

Содержание

Требования к ПО на сервере	3
Подготовка сервера к установке модуля триггеров	3
Компоненты модуля Триггеров	4
Получение архива для распространения с исполняемыми файлами из VisualStudio (для разработчиков)	4
Настройка MS SQL	5
Настройка PostgreSQL	5
Настройка веб-сервера	6
Установка, настройка и запуск "Сайта"	7
Описание настроек "Сайта"	8
Основной раздел настроек "Сайта"	8
Раздел Logging	8
Подраздел File:	9
Раздел Advanta	9
Подраздел Alerts	9
Подраздел Rebus	10
Раздел Database	11
Раздел Module	12
Раздел HashiCorpVault:	12
Развёртывание "Движка" (Engine)	13
Установка и настройка "Движка" (Engine)	13
Описание настроек "Движка" (Engine)	13
Раздел Integration	13
Развёртывание "Агента"	14
Установка и настройка "Агента"	15
Описание настроек "Агента"	15
Раздел Logging	15
Подраздел File	15
Основной раздел	16
Запрос состояния "Агента"	16
Перечень использованных сторонних компонентов	16
Компонент "Сайт" (веб-приложение Панели управления)	16
Компоненты "Движок" (Engine) и "Агент"	16

Инструкция по развертыванию модуля Триггеров в Linux: версия 17.X+

Версия модуля триггеров 17.X и выше работает только с версией системы ADVANTA 3.23 и выше.

Перед началом установки модуля триггеров убедитесь, что установка Системы ADVANTA проведена корректно, иначе модуль триггеров не запустится. Информацию об установке Системы можно найти на страницах:

- [Установка и настройка системы под Windows](#);
- [Установка и настройка системы под Linux](#).

Требования к ПО на сервере

- Операционная система последней стабильной LTS-версии на базе ядра Linux - например, [Astra Linux](#), Red OS, дистрибутивы на базе Debian, Ubuntu, CentOS, Fedora, OpenSUSE, Red Hat и другие производные от них (тестировалось на ОС Ubuntu 22.04.4 LTS)
- Веб-сервер последней стабильной версии - например, Apache, Nginx или аналоги (тестировалось на Nginx 1.18.0)
- Среда выполнения Dot Net 6.0 с поддержкой ASP.NET Core (для версии модуля Триггеров 17.25 и выше. Для версий до 17.22 включительно достаточно Dot Net 5.0)
- Один из вариантов СУБД:
 - [MS SQL](#) (тестировалось на версии 2016 Express Edition+, минимальная версия MSSQL 2012) + Microsoft SQL Server Management Studio (SSMS) соответствующей версии
 - [PostgreSQL](#) версии 13+ (тестировалось на PostgreSQL 14.10)

Подготовка сервера к установке модуля триггеров

- Устанавливаем на сервер [Операционную систему на базе Linux](#)
- Устанавливаем выбранную СУБД. Рекомендуется использовать [PostgreSQL](#) (из пакетов репозитория выбранной ОС или с официального сайта разработчика).
- Устанавливаем Среду выполнения ASP.NET Core 6.0 ([Документация по вариантам установки](#))
- Устанавливаем Веб-сервер (например, Nginx)

Пример для ОС Ubuntu 22+ (репозитории-источники указанных пакетов должны быть настроены в ОС, необходимые зависимости будут установлены автоматически)

```
sudo apt update && \
sudo apt install -y postgresql aspnetcore-runtime-6.0 nginx
```

- Средствами операционной системы добавляем учетную запись пользователя, от имени

которого будут запускаться компоненты Модуля триггеров

Компоненты модуля Триггеров

Версия модуля Триггеров 17.x может корректно работать только с версиями ADVANTA 3.23 и выше.

Начиная с версии 17 модуль Триггеров является многокомпонентным. Каждая компонента должна разворачиваться отдельно.

Перечень компонент:

- **Сайт** панели управления – веб-приложение, через которое осуществляется администрирование и мониторинг исполнения задач/сценариев по обработке событий триггерами. **Сайт** осуществляет компиляцию триггеров и сборки доступа через API к основной системе ADVANTA после её обновления. **Сайт** самостоятельно не обрабатывает события, поэтому его жизненный цикл предусматривает, что он может периодически выгружаться из пула активных приложений web-сервера, что не приводит к остановке работы модуля Триггеров в целом.
- **Движок (Engine)** – консольное приложение, которое обрабатывает события системы ADVANTA, поступающие через шину данных. Подразумевается, что данный модуль работает непрерывно и выгружается, только если произошло обновление основной системы ADVANTA. При выгрузке модуль отправляет команду `restart` для управляющего **Агента**.
- **Агент** – программа, которая может быть запущена как сервис операционной системы или как консольное приложение. Задача данного компонента – управлять модулями **Движка**, развёрнутыми в рамках одного хоста. **Агент** осуществляет старт **Движков**, согласно конфигурации, а также рестарт конкретного **Движка**, если от него поступила соответствующая команда в результате обновления основной системы ADVANTA.

Получение архива для распространения с исполняемыми файлами из VisualStudio (для разработчиков)

- Открываем проект Advanta в Visual Studio 2022
- В проекте Advanta.Triggers.WebClient редактируем `appsettings.json`, в соответствии с требуемыми настройками (редактирование конфигурации можно сделать позже, на этапе установки приложения)
- Вызываем контекстное меню на проекте Advanta.Triggers.WebClient и выбираем пункт `Publish`
- Выбираем профиль «`Distribute`» и нажимаем кнопку `Publish`, после успешного завершения процесса скомпилированные исполняемые файлы будут доступны в папке «`Advanta.Triggers.WebClient/bin/Release/net5/publish/`» относительно пути расположения исходных файлов.
- Данные файлы можно заархивировать и распространять клиентам вместе с данной инструкцией по развертыванию приложения на периферии.

Настройка MS SQL

- Открываем SSMS и подключаемся к нашему инстансу MSSQL
- Заходим в глобальную секцию «Безопасность» и добавляем туда объект «Все пользователи, успешно прошедшие аутентификацию»
- Создаем новую БД.
- Во вновь созданной БД в параметрах «Безопасность» добавляем созданную нами учетную запись пользователя.
- При первичном развёртывании или обновлении мажорной версии модуля триггеров, необходимо обновить схему базы данных. Для этого первый запуск модуля триггеров должен выполняться с ключом Advanta/Database/MigrateOnStartup = true в конфигурационных файлах компонентов **Сайт** и **Движок**. Последующие запуски могут осуществляться с любым значением данного ключа.
- Сохраняем информацию о созданной БД и учетных данных пользователя для использования в строке подключения конфигурационных файлов Модуля триггеры

Подробная настройка БД MS SQL представлена на странице [Настройка базы данных MS SQL](#).

Настройка PostgreSQL

- Открываем pgAdmin либо используем режим командной строки `psql`, запущенной под админом сервера БД
- Добавляем пользователя (например, `CREATE USER user_name WITH PASSWORD 'user_password' ;`)
- Создаем новую Базу данных (например, `CREATE DATABASE db_name;`)
- Предоставляем новому пользователю полные права на созданную Базу данных (например, `GRANT ALL PRIVILEGES ON DATABASE db_name TO user_name;`)
- При первичном развёртывании или обновлении мажорной версии модуля триггеров, необходимо обновить схему базы данных. Для этого первый запуск модуля триггеров должен выполняться с ключом Advanta/Database/MigrateOnStartup = true в конфигурационных файлах компонентов **Сайт** и **Движок**. Последующие запуски могут осуществляться с любым значением данного ключа.
- Сохраняем информацию о созданной БД и учетных данных пользователя для использования в строке подключения конфигурационных файлов Модуля триггеры

Подробная настройка БД PostgreSQL представлена на странице [Настройка базы данных PostgreSQL](#).

При настройке базы данных нужно создать пустую БД и пользователя:

```
su - postgres
psql
CREATE USER user_tr WITH PASSWORD 'P@ssw0rd';
CREATE DATABASE db_tr;
GRANT ALL PRIVILEGES ON DATABASE db_tr TO user_tr;
```

Развертывание компонента "Сайт"

Веб-приложение Панели управления триггерами

Настройка веб-сервера

На примере Nginx в ОС Ubuntu 22+ ([документация](#)).

Для других ОС и веб-серверов настройки выполняются по аналогии.

- Веб-сервер должен быть установлен в операционной системе (см. выше раздел «Подготовка сервера к установке ...»)
- Запускаем его командой

```
sudo service nginx start
```

- Создаем отдельный файл конфигурации nano/etc/nginx/sites-available/advanta_triggers
- Заполняем его следующими данными:

```
map $http_connection $connection_upgrade {  
    "~*Upgrade" $http_connection;  
    default keep-alive;  
}  
  
server {  
    listen 80;  
    server_name example.com *.example.com;  
    location / {  
        proxy_pass http://127.0.0.1:5000/;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection $connection_upgrade;  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
}
```

- В параметрах сервера `listen` и `server_name` указываем корректные данные порта и адреса домена(ов), по которым будет доступно веб-приложение. При необходимости, настройте иные параметры самостоятельно стандартными средствами конфигурирования веб-сервера (например, SSL-шифрование трафика).
- Параметр `proxy_pass` в разделе `location` содержит указание на локальный адрес и порт сервера `http://127.0.0.1:5000/`, на котором будет работать компонент модуля

триггеров **Сайт** после его запуска. Изменяйте данный параметр только при скорректированных параметрах работы компонента **Сайт**, отличных от установленных по умолчанию. Обеспечьте средствами сервера (например, используя firewall) защиту указанного порта (5000) от доступа к нему извне, т.к. он должен быть доступен только локально для работы приложений в рамках одного сервера.

- Сохраняем созданный файл конфигурации и выполняем следующую команду, чтобы проверить корректность синтаксиса файлов конфигурации веб-сервера.

```
sudo nginx -t
```

- При наличии ошибок, исправляем их до тех пор, пока проверка конфигурации не будет проходить успешно.
- Создаем символьическую ссылку на наш файл конфигурации веб-приложения

```
sudo ln -s /etc/nginx/sites-available/advanta_triggers /etc/nginx/sites-enabled/advanta_triggers
```

- Применяем изменения в конфигурационных файлах веб-сервера командой

```
sudo nginx -s reload
```

- Веб-приложение с Панелью управления триггерами будет доступно через браузер по адресу и порту `server_name:listen`, указанным в конфигурационном файле веб-сервера, но только после запуска на сервере компонента **Сайт** (см. далее)
- В случае, если компонент **Сайт** Модуля триггеров не запущен или запущен с ошибками, веб-сервер будет сообщать об ошибке 502 «Bad Gateway».

Установка, настройка и запуск "Сайта"

- Копируем в отдельную папку на сервере все файлы из полученного архива с дистрибутивом приложения Модуль триггеров. Например, создав новую директорию `/home/<username>/triggers/` в домашней папке пользователя сервера, под которым планируется запуск приложения. Для размещения файлов компонента можно использовать любую другую папку на ваш выбор. Главное, чтобы она была доступна запускающему приложение пользователю ОС.
- Проверяем, что именно в этой рабочей папке находится основной исполняемый файл компонента **Сайт** - `Advanta.Triggers.WebClient.dll`
- Открываем в рабочей папке компонента файл `appsettings.json` и находим секцию `ConnectionStrings`. Меняем в ней тип подключения, адрес SQL-сервера, название БД, логин и пароль пользователя, созданные на этапе настройки Базы данных для модуля триггеров.
- При необходимости, вносим изменения в остальные параметры файла `appsettings.json` (см. ниже раздел «Описание настроек Сайта»)
- Запускаем компонент **Сайт** командой в рабочей папке под нужным пользователем ОС

```
dotnet Advanta.Triggers.WebClient.dll
```

- Приложение работает в консольном режиме и может быть остановлено сочетанием клавиш `Ctrl+C`.

- В случае успешного запуска компонента **Сайт** веб-приложение Панели управления триггерами отобразит в браузере форму авторизации для входа.
- Приложение компонента **Сайт** должно быть постоянно запущено на сервере для функционирования панели управления в веб-браузере. При необходимости, средствами операционной системы его можно настроить в виде службы с автозапуском и автоматической перезагрузкой в случае сбоев ([документация](#)).
- В ходе работы компонент **Сайт** сохраняет журналы своей работы в подпапке рабочего каталога, указанной в файле `appsettings.json` в разделе `Logging/File/BasePath`.
- Если необходима диагностика, почему не запускается компонент **Сайт**, то необходимо в файле `web.config` найти секцию `aspNetCore` и поменять параметр `stdoutLogEnabled` на значение `true`

Описание настроек "Сайта"

Настройки компонента приложения **Сайт** осуществляются в файле `appsettings.json`, размещенном в рабочей папке компонента. Так как приложение можно запустить в различных конфигурациях, то в проекте есть несколько файлов типа: `appsettings.ConfigurationName.json`.

При получении архива для распространения с исполняемыми файлами используется файл с настройками `appsettings.json` для запуска компонента в режиме `Production`.

Основной раздел настроек "Сайта"

WorkingDirectory – относительный или абсолютный путь, по которому размещается рабочая папка компонента.

Необязательный параметр. Значение по умолчанию: "`<external>`" (относительный путь к рабочей папке от размещения файла `appsettings.json`).

Раздел `Logging`

```
"Logging": {
  "File": {
    "RootPath": "/home/<username>/triggers",
    "BasePath": "Logs",
    "LogLevel": {
      "Default": "Information"
    }
  }
},
```

Данный раздел настраивается в соответствии с правилами ведения журнала в .NET Core.

Подраздел File:

RootPath – абсолютный путь по которому будет размещаться папка с логами. По умолчанию – полный путь до рабочей папки с компонентом приложения

BasePath – относительный от **RootPath** путь к папке, в которой будут создаваться файлы с логами. По умолчанию - "Logs". Конечный путь до папки с файлами логов будет сформирован объединением путей: [RootPath]/[BasePath]

LogLevel – набор фильтров сообщений поступающих в лог. Формируется по правилам стандартного логирования в Asp.Net Core. По умолчанию - "Default": "Information". Для полного отключения логов работы компонента, необходимо внутри данной секции указать параметр "Default": "None".

Раздел Advanta

```
"RuntimeUser": {
    // "Token": "your_token",
    // или
    "Login": "your_login",
    "Password": "your_password"
},
"Host": "https://your_instance_domain.ru/",
"ApiRequestTimeout": 300, //секунд
"ObjectsSyncTimeout": 60, //минут
```

RuntimeUser – логин и пароль (или токен с версии системы ADVANTA 3.27) пользователя, от чьего имени будут вестись запросы в API системы ADVANTA при исполнении триггеров (указываются либо параметры Login и Password, либо параметр Token). Под этим пользователем через API также осуществляется запрос на извлечение данных о типах объектов.

Host – URL-адрес сервера для осуществления запросов в API системы ADVANTA.

ApiRequestTimeout – период ожидания ответа от SOAP API системы ADVANTA. Задаётся в секундах. Параметр необязательный. Если параметр не задан или его значение равно 0, то будет использовано значение по умолчанию равное 3600 (т.е. 3600 секунд, что равно одному часу).

ObjectsSyncTimeout – период синхронизации типов объектов с системой ADVANTA. Задаётся в минутах. Параметр необязательный. Значение по умолчанию 15 (т.е. 1 раз в 15 минут)

Подраздел Alerts

```
"Alerts": {
    "Emails": "admin_email@my_email_hosting.ru;
admin_email2@my_email_hosting.ru",
    "SmtpServer": {
```

```

        "Address": "smtp.my_email_hosting.ru",
        "Login": "mylogin@my_email_hosting.ru",
        "Password": "12345",
        "Port": "587",
        "UseSSL": "true"
    }
},

```

Emails – список почтовых адресов, на которые необходимо рассылать уведомления о перезапуске или сбое в работе триггеров. Список необходимо указать через любой из разделителей: запятая, точка с запятой, пробел.

SmtpServer - содержит настройки smtp сервера для отправки email уведомлений:

- **Address**: адрес smtp сервера,
- **Login**: логин (email) отправителя,
- **Password**: пароль отправителя,
- **Port**: номер порта, если требуется,
- **UseSSL**: требование использовать SSL (true/false) по умолчанию false

Подраздел Rebus

В данном разделе указываются параметры подключения к Базе данных системы ADVANTA кшине обмена сообщениями Rebus. Вид транспорта и строка подключения к БД ConnectionString должны быть идентичны параметрам, используемым в настройках системы ADVANTA.

```

"Rebus": {
    //Варианты транспорта: "MSSQL", "PostgreSql"
    "Transport": "PostgreSql",

    // пример строки подключения для транспорта "MSSQL"
    // "ConnectionString": "Data Source =
    SqlServerName;Database=AdvantaRebus;Trusted_Connection=True;MultipleActiveRe
    sultSets=true",

    // пример строки подключения для транспорта "PostgreSql"
    "ConnectionString": "User
    ID=username;Password=userpwd;Host=localhost;Port=5432;Database=AdvantaRebus;
    ",

    "InputQueueName": "bus_triggers_queue", // "Bus_TriggersInputQueue" -
    для транспорта "MSSQL"
    "SubscriptionsTableName": "Bus_TriggersInputQueue_Subscriptions"
    "PostgreSqlMessagesTableName": "bus_triggersinputqueue" //используется
    только для транспорта "PostgreSql"
}

```

Transport – тип транспорта для Rebus-шины сообщений. Возможные варианты: "MSSQL" и "PostgreSql". Если параметр не указан или пустой, то по умолчанию используется значение "MSSQL".

Возможен также вариант "Emulator", который используется для запуска приложения без привязки к Rebus.

ConnectionString – строка подключения к базе данных ADVANTA, через которую публикуются сообщения шины Rebus.

InputQueueName – название очереди сообщений (канала), через которую будут поступать сообщения для данного instans-а. Для транспорта "MSSQL" по умолчанию будет "Bus_TriggersInputQueue", для "PostgreSql" по умолчанию будет "bus_triggers_queue".

SubscriptionsTableName – название таблицы БД, в которую будет размещена информация о подписках. По умолчанию будет "Bus_TriggersInputQueue_Subscriptions"(должно совпадать с настройками публикующего сервиса).

PostgreSqlMessagesTableName – название таблицы БД, через которую будет вестись обмен сообщениями в случае "PostgreSql" транспорта. По умолчанию будет использоваться "bus_triggersinputqueue" (должно совпадать с настройками публикующего сервиса). Для "MSSQL" этот параметр игнорируется, так как для "MSSQL" Rebus автоматически использует (и при необходимости создаёт) таблицу с именем, совпадающим с названием входящей очереди сообщений (**InputQueueName**).

Раздел Database

В этом разделе указываются настройки подключения к собственной служебной Базе данных модуля Триггеров, отдельной от БД системы ADVANTA.

```

"Database": {
    //Варианты провайдеров: "MSSQL", "PostgreSql"
    "Provider": "PostgreSql",

    // пример строки подключения для провайдера "MSSQL"
    // "ConnectionString": "Data Source =
    // SqlServerName;Database=AdvantaTriggers;Trusted_Connection=True;MultipleActiv
    // eResultSets=true",

    // пример строки подключения для провайдера "PostgreSql"
    "ConnectionString": "User
    ID=username;Password=userpassword;Host=localhost;Port=5432;Database=AdvataTr
    iggers;",
    "MigrateOnStartup": "true"
},

```

Provider – определяет тип СУБД, используемой для модуля триггеры. Возможные варианты: "MSSQL" и "PostgreSql". Если параметр не указан или пустой, то по умолчанию используется значение "MSSQL". Возможен также вариант "Memory". В этом варианте СУБД использоваться

не будет, а все данные модуля будут храниться в памяти до завершения работы приложения. Данный режим удобно использовать в процессе разработки и тестирования.

ConnectionString – строка подключения к базе данных, в которой размещается информация о настроенных триггерах и их работе.

MigrateOnStartup – признак автоматической миграции структуры базы данных при запуске приложения. Принимаемые значения "true" или "false". Параметр необязательный. Значение по умолчанию "false".

Флаг "true" может потребоваться, только если есть возможность обновления и развёртывания схемы данных через миграции (например, во время разработки или создания локальной базы).

Если значение флага установлено в "true", пользователю базы данных потребуются права на создание, изменение, удаление таблиц и индексов.

Раздел Module

```
"Module": {  
    "InstanceName": "My Triggers Module"  
},
```

InstanceName – наименование запускаемого инстанса модуля Триггеры. Данное наименование добавляется к логам и в тему email-уведомлений.

Раздел HashiCorpVault:

Данный раздел настраивается только в том случае, если требуется часть конфиденциальных строк получать через [HashiCorp Vault](#).

Address – адрес сервера Vault, обязательный параметр. Пример значения: <https://vault.yourcompany.ru:8200>.

RoleId – Идентификатор (GUID) роли, обязательный параметр. Пример значения: «b52f920d-dccf-76ad-e7e0-a2d3eb23abc3».

SecretId – Идентификатор (GUID) секрета, обязательный параметр. Пример значения: «cfa2b4f6-4a92-5ec3-5253-acf5a026ddec».

Version – версия контейнера. Возможные варианты: «V1» или «V2», обязательный параметр.

MountPoint – точка монтирования контейнера секретов, обязательный параметр. Пример значения: «kv-v2».

Path – путь к секретам, обязательный параметр. Пример значения: dbPasswords.

Ссылка на секрет в Vault могут прописываться для любых значений строковых параметров в любых разделах файла конфигурации (кроме раздела HashiCorpVault) в формате: {\$your_hashicorp_vault_key\$}

Развертывание "Движка" (Engine)

Консольное приложение, которое обрабатывает события системы ADVANTA, поступающие в Модуль триггеров через шину данных Rebus

Установка и настройка "Движка" (Engine)

- Исполняемые файлы **Движка** размещаются в подпапку Engine общего дистрибутива при публикации всех компонентов Модуля триггеров.
- Содержимое папки Engine может быть целиком перенесено в любую другую папку сервера (например, в папку /home/<username>/triggers_engine - в имени каталогов рекомендуется использовать буквы только в нижнем регистре).
- Движок** является консольным приложением ОС Linux, которое запускается из рабочей папки данного компонента командой:

```
dotnet Advanta.Triggers.Engine.dll
```

- Движок** должен располагаться на том же хосте (сервере), что и компонент **Сайт**.
- Движок** может быть запущен вручную либо передан под управление **Агенту**.
- Одному **Сайту** должен соответствовать один экземпляр работающего **Движка**.

Описание настроек "Движка" (Engine)

- Настройки компонента **Движок** осуществляются в отдельном файле appsettings.json, который расположен в корне рабочей папки данного компонента. Так как приложение можно запустить в различных конфигурациях, то в проекте есть несколько файлов типа: appsettings.ConfigurationName.json.
При получении архива для распространения с исполняемыми файлами используется файл с настройками appsettings.json для запуска компонента **Движок** в режиме "Production".
- Для запуска **Движка** необходимо использовать те же настройки файла appsettings.json, которые используются для запуска компонента **Сайт**. Отличаться может только необязательный раздел Logging. При отсутствии в настройках данного раздела журналы событий компонента создаются по умолчанию в подкаталоге logs рабочей папки компонента.
- В случае несовпадения ключевых настроек в разделах "Database", "Advanta", "Rebus" конфигурационных файлов appsettings.json **Движка** и **Сайта**, они не смогут «найти» друг друга в процессе инициализации канала коммуникации по UDP-протоколу.
- Начиная с версии Модуля триггеров 18.4 в настройках **Движка** может использоваться дополнительный optionalный раздел "Integration", отвечающий за его подключение к очереди сообщений RabbitMQ.

Раздел Integration

```
"Integration": {
  "RabbitMQ": {
    "Endpoints": [
```

```

{
    "EndpointName": "publisher", // Название точки подключения
    "EndpointMode": "Publish", // Варианты: Publish, Consume
    "ConnectionString": null, // Необходимо указать либо
ConnectionString, либо Connection
    "Connection": { // Если указаны оба параметра, то будет использован
ConnectionString
        "Host": "localhost",
        "Port": 5672,
        "VirtualHost": null
    },
    "Channel": {
        "Exchange": "test",
        "Queue": "test"
    },
    "RoutingKey": null,
    "ReconnectTimeoutInMinutes": 1
},
{
    "EndpointName": "consumer", // Название точки подключения
    "EndpointMode": "Consume", // Варианты: Publish, Consume
    "ConnectionString": null, // Необходимо указать либо
ConnectionString, либо Connection
    "Connection": { // Если указаны оба параметра, то будет использован
ConnectionString
        "Host": "localhost",
        "Port": 5672,
        "VirtualHost": null
    },
    "Channel": {
        "Exchange": "test",
        "Queue": "test"
    },
    "RoutingKey": null,
    "ReconnectTimeoutInMinutes": 1
}
}
}
}

```

RabbitMQ - подраздел описывает параметры точек подключения **Движка** модуля триггеров к очередям брокера сообщений **Endpoints** - массив точек подключения с параметрами публикации и получения сообщений. Информацию для заполнения конфигурации необходимо получить от администратора RabbitMQ

Развертывание "Агента"

Приложение для автоматического перезапуска Движка (Engine) модуля триггеров при обновлении связанной с ним системы ADVANTA

Установка и настройка "Агента"

- Исполняемые файлы **Агента** размещаются в подпапку Agent общего дистрибутива при публикации всех компонентов Модуля триггеров. Содержимое папки Agent может быть целиком перенесено в любую другую папку сервера (например, в папку /home/<username>/triggers_agent).
- По умолчанию **Агент** сконфигурирован как консольное приложение ОС Linux, которое запускается из рабочей папки данного компонента:

```
dotnet Advanta.Triggers.Agent.dll
```

- Запущенный такой командой **Агент** по умолчанию выводит логи своей работы и в консоль, из которой он запущен, и в файл журнала.
- Для прекращения работы **Агента** необходимо в консоли с запущенным приложением нажать Ctrl+C.
- Для развертывания **Агента** как службы можно воспользоваться [инструкцией](#), чтобы обеспечить автоматический запуск компонента при загрузке системы и перезапуск в случае сбоев приложения.
- Для запуска **Агента** в режиме вывода логов только в файл необходимо запустить приложение с ключом -console.

```
dotnet Advanta.Triggers.Agent.dll -console
```

- При старте **Агент** будет пытаться запустить все экземпляры **Движков (Engine)**, перечисленные в разделе "ModulesPaths" файла настроек appsettings.json, которая расположена в корне рабочей папки компонента **Агент**.

Описание настроек "Агента"

Настройки приложения осуществляются в файле appsettings.json. Так как приложение можно запустить в различных конфигурациях, то в проекте несколько файлов типа: appsettings.ConfigurationName.json. При получении архива для распространения с исполняемыми файлами используется файл с настройками appsettings.json.

Раздел Logging

Данный раздел настраивается в соответствии с [правилами ведения журнала в .NET Core](#).

Подраздел File

RootPath – абсолютный путь по которому будет размещаться папка с логами. По умолчанию полный путь до папки с приложением.

BasePath – относительный путь до папки, в которой будут создаваться файлы с логами. По умолчанию Logs. Конечный путь до папки с файлами логов будет сформирован объединением путей: [RootPath]\[BasePath]

LogLevel – набор фильтров сообщений поступающих в лог. Формируется по правилам стан-

дартного логирования в Asp.Net Core. Для полного отключения логов, необходимо внутри данной секции указать параметр "Default": "None".

Основной раздел

ModulesPaths – массив относительных или абсолютных путей, по которым размещаются рабочие папки управляемых Engine модулей.

Данный раздел файл appsettings.json мониторится агентом динамически, поэтому пути до управляемых Engine могут быть прописаны как до старта, так и после старта агента.

При добавлении очередного пути, агент пытается определить есть ли Engine по этому пути. Если есть, то определяет стартовал ли Engine, если нет, то стартует его.

При удалении пути агент посыпает соответствующему Engine модулю команду shutdown.

Запрос состояния "Агента"

Для получения информации по текущему состоянию агента, необходимо в папку агента положить файл "status" без расширения. Содержимое файла не читается. После того как агент обнаружит файл, он сформирует информацию о текущем состоянии движков и создаст или обновит файл output.txt. Информация о статусе так же будет записана в лог. Файл "status" будет автоматически удалён.

Перечень использованных сторонних компонентов

Компонент "Сайт" (веб-приложение Панели управления)

- **Material UI** – Фреймворк для верстки. Содержит в себе набор компонентов в стиле google material. Лицензия MIT.
- **Axios** – HTTP клиент для запросов к API. Лицензия MIT.
- **Classnames** – Библиотека для модификации классов элементов DOM дерева. Лицензия MIT.
- **Lodash** – Библиотека с набором функций для работы с данными. Лицензия MIT.
- **Monaco Editor** – это редактор кода, который можно встроить в браузер. Лицензия MIT.
- **QueryString** – Библиотека для удобного извлечения query параметров из адресной строки браузера. Лицензия MIT.
- **StyledComponents** – Библиотека стилизации компонентов. Лицензия MIT.

Компоненты "Движок" (Engine) и "Агент"

- **Stateless** – Библиотека для описания и исполнения в .net core коде машины состояний. APACHE LICENSE, VERSION 2.0
- **Npgsql.EntityFrameworkCore.PostgreSQL** – PostgreSQL provider для Entity Framework Core ORM. Лицензия по адресу <https://github.com/npgsql/efcore.pg/blob/main/LICENSE> “

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted."

- **Rebus** – Инфраструктура для организации очереди сообщений. Лицензия MIT.

From:

<https://wiki.a2nta.ru/> - Wiki [3.x]



Permanent link:

https://wiki.a2nta.ru/doku.php/product/triggers/installation/17x_linux

Last update: **22.10.2025 04:43**