

## Содержание

<b>Требования к ПО на сервере:</b> .....	3
<b>Настройка и установка системы:</b> .....	3
<b>Компоненты системы</b> .....	3
<b>Получение архива для распространения с исполняемыми файлами из VisualStudio (для разработчиков):</b> .....	4
<b>Настройка MSSQL:</b> .....	4
<b>Настройка PostgreSQL:</b> .....	5
<b>Настройка IIS:</b> .....	5
<b>Установка и настройка сайта:</b> .....	6
<b>Описание настроек сайта:</b> .....	6
Основной раздел: .....	6
Раздел Logging: .....	7
Подраздел File: .....	7
Раздел Advanta: .....	7
Подраздел Alerts: .....	7
Подраздел SmtпServer .....	7
Подраздел Rebus: .....	8
Раздел Database: .....	9
Раздел Module: .....	9
<b>Развёртывание агента</b> .....	10
Установка и настройка: .....	10
Описание настроек агента: .....	10
Раздел Logging: .....	10
Подраздел File: .....	10
Основной раздел: .....	10
Запрос состояния агента: .....	11
<b>Развёртывание Engine</b> .....	11
Установка и настройка: .....	11
Описание настроек агента: .....	11
<b>Перечень использованных сторонних компонентов</b> .....	11
Клиент: .....	11
Сервер (Движок / Engine) .....	12



# Инструкция по развёртыванию модуля Триггеров в Windows: версия 17.X+

Версия модуля триггеров 17.X. работает с версией системы 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:

- IIS 8.5
- Windows server 2012 R2 +
- Один из вариантов СУБД:
  - MSSQL (тестировалось на версии 2016 Express Edition+, минимальная версия MSSQL 2012) + Microsoft SQL Server Management Studio (SSMS) соответствующей версии
  - PostgreSQL версии 13+ (возможна установка на более раннюю версию Postgres, но тогда потребуется вносить правки в код)
- .net 6 hosting bundle (с версии 17.25 и выше, до версии 17.22 включительно достаточно .net 5)

## Настройка и установка системы:

- Устанавливаем ОС Windows Server.
- Устанавливаем СУБД
- Устанавливаем Net 6 hosting bundle
- Добавляем роль IIS через графическую утилиту «Диспетчер Сервера»
- Добавляем учетную запись пользователя в систему, от имени которого будет работать сервис через оснастку «Локальные пользователи и группы»

## Компоненты системы

Начиная с версии 17 система «модуля Триггеров» является многокомпонентной.

Каждая компонента должна разворачиваться отдельно.

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

- **Сайт** – это web-приложение, через которое осуществляется администрирование и мониторинг исполнения задач, по обработке событий триггерами. **Сайт** осуществляет

компиляцию триггеров и сборки доступа через API к основной системе ADVANTA после её обновления. **Сайт** не обрабатывает события, поэтому его жизненный цикл предусматривает, что он будет периодически выгружаться из пула активных приложений web-сервера, что не приводит к остановке работы модуля Триггеров в целом.

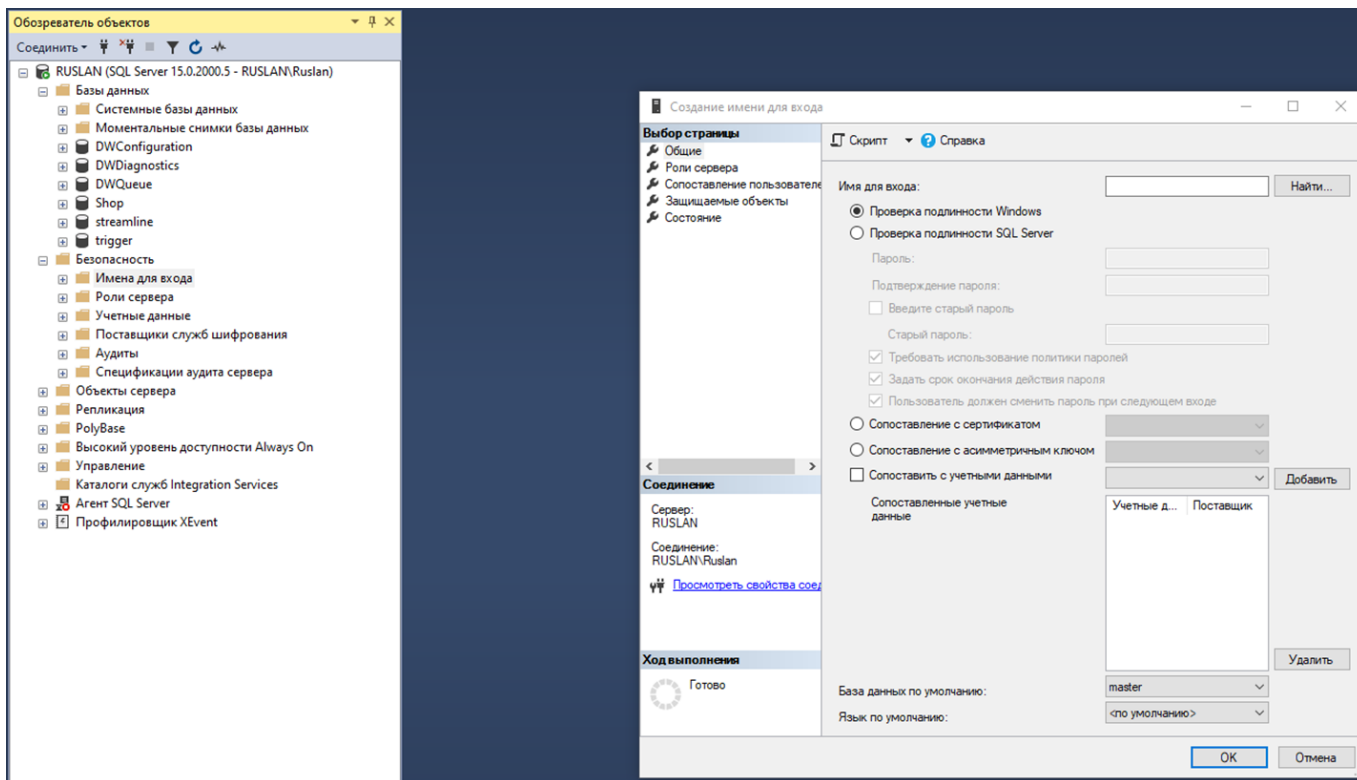
- **Движок (Engine)** – консольное приложение, которое обрабатывает события системы ADVANTA, поступающие через шину данных. Подразумевается, что данный модуль работает непрерывно и выгружается, только если произошло обновление основной системы ADVANTA. При выгрузке модуль отправляет команду restart для управляющего **Агента**.
- **Агент** – сервис, который может быть запущен как Windows-service или как консольное приложение. Задача данного сервиса – управлять модулями **Engine**, развёрнутыми в рамках одного хоста. **Агент** осуществляет старт **Engine**-консолей, согласно конфигурации, а также рестарт конкретного **Engine**, 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 с вышеописанной инструкцией по развертыванию приложения на периферии.

## Настройка MSSQL:

- Открываем SSMS и подключаемся к нашему экземпляру 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, необходимо обновить схему базы данных. Для этого первый запуск должен выполняться с ключом Adavanta/Database/MigrateOnStartup = true, последующие запуски могут осуществляться с любым значением данного ключа.

## Настройка PostgreSQL:

- Открываем pgAdmin
- Добавляем пользователя
- Создаем новую БД
- При первичном развёртывании или обновлении мажорной версии, необходимо обновить схему базы данных. Для этого первый запуск должен выполняться с ключом Adavanta/Database/MigrateOnStartup = true, последующие запуски могут осуществляться с любым значением данного ключа

## Развёртывание сайта

### Настройка IIS:

- Открываем утилиту управления IIS «Диспетчер служб IIS»
- Создаем новый пул для приложения:
- Имя - любое
- Пул приложения - «Без управляемого кода»

- Режим управляемого конвейера - «Встроенный»
- После создания пула заходим в «Дополнительные параметры», находим пункт «Удостоверение» и меняем его на использование «Особой учетной записи», в качестве учетной записи используем созданную выше учетную запись пользователя.
- В этом же окне «Дополнительные параметры» находим пункт «Режим запуска» и выбираем режим «Always running» - необходимо чтобы процесс всегда работал и не завершался при простое (с версии 17.x не актуально).
- Создаем новый сайт:
  - Имя - любое
  - Пул - выбираем только что созданный нами пул приложения
  - Физический путь - выбираем путь, где будет храниться исполняемые файлы приложения
  - Привязка - вводим доменное имя сайта
- Выбираем в списке сайтов вновь созданный сайт, заходим в контекстное меню и выбираем пункт «Редактировать разрешение», переходим на вкладку «Безопасность» и добавляем туда созданную выше учетную запись пользователя
- Заходим в основные настройки сайта и нажимаем «Тест настроек», должно открыться модальное окно, в котором все пункты будут отмечены зелеными индикаторами, говорящие об успешности настройки доступа сайта и файловой системы.

## Установка и настройка сайта:

- Копируем в папку, указанную в настройках сайта исполняемые файлы из полученного архива с приложением.
- Открываем файл appsettings.json и находим секцию «ConnectionStrings», меняем в ней адрес SQL сервера и название БД.
- Если необходима диагностика почему не запускается приложение, необходимо в файле web.config найти секцию «aspNetCore» и поменять параметр «stdoutLogEnabled» на true

## Описание настроек сайта:

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.

## Основной раздел:

**WorkingDirectory** – относительный или абсолютный путь, по которому будет размещаться рабочая папка.

Значение по умолчанию: “external” (относительный путь).

## 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 стандартного логирования в Asp.Net Core. Для полного отключения логов, необходимо внутри данной секции указать параметр “Default”: “None”.

## Раздел Advanta:

**RuntimeUser** – логин и пароль пользователя, от чьего имени будут вестись запросы в API системы Advanta при исполнении триггеров (параметры Login и Password соответственно). Под этим пользователем, через API так же осуществляется запрос на извлечение данных о типах объектов.

**Host** – URL адрес сервера для осуществления запросов в API системы Advanta.

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

**ApiRequestTimeout** – период ожидания ответа 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 ча-су).

### Подраздел Alerts:

**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.

### Подраздел SmtпServer

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 к шине 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_triggers_input_queue_subscriptions", //
"Bus_TriggersInputQueue_Subscriptions" - для транспорта "MSSQL"
  "PostgreSqlMessagesTableName": "bus_triggers_messages" //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 так же вариант “Emulator” используется для запуска приложения без привязки к rebus.

**ConnectionString** – строка подключения к базе данных, через которую публикуются сообщения rebus.

**InputQueueName** – название очереди сообщений (канала), через которую будут поступать сообщения для данного instans-а. Для MSSQL по-умолчанию будет «Bus\_TriggersInputQueue», для PostgreSQL по-умолчанию будет “bus\_triggers\_queue”.

**SubscriptionsTableName** – название таблицы, в которую будет размещена информация о подписках. Для MSSQL по-умолчанию будет «Bus\_TriggersInputQueue\_Subscriptions», для PostgreSQL по-умолчанию будет “ bus\_triggers\_input\_queue\_subscriptions ”(должно совпадать с

настройками публикующего сервиса).

**PostgreSqlMessagesTableName** – название таблицы, через которую будет вестись обмен сообщениями в случае PostgreSQL транспорта. По-умолчанию будет использоваться “bus\_triggers\_messages” (должно совпадать с настройками публикующего сервиса). Для MSSQL параметр игнорируется, так как для MSSQL Rebus 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;MultipleActiveResultSets=true",  
  
  // пример строки подключения для провайдера "PostgreSql"  
  "ConnectionString": "User  
ID=username;Password=userpassword;Host=localhost;Port=5432;Database=AdvantaTriggers";",  
  "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 так же вариант “Memory”. В этом варианте СУБД использоваться не будет, а все будет храниться в памяти до завершения работы приложения. Данный режим удобно использовать в процессе разработки и тестирования.

**ConnectionString** – строка подключения к базе данных, в которой размещаются триггеры.

**MigrateOnStartup** – признак автоматической миграции структуры базы данных. 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”, потребуются права на создание, изменение, удаление таблиц и индексов.

## Раздел Module:

**InstanceName** – наименование запускаемого инстанса модуля Триггеры. Данное наимено-

вание добавляется к логам и в тему email уведомлений.

## Развёртывание агента

### Установка и настройка:

- Исполняемые файлы публикуются в подпапку Agent после публикации. Содержимое папки мо-жет быть перенесено в любую другую папку хоста.
- По умолчанию агент сконфигурирован как windows service, для развёртывания агента как службы можно воспользоваться инструкцией компании Microsoft. Для запуска агента в консольном режиме необходимо запустить приложение Advanta.Triggers.Agent с ключом - console.
- При старте агент будет пытаться запустить все экземпляры движков, перечисленные в разделе ModulesPaths файла appsettings.json.

### Описание настроек агента:

Настройки приложения осуществляются в файле appsettings.json. Так как приложение мож-но запустить в различных конфигурациях, то в проекте несколько файлов типа: appsettings.ConfigurationName.json. При получении архива для распространения с исполня-емыми 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 логам. По умолча-нию полный путь до папки с приложением.

**BasePath** – относительный путь до папки, в которой будут создаваться файлы с логам. По умолчанию Logs. Конечный путь до папки с файлами логов будет сформирован объедине-нием 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 ра-

бочие 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 посылает 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.

## Развёртывание Engine

### Установка и настройка:

- Исполняемые файлы публикуются в подпапку Engine после публикации. Содержимое папки мо-жет быть перенесено в любую другую папку хоста.
- Движок должен располагаться на том же хосте, что и сайт.
- Движок может быть запущен вручную либо передан под управление агенту.
- Одному сайту должен соответствовать один экземпляр работающего движка.

### 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.

Для запуска движка необходимо использовать те же настройки appsettings.json, которые используются для запуска сайта.

Отличаться может только раздел Logging. В противном случае движок и сайт не смогут найти друг друга в процессе инициализации канала коммуникации по UDP протоколу.

## Перечень использованных сторонних компонентов

### Клиент:

- **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.

## Сервер (Движок / Engine)

- **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?rev=1736834729>

Last update: **14.01.2025 06:05**

