

Логирование

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

NLog	4
Библиотека логирования NLog	4
Настройка логирования в Creatio	4
Логирование для приложений on-site	4
Логирование для приложений cloud	7
Настроить логирование NLog	7
Настройка логирования через конфигурационный файл	7
Настройка логирования через конфигурационный объект LoggingConfiguration	8
Логирование входящих http-запросов в Creatio для .NET Core	8
Стандартное логирование входящих http-запросов	9
Расширенное логирование входящих http-запросов	9

NLog



Средний

Библиотека логирования NLog

Для проверки корректности работы новой функциональности, мы рекомендуем включить логирование. Чтобы избежать ухудшения производительности, рекомендуется включать логирование только при тестировании и отладке приложения. В Creatio выполняется логирование всех основных операций. Для этого используется [NLog](#) — бесплатная библиотека логирования для .NET с широкими возможностями маршрутизации и управления высококачественными логами для приложения, независимо от его размера или сложности. Библиотека может обрабатывать диагностические сообщения, отправляемые с любого языка .NET, дополнять контекстной информацией, форматировать в соответствии с предпочтениями пользователя и отправлять одному или нескольким получателям сообщений, таким как файл или база данных.

Документация по работе с библиотекой NLog доступна на [сайте GitHub](#).

Настройка логирования в Creatio

Логирование ведется отдельно для загрузчика приложения и для конфигурации `Default`. Настройки логирования выполняются в каталоге `..\Terrasoft.WebApp` в следующих конфигурационных файлах:

- `nlog.config`.
- `nlog.targets.config`.
- `nlog.cloud.config` (для приложений, развернутых в облаке).
- `nlog.cloud.settings.config` (для приложений, развернутых в облаке).

Пример настройки логирования содержится в статье [Реализовать модальное окно](#).

Логирование для приложений on-site

Хранение логов

Местоположение файлов логов зависит от значения системных переменных Windows.

Путь к файлам с логами загрузчика (по умолчанию)

```
[TEMP]\Creatio\Site_{SiteId}\{ApplicationName}\Log\{DateTime.Today}
```

Пример пути

```
C:\Windows\Temp\Creatio\Site_1\creatio7121\Log\2018_05_22
```

Путь к файлам с логами конфигурации Default

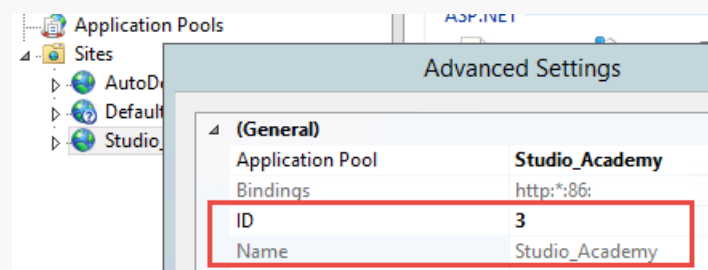
```
[TEMP]\Creatio\Site_{SiteId}\[{ApplicationName}]\[ConfigurationNumber]\Log\[DateTime.Today]
```

Пример пути

```
C:\Windows\Temp\Creatio\Site_1\creatio7121\0\Log\2018_05_22
```

В приведенных выше примерах в квадратных скобках указаны следующие переменные:

- [*TEMP*] — базовый каталог. По умолчанию для IIS используется каталог `C:\Windows\Temp`, а для Visual Studio (IIS Express) — `C:\Users\{Имя пользователя}\AppData\Local\Temp`.
- [*{SiteId}*] — номер сайта. Для IIS указан в расширенных настройках сайта. Для Visual Studio номер содержит значение 2.
- [*{ApplicationName}*] — название приложения.
- [*ConfigurationNumber*] — номер конфигурации. Конфигурация `Default`, как правило, имеет номер 0.
- [*{DateTime.Today}*] — дата логирования.



Получатели логирования

Получатели логирования используются для отображения, хранения или передачи сообщений лога другому получателю. Есть получатели, которые получают и обрабатывают сообщения и те, которые буферизируют или направляют сообщения другому получателю.

Получатели логирования задаются в файле `nlog.targets.config` с помощью следующих атрибутов:

- `name` — имя получателя.
- `xsi:type` — тип получателя. Может принимать значения "File", "Database", "Mail".
- `fileName` — файл и путь к файлу, куда будут записываться логи.
- `layout` — шаблон, по которому будет заполняться файл.

Получатели логирования подробно описаны в [документации NLog](#).

Правила логирования

Правила логирования устанавливаются в файле `nlog.config` с помощью следующих атрибутов:

- `name` — имя лога.
- `minlevel` — минимальный уровень логирования.
- `maxlevel` — максимальный уровень логирования.
- `level` — логирование событий одного уровня логирования.
- `levels` — логирование событий нескольких уровней логирования (записываются через запятую).
- `writeTo` — имя получателя логирования.
- `final` — после совпадения окончательного правила правила не обрабатываются.
- `enabled` — отключить правило (установить в значение `false`), не удаляя его.
- `ruleName` — идентификатор правила.

Атрибуты правил обрабатываются в следующем порядке:

1. `level`.
2. `levels`.
3. `minlevel` и `maxlevel` (имеют одинаковый приоритет).

Если `minLevel = "Warn"` `level = "Info"`, то будет использоваться только уровень логирования `Info`. Атрибут `level` имеет высший приоритет обработки, чем `minLevel`.

Уровни логирования

Уровни логирования устанавливаются в файле `nlog.config`. По умолчанию уровень логирования для всех компонентов Creatio установлен таким образом, чтобы обеспечить максимальную производительность приложения. Возможные уровни логирования в порядке возрастания приоритета:

- `Trace` — логирование событий при отладке трассировки (указываются начальный и конечный методы).
- `Debug` — логирование всех событий при отладке.
- `Info` — нормальная работа приложения.
- `Warn` — логирование предупреждений (приложение будет продолжать работу).
- `Error` — логирование ошибок (возможно прекращение работы приложения).
- `Fatal` — логирование ошибок, приводящих к прекращению работы приложения.
- `off` — логирование отключено, не используется для записей.

Способы настройки логирования

Логирование можно настраивать следующими способами:

- через конфигурационный файл;
- через конфигурационный объект `LoggingConfiguration`.

Логирование для приложений cloud

Для настройки логирования в приложении, развернутом в облаке, необходимо обратиться в службу технической поддержки Terrasoft.

Важно. При настройке логирования приложения cloud нет возможности добавить дополнительный получатель логирования.

Настроить логирование NLog

 Средний

Настройка логирования через конфигурационный файл

1. Задать путь к файлу с настройками логирования

Путь к файлу `nlog.config` задается в конфигурационном файле `..\Terrasoft.WebApp\Web.config`.

```
..\Terrasoft.WebApp\Web.config
```

```
<common>
  <logging>
    <factoryAdapter type="Common.Logging.NLog.NLogLoggerFactoryAdapter, Common.Logging.NLog45"
      <arg key="configType" value="FILE" />
      <arg key="configFile" value="~/nlog.config" />
    </factoryAdapter>
  </logging>
</common>
```

2. Задать получателей логирования

Для этого необходимо в XML-элементе `<target></target>` файла `..\Terrasoft.WebApp\nlog.targets.config` определить свойства получателя логирования.

```
..\Terrasoft.WebApp\nlog.targets.config
```

```
<target name="universalAppender" xsi:type="File"
  layout="${DefaultLayout}"
  fileName="${LogDir}${LogDay}${logger:shortName=True}.log" />
```

3. Задать правило логирования

Пример правила, которое реализует запись лога в базу данных

```
<logger name="IncidentRegistration" writeTo="AdoNetBufferedAppender" minlevel="Trace" final="tru
```

Настройка логирования через конфигурационный объект LoggingConfiguration

Для программной настройки логирования необходимо выполнить следующие шаги:

1. Создать объект `LoggingConfiguration`, который будет описывать конфигурацию.
2. Создать одного или нескольких получателей.
3. Установить свойства получателей.
4. Определить правила с помощью объектов `LoggingRule` и добавить их в конфигурационные `LoggingRules`.
5. Активировать конфигурацию, указав в `LogManager.Configuration` созданный объект конфигурации.

Логирование входящих http-запросов в Creatio для .NET Core



Легкий

Возможность логирования входящих http-запросов доступна для приложений Creatio на платформе **.NET Core** с версии 7.17.3.

Для приложений Creatio на платформе **.NET Core** стандартное логирование включено по умолчанию. Для использования расширенного логирования необходимо выполнить его настройку. Настройка логирования входящих http-запросов выполняется в секции `RequestLogging` конфигурационного файла `appsettings.json`, который находится в корневом каталоге приложения.

Секция настройки логирования файла `appsettings.json`

```
"RequestLogging": {
  "Standard": {
    "Enabled": true,
    "StatusCodes": [ 200, 300, 302 ]
```



```

    },
    "Extended": {
        "Enabled": true,
        "LogRequestBody": false,
        "MaxBodySizeBytes": 500,
        "StatusCodes": [ 400, 401, 403 ]
    }
}

```

Стандартное логирование входящих http-запросов

Стандартное логирование позволяет получить информацию о логах входящих http-запросов. Стандартные логи записываются в файл `Request.log`, который находится в каталоге `Logs` корневого каталога приложения.

Настроить стандартное логирование:

1. Откройте конфигурационный файл `appsettings.json`, который находится в корневом каталоге приложения.
2. Перейдите в секцию `Standard`. Стандартное логирование по умолчанию включено (значение `true` флага `Enabled`).
3. В флаге `StatusCodes` перечислите http-коды ответа, для которых необходимо выполнять запись логов. Если включить флаг `Enabled` и оставить незаполненным флаг `StatusCodes`, то запись логов будет выполняться для всех входящих http-запросов.

Для анализа логов можно использовать утилиту Log Parser Studio, поскольку формат логов аналогичен формату логов для IIS.

Расширенное логирование входящих http-запросов

Расширенное логирование позволяет получить детальную информацию о логах входящих http-запросов. Расширенные логи записываются в файл `ExtendedRequest.log`, который находится в каталоге `Logs` корневого каталога приложения. При включенном расширенном логировании информация о теле ответа входящего http-запроса будет записана автоматически.

Настроить расширенное логирование:

1. Откройте конфигурационный файл `appsettings.json`, который находится в корневом каталоге приложения.
2. Перейдите в секцию `Extended`. Расширенное логирование по умолчанию отключено (значение `false` флага `Enabled`).
3. Включите логирование, установив значение `true` для флага `Enabled`.
4. При необходимости получить в логе информацию о теле http-запроса, установите значение `true` для флага `LogRequestBody` (по умолчанию — `false`).
5. Задайте размер выводимого тела запроса/ответа (первые N байт) для флага `MaxBodySizeBytes`.

6. В флаге `StatusCodes` перечислите http-коды ответа, для которых необходимо выполнять запись логов. Если включить флаг `Enabled` и оставить незаполненным флаг `StatusCodes`, то запись логов будет выполняться для всех входящих http-запросов.

Допускается одновременное использование стандартного и расширенного логирования, которые могут отличаться значением флага `StatusCodes`. Если для стандартного и расширенного логирования совпадают значения флага `StatusCodes`, то логи входящих http-запросов будут продублированы в файлах `Request.log` и `ExtendedRequest.log` с разной степенью детализации.

Важно. Расширенное логирование влияет на производительность при большом количестве запросов или большом значении флага `MaxBodySizeBytes`.