa14-721d9190e9d2",
  "isFolder": false,
  "isFile": false,
  "image": "7585a798-cb8a-439e-a306-b06e0073db6d",
  "fieldList": [
    {
      "internalName": "name",
      "description": "Название",
      "fieldTypeId": "f1e86fdc-58b8-eb11-bfe3-00155d013401",
      "fieldType": {
        "dbType": "VARCHAR(255)",
        "dotNetTypeName": "System.String",
        "reactComponent": "Input",
        "id": "f1e86fdc-58b8-eb11-bfe3-00155d013401",
        "name": "String",
        "isNew": false,
        "insertUser": {
          "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
          "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
          "fieldValueMap": {
            "name": "admin",
            "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
          }
        },
        "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
        "insertUserName": "admin",
        "insertDate": "2023-11-30T07:02:44.776582Z",
        "updateUser": {
          "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
          "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
          "fieldValueMap": {
            "name": "admin",
            "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
          }
        }
      },
    },
  ],
}
```

```
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2023-11-30T07:02:44.776582Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {}
},
"isRequired": true,
"isVisible": false,
"isReadOnly": false,
"component": null,
"id": "0d284b2f-5d1c-ed11-8015-00155d013401",
"name": "Название",
"isNew": false,
"insertUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",
"insertDate": "2023-11-30T07:02:44.776582Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
```

```

      "updateDate": "2023-11-30T07:02:44.776582Z",
      "deleteUser": null,
      "deleteUserId": "00000000-0000-0000-0000-000000000000",
      "deleteUserName": null,
      "deleteDate": null,
      "status": 1,
      "fieldValueMap": {
        "hidden": false
      }
    },
.....
  ],
  "viewId": "5895263d-fb23-4e9a-b7c0-199999421bd4",
  "parentContentTypeIdList": [
    "e1a0f60e-cafb-459f-aa14-721d9190e9d2"
  ],
  "id": "274e0738-d9ff-4482-84df-7b845d47fd8b",
  "name": "Функциональная группа",
  "isNew": false,
  "insertUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "insertUserName": "admin",
  "insertDate": "2023-11-30T07:02:44.776582Z",
  "updateUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  }
},

```

```

"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2023-11-30T07:02:44.776582Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {}
},
{
"description": "Подразделение",
"parentId": "e1a0f60e-cafb-459f-aa14-721d9190e9d2",
"isFolder": true,
"isFile": false,
"image": "31e99b39-fc33-4f3e-9974-b06e0073db6d",
"fieldList": [
{
"internalName": "name",
"description": "Название",
"fieldTypeId": "f1e86fdc-58b8-eb11-bfe3-00155d013401",
"fieldType": {
"dbType": "VARCHAR(255)",
"dotNetTypeName": "System.String",
"reactComponent": "Input",
"id": "f1e86fdc-58b8-eb11-bfe3-00155d013401",
"name": "String",
"isNew": false,
"insertUser": {
"id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
"fieldValueMap": {
"name": "admin",
"image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
}
}
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",

```

```
"insertDate": "2023-11-30T07:02:44.776582Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2023-11-30T07:02:44.776582Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {}
},
"isRequired": true,
"isVisible": false,
"isReadOnly": false,
"component": null,
"id": "0d284b2f-5d1c-ed11-8015-00155d013401",
"name": "Название",
"isNew": false,
"insertUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",
"insertDate": "2023-11-30T07:02:44.776582Z",
"updateUser": {
```

```

        "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
        "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
        "fieldValueMap": {
            "name": "admin",
            "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
        }
    },
    "updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "updateUserName": "admin",
    "updateDate": "2023-11-30T07:02:44.776582Z",
    "deleteUser": null,
    "deleteUserId": "00000000-0000-0000-0000-000000000000",
    "deleteUserName": null,
    "deleteDate": null,
    "status": 1,
    "fieldValueMap": {
        "hidden": false
    }
},
.....
],
"viewId": "a9aa69be-fd08-435d-8df1-01248294d709",
"parentContentTypeIdList": [
    "e1a0f60e-cafb-459f-aa14-721d9190e9d2"
],
"id": "733af6e2-e187-4e75-8a2b-ae4072f402e0",
"name": "Подразделение",
"isNew": false,
"insertUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
        "name": "admin",
        "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",

```

```

"insertDate": "2023-11-30T07:02:44.776582Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2023-11-30T07:02:44.776582Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {}
}
]

```

Папки /
элементы

Перемещение

URL: `api/item/move/{itemId}/{toItemId}/{moveItemType}`
Метод: POST
Описание: Переместить элемент
Входные параметры:

itemId - ID перемещаемого элемента
toItemId - ID элемента назначения
moveItemType - тип перемещения:
1 - выше элемента назначения
2 - в конец дочерних элементов элемента назначения
3 - ниже элемента назначения

Пример:

Переместить папку с id `b128f641-7169-419f-b84f-99fc34b51aff` в папку с id `4a862423-20fe-4798-b013-1ea0471d6552`

URL:

<http://my-site.ru/api/item/move/b128f641-7169-419f-b84f-99fc34b51aff/4a862423-20fe-4798-b013-1ea0471d6552/2>
{

```

"siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
"listId": "966e62c5-a803-49a0-a1be-e680d130c481",
"parentId": "4a862423-20fe-4798-b013-1ea0471d6552",
"itemPath": {
  "path": [
    "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",

```

```
"966e62c5-a803-49a0-a1be-e680d130c481",
"4a862423-20fe-4798-b013-1ea0471d6552"
},
"scopeId": "4a862423-20fe-4798-b013-1ea0471d6552",
"childCount": 0
},
"contentType": "09ad2c16-6047-4dfb-8274-6c4a5f1edbe5",
"permissionList": [
  "read",
  "create",
  "download",
  "update",
  "delete"
],
"id": "b128f641-7169-419f-b84f-99fc34b51aff",
"name": null,
"isNew": false,
"insertUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",
"insertDate": "2024-10-04T06:43:30.960319Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin",
    "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
```

		<pre> "updateDate": "2024-10-04T09:48:20.166882Z", "deleteUser": null, "deleteUserId": "00000000-0000-0000-0000-000000000000", "deleteUserName": null, "deleteDate": null, "status": 1, "fieldValueMap": { "name": "Test", "sequence": 1050 } } </pre>
Файлы	Создание	<p>URL: <code>api/item/update</code> Метод: POST Описание: Создать/обновить элемент списка (или массив элементов) Входные параметры: multipart/form-data</p> <p>itemListJson - строка, сериализованный в json массив элементов ItemUpdateRequest fileList - двоичные данные - массив файлов (экземпляр File)</p> <pre> { id: Guid - ID обновляемого элемента. Если создается новый элемент, то это поле не указывается (не обязательное) list_id: Guid - ID списка, элемент которого нужно обновить (обязательное) content_type_id: Guid - ID типа контента элемента. Указывается только при создании или при изменении типа контента существующего элемента (не обязательное) parent_id: Guid - ID типа родительского элемента. Указывается только при создании или при изменении родителя существующего элемента (не обязательное) file: int - ID тела файла, индекс из массива fileList . Указывается только при добавлении тела файла или при изменении тела существующего файла (не обязательное) ...поля типа контента элемента, которые нужно изменить, например: name: "новое имя" } </pre> <p>Пример:</p> <p>Загрузить файл список "Файлы" (id: 966e62c5-a803-49a0-a1be-e680d130c481) в папку с id 0b19b8bb-db6c-4057-9fdd-b79c2a0262e4</p> <p>URL: http://my-site.ru/api/item/update</p> <p>itemListJson:</p> <pre> [{ "list_id":"966e62c5-a803-49a0-a1be-e680d130c481", "content_type_id":"6065b2ae-63e3-44aa-a92d-2d414c30f808", "parent_id":"0b19b8bb-db6c-4057-9fdd-b79c2a0262e4", "name":"Пример ответа.xml","file":0 }] </pre> <p>fileList: (binary)</p> <pre> [</pre>

```

{
  "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
  "listId": "966e62c5-a803-49a0-a1be-e680d130c481",
  "parentId": "0b19b8bb-db6c-4057-9fdd-b79c2a0262e4",
  "itemPath": {
    "path": [
      "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
      "966e62c5-a803-49a0-a1be-e680d130c481",
      "c7196d6a-9918-4e14-a8bb-25eab2020524",
      "0b19b8bb-db6c-4057-9fdd-b79c2a0262e4"
    ],
    "scopeId": "4a862423-20fe-4798-b013-1ea0471d6552",
    "childCount": 0
  },
  "contentType": "6065b2ae-63e3-44aa-a92d-2d414c30f808",
  "permissionList": [
    "read",
    "create",
    "download",
    "update",
    "delete"
  ],
  "id": "8764e7a0-9a51-456b-81fe-9145345e17da",
  "name": null,
  "isNew": false,
  "insertUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin",
      "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
    }
  },
  "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "insertUserName": "admin",
  "insertDate": "2024-10-04T10:01:03.145883Z",
  "updateUser": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",

```

```
"contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
"fieldValueMap": {
  "name": "admin",
  "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"
}
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2024-10-04T10:01:03.145883Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "name": "Тестовый документ.docx",
  "document_status": {
    "id": "b9d061b1-7ce7-4756-8230-e502cfe3d8d8",
    "contentType": "2067cdf9-2683-4658-9eef-ae6800802a2f",
    "fieldValueMap": {
      "name": "Размещено"
    }
  }
},
"approval_count": 0,
"calc_field_test1": "--/Пример ответа.xml",
"sequence": 1070,
"file": {
  "id": "a3c65fbd-4e44-408d-91e7-fe5e6e6ad586",
  "contentType": "587f682a-fbf2-4647-8d91-e2f6460f59bf",
  "fieldValueMap": {
    "file_size": 6675,
    "upd_date": "2024-10-04T10:01:03.137597Z"
  }
},
"version": {
  "id": "b19601ab-a272-4db1-9ba4-332cc133cd1c",
  "contentType": "61895e0d-ba2b-48f3-8864-e0d0836c2e53",
  "fieldValueMap": {
```

```

        "name": "0.1",
        "version_minor": 1,
        "version_major": 0
    }
}
}
}
]

```

Получить список версий файла

URL: `api/fileversion/get/{sourceId}`

Метод: GET

Описание: получить список версий файла по ID файла

Входные параметры:

`sourceId`: Guid - ID элемента/файла

Выходные параметры:

массив – список версий файла

Пример:

Получить список версий файла с id `5bca4c6f-b7f6-47c8-974a-62d1622faf7e`

URL:

<http://my-site.ru/api/fileVersion/get/5bca4c6f-b7f6-47c8-974a-62d1622faf7e>

```

[
  {
    "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
    "listId": "2c269439-6848-4b3d-a7ad-4d55f1b9e499",
    "parentId": "2c269439-6848-4b3d-a7ad-4d55f1b9e499",
    "itemPath": {
      "path": [
        "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
        "2c269439-6848-4b3d-a7ad-4d55f1b9e499"
      ],
      "scopeId": "2c269439-6848-4b3d-a7ad-4d55f1b9e499",
      "childCount": 0
    },
    "contentType": "61895e0d-ba2b-48f3-8864-e0d0836c2e53",
    "permissionList": [
      "read",
      "create",
      "download",
      "update",
      "delete"
    ]
  }
]

```

```
},
  "id": "85945d0b-72fa-421c-b772-9ac10dc13652",
  "name": null,
  "isNew": false,
  "insertUser": {
    "id": "ba2bdc08-f633-5666-94f7-a91901fdf3da",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "Мисюра Дмитрий"
    }
  },
  "insertUserId": "ba2bdc08-f633-5666-94f7-a91901fdf3da",
  "insertUserName": "Мисюра Дмитрий",
  "insertDate": "2024-08-21T10:15:41.463385Z",
  "updateUser": {
    "id": "ba2bdc08-f633-5666-94f7-a91901fdf3da",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "Мисюра Дмитрий"
    }
  },
  "updateUserId": "ba2bdc08-f633-5666-94f7-a91901fdf3da",
  "updateUserName": "Мисюра Дмитрий",
  "updateDate": "2024-08-21T10:15:41.463385Z",
  "deleteUser": null,
  "deleteUserId": "00000000-0000-0000-0000-000000000000",
  "deleteUserName": null,
  "deleteDate": null,
  "status": 1,
  "fieldValueMap": {
    "source_id": "5bca4c6f-b7f6-47c8-974a-62d1622faf7e",
    "source_list_id": "966e62c5-a803-49a0-a1be-e680d130c481",
    "source_site_id": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
    "version_minor": 1,
    "version_major": 0,
    "file": {
      "id": "eabad87c-4329-495f-bc25-3454c909afae",
      "contentType": "587f682a-fbf2-4647-8d91-e2f6460f59bf",
```

		<pre> "fieldValueMap": { "file_size": 714488, "upd_date": "2024-08-21T10:15:42.956058Z" } }, "name": "0.1" } }] </pre>
Скачивание файла		<p>URL: <code>api/file/getbyitemid/{itemId}/{version}</code> Метод: GET Описание: получить файл по ID элемента/файла Входные параметры: itemId: Guid - ID элемента/файла</p> <p>version: имя версии файла (0.1, 1.0, 1.1, ...) Выходные параметры: тело файла с добавленным номером версии в конце имени файла.</p> <p>Пример:</p> <p>Скачать версию 0.1 файла с id <code>7efc4818-5e79-41ea-86d8-cf9bbcb3269c</code></p> <p>URL:</p> <p>http://my-site.ru/api/file/getbyitemid/7efc4818-5e79-41ea-86d8-cf9bbcb3269c/0.1 тело файла</p>
Бизнес-процессы	Создание бизнес-процесса	<p>URL: <code>api/item/update</code> Метод: POST Описание: Создать/обновить бизнес-процесс Входные параметры: multipart/form-data</p> <p>itemListJson - строка, сериализованный в json массив элементов <code>ItemUpdateRequest</code></p> <pre> { list_id: Guid – ID списка «Процессы», content_type_id: Guid – ID типа элемента запускаемого процесса, name: Наименование создаваемого процесса, start_date_plan: дата и время запуска процесса в формате "YYYY-MM-DDThh:mm:ss[.f{1,7}]<TZDSuffix>", importance: Guid – ID уровня важности процесса из списка «Важность», workflow_template: Guid – ID шаблона проесса из списка «Шаблоны процессов», (поля типа контента процесса, которые нужно изменить), reference_list: список вложений рабочего процесса [{ content_type_id: Guid – ID типа элемента «Вложение рабочего процесса» из списка «Типы контента», </pre>

destination: Guid – ID элемента, вкладываемого в процесс

```
},  
]  
}
```

Пример:

Создать бизнес-процесс согласования (id типа процесса b3d71386-28c7-4e13-aab8-f11a65d69dd6) с одним этапом, в качестве исполнителя указать единственного пользователя с id aab39600-92b4-4c38-a55c-53efac8db9cc, продолжительность этапа задать в 1 час, вложение в процесс - документс id 56fc1764-06ef-4bbc-ada2-88d16b4aa736

URL:

<http://my-site.ru/api/item/update>

itemListJson:

```
[  
  {  
    "list_id": "c63507f9-c892-4685-92e4-97b787199422",  
    "content_type_id": "b3d71386-28c7-4e13-aab8-f11a65d69dd6",  
    "parent_id": "c63507f9-c892-4685-92e4-97b787199422",  
    "name": "Согласовать /TEST", "start_date_plan": "2024-10-07T10:40:21.1307321Z",  
    "importance": "34ba677c-bd32-4322-a47e-28add7e09a85",  
    "initiator": "aab39600-92b4-4c38-a55c-53efac8db9cc",  
    "workflow_status": "e5ac8947-7a27-4a12-8e88-09c297377890",  
    "approval_user_list_1": [],  
    "approval_user_list_2": [],  
    "approval_user_list_final": ["aab39600-92b4-4c38-a55c-53efac8db9cc"],  
    "duration_plan_stage_final": 1,  
    "reference_list":  
    [  
      {  
        "content_type_id": "9ea7ecc3-0886-4b4b-9e40-b0a957d689dd",  
        "destination": "56fc1764-06ef-4bbc-ada2-88d16b4aa736"  
      }  
    ]  
  }  
]  
[  
  {  
    "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",  
    "listId": "c63507f9-c892-4685-92e4-97b787199422",  
    "parentId": "c63507f9-c892-4685-92e4-97b787199422",  
    "itemPath": {  
      "path": [  
        "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",  
        "c63507f9-c892-4685-92e4-97b787199422"  
      ],  
      "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",  
      "childCount": 0  
    }  
  }  
]
```

```

},
"contentTypeid": "b3d71386-28c7-4e13-aab8-f11a65d69dd6",
"permissionList": [
  "read",
  "create",
  "download",
  "update",
  "delete"
],
"id": "bf9ffbae-f773-4f63-af15-38b17036ec36",
"name": null,
"isNew": false,
"insertUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentTypeid": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin"
  }
},
"insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"insertUserName": "admin",
"insertDate": "2024-10-07T10:40:32.023427Z",
"updateUser": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentTypeid": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin"
  }
},
"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2024-10-07T10:40:32.023427Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "approval_user_list_1": [],
  "approval_user_list_2": [],
  "approval_user_list_final": [
    {
      "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
      "contentTypeid": "99c92e22-4e60-48c0-ab70-add500e71a45",
      "fieldValueMap": {
        "name": "admin"
      }
    }
  ]
},
"name": "Согласовать /TEST",
"sequence": 780,
"item_id": 78,
"start_date_plan": "2024-10-07T10:40:21.130732Z",
"importance": {
  "id": "34ba677c-bd32-4322-a47e-28add7e09a85",
  "contentTypeid": "55231344-ec21-46e5-ac17-a4d00dc21a79",
  "fieldValueMap": {
    "name": "Обычная"
  }
},
"initiator": {
  "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
  "contentTypeid": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "admin"
  }
},
"workflow_status": {
  "id": "e5ac8947-7a27-4a12-8e88-09c297377890",
  "contentTypeid": "dbcfff84-772e-4682-bd51-496369b43f2f",

```

```

        "fieldValueMap": {
            "name": "Не начат"
        }
    },
    "approval_user_list_1_lookup": [],
    "approval_user_list_2_lookup": [],
    "approval_user_list_final_lookup": [
        "aab39600-92b4-4c38-a55c-53efac8db9cc"
    ],
    "duration_plan_stage_final": 1
}
}
]

```

Запуск бизнес-процесса

URL: workflow/api/workflow/start/{processId}

Метод: POST

Описание: Запустить бизнес-процесс

Входные параметры:

processId - Guid – ID запускаемого процесса

Пример:

Запустить созданный в предыдущем примере процесс с id bf9ffbae-f773-4f63-af15-38b17036ec36

URL:

<http://my-site.ru/workflow/api/workflow/start/bf9ffbae-f773-4f63-af15-38b17036ec36>

```

{
    "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
    "listId": "c63507f9-c892-4685-92e4-97b787199422",
    "parentId": "c63507f9-c892-4685-92e4-97b787199422",
    "itemPath": {
        "path": [
            "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",
            "c63507f9-c892-4685-92e4-97b787199422"
        ],
        "scopeId": "bf9ffbae-f773-4f63-af15-38b17036ec36",
        "childCount": 0
    },
    "contentType": "b3d71386-28c7-4e13-aab8-f11a65d69dd6",
    "permissionList": [
        "read",
        "create",
        "download",
        "update",
        "delete"
    ],
    "id": "bf9ffbae-f773-4f63-af15-38b17036ec36",
    "name": null,
    "isNew": false,
    "insertUser": {
        "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
        "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
        "fieldValueMap": {
            "name": "admin"
        }
    },
    "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "insertUserName": "admin",
    "insertDate": "2024-10-07T10:40:32.023427Z",
    "updateUser": {
        "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
        "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
        "fieldValueMap": {
            "name": "admin"
        }
    }
}

```

```

"updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",
"updateUserName": "admin",
"updateDate": "2024-10-07T10:40:35.008649Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "approval_user_list_1": [],
  "approval_user_list_2": [],
  "approval_user_list_final": [
    {
      "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
      "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
      "fieldValueMap": {
        "name": "admin"
      }
    }
  ],
  "name": "Согласовать /TEST",
  "sequence": 780,
  "item_id": 78,
  "start_date_plan": "2024-10-07T10:40:21.130732Z",
  "start_date_fact": "2024-10-07T00:00:00Z",
  "end_date_plan": "2024-10-07T00:00:00Z",
  "importance": {
    "id": "34ba677c-bd32-4322-a47e-28add7e09a85",
    "contentType": "55231344-ec21-46e5-ac17-a4d00dc21a79",
    "fieldValueMap": {
      "name": "Обычная"
    }
  },
  "initiator": {
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",
    "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
    "fieldValueMap": {
      "name": "admin"
    }
  },
  "workflow_status": {
    "id": "e4334055-7f27-4aa4-8de7-bde7e77b330e",
    "contentType": "dbcfff84-772e-4682-bd51-496369b43f2f",
    "fieldValueMap": {
      "name": "Утверждающие"
    }
  },
  "workflow_route": {
    "id": "4c4d2875-be96-49c9-915f-bb92b7fa27c0",
    "contentType": "8698be63-2959-46e1-b511-3dd379c0b194",
    "fieldValueMap": {
      "name": "Согласовать /TEST"
    }
  },
  "approval_user_list_1_lookup": [],
  "approval_user_list_2_lookup": [],
  "approval_user_list_final_lookup": [
    "aab39600-92b4-4c38-a55c-53efac8db9cc"
  ],
  "duration_plan_stage_final": 1
}
}

```

Связи
Получить список
связей элемента

URL: `api/reference/ getByDestination/{itemId}`
Метод: POST
Описание: Получить список связей элемента по id элемента и id типа связи.
Входные параметры:
itemId: Guid - ID элемента

Json:

Список id типов связей, которые надо получить

```
["referenceTypeId1", "referenceTypeId2"]
```

Пример:

Получить список замечаний (id типа связи 4bc42916-9e3c-4c96-a25d-8a1b9b4bbc35) к файлу с id 4922dc08-bb93-ec14-a0a9-eae1ebf177bb

URL:

<http://my-site.ru/api/reference/getByDestination/4922dc08-bb93-ec14-a0a9-eae1ebf177bb>

Тело:

```
["4bc42916-9e3c-4c96-a25d-8a1b9b4bbc35"]
[
  {
    "siteId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
    "listId": "02b4fda6-01b9-4ddb-a7cd-7a56fa415f11",
    "parentId": "02b4fda6-01b9-4ddb-a7cd-7a56fa415f11",
    "itemPath": {
      "path": [
        "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",
        "02b4fda6-01b9-4ddb-a7cd-7a56fa415f11"
      ],
      "scopeId": "02b4fda6-01b9-4ddb-a7cd-7a56fa415f11",
      "childCount": 0
    },
    "contentType": "4bc42916-9e3c-4c96-a25d-8a1b9b4bbc35",
    "permissionList": [
      "read",
      "create",
      "download",
      "update",
      "delete"
    ],
    "id": "1f32dc08-63a6-7c67-afa3-b060f96084d1",
    "name": null,
    "isNew": false,
    "insertUser": {
      "id": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
      "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
      "fieldValueMap": {
        "name": "Левитин Андрей"
      }
    }
  }
]
```

```

},
"insertUserId": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
"insertUserName": "Левитин Андрей",
"insertDate": "2024-02-20T14:24:29.858865Z",
"updateUser": {
  "id": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
  "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
  "fieldValueMap": {
    "name": "Левитин Андрей"
  }
},
"updateUserId": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
"updateUserName": "Левитин Андрей",
"updateDate": "2024-02-20T14:24:29.858865Z",
"deleteUser": null,
"deleteUserId": "00000000-0000-0000-0000-000000000000",
"deleteUserName": null,
"deleteDate": null,
"status": 1,
"fieldValueMap": {
  "source": {
    "id": "1f32dc08-63a6-7c66-8379-d422ba2491d2",
    "contentType": "8231527e-3cba-4cb3-8196-4e2b13db362c",
    "fieldValueMap": {
      "end_date_fact": "2024-07-15T10:47:16.735948Z",
      "assignedto": {
        "id": "3a9a648b-98a0-43e2-b5bf-b06d0084303a",
        "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",
        "fieldValueMap": {
          "name": "Левитин Андрей"
        }
      }
    }
  }
},
"markup": "{\n  \"id\": \"new-1708439228600\", \"pageNr\": 1,\n    \"position\": {\n      \"x\": 343.1880005720824, \"y\": 390.1762013729977,\n      \"content\": {\n        \"creator\": \"\", \"date\": \"20.02.2024 17:27\",\n        \"label\": \"1-[object Promise]1\", \"desc\": \"\", \"svg\": {\n          \"id\": \"new-1708439228600\", 1708439228600,\n          \"pageNr\": 1, \"type\": \"rect\",\n          \"position\": {\n            \"x\": 363.1880005720824, \"y\": 405.1762013729977,\n
```

```

\width\":381.55606407322654,

      \"height\":279.80778032036613,

      \"stroke\":{\color\":\tibr\",width\":3},fill\":\none\"}},

      \"answer\": \"\",

      \"question_for_designer\": \"\",

      \"task_status\": {

        \"id\": \"646ac5e3-d5d4-4f88-b213-af0a0058c25c\",

        \"contentType\": \"2067cdf9-2683-4658-9eef-ae6800802a2f\",

        \"fieldValueMap\": {

          \"name\": \"Не начато\",

          \"color\": \"#2d9cdb\"

        }

      },

      \"item_id\": 259,

      \"sequence\": 2590,

      \"importance\": {

        \"id\": \"34ba677c-bd32-4322-a47e-28add7e09a85\",

        \"contentType\": \"55231344-ec21-46e5-ac17-a4d00dc21a79\",

        \"fieldValueMap\": {

          \"name\": \"Обычная\"

        }

      },

      \"name\": \"1\",

      \"description\": \"\"

    }

  },

  \"destination\": \"4922dc08-bb93-ec14-a0a9-eae1ebf177bb\",

  \"destination_version\": \"f2c01a74-c06a-473a-8ce3-b8ecf49e6837\"

}

]

```

Пользователи

Задать/изменить пароль пользователя

URL: api/security/setPassword
 Метод: POST
 Описание: Задать/изменить пароль пользователя (требуется права админа).
 Входные параметры:

json объект

```

{

  \"principal\": id пользователя, для которого надо задать пароль,

  \"password\": новый пароль пользователя,

```

"is_change_password_at_next_login": bool, требовать ли смену пароля при следующем входе в систему,

"is_two_factor_authentication_enabled": bool, включить двухфакторную авторизацию

}

Пример:

Установить пароль для пользователя с id ba3bdc08-f633-5666-94f7-a91901fdf3da

URL:

<http://my-site.ru/api/security/setPassword>

Тело:

{

"principal": "ba3bdc08-f633-5666-94f7-a91901fdf3da",

"password": "Pass123word",

"is_change_password_at_next_login": false,

"is_two_factor_authentication_enabled": false

}

-

Описание - версия 24.1.23

Сущность	Операции доступные через API	REST API
Все элементы	Чтение метаданных	<p>URL: api/item/get/{itemId}</p> <p>Метод: POST</p> <p>Описание: получить информацию об элементе по ID элемента</p> <p>Входные параметры: itemId: Guid - ID элемента списка</p> <p>Выходные параметры: json объект:</p> <pre>{ id: Guid - ID элемента siteId: Guid - ID сайта listId: Guid - ID списка parentId: Guid - ID родительского элемента itemPath: ItemPath - путь до элемента + количество дочерних элементов contentTypeId - ID типа контента элемента fieldValueMap - объект ключ-значение. Ключ - internalName поля, значение - значение этого поля в элементе }</pre>
	Создание/изменение	<p>URL: api/item/update</p> <p>Метод: POST</p> <p>Описание: Создать/обновить элемент списка (или массив элементов)</p> <p>Входные параметры: multipart/form-data</p> <p>itemListJson - строка, сериализованный в json массив элементов ItemUpdateRequest</p> <pre>{ list_id: Guid - ID списка, элемент которого нужно создать (обязательное) content_type_id: Guid - ID типа контента элемента (обязательное при создании элемента). parent_id: Guid - ID родительского элемента. ...поля типа контента элемента, которые нужно изменить, например: name: "новое имя" }</pre>
	Удаление	<p>URL: api/item/delete</p> <p>Метод: POST</p> <p>Описание: Удалить элемент (массив элементов)</p> <p>Входные параметры: itemListJson - строка, сериализованный в json массив элементов</p> <pre>[{ id : Guid - ID первого удаляемого элемента }, { id : Guid - ID второго удаляемого элемента }, ...]</pre>
	Установить уникальные права доступа к элементу	<p>URL: api/security/setItemUniquePermission/{itemId}</p> <p>Метод: POST</p> <p>Описание: Прервать наследование прав/ установить уникальные права доступа к элементу</p> <p>Входные параметры: itemId - ID элемента</p>
	Удалить уникальные права доступа к элементу	<p>URL: api/security/removeItemUniquePermission/{itemId}</p> <p>Метод: POST</p> <p>Описание: Восстановить наследование прав/ Удалить уникальные права доступа к элементу</p> <p>Входные параметры: itemId - ID элемента</p>

<p>Назначение прав доступа к элементу пользователю или группе</p>	<p>URL: api/item/update Метод: POST Описание: Задать права пользователю на элемент с уникальными правами. Задание прав происходит путем создания элемента «Разрешение пользователя» в списке «Разрывы прав». Входные параметры: multipart/form-data</p> <p>itemListJson - строка, сериализованный в json массив элементов ItemUpdateRequest</p> <pre>{ list_id: Guid - ID списка «Разрывы прав» (обязательное) , content_type_id: Guid – ID типа элемента «Разрешение пользователя» (обязательное), source: Guid - ID элемента, для которого задаются права (обязательное), principal: Guid - ID пользователя или группы (обязательное), permission_level: Guid - ID уровня доступа (обязательное) }</pre>
<p>Списки</p> <p>Чтение метаданных</p>	<p>URL: api/list/get/{listId} Метод: POST Описание: получить информацию о списке по ID списка Входные параметры: listId: Guid - ID списка Выходные параметры: json объект List</p> <pre>{ id: Guid - ID списка name: string - Название списка description: string - описание списка }</pre>
<p>Элементы списка</p>	<p>URL: api/item/getList/{parentId} Метод: POST Описание: получить информацию о всех дочерних элементах по ID родительского элемента Входные параметры: parentId - ID родительского элемента (ID списка или ID папки внутри списка) Выходные параметры: json объект:</p>
<p>Типы элементов списка</p>	<p>URL: api/contenttype/getByList/{listId} Метод: POST Описание: получить информацию о типах контента списка по ID списка Входные параметры: listId: Guid - ID списка Выходные параметры: json объект ContentType[] - массив типов контента списка</p>

		<pre> ContentType { id: Guid - ID типа контента name: string - Название типа контента description: string - описание типа контента isFolder: boolean - могут ли элементы этого типа контента содержать в себе дочерние элементы fieldList: Field[] - массив полей, которые привязаны к типу контента } </pre>
Папки / элементы	Перемещение	<p>URL: api/item/move/{itemId}/{toItemId}/{moveItemType} Метод: POST Описание: Переместить элемент Входные параметры:</p> <p>itemId - ID перемещаемого элемента toItemId - ID элемента назначения moveItemType - тип перемещения: 1 - выше элемента назначения 2 - в конец дочерних элементов элемента назначения 3 - ниже элемента назначения</p>
Файлы	Создание	<p>URL: api/item/update Метод: POST Описание: Создать/обновить элемент списка (или массив элементов) Входные параметры: multipart/form-data</p> <p>itemListJson - строка, сериализованный в json массив элементов ItemUpdateRequest fileList - двоичные данные - массив файлов (экземпляр File)</p> <pre> { id: Guid - ID обновляемого элемента. Если создается новый элемент, то это поле не указывается (не обязательное) list_id: Guid - ID списка, элемент которого нужно обновить (обязательное) content_type_id: Guid - ID типа контента элемента. Указывается только при создании или при изменении типа контента существующего элемента (не обязательное) parent_id: Guid - ID типа родительского элемента. Указывается только при создании или при изменении родителя существующего элемента (не обязательное) file: int - ID тела файла, индекс из массива fileList . Указывается только при добавлении тела файла или при изменении тела существующего файла (не обязательное) ...поля типа контента элемента, которые нужно изменить, например: name: "новое имя" } </pre>
	Получить список версий файла	<p>URL: api/fileversion/get/{sourceId} Метод: GET</p> <p>Описание: получить список версий файла по ID файла</p> <p>Выходные параметры: массив – список версий файла</p>
	Скачивание файла	<p>URL: api/file/getbyitemid/{itemId}/{version} Метод: GET Описание: получить файл по ID элемента списка Входные параметры: itemId: Guid - ID элемента списка</p> <p>version: имя версии файла (0.1, 1.0, 1.1, ...) Выходные параметры: тело файла</p>
Бизнес-процессы	Создание бизнес-процесса	<p>URL: api/item/update Метод: POST Описание: Создать/обновить бизнес-процесс</p>

		<p>Входные параметры: multipart/form-data</p> <p>itemListJson - строка, сериализованный в json массив элементов ItemUpdateRequest</p> <pre>{ list_id: Guid – ID списка «Процессы», content_type_id: Guid – ID типа элемента запускаемого процесса, name: Наименование создаваемого процесса, start_date_plan: дата и время запуска процесса в формате "YYYY-MM-DDThh:mm:ss[.f{1,7}]<TZDSuffix>", importance: Guid – ID уровня важности процесса из списка «Важность», workflow_template: Guid – ID шаблона проесса из списка «Шаблоны процессов», (поля типа контента процесса, которые нужно изменить), reference_list: список вложений рабочего процесса [{ content_type_id: Guid – ID типа элемента «Вложение рабочего процесса» из списка «Типы контента», destination: Guid – ID элемента, вкладываемого в процесс },] }</pre>
	<p>Запуск бизнес-процесса</p>	<p>URL: workflow/api/workflow/start/{processId}</p> <p>Метод: POST Описание: Запустить бизнес-процесс</p> <p>Входные параметры:</p> <p>processId - Guid – ID запускаемого процесса</p>
<p>Вход в систему</p>	<p>Получение токена</p>	<p>URL: api/user/login Метод: POST Описание: Авторизация Входные параметры: json объект</p> <pre>{ "UserName": "логин", "Password": "пароль" }</pre> <p>Ответ: токен пользователя</p> <p>(Токен пользователя передается в заголовке Authorization)</p>

Пример работы с API на Python

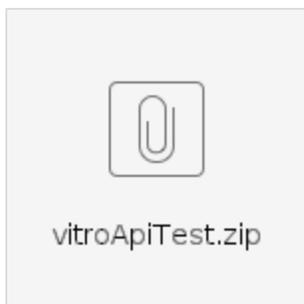
В данном примере код выполняет следующие действия:

1. Авторизация. Получение токена (login).
2. Создание папки на списке "Файлы" (createFolder).
3. Загрузка файла в созданную папку (uploadFile).
4. Изменение статуса файла на "Согласовано" (updateStatus).
5. Удаление файла и папки (delete).

Для запуска необходимо:

1. Установить Python и библиотеку requests.
2. Скачать и распаковать архив "vitroApiTest.zip".
3. В файле "vitroApiTest.py":
 - a. Подставить свои данные для входа в систему (loginData).
 - b. Подставить адрес системы (appBaseUrl).
4. Запустить выполнение программы.

Скачать архив с примером:



Настройка системы с использованием командной строки (CLI)

Настройка системы с использованием командной строки осуществляется путем отправки соответствующего HTTP-запроса к REST-сервису системы.

Для отправки запросов можно использовать популярную кроссплатформенную программу Curl (доступна для Bash, Windows PowerShell и т.д.).

Содержание статьи:

- [Авторизация в системе и получение токена](#)
- [Создание/изменение элементов](#)
- [Создание атрибутов](#)
- [Создание типов элементов](#)
- [Добавление атрибута к типу элемента](#)
- [Создание списков](#)
- [Добавление типов элементов к списку](#)

Авторизация в системе и получение токена

Для выполнения любых настроек системы необходима первичная авторизация в системе. Настройки должны выполняться от имени пользователя с правами администратора.

Авторизация в системе осуществляется путем указания токена авторизации в заголовке Authorization.

Для получения токена необходимо выполнить POST запрос к эндпоинту `api/user/login` в котором передать пару логин-пароль пользователя, от имени которого должна происходить авторизация.

```
curl -X POST http://{site_url}/api/user/login \  
-H "Content-Type: application/json" \  
-d '{"Username": "{my_login}", "Password": "{my_password}"}'
```

Где:

`site_url` – адрес сайта Vitro

`Username: {my_login}` – логин пользователя

`Password: {my_password}` – пароль пользователя

В ответ будет получен JSON в котором содержится список атрибутов пользователя и их значение, а также атрибут `token` значение которого потребуется указывать в заголовке `Authorization` при последующих обращениях к системе.

Также в ответе содержится атрибут `expires`, в котором указана дата действия полученного токена и после которой его необходимо будет получить заново.

```
{  
  "id": "516fdc08-42d3-113f-82d5-2487742d9725",  
  "user": {  
    "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",  
    "name": "admin",  
    "email": "adminMP@vitrosoft.ru",  
    "item": {  
      "siteId": "3064dc08-2e02-8de4-aa70-1b2ae9eb890b",  
      "listId": "e3a94bde-0ca9-456f-b338-4465d40389ee",  
      "parentId": "c0857f66-fbfa-448f-a35d-afac00a3a9cb",  
      "itemPath": {  
        "path": ["3064dc08-2e02-8de4-aa70-1b2ae9eb890b",  
          "e3a94bde-0ca9-456f-b338-4465d40389ee",  
          "c0857f66-fbfa-448f-a35d-afac00a3a9cb"],  
        "scopeId": "8d0bc7d7-f8af-ec11-bfeb-000c29f44205",  
        "childCount": 0,  
        "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",  
        "permissionList": null,  
        "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",  
        "name": null,  
        "isNew": false,  
        "insertUser": {  
          "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",  
          "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",  
          "fieldValueMap": {  
            "name": "admin",  
            "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"}  
          },  
          "insertUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",  
          "insertUserName": "admin",  
          "insertDate": "2023-11-30T07:02:44.776582Z",  
          "updateUser": {  
            "id": "aab39600-92b4-4c38-a55c-53efac8db9cc",  
            "contentType": "99c92e22-4e60-48c0-ab70-add500e71a45",  
            "fieldValueMap": {  
              "name": "admin",  
              "image": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac"}  
            },  
            "updateUserId": "aab39600-92b4-4c38-a55c-53efac8db9cc",  
            "updateUserName": "admin",  
            "updateDate": "2024-04-10T14:33:53.083293Z",  
            "deleteUser": null,  
            "deleteUserId": "00000000-0000-0000-0000-000000000000",  
            "deleteUserName": null,  
            "deleteDate": null,  
            "status": 1,  
            "fieldValueMap": {  
              "sequence": 60,  
              "email": "adminMP@vitrosoft.ru",  
              "login": "admin",  
              "image": {  
                "id": "ee17dc08-4e9e-2c7d-99b7-db3c3e1fa4ac",  
                "contentType": "587f682a-fbf2-4647-8d91-e2f6460f59bf",  
                "fieldValueMap": {  
                  "name": "709921583.jpeg"}  
                },  
                "name": "admin"}  
              },  
              "isActive": true,  
              "isAdmin": true,  
              "groupList": ["3064dc08-2e02-8de4-aa70-1b2ae9eb890b",  
                "e3a94bde-0ca9-456f-b338-4465d40389ee",  
                "c0857f66-fbfa-448f-a35d-afac00a3a9cb",  
                "aab39600-92b4-4c38-a55c-53efac8db9cc",  
                "6ea6149b-5eac-4636-adf1-b01b00b7de4e"}  
            },  
            "token": "qN8CHhUGRn5OXpF+g56kDHIvRDr+LF3gaqqdu1B6LQY_",  
            "expires": "2024-05-11T11:27:20.912149Z",  
            "status": 1,  
            "loginTypeList": [{"id": "05ef8c8e-f104-4dc0-8887-b97c1475dd7a",  
              "hidden": false}, {"id": "17a6ceda-cc38-416c-9f72-379d978a142f",  
              "hidden": false, "is_two_factor_authentication_enabled": true}],  
            "requireLoginInactivityTimeoutMinuteCount": 0  
          }  
        }  
      }  
    }  
  }  
}
```

Создание/изменение элементов

Базовой сущностью в системе Vitro является элемент. Частными случаями элементов являются:

1. Атрибут

2. Тип элемента
3. Список
4. Документ
5. Папка

Создание или изменение любых элементов системы происходит путем отправки POST-запроса к эндпоинту `api/item/update`, в котором передается список создаваемых элементов, с указанием обязательных параметров:

1. Guid родительского списка (списка в котором они должны находиться)
2. Guid типа создаваемого элемента
3. Набора атрибутов и их значений, соответствующих указанному типу элемента.

Также могут быть указаны дополнительные параметры:

1. Guid родительского элемента (если создаваемый элемент должен находится внутри другого элемента – папки)
2. Guid изменяемого элемента (при изменении существующего элемента)

Для примера приведено создание элемента типа «Папка» с именем «Папка_1» в списке «Файлы» внутри другого элемента этого списка.

```
curl -X POST http://{site_url}/api/item/update \
-H "Authorization: {token}" \
-H "Content-Type: multipart/form-data" \
-F "itemListJson=[{\\"list_id\\":\\"966e62c5-a803-49a0-a1be-e680d130c481\\",\\"parent_id\\":\\"5164dc08-adf0-e80d-89e8-3f0ca000ed98\\",\\"content_type_id\\":\\"09ad2c16-6047-4dfb-8274-6c4a5f1edbe5\\",\\"name\\":\\"_1\\"}]"
```

Где:

site_url – адрес сайта Vitro

Authorization: {token} – токен, полученный на этапе авторизации

list_id: "966e62c5-a803-49a0-a1be-e680d130c481" - Guid списка «Файлы»

parent_id: "5164dc08-adf0-e80d-89e8-3f0ca000ed98" - Guid элемента, внутри которого хотим создать наш элемент

content_type_id: "ea164a18-7eb5-44f7-a638-5bb7c05221e6" – Guid типа "Папка"

name: "Папка_1" – название типа элемента

Для примера приведено изменение имени элемента созданного в предыдущем примере.

```
curl -X POST http://{site_url}/api/item/update \
-H "Authorization: {token}" \
-H "Content-Type: multipart/form-data" \
-F "itemListJson=[{\\"list_id\\":\\"966e62c5-a803-49a0-a1be-e680d130c481\\",\\"parent_id\\":\\"5164dc08-adf0-e80d-89e8-3f0ca000ed98\\",\\"id\\":\\"4673dc08-3441-f03d-9e46-5866102fa3a2\\",\\"content_type_id\\":\\"09ad2c16-6047-4dfb-8274-6c4a5f1edbe5\\",\\"name\\":\\"_2\\"}]"
```

Где:

site_url – адрес сайта Vitro

Authorization: {token} – токен, полученный на этапе авторизации

list_id: "966e62c5-a803-49a0-a1be-e680d130c481" - Guid списка «Файлы»

parent_id: "5164dc08-adf0-e80d-89e8-3f0ca000ed98" - Guid элемента, внутри которого хотим создать наш элемент

id: "4673dc08-3441-f03d-9e46-5866102fa3a2" - Guid элемента, который требуется изменить

content_type_id: "ea164a18-7eb5-44f7-a638-5bb7c05221e6" – Guid типа "Папка"

name: "Папка_2" – новое название типа элемента

Создание атрибутов

Создание атрибута происходит путем создания элемента требуемого типа в списке "Атрибуты" через отправку POST-запроса, в котором передается список создаваемых атрибутов, с указанием их родительского списка, типа элемента (атрибута) и набора свойств и их значений, соответствующих указанному типу атрибута.

```
curl -X POST http://{site_url}/api/item/update \
-H "Authorization: {token}" \
-H "Content-Type: multipart/form-data" \
-F "itemListJson=[{\"list_id\":\"a01f8ee1-a912-41b9-b87b-312d6583e38a\",\"content_type_id\":\"{attribute_type_id}\",\"name\":\"{attribute_name}\",\"internal_name\":\"{attribute_internal_name}\"}]"
```

Где:

site_url – адрес сайта Vitro

Authorization: {token} – токен, полученный на этапе авторизации

list_id: "a01f8ee1-a912-41b9-b87b-312d6583e38a" - Guid списка "Атрибуты"

content_type_id: {attribute_type_id} – Guid типа атрибута

name: {attribute_name} – название атрибута

internal_name: {attribute_internal_name} – системное имя атрибута в формате snake_case

помимо указанных обязательных свойств (имя и системное имя) можно указать также остальные, специфичные для создаваемого типа атрибута свойства и их значения.

При необходимости можно получить список доступных для создания типов атрибутов и их свойств через POST-запроса к эндпоинту *api/contenttype/getByList*

```
curl -X POST http://{site_url}/api/contenttype/getByList/a01f8ee1-a912-41b9-b87b-312d6583e38a \
-H "Content-Type: application/json" \
-H "Authorization: {token}"
```

Где:

site_url – адрес сайта Vitro

Authorization: {token} – токен, полученный на этапе авторизации

a01f8ee1-a912-41b9-b87b-312d6583e38a – Guid списка "Атрибуты"

Создание типов элементов

Создание типов элементов системы происходит путем создания элемента в списке "Тип контента" через отправку POST-запроса, в котором передается список создаваемых элементов, с указанием родительского списка, родительского типа элемента (элемента от которого создаваемый наследует атрибуты), типа создаваемого элемента и набора атрибутов и их значений, соответствующих указанному типу элемента.

```
curl -X POST http://{site_url}/api/item/update \
-H "Authorization: {token}" \
-H "Content-Type: multipart/form-data" \
-F "itemListJson=[{\"list_id\":\"27155e95-1874-47f7-9ef7-042e3631159f\",\"content_type_id\":\"ea164a18-7eb5-44f7-a638-5bb7c05221e6\",\"name\":\"{element_name}\",\"folder\":{\"true/false\"},\"description\":\"{element_description}\"}]"
```

Где:

site_url – адрес сайта Vitro

Authorization: {token} – токен, полученный на этапе авторизации

list_id: "27155e95-1874-47f7-9ef7-042e3631159f" - Guid списка "Тип контента"

content_type_id: "ea164a18-7eb5-44f7-a638-5bb7c05221e6" – Guid типа элемента "Тип контента"

name: {element_name} – название создаваемого типа элемента

folder: {true/false} – является ли создаваемый тип элемента папкой

description: {element_description} – описание типа элемента

Добавление атрибута к типу элемента

Добавление атрибутов к типу элемента происходит путем создания элемента типа «Поля типов контента» (связь типа элемента и атрибута) в списке «Поля типов контента» через отправку POST-запроса, в котором передается список элементов, для которых требуется добавить атрибут, с указанием типа элемента и типа добавляемого атрибута.

```
curl -X POST http://{site_url}/api/item/update \  
-H "Authorization: {token}" \  
-H "Content-Type: multipart/form-data" \  
-F "itemListJson=[{\"list_id\":\"d702e382-b9a9-456b-b94f-66ad1df2eb0e\",\"content_type_id\":\" c0639c62-a384-4676-8002-5d9edcfdc473\",\"name\":\"{element_name}\",\"content_type\":\"{id_elem_to_be_updated}\",\"field\":\"{id_attribute_to_be_added}\",\"required\":{\"true/false},\"read_only\":{\"true/false}}]"
```

Где:

site_url – адрес сайта Vitro

Authorization: {token} – токен, полученный на этапе авторизации

list_id: "d702e382-b9a9-456b-b94f-66ad1df2eb0e" – Guid списка «Поля типов контента»

content_type_id: "c0639c62-a384-4676-8002-5d9edcfdc473" – Guid типа элемента «Поля типов контента»

content_type: {id_elem_to_be_updated} – Guid типа элемента к которому требуется добавить атрибут

field: {id_attribute_to_be_added} – Guid типа добавляемого атрибута

name: {element_name} – название создаваемого элемента

required: {true/false} – является ли добавляемый атрибут обязательным

read_only: {true/false} – является ли добавляемый атрибут редактируемым

Создание списков

Создание списка происходит путем создания элемента типа «Список» (связь типа элемента и атрибута) в списке «Списки» через отправку POST-запроса, в котором передается список создаваемых элементов (списков), с указанием их названий и других параметров.

```
curl -X POST http://{site_url}/api/item/update \  
-H "Authorization: {token}" \  
-H "Content-Type: multipart/form-data" \  
-F "itemListJson=[{\"list_id\":\"3c36345d-599b-4711-a290-66e48015ab32\",\"content_type_id\":\" cbdb9089-49c4-4742-8ce6-5f1f7a01e679\",\"name\":\"{list_name}\",\"table_name\":\"{list_table_name}\",\"hidden\":{\"true/false},\"document_archive\":{\"true/false}}]"
```

Где:

site_url – адрес сайта Vitro

Authorization: {token} – токен, полученный на этапе авторизации

list_id: "3c36345d-599b-4711-a290-66e48015ab32" – Guid списка «Списки»

content_type_id: "cbdb9089-49c4-4742-8ce6-5f1f7a01e679" – Guid типа элемента «Список»

name: {list_name} – название создаваемого списка

table_name: {list_table_name} – системное имя таблицы, создаваемой в БД для данного списка. Имя таблицы должно указываться в формате snake_case.

hidden: {true/false} – является ли создаваемый список скрытым

document_archive: {true/false} – является ли создаваемый список хранилищем файлов

Добавление типов элементов к списку

Для того, чтобы список мог содержать элементы определенных типов их необходимо добавить к нужному списку. Добавление элементов к типу элемента происходит путем создания элемента типа «Типы контента списка» (связь списка и типа элемента) в списке «Типы контента списка» через отправку POST-запроса, в котором передается список со значениями Guid привязываемых элементов и Guid списков, к которым эти элементы привязываются.

```
curl -X POST http://{site_url}/api/item/update \  
-H "Authorization: {token}" \  
-H "Content-Type: multipart/form-data" \  
-F "itemListJson=[{\"list_id\":\"a15fc818-68bf-4593-ad80-e098bea64d6c\",\"content_type_id\":\"72a9a18a-c27a-448c-a88e-c7e5f83d287e \",\"name\":\"{element_name}\",\"content_type\":\"{id_elem_to_be_linked}\",\"list\":\"{id_list_to_be_linked}\"}]"
```

Где:

site_url – адрес сайта Vitro

Authorization: {token} – токен, полученный на этапе авторизации

list_id: "a15fc818-68bf-4593-ad80-e098bea64d6c" - Guid списка "Типы контента списка"

content_type_id: "72a9a18a-c27a-448c-a88e-c7e5f83d287e " – Guid типа элемента "Типы контента списка"

content_type: {id_elem_to_be_linked} – Guid типа элемента который требуется привязать к списку

list: {id_list_to_be_linked} - Guid списка, к которому привязывается элемент

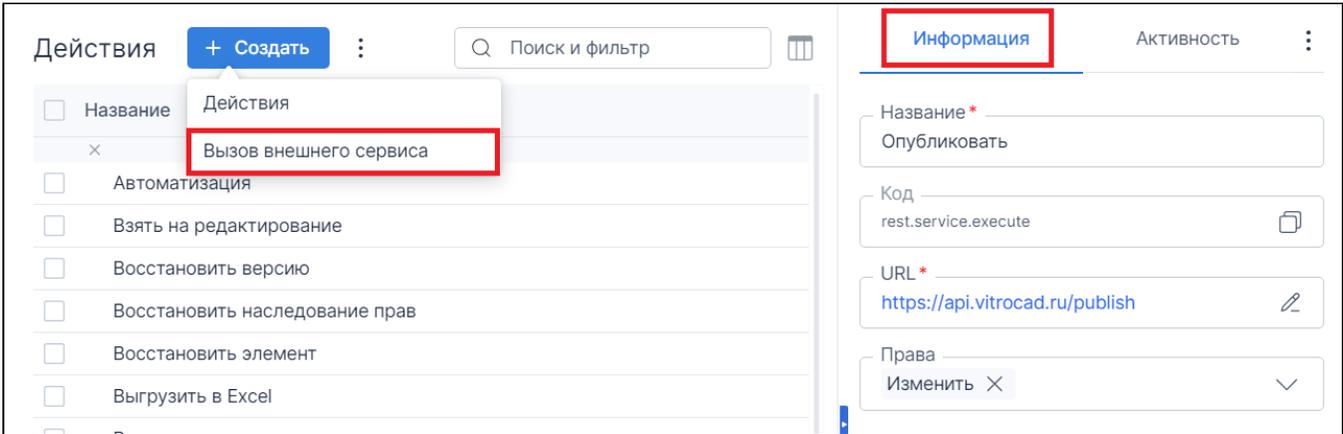
name: {element_name} – название создаваемого элемента (связи)

Создание действия с вызовом REST сервиса

Создание действия

1. В настройках пространства Центр администрирования в разделе Списки по двойному клику ЛКМ перейти на список Действия
2. Создать элемент с типом Вызов внешнего сервиса
3. На вкладке Информация заполнить Атрибуты:

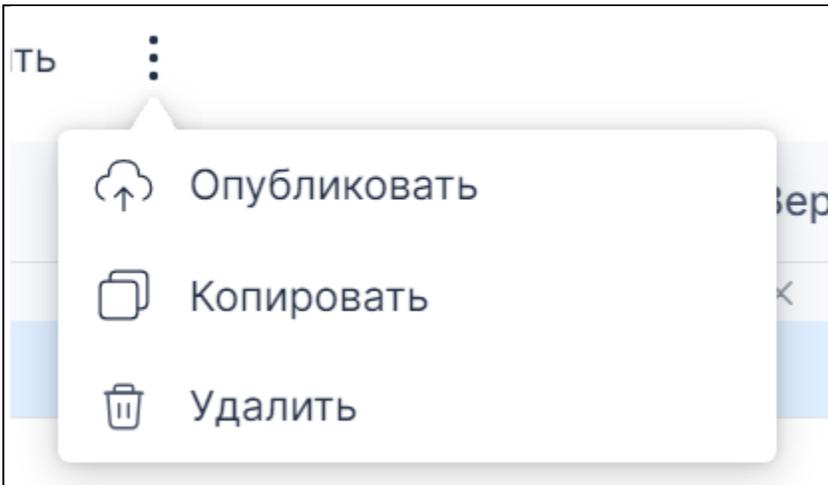
- Название
- URL - url сервиса. Пример: <https://api.vitrocad.ru/publish>
- Права - права, требуемые пользователю для выполнения действия



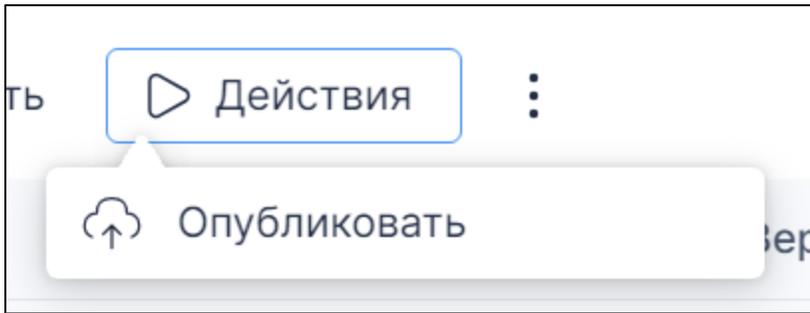
Привязка действия к типу элемента

1. В настройках пространства Центр администрирования в разделе Списки по двойному клику ЛКМ перейти на список Действие типа элемента
2. Создать элемент с типом Действие типа элемента
3. На вкладке Информация заполнить атрибуты:

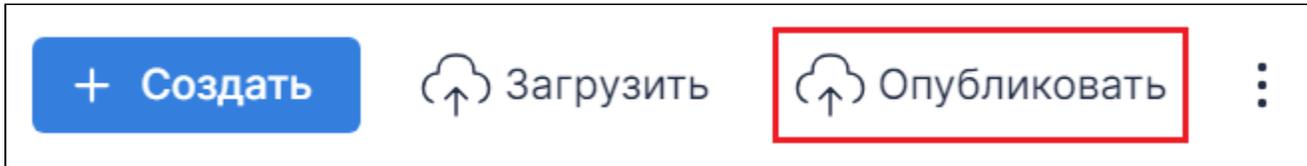
- Название
- Действие - из списка выбрать действие, которое необходимо привязать к типу элемента
- Тип элемента - из списка выбрать тип элемента, к которому привязывается действие
- Изображение - иконка действия
- Подтверждение - требует ли действие подтверждение пользователя
- Показать после действия - отображать ли сообщение об успешности выполнения действия
- Показать прогресс - отображать ли прогресс выполнения действия
- Контекст - отображать ли действие в контекстном меню



- Дополнение - отображать ли действие в дополнительной кнопке "Действия" в командном меню



- Если атрибуты "Контекст" и "Дополнение" имеют значение false, действие будет отображаться отдельной кнопкой в командном меню

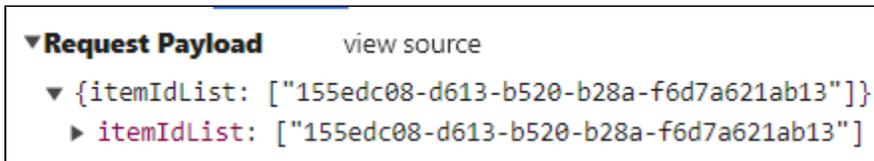


Вызов действия

При вызове действия отправляется POST запрос на указанный в действии URL.
В заголовке Authorization передается токен пользователя.

В теле запроса передается JSON вида:

```
{
  itemIdList: [...] - id
}
```



Для отображения текста ошибки, ответ должен приходить в виде строки, либо в формате JSON вида:

```
{
  exception: {
    message: ''
  }
}
```

```
▼ {exception: {itemId: "2e5edc08-1424-ad96-b902-fd57962b80b0",...}, status: 500,...}
▼ exception: {itemId: "2e5edc08-1424-ad96-b902-fd57962b80b0",...}
  ▶ field: {internalName: "name", description: "Название", fieldType: "f1e86fdc-58b8-e..."}
    hResult: -2146233088
    innerException: null
    itemId: "2e5edc08-1424-ad96-b902-fd57962b80b0"
    message: "Exception of type 'Vitro.Server.Core.Exceptions.ItemExistsFieldValueException'."
    source: "Vitro.Server.Core.Impl"
    stackTrace: "    at Vitro.Server.Core.Services.Impl.DocumentArchiveItemNameValidation.ValidateItemName(String name, String itemId)"
    value: "nanka"
  status: 500
  title: "Exception of type 'Vitro.Server.Core.Exceptions.ItemExistsFieldValueException'."
  traceId: "0HN5ACRILEN2T:00000005"
  type: "Vitro.Server.Core.Exceptions.ItemExistsFieldValueException"
```

Вызов REST сервиса из обработчика действия

Интерфейсная библиотека Vitro.Server.Core.dll содержит ряд интерфейсов для реализации логики обработки определенных событий в системе.

- `ItemEventReceiverBeforeCreateService` - вызов перед созданием нового элемента.
- `ItemEventReceiverAfterCreateService` - вызов после создания нового элемента.
- `ItemEventReceiverBeforeUpdateService` - вызов перед обновлением элемента.
- `ItemEventReceiverAfterUpdateService` - вызов после обновления элемента.
- `ItemEventReceiverBeforeDeleteService` - вызов перед удалением элемента.

Интерфейсы с префиксом "Before", как правило, используются, если необходимо реализовать дополнительные проверки (валидацию).

Демонстрационный проект - [VitroMPReceiverDemo.zip](#)

В проекте реализованы 3 примера. В каждом из примеров вызывается внешний REST сервис, для примера, используются REST сервисы самого портала.

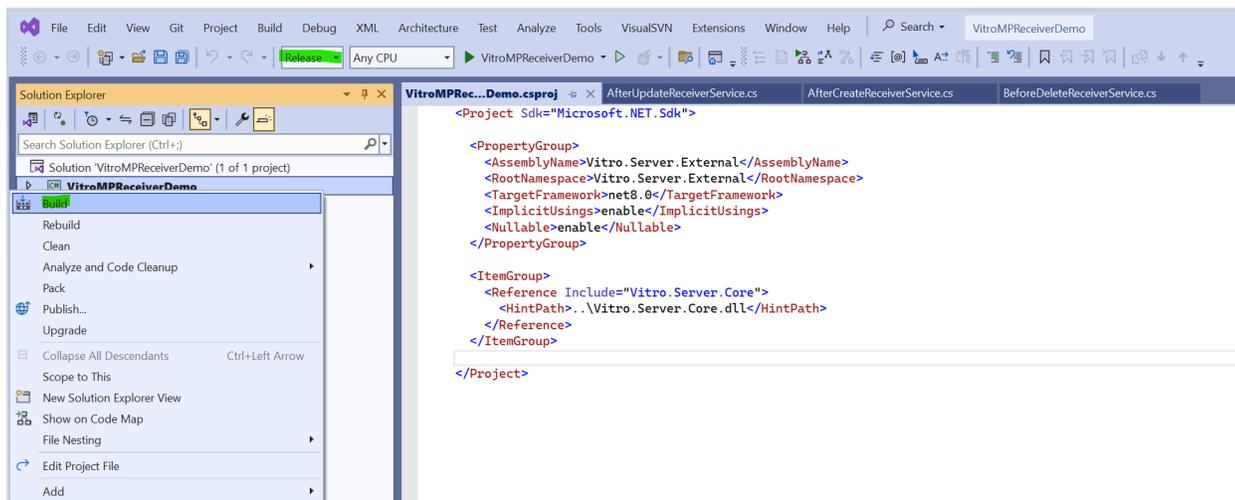
- `AfterCreateReceiverService` - публикация чистой версии документа 1.0 после добавления.
- `AfterUpdateReceiverService` - при смене статуса документа с "Размещено" на "Отклонено" выполняется прерывание наследования прав на текущем документе и права очищаются.
- `BeforeDeleteReceiverService` - перед удалением документа выполняется проверка на наличие не закрытых замечаний для документа и генерируется исключение, если замечания есть и документ не удаляется.

Сборка и размещение "Vitro.Server.External.dll".

1. В архиве проект библиотеки (Vitro.Server.External.dll) с исходниками. Проект сделан на языке c# в среде выполнения .net 8.0 и с помощью среды разработки Visual Studio 2022.

2. Проект необходимо открыть в VS 2022 и выполнить его сборку командой "Build", предварительно выбрав конфигурацию сборки "Release", как показано на рисунке. После сборки проекта в папке "\bin\Release" должен появиться файл "Vitro.Server.External.dll".

Примечание! Перед сборкой проекта надо в исходниках заменить все URL-адреса при вызове REST-сервисов на актуальные, соответствующие текущей площадке.



3. Скомпилированную библиотеку из п.2 необходимо разместить по пути и выполнить перезапуск процесса портала Vitro.Server.Core.Web.

- для Windows: "c:\Program Files\Vitro Software\Vitro\Server\Solutions\External"
- для Linux : "/usr/local/Vitro.Server.Core.Web/Vitro/Server/Solutions/External"

Примечание: в папку "\Solutions\External" надо скопировать только библиотеку "Vitro.Server.External.dll", другие файлы копировать не надо.

Поиск элементов по структуре - версия 25.1.39.16

Есть два способа получения списка элементов:

1. URL: `api/item/getList/{parentId}`
Метод: POST
Описание: получить информацию о всех дочерних элементах по ID родительского элемента
Входные параметры:
`parentId` - ID родительского элемента (ID списка или ID папки внутри списка)
2. URL: `api/item/getRecursive/{parentId}`
Метод: POST
Описание: рекурсивно получить информацию о всех дочерних элементах по ID родительского элемента (дочерние элементы + их дочерние элементы и т.д.)
Входные параметры:
`parentId` - ID родительского элемента (ID списка или ID папки внутри списка)

В теле POST запроса для этих методов можно передать строку с фильтром, который будет применяться для поиска элементов:
{ "query": "item => условия, которым должны удовлетворять искомые элементы"}

Синтаксис строки фильтра

Пример. Найти всех пользователей, у которых email равен "test@vitrocad.ru".

URL: `/api/item/GetRecursive/e3a94bde-0ca9-456f-b338-4465d40389ee`

Тело POST запроса:

```
{ "query": "item => item.ContentTypeId == Guid(\"99c92e22-4e60-48c0-ab70-add500e71a45\") && item.GetValueAsString(\"email\") == \"test@vitrocad.test\"" }
```

`e3a94bde-0ca9-456f-b338-4465d40389ee` - ID списка "Пользователи"

`99c92e22-4e60-48c0-ab70-add500e71a45` - это ID типа контента пользователей

Поля и методы, которые доступны у `item`:

Поле или метод	Тип	Описание
<code>item.ContentTypeId</code>	Guid	ID типа элемента
<code>item.InsertDate</code>	Дата/время	Дата/время создания элемента
<code>item.UpdateDate</code>	Дата/время	Дата/время обновления элемента
<code>item.GetValueAsDateTime("название поля")</code>	Дата/время	Получить значение поля типа "дата"
<code>item.GetValueAsString("название поля")</code>	Строка	Получить значение поля типа "однорочный текст"/"многострочный текст"
<code>item.GetValueAsBool("название поля")</code>	Флаг	Получить значение поля типа "флаг"
<code>item.GetValueAsInt("название поля")</code>	Целое число	Получить значение поля типа "целое число"
<code>item.GetLookupId("название поля")</code>	GUID	Получить значение поля типа "ссылка на элемент списка"

Операторы сравнения

Оператор	Описание	Пример
<code>==</code>	Равно	<code>item.GetValueAsString("email") == "test@vitrocad.ru"</code>
<code>!=</code>	Не равно	<code>item.ContentTypeId != Guid("99c92e22-4e60-48c0-ab70-add500e71a45")</code>
<code>></code>	Больше	<code>item => item.InsertDate > DateTime(2024, 10, 1, 10, 0, 0)</code>
<code>>=</code>	Больше или равно	<code>item => item.InsertDate >= DateTime(2024, 10, 1, 10, 0, 0)</code>
<code><</code>	Меньше	<code>item => item.InsertDate < DateTime(2024, 10, 1, 10, 0, 0)</code>
<code><=</code>	Меньше или равно	<code>item => item.InsertDate <= DateTime(2024, 10, 1, 10, 0, 0)</code>

Искомые значения

--	--	--

Значения	Описание	Пример
"Строка"	Строка текста	item.GetValueAsString("email") == "test@vitrocad.ru"
Guid("99c92e22-4e60-48c0-ab70-add500e71a45")	Создать объект типа GUID из указанной строки	item.ContentTypeId == Guid("99c92e22-4e60-48c0-ab70-add500e71a45")
DateTime(2024, 10, 1, 10, 0, 0)	Создать объект типа Дата/Время По указанным параметрам DateTime(Год, Месяц, День, Час, Минута, Секунда) Обратите внимание: время передается в UTC	item.InsertDate > DateTime(2024, 10, 1, 10, 0, 0)
new Guid[] { Guid("b9d061b1-7ce7-4756-8230-e502cfe3d8d8"), Guid("5cd0640c-1e57-4e09-b0ec-babfb5c72680") }	Создать объект типа массив GUID	new Guid[] { Guid("b9d061b1-7ce7-4756-8230-e502cfe3d8d8"), Guid("5cd0640c-1e57-4e09-b0ec-babfb5c72680") }.Contains(item.GetLookupId("document_status"))

Логические операторы

Оператор	Описание	Пример
&&	Логическое И	item.ContentTypeId == Guid("99c92e22-4e60-48c0-ab70-add500e71a45") && item.GetValueAsString("email") == "admin@email.test"
	Логическое ИЛИ	item.ContentTypeId == Guid("99c92e22-4e60-48c0-ab70-add500e71a45") item.ContentTypeId == Guid("733af6e2-e187-4e75-8a2b-ae4072f402e0")

Функции поиска по подстроке

Функция	Описание	Пример
Contains	Значение содержит подстроку	item.GetValueAsString("email").Contains("test")
StartsWith	Значение начинается со строки	item.GetValueAsString("email").StartsWith("test")
EndsWith	Значение заканчивается на строку	item.GetValueAsString("name").EndsWith(".pdf")

Примеры

1. Найти в подразделении пользователя с указанным именем.
URL: /api/item/GetList/c0857f66-fbfa-448f-a35d-afac00a3a9cb
Тело запроса: {"query": "item => item.ContentTypeId = Guid(\"99c92e22-4e60-48c0-ab70-add500e71a45\") && item.GetValueAsString(\"name\") == \"admin\""}
c0857f66-fbfa-448f-a35d-afac00a3a9cb - ID подразделения пользователя

Обратите внимание: GetList ищет только тех пользователей, которые находятся непосредственно в самом подразделении. Если требуется искать по всей структуре подразделения, то нужно вместо GetList использовать GetRecursive

2. Найти во всем списке пользователей пользователя с указанным Email.
URL: /api/item/GetRecursive/e3a94bde-0ca9-456f-b338-4465d40389ee
Тело запроса: {"query": "item => item.ContentTypeId = Guid(\"99c92e22-4e60-48c0-ab70-add500e71a45\") && item.GetValueAsString(\"email\") == \"admin@email.test\""}
e3a94bde-0ca9-456f-b338-4465d40389ee - ID списка "Пользователи"

99c92e22-4e60-48c0-ab70-add500e71a45 - это ID типа контента пользователей

3. В списке "Файлы" найти элемент по имени в заданной папке.
URL: /api/item/GetList/202b3ffa-48b9-4040-a7db-f8d688166f51
Тело запроса: {"query": "item => item.GetValueAsString(\"name\") == \"Test 002-0.1.pdf\""}
202b3ffa-48b9-4040-a7db-f8d688166f51 - ID папки

Обратите внимание: GetList ищет только те элементы, которые лежат непосредственно в указанной папке. Если требуется искать во вложенных папках, то нужно вместо GetList использовать GetRecursive

4. В списке "Файлы" найти элементы, у которых имя заканчивается на ".pdf".
URL: /api/item/GetList/202b3ffa-48b9-4040-a7db-f8d688166f51
{"query": "item => item.GetValueAsString(\"name\").EndsWith(\".pdf\")"}
202b3ffa-48b9-4040-a7db-f8d688166f51 - ID папки

5. В списке "Файлы" найти все элементы, у которых атрибут "Статус документа" = "Размещено".
 URL: /api/item/GetRecursive/966e62c5-a803-49a0-a1be-e680d130c481
 Тело запроса: {"query": "item => item.GetLookupId(\"document_status\") == Guid(\"b9d061b1-7ce7-4756-8230-e502cfe3d8d8\")"}

 966e62c5-a803-49a0-a1be-e680d130c481 - ID списка "Файлы"
 b9d061b1-7ce7-4756-8230-e502cfe3d8d8 - ID статуса "Размещено"

 Вместо ID списка "Файлы" можно передать ID папки (ID папки проекта, например). В этом случае поиск будет проводится внутри заданной папки
6. В списке "Файлы" найти все элементы, у которых атрибут "Статус документа" = "Размещено" или "На входном контроле".
 URL: /api/item/GetRecursive/966e62c5-a803-49a0-a1be-e680d130c481
 Тело запроса: {"query": "item => new Guid[] { Guid(\"b9d061b1-7ce7-4756-8230-e502cfe3d8d8\"), Guid(\"5cd0640c-1e57-4e09-b0ec-babfb5c72680\") }. Contains(item.GetLookupId(\"document_status\"))"}
 Contains(item.GetLookupId(\"document_status\"))
7. В списке "Файлы" найти элементы, у которых значение поля "Трудозатраты этап 2" больше 1.
 URL: /api/item/GetRecursive/966e62c5-a803-49a0-a1be-e680d130c481
 Тело запроса: {"query": "item => item.GetValueAsInt(\"duration_plan_stage_2\") > 1"}
 item.GetValueAsInt(\"duration_plan_stage_2\") > 1
8. В списке "Файлы" найти элементы по дате создания >= 01.10.2024 10:00:00.
 URL: /api/item/GetRecursive/966e62c5-a803-49a0-a1be-e680d130c481
 Тело запроса: {"query": "item => item.InsertDate >= DateTime(2024, 10, 1, 10, 0, 0)"}
 item.InsertDate >= DateTime(2024, 10, 1, 10, 0, 0)
 Обратите внимание: время передается в UTC
9. Нужно выводить только пользователей (не подразделения и не группы):
 item => item.ContentTypeId = Guid("99c92e22-4e60-48c0-ab70-add500e71a45")
 Здесь 99c92e22-4e60-48c0-ab70-add500e71a45 - это ID типа элемента "Пользователь"

Резервное копирование и восстановление данных

Резервное копирование — это процедура, сохраняющая данные в виде, допускающем их последующее восстановление.

Процедуры резервного копирования и последующего восстановления могут выполняться для следующих объектов системы:

- базы данных контента (БД);
- файлы

Только спланированная стратегия выполнения резервного копирования обеспечит корректное восстановление данных.

Основные рекомендации по резервному копированию данных:

1. План резервирования должен зависеть от режима использования системы и того, какие задачи ставятся при восстановлении.
2. Рекомендуется применять раздельное копирование базы данных и файлов хранилища.
3. Стандартный режим резервирования: ежедневный полный бэкап БД (для БД небольшого и среднего размера) и инкрементный бэкап хранилища файлов и БД большого размера.
4. Для надежного хранения данных следует хранить не менее 2-х полных резервных копий БД. Хотя бы одна из резервных копий должна находиться на удалении, исключая любую вероятность одновременного повреждения в случае непредвиденных обстоятельств.
5. Резервирование БД и файлов хранилища файлов осуществляется специализированными инструментами, например `pg_dump` (для БД) и `rsync` (для хранилища файлов).
6. При использовании средств виртуализации серверов допускается проводить резервирования средствами виртуализации.

[Резервное копирование данных](#)

[Восстановление баз данных](#)

Резервное копирование данных

Резервное копирование базы данных с помощью встроенных средств резервирования

В данном разделе показано как сделать дамп базы данных PostgreSQL в консоли при подключении по SSH с помощью утилит `pg_dump` и `pg_dumpall`.

Утилита `pg_dump`

В PostgreSQL есть встроенный инструмент для создания резервных копий — утилита `pg_dump`. Утилита имеет простой синтаксис:

```
pg_dump -options < > > < >
```

Где:

`-options` - параметры, с которыми должна выполняться утилита.

`<имя базы>` - название базы, которую необходимо сохранить.

`<файл для сохранения копии>` - название файла копии.

В простейшем случае достаточно указать имя базы данных, которую в дальнейшем нужно будет восстановить. Резервная копия создается следующей командой:

```
pg_dump -U vitrodbuser -W vitrodb > /tmp/vitrodb.dump # pg_dump u postgres
```

Ключ `-U` определяет пользователя, а `-W` требует ввести пароль.

Чтобы сэкономить место на диске, можно сразу же сжимать дампы:

```
pg_dump -U vitrodbuser -W vitrodb | gzip > /tmp/vitrodb.gz
```

Для выполнения резервного копирования по расписанию, например, можно использовать скрипт. Нижеприведённый пример скрипта выполняет бэкап и удаляет все файлы старше 31 дня:

```
#!/bin/sh

PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin # PATH

PGPASSWORD=vitrodbuser_password # PGPASSWORD PostgreSQL
export PGPASSWORD # PGPASSWORD,
pathB=/mnt/backup # pathB
dbUser=vitrodbuser # dbUser
database=vitrodb # database

find $pathB \( -name "*-[0123]?.*" \) -ctime +31 -delete # , 31 ,
pg_dump -U $dbUser $database | gzip > $pathB/pgsql_$(date "+%Y-%m-%d").sql.gz # , gzip ,

unset PGPASSWORD # PGPASSWORD
```

Для настройки работы данного скрипта по расписанию нужно настроить планировщик `crontab`:

```
crontab -e 3 0 * * * /etc/scripts/pgsql_dump.sh # postgres pg_dump
```

Данная команда добавляет задание в cron, которое запускает скрипт `/etc/scripts/pgsql_dump.sh` и будет выполняться каждый день в 3:00.

Более подробное описание работы и параметров утилиты `pg_dump` см. по [ссылке](#).

Утилита `pg_dumpall`

Утилита `pg_dumpall` реализует резервное копирование всего экземпляра (кластера или инстанса) базы данных без указания конкретной базы данных на инстансе:

```
pg_dumpall > /tmp/instance.bak
```

Чтобы сразу скачать резервную копию экземпляра базы данных, нужно передать вывод на архиватор `gzip`:

```
pg_dumpall | gzip > /tmp/instance.tar.gz
```

Более подробное описание работы и параметров утилиты `pg_dumpall` см. по [ссылке](#).

Резервное копирование файлов

Утилита `rsync`

`Rsync` — утилита для удаленной синхронизации и копирования файлов. Одним из её преимуществ является то, что она позволяет не передавать файлы полностью, а только изменения в них, синхронизирует директории и файлы, имеет встроенный механизм сжатия.

Синтаксис утилиты в общем виде выглядит так:

```
rsync -options <source> <destination>
```

Где:

`-options` — параметры, с которыми должна выполняться утилита.

`<source>` — каталог или файл, который является источником.

`<destination>` — каталог или файл, который является приемником.

При локальном режиме работы достаточно задать каталог-источник и каталог-приемник:

```
rsync -avzh1 /path/of/source/folder /path/to/destination/folder
```

Заданные параметры:

- `-a` - режим архивирования, который включает рекурсивное копирование, сохранение прав доступа, временных меток и т.д.
- `-v` - уровень подробности вывода (verbose), позволяет видеть подробную информацию о процессе копирования.
- `-z` - сжатие данных во время передачи.
- `-h` - вывод размеров файлов в удобном для чтения формате (human-readable).
- `-H` - сохранение жестких ссылок.
- `-l` - сохранение символических ссылок.

Если копирование нужно выполнить с удаленным узлом, добавляется имя пользователя и IP-адрес или имя узла:

```
rsync -avzh1 /path/of/source/folder root@192.168.52.1:/path/to/destination/folder
```

Автоматическая синхронизация папок выполняется штатным способом — планировщиком заданий (cron). Создадим скрипт, который будет копировать файлы на удаленный узел:

```
#!/bin/sh

RSYNC=/usr/bin/rsync
SSH=/usr/bin/ssh
KEY=/root/.ssh/id_rsa # ssh
RUSER=root #
RHOST=192.168.56.1 #
RPATH=/remote/dir #,
LPATH=/local/dir #,

$RSYNC -az -e "$SSH -i $KEY" $LPATH $RUSER@$RHOST:$RPATH
```

Настройка планировщика crontab:

```
crontab -e 0 22 * * * /root/scripts/rsync_to_cron.sh
```

Данная команда добавляет задание в cron, которое запускает скрипт /root/scripts/rsync_to_cron.sh и будет выполняться каждый день в 22:00.

Более подробное описание работы и параметров утилиты `pg_dump` см. по [ссылке](#).

Восстановление баз данных

Восстановление данных из резервной копии базы данных с помощью встроенных средств восстановления

Утилита pg_restore

Утилита позволяет восстанавливать данные из резервных копий, созданных утилитой pg_dump. Данная утилита не подходит для восстановления дампов созданных в формате sql. В этом случае следует воспользоваться утилитой psql.

Команда для восстановления базы данных:

```
# pg_restore -v -U vitrodbuser -h localhost -d vitrodb /tmp/vitrodb.bak
```

ключ -v включает режим подробных сообщений, -U дает возможность указать пользователя, от имени которого производится подключение к БД, -d позволяет восстановить в указанную БД

Более подробное описание работы и параметров утилиты pg_restore см по [ссылке](#).

Утилита psql

Утилита psql — это терминальный клиент для работы с PostgreSQL. Она позволяет интерактивно вводить запросы, передавать их в PostgreSQL и видеть результаты. Также запросы могут быть получены из файла или из аргументов командной строки. Кроме того, psql предоставляет ряд метакоманд и различные возможности, в т.ч. возможность восстанавливать данные из резервной копии. Данную утилиту можно использовать только для восстановления данных из дампа в формате sql.

Команда для восстановления выглядит так:

```
# psql -U vitrodbuser -h localhost -d vitrodb < /tmp/vitrodb.bak.sql
```

Более подробное описание работы и параметров утилиты psql см по [ссылке](#).

Прочие настройки

- [Настройка расписания в ОС Linux на примере задания на запуск утилиты синхронизации ОШС с AD/LDAP](#)
- [Настройка функционала проверки QR-кодов](#)
- [Настройки безопасности учетных записей](#)
- [Перенос базы данных в другое расположение \(Linux\)](#)
- [Настройка Приложения выпуска](#)

Настройка расписания в ОС Linux на примере задания на запуск утилиты синхронизации ОШС с AD/LDAP

1. создать файл Bash-скрипта для запуска утилиты

```
sudo touch /usr/sbin/Vitro.Server.OrgStructureLdap.sh
```

2. Отредактировать файл Bash-скрипта

```
sudo nano /usr/sbin/Vitro.Server.OrgStructureLdap.sh
```

вставив следующее содержимое:

```
Vitro.Server.OrgStructureLdap.sh
```

```
#!/bin/sh  
  
#   :  
cd /usr/local/Vitro.Server.OrgStructureLdap.Linux  
  
#   :  
./Vitro.Server.OrgStructureLdap
```

3. Разрешить исполнение для файл Bash-скрипта

```
sudo chmod +x /usr/sbin/Vitro.Server.OrgStructureLdap.sh
```

4. Запустить редактор расписания.

если планируем запуск по расписанию команды/приложения/скрипта от имени суперпользователя:

```
sudo crontab -e
```

если планируем запуск по расписанию команды/приложения/скрипта от имени другого пользователя:

```
sudo crontab -u <username> -e
```

5. Структура пользовательского файла (таблицы) расписания:

```
crontab
```

```
m h dom mon dow  command  
0 2 * * * Vitro.Server.OrgStructureLdap.sh 2>&1 | /usr/bin/logger -t vitro_org_sync
```



Описание структуры записи в таблице расписания

Каждая запись в *crontab* состоит из шести полей, указываемых в следующем порядке:

m - минуты
h - часы
dom - день месяца (число по календарю)
mon - месяц (порядковый номер месяца в году)
dow - день недели (порядковый номер дня недели)

Символы для настройки шаблонов значений в календарных полях:

"*" - любое числовое значение валидное для данного типа поля

"," - разделитель в списке значений поля (0 2 * * 1,3,5 запись означает запуск задания в 2:00 по Пн,Ср,Пт еженедельно)

"-" - диапазон значений (0 2 * * 1-5 запись означает запуск задания в 2:00 по будням еженедельно)

"/" - шаг выполнения повторов (0 0-13/2 * * * запись означает запуск задания в 00 минут каждого 2-го часа в промежутке суток с 0:00 по 13:00 ежедневно в течение всего года)

поле "comand" - команда/приложение/скрипт выполняемый в задании расписания

в примере команда исполнения Bash-скрипта запуска утилиты синхронизации ОШС с AD/LDAP с перенаправлением стандартного вывода в системный лог (/var/log/syslog).

Структура команды в примере:

Vitro.Server.OrgStructureLdap.sh - вызов на выполнение Bash-скрипта, размещенного в /usr/sbin

2>&1 - перенаправление стандартного потока ошибок "2" (stderr) в поток стандартного вывода "1" (stdout)

| - конвейер (перенаправление стандартного вывода на вход следующей команды)

/usr/bin/logger - вызов системной утилиты Linux для записи в лог

-t vitro_org_sync - аргумент для команды logger, добавляющий маркировку строк тегом "vitro_org_sync", записываемых в системный лог.

6. Проверить настроенное расписание

если настраивали для суперпользователя:

```
sudo crontab -l
```

если настраивали для другого пользователя:

```
sudo crontab -u <username> -l
```

7. Проверка сообщений, фиксируемых при выполнении задания в системном логге:

```
grep vitro_org_sync /var/log/syslog
```

Настройка функционала проверки QR-кодов

Страница проверки QR кода разворачивается вместе с основным веб приложением.

Настройка QR кода производится в списке Конфигурация /list/f5610aa8-24b8-4243-b418-a74b139029bd

URL проверки

P > QR > URL (Workflow > QRCode > PageUrl)

Значение:

{URL }/qrcode/check/

Пример значения:

http://79.143.74.27/qrcode/check/

Пример сформированного URL проверки QR кода на элементе:

http://79.143.74.27/qrcode/check?Rev=1&Id=b020dc08-1250-c5de-86b0-b381acef492f

Пример настроек:



Размер

P > QR > (Workflow > QRCode > Size)

Размер QR кода на PDF документе (в пунктах)

Значение:

120;120

Расположение

P > QR > (Workflow > QRCode > Position)

Расположение QR кода на PDF документе (в пунктах, от правого нижнего угла)

Значение:

158;207

Актуальные статусы

P > QR > (Workflow > QRCode > StatusListValid)

Список ID статусов документа, при которых документ будет считаться актуальным. Статусы: `Согласовано`, `Согласовано с замечаниями`

Значение:

`84124eed-7049-46a8-bac9-a179ff00efe3,132fd34b-b757-459e-8191-4a264c91a5b3`

Статусы корректировки

P > QR > (Workflow > QRCode > StatusListWarning)

Список ID статусов документа, при которых документ находится на корректировке. Статусы: `На корректировке`

Значение:

`22a8669a-9674-4aac-950b-3379df1059bd`

Ревизия

P > > > (Workflow > Item > Field > Revision)

Системное имя атрибута документа, в котором хранится номер ревизии документа. Атрибут: `Цикл согласования (approval_count)`

Значение:

`approval_count`

Статус документа

P > > > (Workflow > Item > Field > Status)

Системное имя атрибута документа, в котором хранится статус документа. Атрибут: `Статус документа (document_status)`

Значение:

`document_status`

Настройки безопасности учетных записей

Для детальной настройки параметров безопасности учетных записей необходимо добавить ключи в реестр конфигурации.

В списке Конфигурация создается следующая структура:

Список Конфигурация
<pre>- (- , - Security) -- (- , - Authentication) --- (- , - Password)</pre>

После создания структуры в папку конфигурации "По паролю" добавляются ключи с указанием значения (см. таблица ключей безопасности учетных записей).

Если ключ отсутствует, то автоматически срабатывает значение по умолчанию.

Таблица ключей безопасности учетных записей

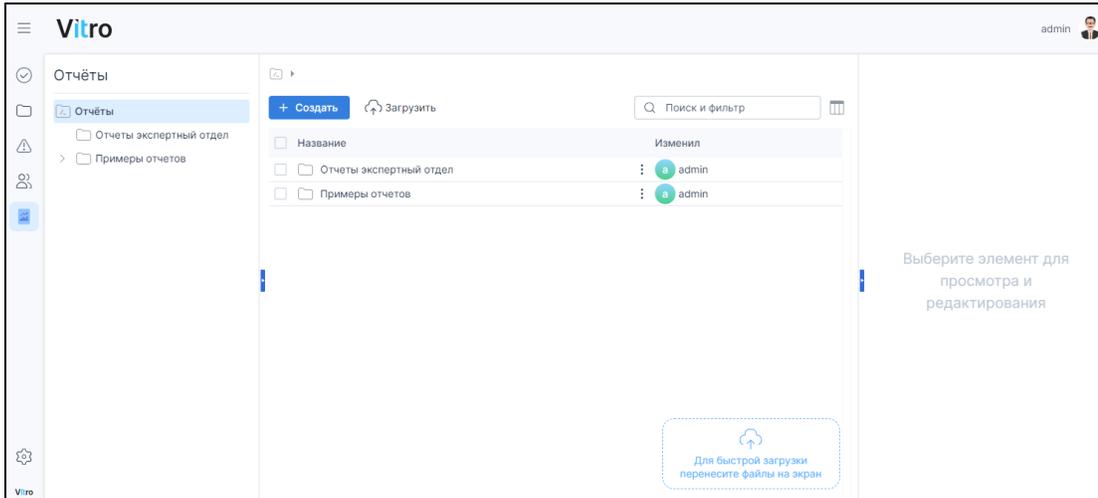
Название	Тип элемента	Код	Значение по умолчанию	Ед. изм.	Примечание
Минимальная длина пароля	Значение конфигурации Число	MinLength	5	символ	
Максимальная длина пароля	Значение конфигурации Число	MaxLength	50	символ	
Проверка совпадений по истории паролей	Значение конфигурации Число	HistoryCheckCount	1	запись	новый пароль не должен совпадать только с последним
Минимальное количество символов в нижнем регистре в пароле	Значение конфигурации Число	MinLowerCaseCharCount	0	символ	
Минимальное количество символов в верхнем регистре в пароле	Значение конфигурации Число	MinUpperCaseCharCount	0	символ	
Минимальное количество цифр в пароле	Значение конфигурации Число	MinDigitCharCount	0	символ	
Минимальное количество спец. символов в пароле	Значение конфигурации Число	MinSpecialCharCount	0	символ	
Срок действия паролей	Значение конфигурации Число	ExpireDayCount	30	день	
Время жизни сессии, если не активна	Значение конфигурации Число	SessionInactivityTimeoutMinuteCount	4320	минута	3 дня
Блокировка страницы браузера при бездействии пользователя	Значение конфигурации Число	RequireLoginInactivityTimeoutMinuteCount	0	минута	не блокируется
Включена двухфакторная аутентификация	Значение конфигурации Флаг	TwoFactorAuthenticationEnabled	true		

Отчеты

1. В систему интегрирован Telerik Report Server в качестве решения для управления отчетами
2. Сервисы Telerik вшиты в дистрибутив системы, шаблоны отчетов сохраняются в системе в виде файлов формата trdp

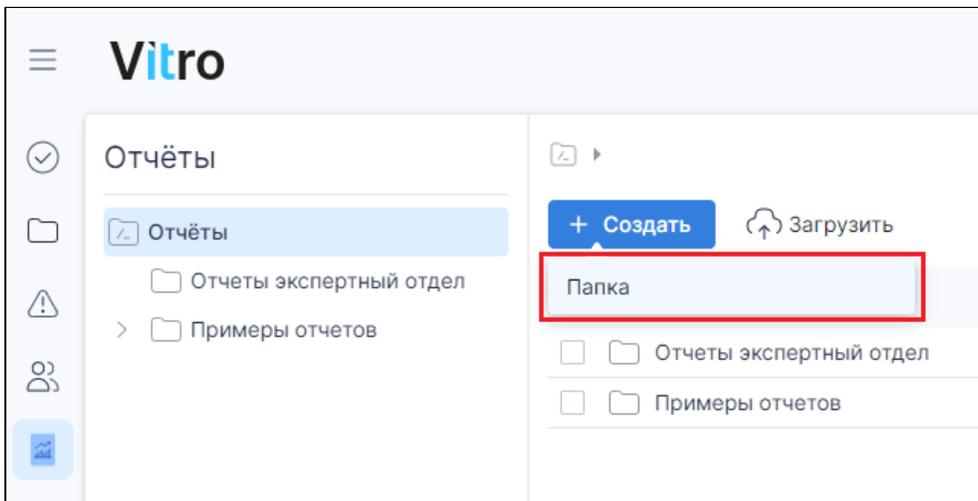
Размещение шаблонов отчетов в систему

В данном Отчеты, в структурированном виде расположены все предварительно настроенные и добавленные в систему отчеты.

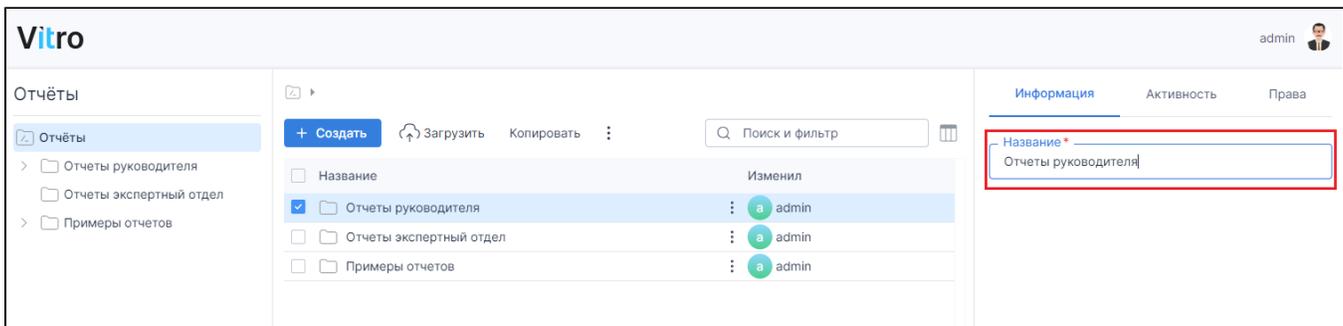


Создание структуры отчетов

Для создания папки необходимо в дереве каталогов выбрать корневую папку, нажать кнопку "Создать" и из выпадающего меню выбрать "Папка".



В области редактирования заполнить необходимую информацию.



После создания папки, необходимо настроить права и выдать разрешения на действия в папке (данная возможность есть у Администратора системы). Подробнее узнать про настройку прав можно в разделе документации - [Настройка прав доступа](#)

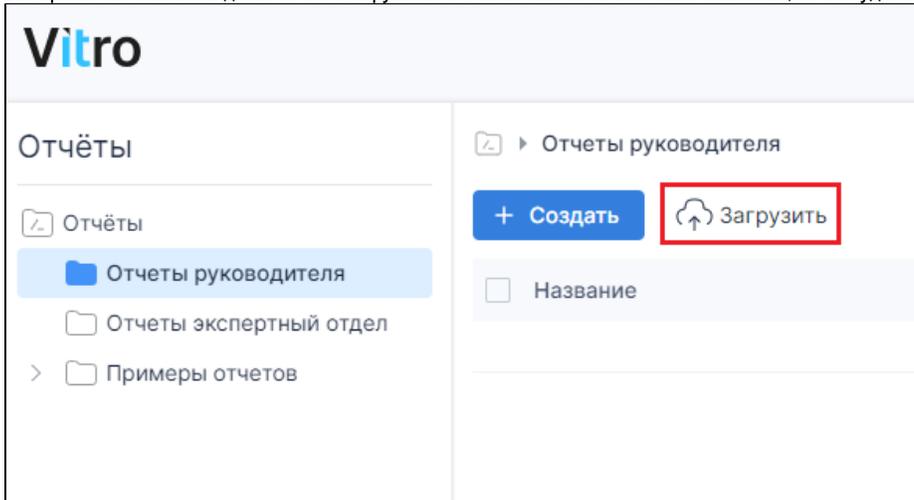
Загрузка отчета

После создания структуры, подготовленные файлы отчетов необходимо загрузить в систему. Это действие можно выполнить двумя способами:

По кнопке "Загрузить"

После перехода в нужную папку, для загрузки отчета нажимаем на кнопку "Загрузить"

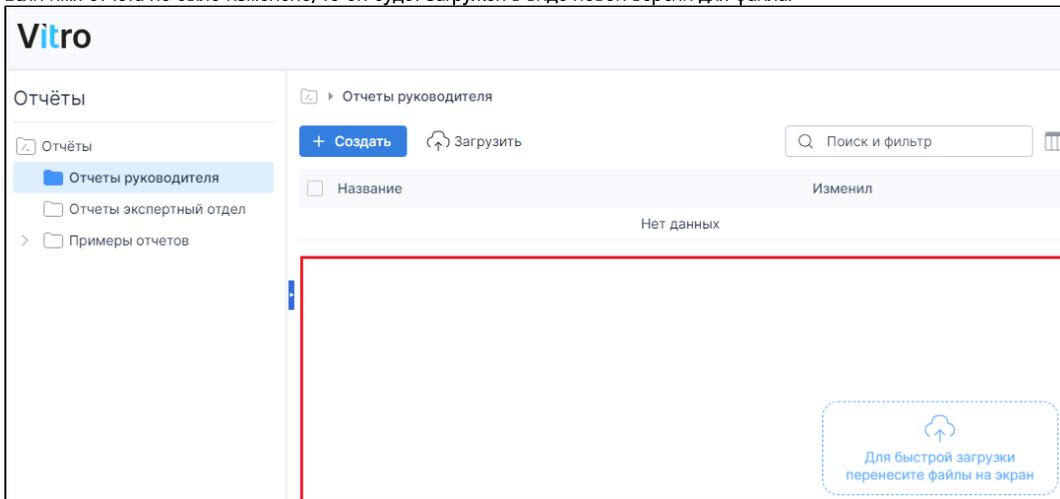
Выбираем на локальном диске отчет и загружаем его. Если имя отчета не было изменено, то он будет загружен в виде новой версии отчета.



Быстрая загрузка через функционал "drag'and'drop"

После перехода в нужную папку, для загрузки отчета перетаскиваем подготовленный файл в область для загрузки файлов.

Если имя отчета не было изменено, то он будет загружен в виде новой версии для файла.

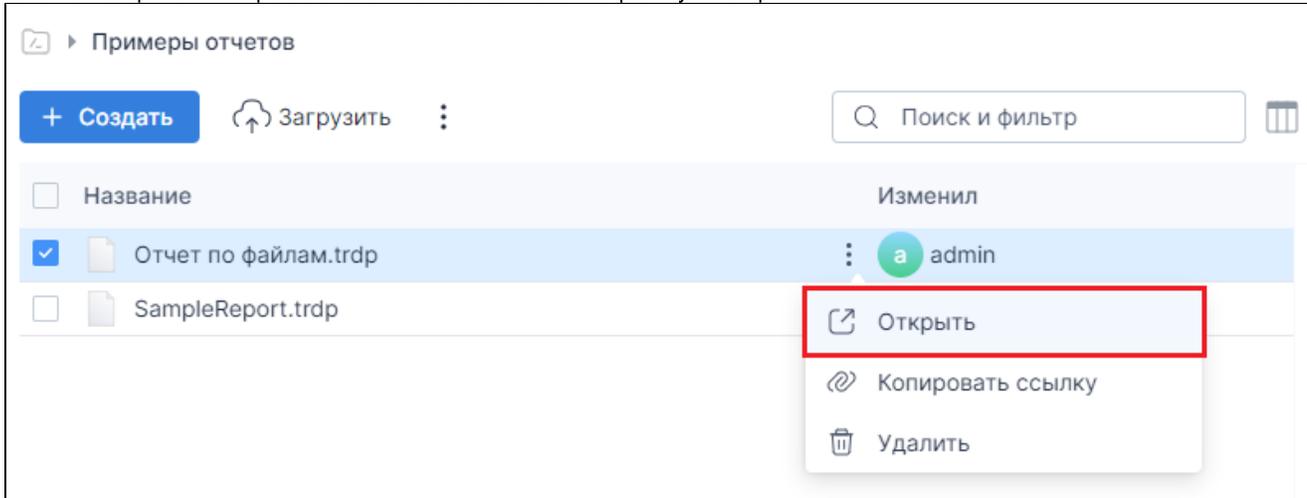


Просмотр отчета

Для того, чтобы просмотреть отчет необходимо его открыть. Это действие можно выполнить двумя способами:

Через контекстное меню

Нажимаем на "три точки" в строке с отчетом в контекстном меню выбираем пункт "Открыть"



Двойным кликом на строке с отчетом

Нажимаем двойным кликом ЛКМ по названию отчета. Отчет откроется в новом окне.

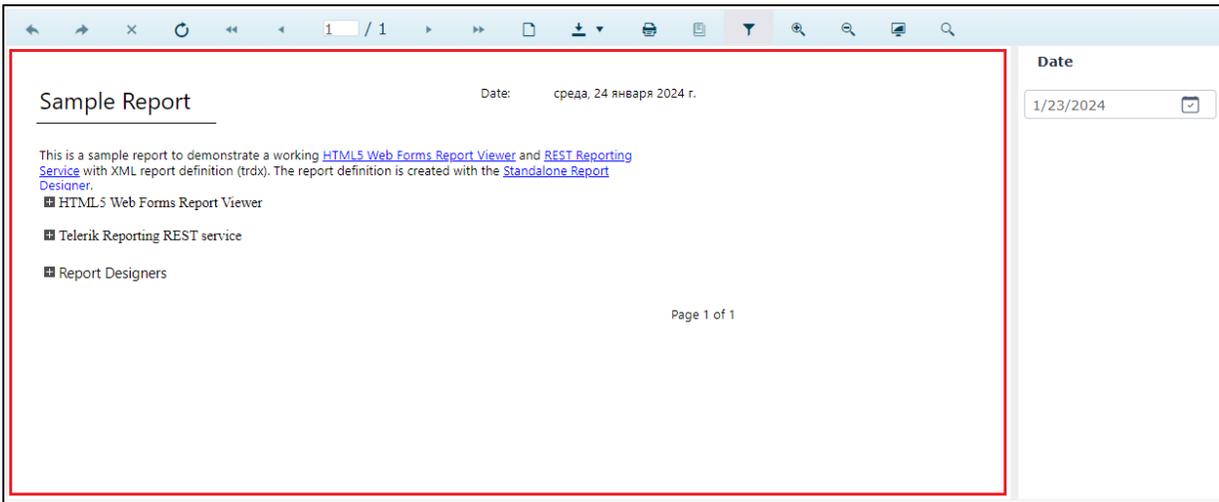


Внешний вид отчета

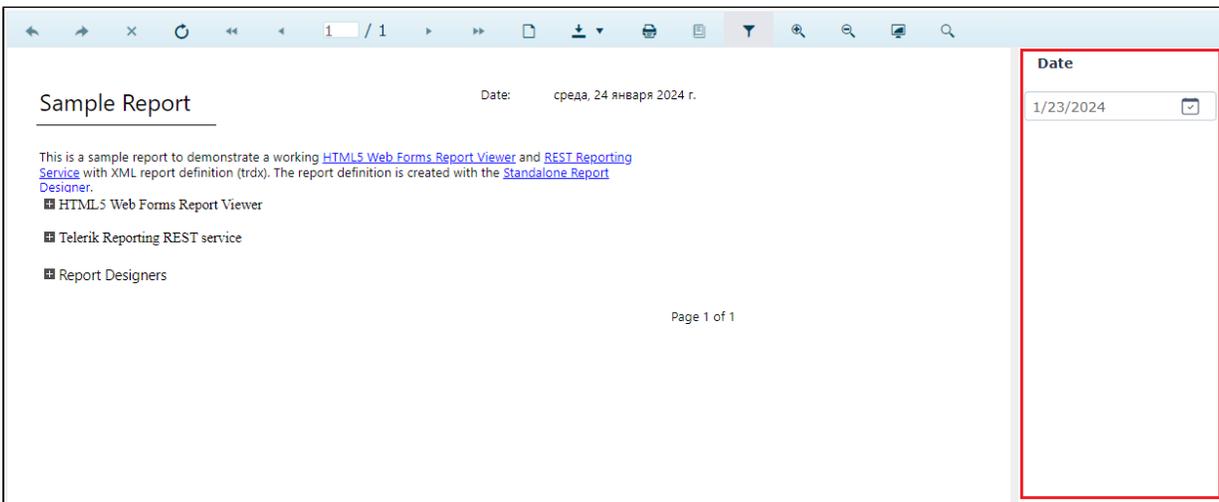
У каждого отчета присутствует стандартный набор элементов управления. К ним относятся:
Инструменты отчета - верхняя панель управления отчетом (экспорт, масштабирование,...)



Основная часть - отображение подготовленной информации

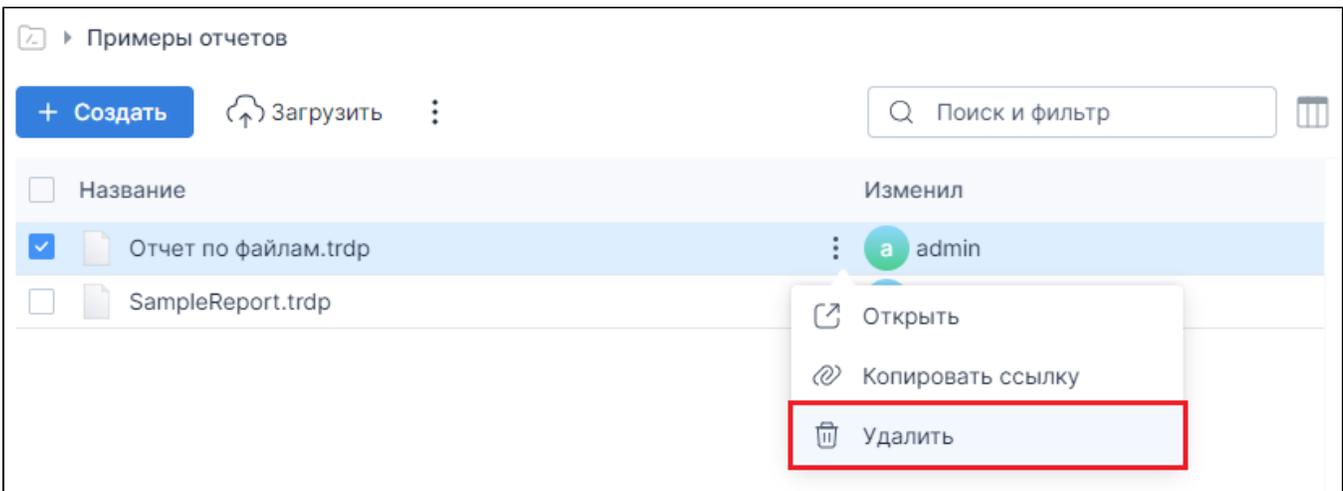


Панель фильтров - фильтрация значений осуществляется в основной части отчета по выбранным параметрам в полях панели фильтров (параметр настраивается опционально при подготовке файла отчета)

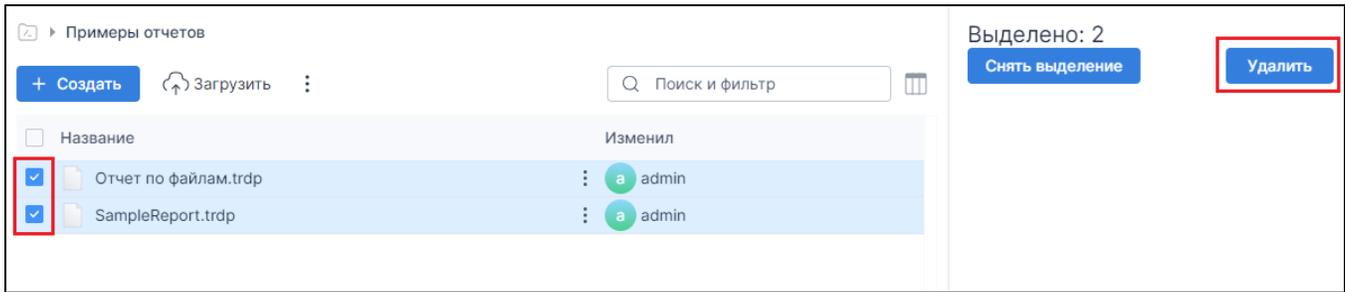


Удаление отчета

Для того, чтобы удалить отчет, необходимо его выбрать в списке отчетов. Нажимаем на "три точки" в контекстном меню выбрать пункт "Удалить".



Если выбрать несколько отчетов, то в области редактирования появится возможность группового удаления отчетов.

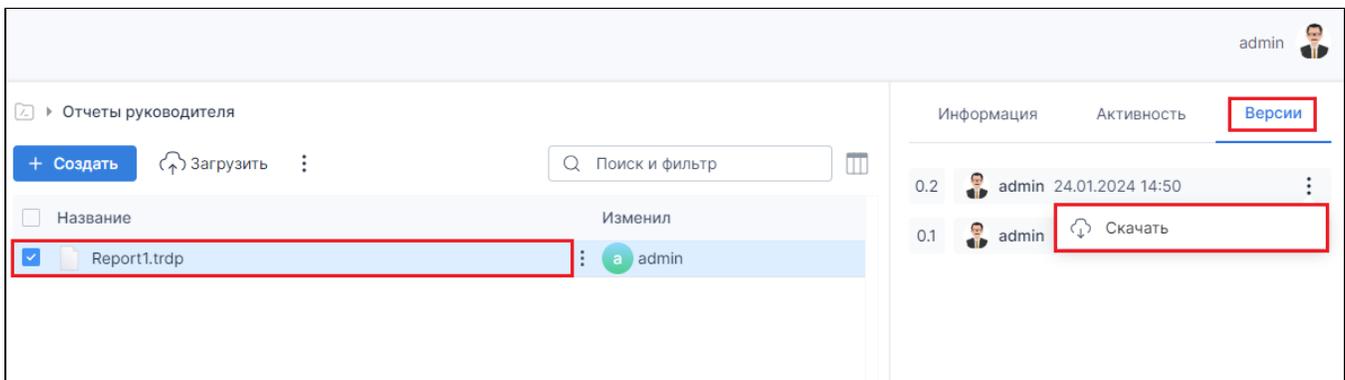


Версионность отчетов

Каждый отчет позволяет хранить в себе различные версии, которые были загружены ранее. Данный функционал будет необходим прежде всего при настройке и внесении изменений в отчет. Загруженный ранее отчет можно "Скачать" или "Восстановить"

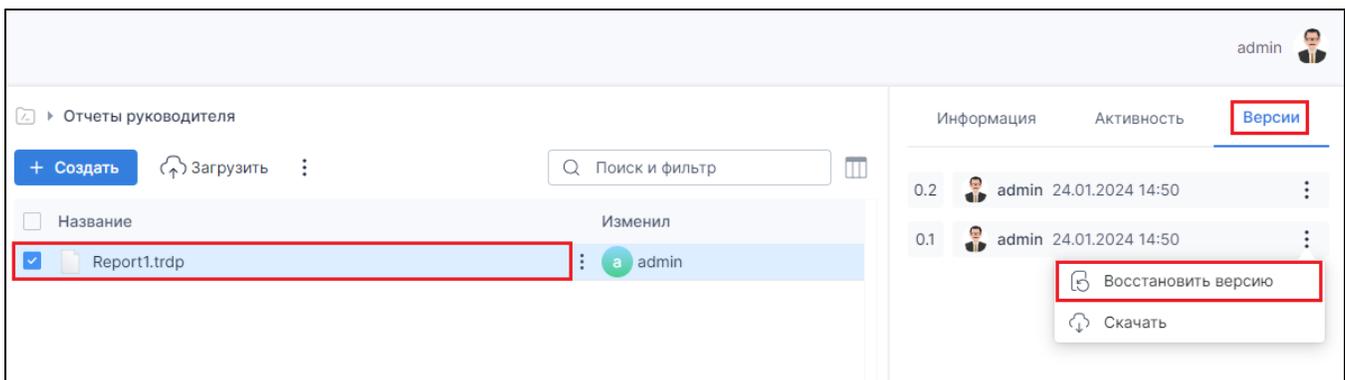
Скачивание

Для внесения изменений в отчет необходимо его скачать. Для этого необходимо: Выбрать нужный отчет и открыть вкладку "Версии". В области редактирования напротив необходимой версии нажать на "три точки". В контекстном меню выбрать пункт "Скачать"



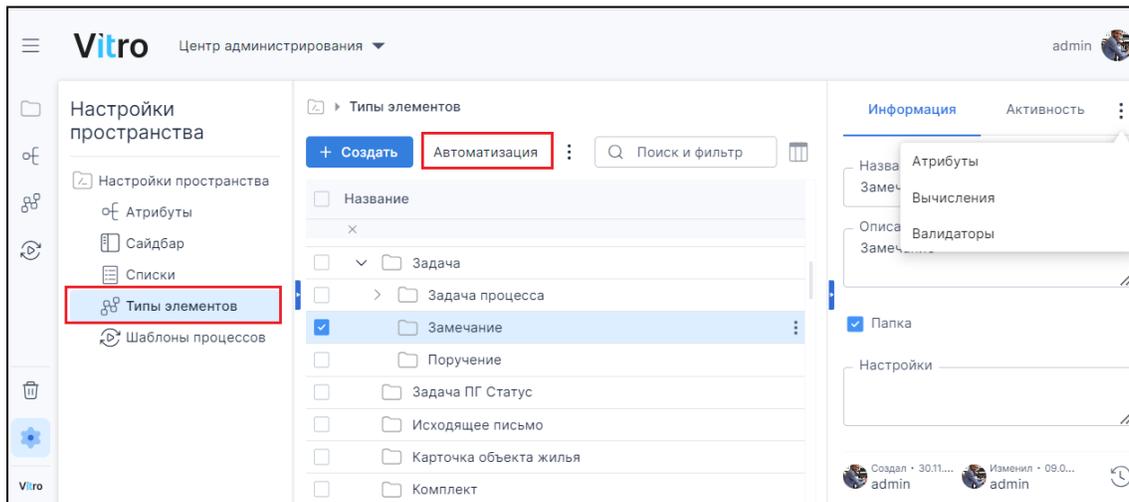
Восстановление

Если созданный отчет был загружен в качестве новой версии, но в нем была обнаружена ошибка, то можно восстановить корректно работающую версию отчета. Для этого необходимо: Выбрать нужный отчет и открыть вкладку "Версии". В области редактирования напротив необходимой версии нажать на "три точки". В контекстном меню выбрать пункт "Восстановить версию"



Автоматизация

1. Автоматизации - инструмент гибкой настройки автоматизированных действий системы, который позволяет гибко настраивать события в системе
2. Автоматизация состоит из трех основных компонентов: Триггеров (или событий), Условий и Действий
3. При настройке одной записи (одного правила) Автоматизации прописывается при каком событии и каких условиях какие действия одного типа выполнит система
4. Количество записей (правил автоматизации) в списке Автоматизаций не ограничено. Правила автоматизации привязаны к Типу элемента



5. Компоненты Автоматизации в сочетании с [Валидаторами](#) позволяют автоматизировать:

- жизненный цикл процесса
- жизненный цикл задачи
- жизненный цикл документации
- жизненный цикл замечания
- рассылку оповещений по событиям
- получение/передача команд по REST API
- прочие задачи

Триггеры

1. Триггер (Событие) – в широком понимании, это событие в системе, которое влечет за собой определенное действие системы. В контексте данной статьи триггером является смена значения выбранного Атрибута для выбранного Типа элемента. Например, смена значение статуса для замечания (шаг жизненного цикла замечания).
2. При наступлении данного события в системе мы ожидаем, что автоматически должно быть выполнено действие определенного типа

Действия

1. Автоматические действия в системе имеют несколько типов:
 - "Установить значения атрибута" (Установить значение поля) - присвоение значения указанному атрибуту
 - "Установить права доступа" (Установить права на элемент) - установка прав на указанный элемент
 - "Привязать файлы" (Установить вложения) - добавление вложения к указанному элементу
 - "Установить значения атрибута связанного элемента" (Установить значение поля связанного элемента) - присвоение значения указанному атрибуту у связанного элемента
 - "Настроить этап процесса" (Этап процесса) - привязка события к этапу процесса
 - "Вызов внешнего сервиса" - вызов внешнего сервиса по REST API
2. При действиях присвоения значения атрибутов можно использовать вычисления по формуле

Условия

1. Для автоматических действий в системе можно настроить условия выполнения действия

Пример условия: "выполнить действие только если статус процесса равен значению *Завершен*"

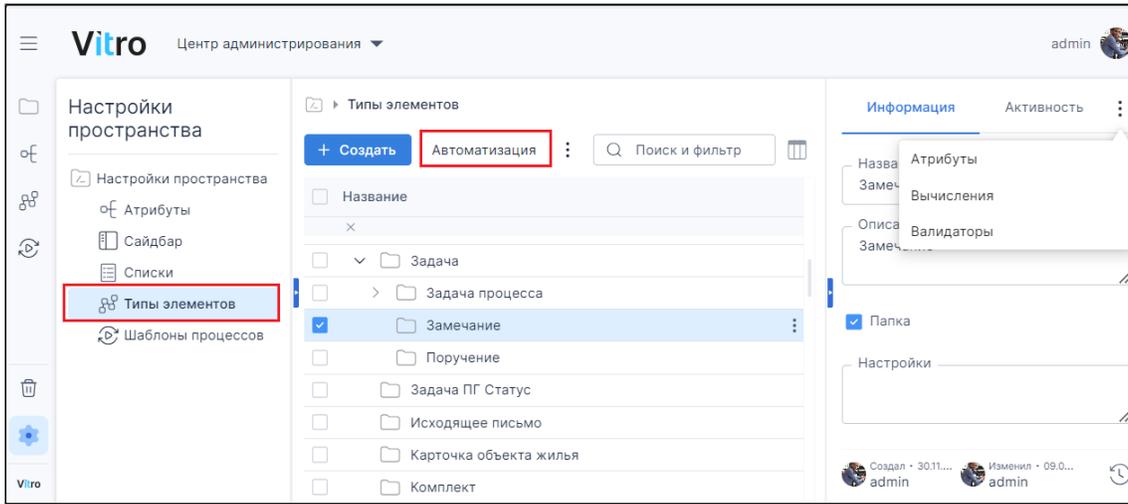
2. При настройке условия используется формула с использованием операторов сервиса [Вычисления](#)

Статьи по теме:

- [Вид Канбан для настройки правил автоматизации](#)
- [Настройка процесса согласования с использованием сервиса Автоматизации](#)

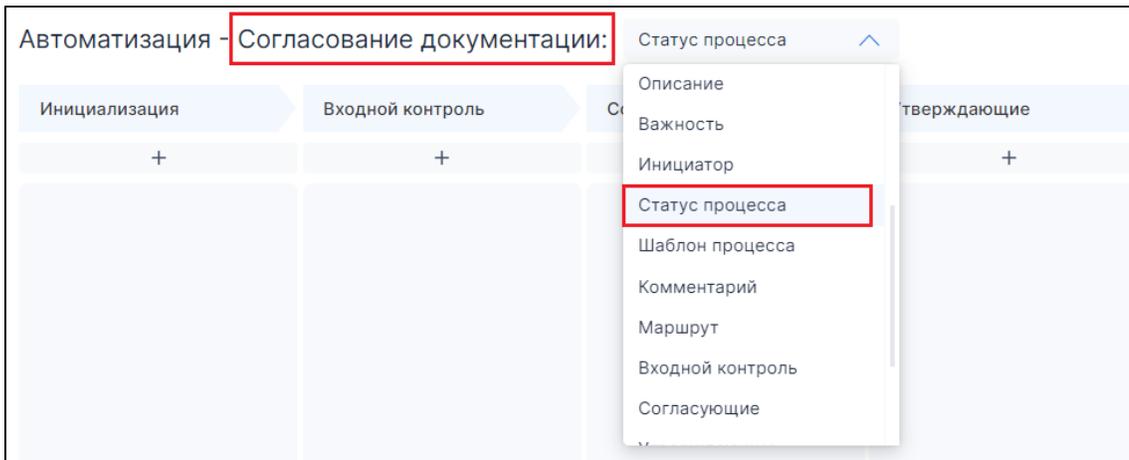
Вид Канбан для настройки правил автоматизации

1. Правила автоматизации привязаны к Типу элемента и настраиваются в отдельном окне в виде доски Канбан. Для перехода к правилам автоматизации на выбранном Типе элемента нажмите кнопку "Автоматизация" в верхнем меню команд



2. В отдельном окне настройки правил Автоматизации для выбранного Типа элемента следует выбрать тот атрибут, изменения которого станут триггером для правил автоматизации.

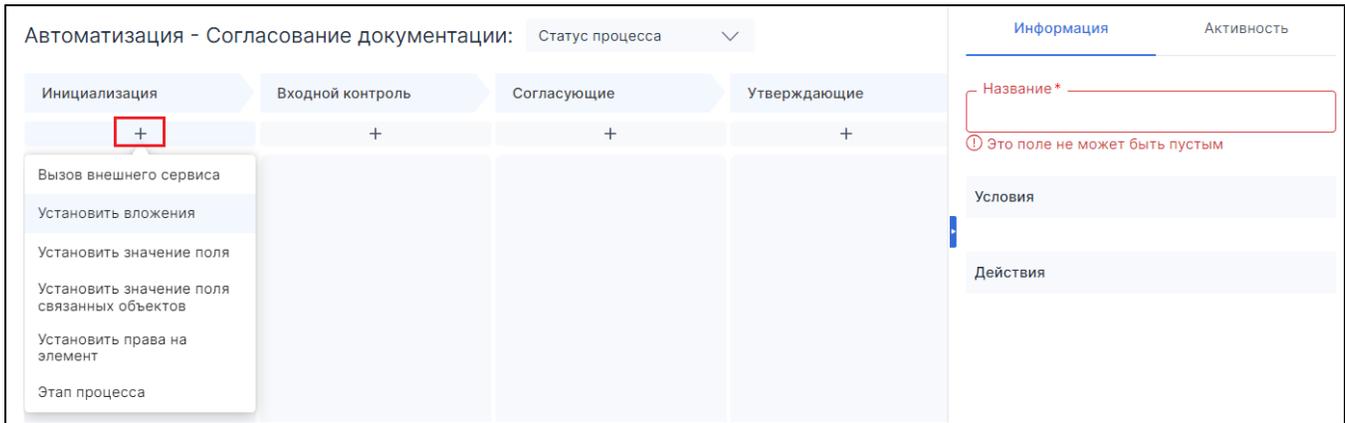
Например, на скрине ниже выбран Тип элемента процесса "Согласование документации", для его автоматизации выбран атрибут "Статус процесса" - это означает, что дальше будут настроены правила автоматизации, привязанные к смене значения статуса данного типа процесса.



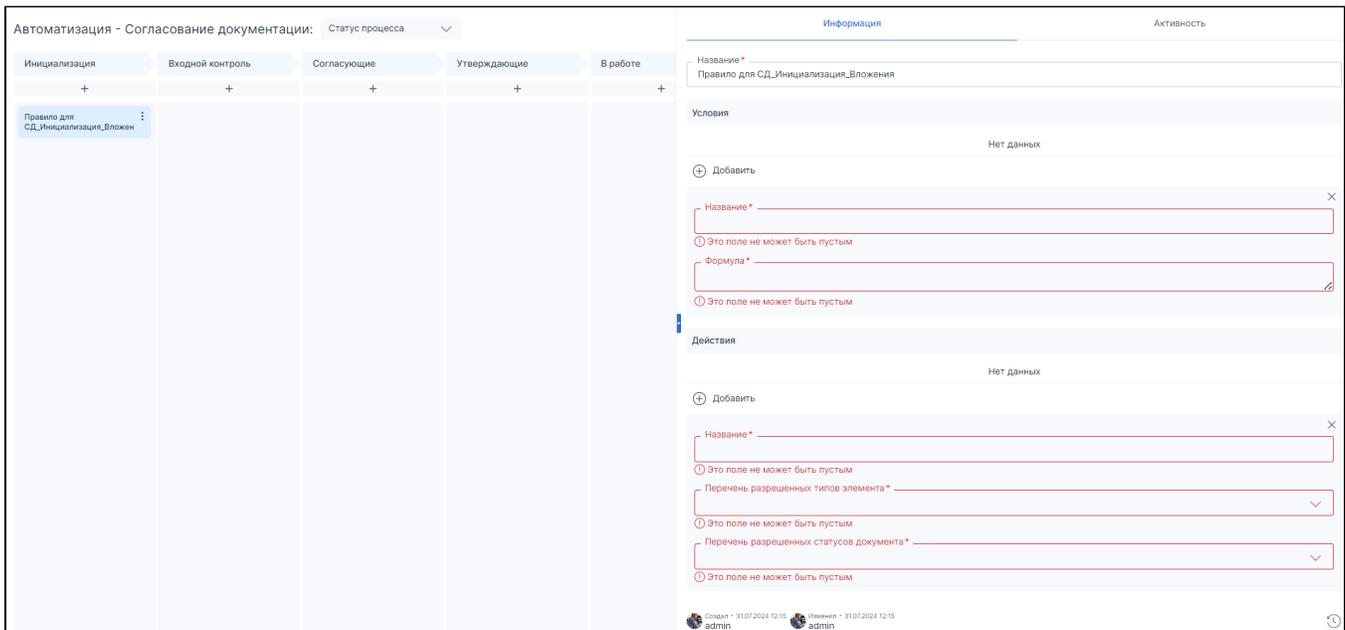
3. Значения выбранного атрибута автоматически разворачиваются в колонки доски вида Канбан. Например, на скрине выше колонками становятся значения списка статусов, из которого заполняется атрибут "Статус процесса"

4. Для создания правила автоматизации в колонке со значением атрибута, который будет триггером для выполнения автоматического действия, необходимо

- нажать плюсик и выбрать тип правила (равно типу автоматического действия) из предложенного выпадающего списка



- дать в открывшейся справа зоне свойств название правилу автоматизации (рекомендация - подробно)
- добавить по плюсику условие выполнения действия и заполнить его название и формулу проверки условия (если плюсики не появились - обновить страницу и нажать на плитку только что созданного правила автоматизации). Формула - выражение, вычисление которого должно вернуть true или false. Если условие выполняется, то действия автоматизации будут выполнены, если нет, то не будут выполняться. Условий для выполнения правила может быть несколько
- добавить по плюсику само автоматическое действие и заполнить его атрибуты (набор атрибутов зависит от типа выбранного правила). Описание атрибутов по типам правил и действий дано в статье [Типы автоматических действий](#). В одном правиле автоматизации может быть несколько действий одного типа



Типы автоматических действий

1. Автоматические действия в системе имеют несколько типов:

- "Установить значения атрибута" (Установить значение поля) - присвоение значения указанному атрибуту
- "Установить права доступа" (Установить права на элемент) - установка прав на указанный элемент
- "Привязать файлы" (Установить вложения) - добавление вложения к указанному элементу
- "Установить значения атрибута связанного элемента" (Установить значение поля связанного элемента) - присвоение значения указанному атрибуту у связанного элемента
- "Настроить этап процесса" (Этап процесса) - привязка события к этапу процесса
- "Вызов внешнего сервиса" - вызов внешнего сервиса по REST API

2. Описание типов действий - ниже в таблице

№	Название типа действия	Атрибут для типа действия	Описание атрибута
1	Вызов внешнего сервиса	Название	Текст. Название действия
		Сервис обработки	Адрес веб-сервиса, который должен быть запрошен в случае срабатывания действия На указанный адрес будет отправлен REST-запрос тело которого содержит список id элементов, для которых вызвано действие
2	Привязать файлы (Установить вложения)	Название	Текст. Название действия
		Перечень разрешенных типов документа	Выбор из списка типов элементов. Допустимо несколько значений
		Перечень разрешенных статусов документа	Выбор из списка статусов документов. Допустимо несколько значений
3	Установить значения атрибута (Установить значение поля)	Название	Текст. Название действия
		Атрибут	Выбор из списка атрибутов текущего типа элемента. Допустимо одно значение
		Формула	Значение атрибута или формула для его расчета (зависит от типа атрибута)
4	Установить значения атрибута связанного элемента (Установить значение поля связанного элемента)	Название	Текст. Название действия
		Атрибут	Выбор из списка атрибутов связанного типа элемента. Допустимо одно значение
		Формула	Значение атрибута или формула для его расчета (зависит от типа атрибута)
5	Установить права доступа (Установить права на элемент)	Название	Текст. Название действия
		Уровень доступа	Выбор из списка "Уровни доступа". Допустимо одно значение
		Формула	Формула расчета значения элемента/элементов списка Пользователи, для которого будет присвоен новый уровень доступа на элемент текущего типа элементов. Формула должна вернуть объект списка
6	Настроить этап процесса (Этап процесса)	Название	Текст. Название действия
		Исполнители	Формула расчета значения элемента/элементов списка Пользователи, для которого будет присвоен новый уровень доступа на элемент текущего типа элементов. Формула должна вернуть объект списка
		Шаблон процесса	Выбор из списка "Шаблоны процесса". Допустимо одно значение
		Плановая продолжительность этапа	Значение нормативной продолжительности этапа в днях или формула для его расчета

Настройка процесса согласования с использованием сервиса Автоматизации

Пошаговый план настройки процесса согласования

1. Определить требования к процессу:
 - a. Количество шагов и их названия (статусы процесса).
 - b. Название участников каждого шага.
 - c. Типы элементов, которые попадут во вложения процесса.
 - d. Типы элементов, на которых возможен вызов запуска процесса.
 - e. Значения статусов элементов, которые попадут во вложения процесса.
 - f. Дополнительные требования к логике процесса, такие как валидаторы, вычисления, оповещения и другие.
2. Создать тип элемента для процесса (внутри группы "Рабочий процесс").
 - a. Настроить представления (виды) для детального отображения в правой области и для шага создания. Если в системе отсутствуют подходящие виды, то создать.
 - b. Настроить атрибуты типа элемента в соответствии с требованиями к процессу. Если в системе отсутствуют подходящие атрибуты, то создать.
 - c. Настроить вычисления типа элемента в соответствии с требованиями к процессу (опционально).
 - d. Настроить валидаторы типа элемента в соответствии с требованиями к процессу (опционально).
 - e. Настроить привязку типа элемента в соответствии с требованиями к процессу. Привязка определяет на каких типах элемента будет отображаться кнопка запуска данного процесса и как она будет называться.
3. Привязать созданный тип элемента к списку Процессы (основное пространство, по умолчанию "Портал").
4. Настроить логику процесса через Автоматизацию в виде канбан доски по атрибуту "Статус процесса" в соответствии с требованиями к процессу.
5. Проверить результат настройки.
6. Подготовить документацию по процессу.

Принципиальное описание жизненного цикла процесса

1. Процесс создается со статусом процесса "Не начат". К этому статусу необходимо привязывать валидаторы (к другим статусам привязывать валидаторы в автоматических процессах нет смысла).
2. При успешном запуске статус процесса переходит в значение "Инициализация" - первому этапу Автоматизации. К этому статусу надо привязывать все подготовительные действия, такие как заполнение первичных атрибутов процессов и так далее.
3. Далее выполняется поэтапные переходы по статусам процесса в соответствии с настроенной логикой автоматизации.
4. В финале необходимо настроить завершение процесса через установку статуса процесса "Завершен".

Пример настройки

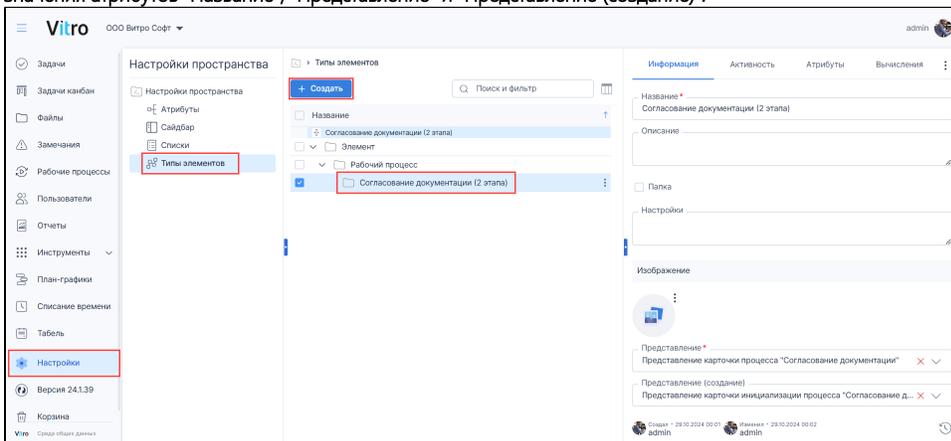
В рамках данного примера рассмотрим создание простого двухэтапного последовательно-параллельного процесса согласования.

Требования к процессу

1. Процесс "Согласование документации (2 этапа)" должен выполняться в два шага.
2. Название процесса должно вычисляться с указанием принадлежности вложенных файлов к Проекту, Стадии и Комплекту.
3. Запуск процесса должен быть доступен на файлах с типом элемента "Документ" в статусах "Размещено" и "На корректировке".
4. Запуск процесса должен быть заблокирован, если в него не попало ни одного подходящего вложения.
5. Первый шаг процесса "Согласующие":
 - a. Выполняется в статусе процесса "На согласовании".
 - b. У вложенных файлов должно устанавливаться значение "На согласовании" для атрибута "Статус документа".
 - c. Создает задачи на пользователей, указанных в атрибуте процесса "Согласующие". Допускается множественный выбор "Согласующих", в этом случае задачи должны создаваться параллельно.
 - d. Срок выполнения задач вычисляется из количества дней, указанных в атрибуте "Трудозатраты этап 1".
 - e. Завершает процесс с отклонением файлов, если хотя бы в одной из задач этапа отклонены все файлы.
6. Второй шаг процесса "Утверждающие":
 - a. Выполняется в статусе процесса "На утверждении".
 - b. Создает задачи на пользователя, указанного в атрибуте процесса "Утверждающие", допускается только одиночный выбор.
 - c. Срок выполнения задачи этапа вычисляется из количества дней, указанных в атрибуте "Трудозатраты этап 2".
 - d. При завершении этапа файлы должны получать "Статус документа" в соответствии с принятыми решениями.

Пошаговая настройка типа элемента процесса

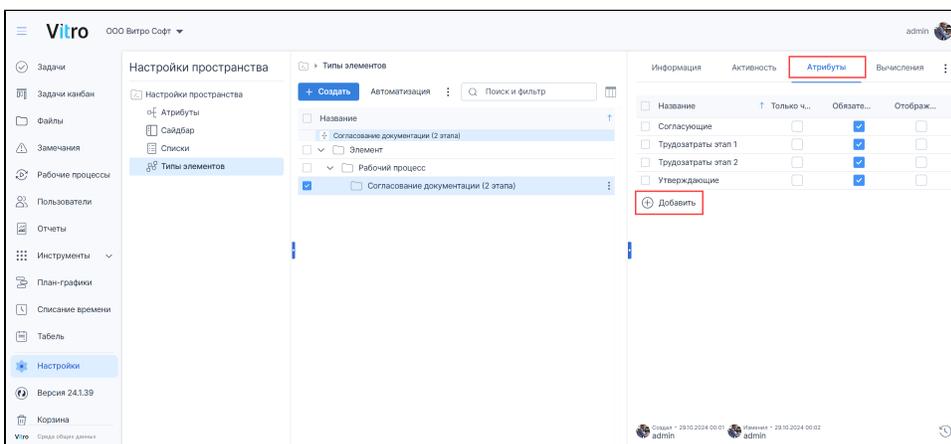
1. Создать тип элемента "Согласование документации (2 этапа)" **внутри типа элемента "Рабочий процесс"**.
Перейти в Настройки → Типы элементов, выделить тип элемента "Рабочий процесс" и выполнить действие Создать → Тип элемента. Указать значения атрибутов "Название", "Представление" и "Представление (создание)".



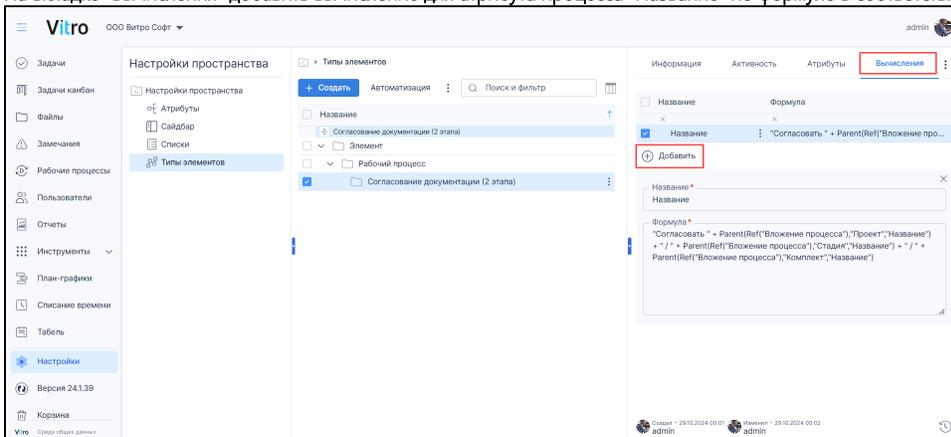
2. На вкладке "Атрибуты" добавить необходимые атрибуты в соответствии с требованиями к процессу.

Примечание!

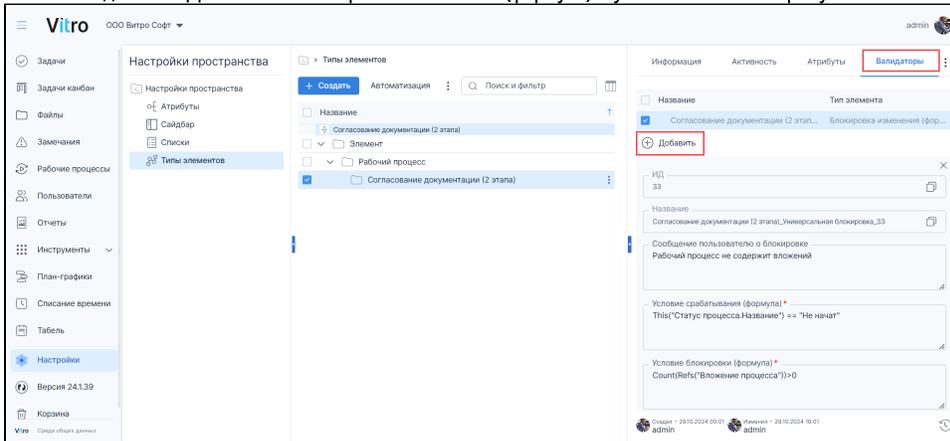
Базовые атрибуты, необходимые для работы процессов, будут унаследованы от родительских типов элементов: Элемент → Рабочий процесс → Согласование документации (2 этапа)
Добавлять нужно только уникальные для данного процесса атрибуты.



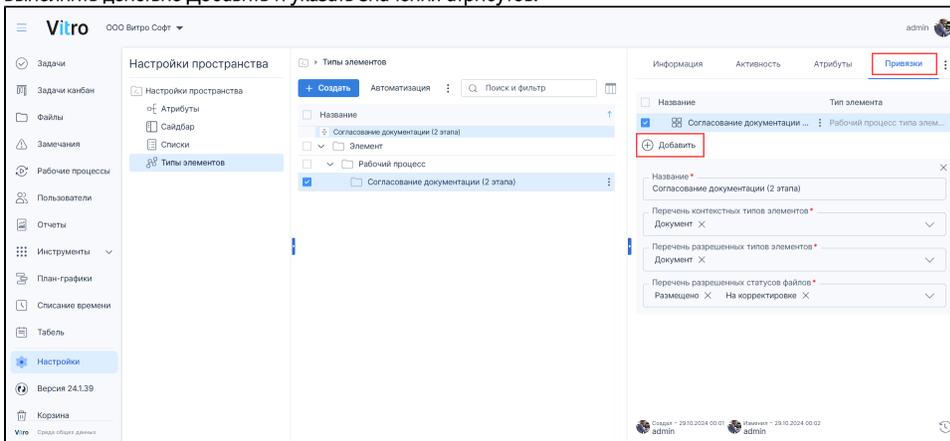
3. На вкладке "Вычисления" добавить вычисление для атрибута процесса "Название" по формуле в соответствии с требованиями к процессу.



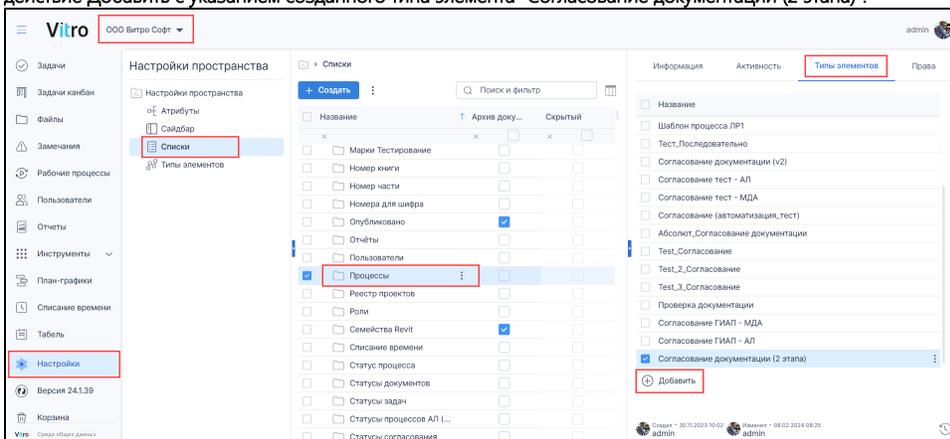
4. На вкладке "Валидаторы" добавить блокировку по условию в соответствии с требованиями к процессу. Выполнить действие **Добавить** → Блокировка изменения (формула) и указать значения атрибутов:



- "Сообщение пользователю о блокировке" - какой текст будет выведен пользователю в случае срабатывания блокировки.
 - "Условие срабатывания (формула)" - по какому событию работает проверка, в данном примере при переходе процесса в статус "Не начат".
 - "Условие блокировки (формула)" - условие проверки, в данном примере проверяется наличие хотя бы одного вложенного файла.
5. На вкладке "Привязки" добавить настройку в соответствии с требованиями к процессу. Выполнить действие **Добавить** и указать значения атрибутов:



- "Название" - название действия запуска процесса в контекстном меню.
 - "Перечень контекстных типов элементов" - типы элементов, в контекстном меню которых отобразится действие запуска процесса.
 - "Перечень разрешенных типов элементов" - типы элементов, которые попадут во вложения процесса (элементы с другими типами не попадут в процесс).
 - "Перечень разрешенных статусов файлов" - статусы элементов, которые попадут во вложения процесса (элементы с другими статусами не попадут в процесс).
6. Привязать созданный тип элемента к списку "Процессы".
В Основном пространстве перейти в Настройки → Списки, выделить список "Процессы", перейти на вкладку "Типы элементов" и выполнить действие **Добавить** с указанием созданного типа элемента "Согласование документации (2 этапа)".



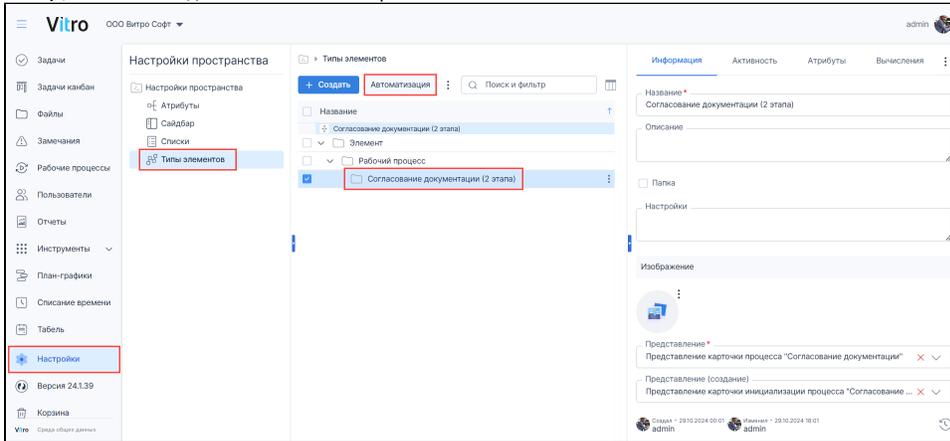
7. Настроить правила автоматизации для типа элемента в соответствии с требованиями к процессу. *Подробнее смотри раздел "Пошаговая настройка Автоматизации для типа элемента".*

- Проверить работу процесса, чтобы убедиться в корректности настройки.
- Подготовить документацию по выполненным настройкам процесса в соответствии с требованиями компании.

Пошаговая настройка Автоматизации для типа элемента

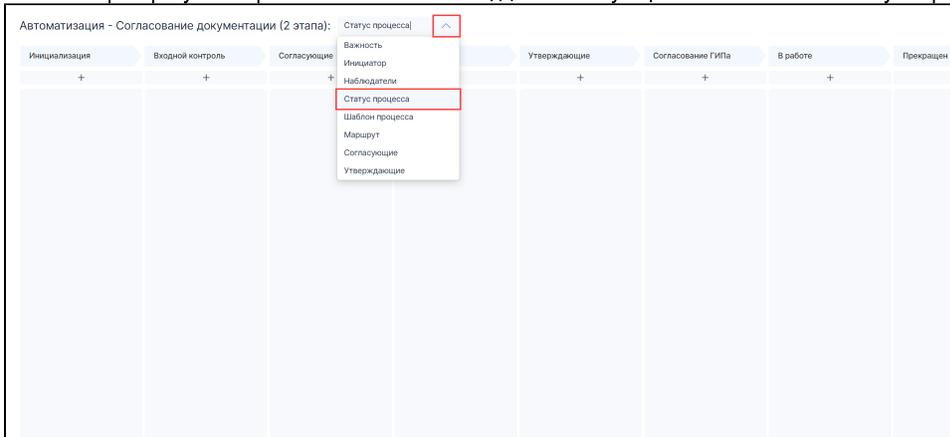
- Открыть канбан вид автоматизации для настройки жизненного цикла процесса.

Перейти в Настройки → Типы элементов, выделить тип элемента процесса, в данном примере тип элемента "Согласование документации (2 этапа)", и выполнить действие Автоматизация.

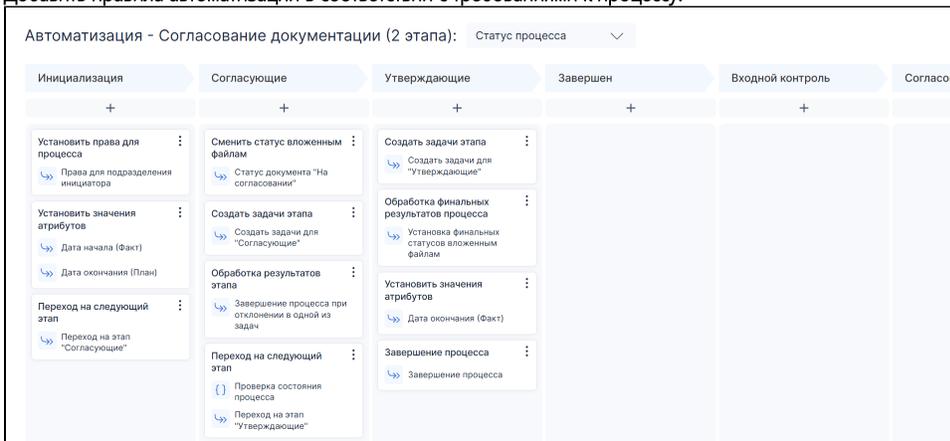


- Выбрать атрибут "Статус процесса" для построения канбан вида автоматизации.

После выбора атрибута отобразятся колонки канбан вида, соответствующие значениям списка "Статусы процесса".



- Добавить правила автоматизации в соответствии с требованиями к процессу.



Статус процесса	Название	Тип блока	Условие	Действие	Примечание
Инициализация	Установить права для процесса	Установить права на элемент		Права для подразделения инициатора	Устанавливает права на процесс для подразделения инициатора процесса.

				Уровень доступа: Изменить Формула: Parent(This("Инициатор"))	
Инициализация	Установить значения атрибутов	Установить значение поля		Дата начала (Факт) Атрибут: Дата начала (Факт) Формула: Date() Дата окончания (План) Атрибут: Дата окончания (План) Формула: AddDay(Date(), if(IsNull(This("Согласующие")), 0, This("Трудозатраты этап 1")) + if(IsNull(This("Утверждающие")), 0, This("Трудозатраты этап 2"))	Вычисляет значения для атрибутов процесса: "Дата начала (Факт)" и "Дата окончания (План)".
Инициализация	Переход на следующий этап	Установить значение поля		Переход на этап "Согласующие" Атрибут: Статус процесса Формула: Lookup("Статус процесса. Согласующие")	Переводит процесс на следующий этап "Согласующие".
Согласующие	Сменить статус вложенным файлам	Установить значение поля связанных объектов		Статус документа "На согласовании" Атрибут: Статус документа Формула: Lookup("Статусы документов. На согласовании")	Устанавливает вложенным в процесс файлам значение "На согласовании" для атрибута "Статус документа".
Согласующие	Создать задачи этапа	Этап процесса		Создать задачи для "Согласующие" Исполнители: This("Согласующие") Шаблон процесса: Параллельный этап процесса Плановая продолжительность этапа: This("Трудозатраты этап 1") Подписание ЭЦП: Нет Название задачи: Format("{0}: {1}", "Согласовать файлы", This("Статус процесса.Название")) Тип элемента: Задача на согласование	Формирует задачи на исполнителей этапа "Согласующие". В данном примере атрибут "Согласующие" позволяет указывать множественные значения, в этом случае задачи будут созданы параллельно.
Согласующие	Обработка результатов этапа	Вызов внешнего сервиса		Завершение процесса при отклонении в одной из задач Сервис обработки: Format("{0}/api/automation/completestage", Cfg("SitePortalUrl"))	Сервис "completestage" завершает процесс с результатом отклонено, если хотя бы в одной из задач этапа отклонили все файлы.
Согласующие	Переход на следующий этап	Установить значение поля	Проверка состояния процесса Формула: This("Статус процесса. Название") != "Завершен"	Переход на этап "Утверждающие" Атрибут: Статус процесса Формула: Lookup("Статус процесса. Утверждающие")	При условии, что процесс не завершен (предыдущим правилом "Обработка результатов этапа"), переводит процесс на следующий этап "Утверждающие".
Утверждающие	Создать задачи этапа	Этап процесса		Создать задачи для "Утверждающие" Исполнители: This("Утверждающие") Шаблон процесса: Параллельный этап процесса Плановая продолжительность этапа: This("Трудозатраты этап 2") Подписание ЭЦП: Нет Название задачи: Format("{0}: {1}", "Утвердить файлы", This("Статус процесса.Название")) Тип элемента: Задача на согласование	Формирует задачи на исполнителей этапа "Утверждающие".

Утверждающие	Обработка финальных результатов процесса	Вызов внешнего сервиса		<p>Установка финальных статусов вложенным файлам</p> <p>Сервис обработки: Format("{0}/api/automation/completeapprovalstage", Cfg("Site.PortalUri"))</p>	Сервис "completeapprovalstage" устанавливает вложенным файлам итоговые результаты на основании решений на данном этапе процесса.
Утверждающие	Установить значения атрибутов	Установить значение поля		<p>Дата окончания (Факт)</p> <p>Атрибут: Дата окончания (Факт) Формула: Date()</p>	Вычисляет значение для атрибута процесса "Дата окончания (Факт)".
Утверждающие	Завершение процесса	Установить значение поля		<p>Завершение процесса</p> <p>Атрибут: Статус процесса Статус процесса: Lookup("Статус процесса.Завершен")</p>	Переводит процесс в статус завершено, таким образом завершая процесс.

Валидаторы

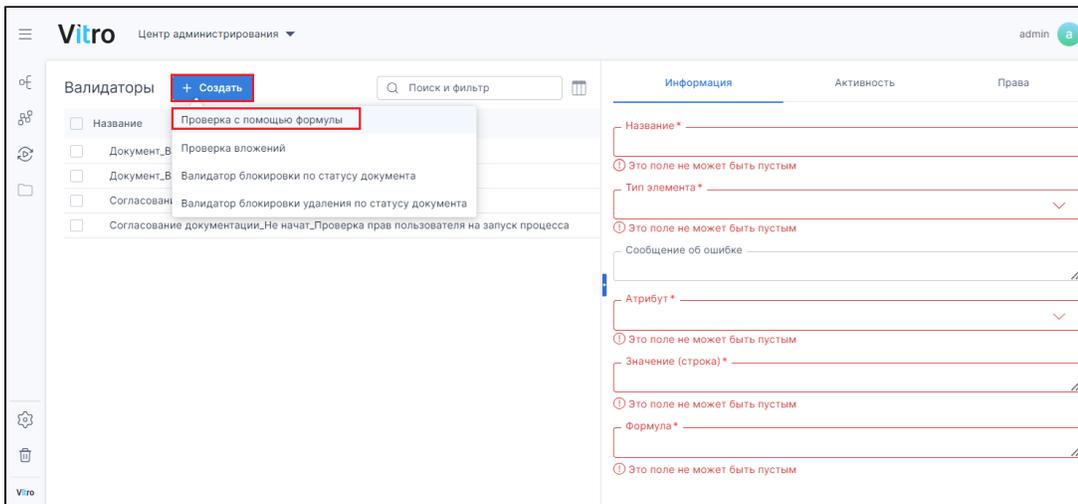
1. Валидаторы - инструмент, позволяющий осуществлять проверку по заданному условию при осуществлении определенного действия над элементом системы и в зависимости от результатов проверки разрешить данное действие или заблокировать его.

Валидаторы концептуально состоят из двух основных частей:

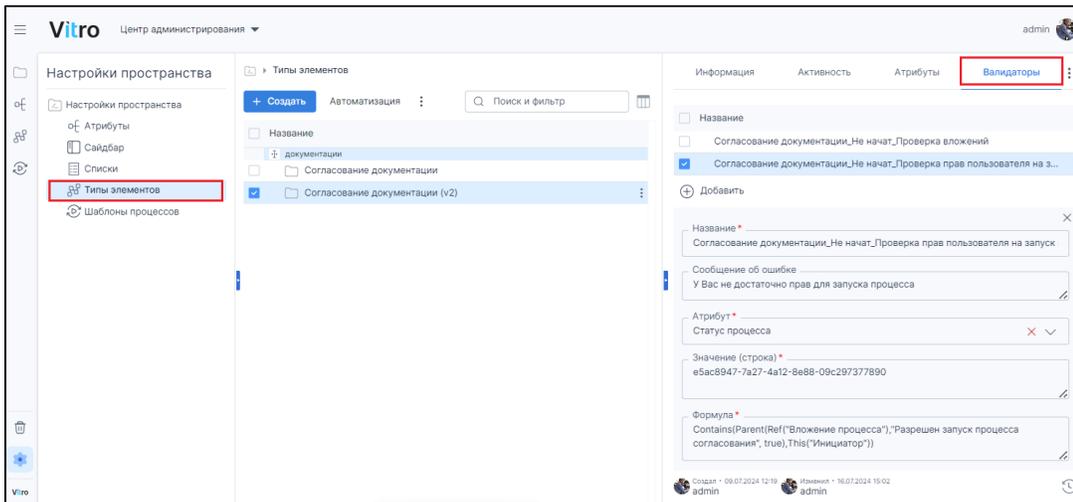
- Триггера, т.е. попытки действия над элементом, при котором срабатывает валидатор.
- Условия, которое проверяется, прежде чем данное действие будет выполнено.

2. Для создания Валидатора надо

- войти в Центр администрирования в Списки
- открыть список Валидаторы
- нажать кнопку Создать и выбрать тип проверки (тип валидатора) - описание ниже по тексту
- заполнить обязательные атрибуты в правой части



3. Количество записей (правил проверки) в списке Валидаторов не ограничено. Все Валидаторы привязаны к Типу элемента



На закладке сверху в таблице список валидаторов (проверок) для типа элемента, ниже - атрибуты выделенного валидатора из списка. Набор атрибутов зависит от типа валидатора

3. Доступные в системе типы валидаторов и их описание сведены в таблицу.

№ п/п	Тип валидатора	Описание	Атрибут	Описание атрибута	Условие срабатывания валидатора	Результат срабатывания
1	Универсальная блокировка	Валидатор блокировки действия пользователя	Название	Название валидатора (любое)	Изменение значения	Блокировка действия пользователя по изменению значения атрибута в

	(Проверка с помощью формулы)	по изменению значения заданного атрибута	Тип элемента	Тип элемента, на котором должен срабатывать валидатор Допустимые типы: любой	указанного атрибута на заданное в поле "Значение"	случае, если результатом проверки будет false
			Сообщение об ошибке	Сообщение, в случае если результатом проверки будет false		
			Атрибут	Атрибут, при указанном значении которого должна быть произведена проверка		
			Значение (строка)	Значение атрибута при котором должна быть произведена проверка		
			Формула	Формула, по которой производится проверка. Результатом вычисления формулы должно быть true или false		
2	Блокировка привязки файлов (Проверка вложений)	Валидатор проверки допустимых типов вложений и их статусов	Название	Название валидатора (любое)	Добавление связей к данному элементу (процессу, задаче)	Элементы, тип и/или статус которых будет отличаться от разрешенных не будут добавлены в качестве вложения (связи) к данному элементу
			Тип элемента	Тип элемента, на котором должен срабатывать валидатор Допустимые типы: рабочий процесс, задача		
			Сообщение об ошибке	Сообщение, в случае если результатом проверки будет false		
			Атрибут	Атрибут, при указанном значении которого должна быть произведена проверка		
			Значение (строка)	Значение атрибута при котором должна быть произведена проверка		
			Перечень разрешенных типов элемента	Перечень разрешенных типов элемента, которые могут быть добавлены в качестве вложений к данному элементу		
			Перечень разрешенных статусов документа	Перечень статусов из списка "Статусы документов", при которых разрешенный тип элемента может быть добавлен в качестве вложения к данному элементу		
3	Блокировка изменения файла по статусу (Валидатор блокировки по статусу документа)	Валидатор блокировки изменений атрибутов элемента	Название	Название валидатора (любое)	Изменение значения любого из атрибутов данного элемента	Элементы, статус которых будет отличаться от разрешенного будут заблокированы от изменений атрибутов пользователем
			Тип элемента	Тип элемента, на котором должен срабатывать валидатор Допустимые типы: любой		
			Сообщение об ошибке	Сообщение, в случае если результатом проверки будет false		
			Перечень разрешенных статусов документа	Перечень статусов из списка "Статусы документов", при которых атрибуты данного элемента могут быть изменены		
4	Блокировка удаления файла по статусу (Валидатор блокировки удаления по статусу документа)	Валидатор блокировки удаления в зависимости от статуса документа	Название	Название валидатора (любое)	Попытка удаления данного элемента	Элементы, статус которых будет отличаться от разрешенного будут заблокированы от удаления пользователем
			Тип элемента	Тип элемента, на котором должен срабатывать валидатор Допустимые типы: любой		
			Сообщение об ошибке	Сообщение, в случае если результатом проверки будет false		
			Перечень разрешенных статусов документа	Перечень статусов из списка "Статусы документов", при которых данный элемент может быть удален		

Рекомендации по обеспечению информационной безопасности

Рекомендация	Краткая инструкция
Использование трехзвенной архитектуры серверов (browser - front-server - database)	Подробнее в требованиях к конфигурации .
Использование антивируса на веб-серверах	Необходимо добавить в исключения используемые порты и протоколы .
Использование https протокола и сертификатов SSL	На примере Linux настройка сертификатов и nginx описана в статье по установке .
Смена пароля администратора, установленного по умолчанию	Для смены пароля после установки: <ol style="list-style-type: none">1. Перейти в структуру <i>Пользователи</i>2. Перейти в контейнер <i>ООО "Компания"</i>3. Вызвать контекстное меню на пользователе <i>admin</i>4. Выбрать <i>Установить пароль</i>5. Ввести старый и новый пароли и нажать <i>Ок</i>
Настройка безопасности учетных записей	Необходимо настроить параметры в соответствии с требованиями к паролю. Базовые требования: <ol style="list-style-type: none">1. Минимальная длина пароля - 12 символов2. Минимальное количество символов в нижнем регистре - 1 символ3. Минимальное количество символов в верхнем регистре - 1 символ4. Минимальное количество цифр в пароле - 1 символ5. Минимальное количество спец. символов в пароле - 1 символ Подробнее в статье о настройке .
Настройка разграничения прав доступа	Не выдавать пользователям права на изменение и удаление на списки Центра Администрирования (за исключением необходимого). Не выдавать пользователям права на изменение типов элементов и атрибутов. Подробнее о настройке и список необходимых прав доступа в статье .
Настройка протокола Kerberos для использования доменной аутентификации в Linux (опционально)	Подробнее указаны в статьях по настройке для Linux и Windows .

Журналы регистрации событий в системе

Общая информация

События в системе фиксируются в двух списках - Аудит действий пользователя и Аудит безопасности.

В список Аудит действий пользователя попадают изменения элементов системы.

В список Аудит безопасности попадают события, связанные с входом в систему.

Список Аудит действий пользователя

При внесении изменений в списки системы активируется триггер, который добавляет информацию об изменении в таблицу audit.

В списке Аудит действий пользователя отображается информация из этой таблицы.

В списке создаются записи при:

- создании\изменении\удалении элементов
- внесении изменений в атрибуты элементов
- изменении уровня доступа к элементам
- создании\удалении пользователей или изменении их параметров
- создании\удалении групп доступа или изменении их параметров
- добавлении\исключении из группы доступа

В списке по умолчанию отображается следующая информация о событии:

- ID элемента
- тип события
- имя списка
- имя элемента
- автор события
- дата события

В таблице есть более подробная информация о событии. Для вывода этой информации в списке можно использовать столбец Data.

Выбрать элементы

ID элемента	<input checked="" type="checkbox"/>	Data	<input type="checkbox"/>
Автор события	<input checked="" type="checkbox"/>	OldItem	<input type="checkbox"/>
Дата события	<input checked="" type="checkbox"/>	Идентификатор сайта	<input type="checkbox"/>
Имя списка	<input checked="" type="checkbox"/>	Тип события	<input type="checkbox"/>
Имя элемента	<input checked="" type="checkbox"/>	Тип элемента	<input type="checkbox"/>
Тип события	<input checked="" type="checkbox"/>		

Отмена

В поле Data указан список атрибутов, которые подверглись изменению.

Для каждого атрибута указано:

- имя атрибута
- старое значение
- новое значение

Пример значения поля Data для записи о создании элемента:

```
[
  {"new": "01_01", "old": null, "field": "name"},
  {"new": "1", "old": null, "field": "status"},
  {"new": "11111111-1111-1111-1111-111111111111", "old": null, "field": "list_id"},
  {"new": [{"id": "22222222-2222-2222-2222-222222222222", "name": "02_ "}], "old": null, "field": "parent_id"},
  {"new": "33333333-3333-3333-3333-333333333333", "old": null, "field": "id"},
  {"new": [{"id": "44444444-4444-4444-4444-444444444444", "name": ""}], "old": null, "field": "content_type_id"},
  {"new": "55555555-5555-5555-5555-555555555555", "old": null, "field": "site_id"}
]
```

Список Аудит безопасности

При попытке пользователя войти в систему (успешной или нет) в системе не происходит изменений.

Следовательно, такие события не могут быть обработаны триггером и не будут попадать в список Аудит действий пользователя.

По этой причине был реализован механизм для отслеживания попыток входа в систему.

Он работает на уровне веб-приложения и при попытке входа добавляет запись в таблицу audit_security.

Информация из этой таблицы отображается в списке Аудит безопасности.

В списке создаются записи при:

- успешном входе
- входе с использованием неправильного логина
- входе с использованием неправильного пароля

В списке возможно отображать следующую информацию по событию:

- тип события
- дата события
- пользователь системы, совершивший попытку входа
- логин, вводимый при попытке входа
- IP адрес, с которого совершена попытка входа

Список выявленных уязвимостей

Дата бюллетеня	ID уязвимости	Наименование ПО	Уровень угрозы	Описание уязвимости	Рекомендуемые действия
2024-02-14	CVE-2024-24990	Nginx	Высокий	Уязвимость модуля ngx_http_v3_module серверов NGINX и NGINX Plus связана с использованием памяти после ее освобождения. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании.	Обновить до версии 1.25.4 или более поздней. Ссылка на загрузку актуальной версии.
2024-01-15	CVE-2024-24989	Nginx	Средний	Уязвимость модуля HTTP/3 QUIC, позволяющая нарушителю вызвать отказ в обслуживании.	Обновить до версии 1.25.4 или более поздней. Ссылка на загрузку актуальной версии.
2024-01-22	CVE-2024-34161	Nginx	Высокий	Уязвимость модуля HTTP/3 QUIC (ngx_http_v3_module) связана с использованием памяти после ее освобождения. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, получить несанкционированный доступ к защищаемой информации с помощью специально созданных HTTP/3- запросов.	Обновить до версии 1.27.0 или более поздней либо 1.26.1 или более поздней. Ссылка на загрузку актуальной версии.
2024-02-05	CVE-2024-35200	Nginx	Высокий	Уязвимость модуля HTTP/3 QUIC, позволяющая нарушителю вызвать отказ в обслуживании.	Обновить до версии 1.27.0 или более поздней либо 1.26.1 или более поздней. Ссылка на загрузку актуальной версии.
2024-02-12	CVE-2024-31079	Nginx	Средний	Уязвимость модуля HTTP/3 QUIC (ngx_http_v3_module) связана с переполнением буфера в стеке. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании с помощью специально созданных HTTP/3- запросов.	Обновить до версии 1.27.0 или более поздней либо 1.26.1 или более поздней. Ссылка на загрузку актуальной версии.
2024-02-20	CVE-2024-32760	Nginx	Высокий	Уязвимость модуля HTTP/3 QUIC (ngx_http_v3_module) связана с записью за пределами буфера в памяти. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании с помощью специально созданных HTTP/3- запросов.	Обновить до версии 1.27.0 или более поздней либо 1.26.1 или более поздней. Ссылка на загрузку актуальной версии.
2024-03-03	CVE-2024-7347	Nginx	Средний	Уязвимость модуля ngx_http_v4_module связана с чтением вне границ памяти. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, получить несанкционированный доступ к защищаемой информации.	Обновить до версии 1.27.1 или более поздней либо 1.26.2 или более поздней. Ссылка на загрузку актуальной версии.