



ExpoServer

(внешняя компонента)

20.04.2013
Exponenta.info
ver. 1.5.0.101

Оглавление

Введение	2
Установка	2
Системные требования.....	4
Регистрация и пробный период	4
Свойства	5
int AuthentType { get; set; }	5
int PortNum { get; set; }	5
int HashCount { get; }	5
int HashPeriod { get; set; }	5
int WaitPeriod { get; set; }	5
int FromHash { get; set; }	6
int FromBase { get; set; }	6
int SendError { get; set; }	6
int WathAll { get; set; }	6
int WathBase { get; set; }	6
string Log { get; set; }	6
Методы	7
bool Start (string RegName, string RegCode)	7
bool Stop ()	7
bool Ready ()	7
void SetResponse (string Response, string RespCode, int Status, bool NoHash, string mime, string cookie)	7
string GetHash()	8
void AddHash(string Key, string Value, string Mime)	8
void ClearHash()	8
string MD5(string input)	8
string POST(string URL, string Data, string ContentType, int Timeout);	8
string GET(string URL, int Timeout);	9
void ShowMainWindow();	9
void HideMainWindow(string text);	9
void AddTrayCommand(string text, string msg);	9
События	10
Событие «Request»	10

Введение

Библиотека ExpoServer – многопоточный кэширующий веб сервер. Эта внешняя компонента для 1С:Предприятие позволяет получать запросы по http протоколу и отвечать на них. Высокую производительность серверу обеспечивает одновременная обработка нескольких клиентских запросов и внутренний кэш данных компоненты. Таким образом, компоненте при получении запроса нужно только один раз обратиться к 1С:Предприятию за ответом, а далее практически мгновенно отсылать данные клиенту. При необходимости, возвращая ответ из 1С:Предприятие, можно запретить помещать данные в кэш.

ExpoServer имеет возможность отсылать как текстовые данные, так и двоичные (картинки, файлы).

Текущая версия компоненты поддерживает Anonymous, Basic и Windows аутентификацию (последние 2 только в зарегистрированной версии).

Общая последовательность обработки запроса:

1. Получение запроса. Создание нового программного потока для обработки.
2. Проверка аутентификации (при необходимости).
3. Поиск в кэше записи с запрашиваемым URL. Если запись найдена и не устарела, переход к пункту 7.
4. Формирование из данных запроса xml структуры.
5. Вызов в 1С:Предприятие внешнего события и передача xml данных.
6. Получение ответа из 1С:Предприятие.
7. Отправка ответа клиенту.

Компонента имеет и дополнительные методы, которые могут быть полезны для работы: хэш строк по алгоритму md5, сворачивание главного окна 1С в трей и другие.

Установка

Компонента должна быть зарегистрирована в системном реестре Windows. Для регистрации можно использовать прилагаемый ExpoServer.bat, предварительно указав в нем правильный путь к ExpoServer.dll. Запускать скрипт регистрации нужно с правами администратора.

Так же можно организовать распаковку компоненты из макета в конфигурации 1С:Предприятие и ее регистрацию. Пример подключения компоненты к 1С:

```
Если Компонента = Неопределено Тогда
    Если Не ПодключитьВнешнююКомпоненту("AddIn.ExpoServer") Тогда
        Каталог = КаталогПрограммы();
        ФайлКомпоненты = ПолучитьИзВременногоХранилища(ПолучитьКомпоненту());
        ФайлКомпоненты.Записать(Каталог + "ExpoServer.dll");

        ТекстКоманды = Новый ТекстовыйДокумент;
```

```

ТекстКоманды.ДобавитьСтроку("@ECHO OFF");
ТекстКоманды.ДобавитьСтроку("set DOTNETFX4 =
%SystemRoot%\Microsoft.NET\Framework\v4.0.30319");
ТекстКоманды.ДобавитьСтроку("set PATH=%PATH%;%DOTNETFX4%");
ТекстКоманды.ДобавитьСтроку("regasm.exe "" + КаталогПрограммы() +
"ExpoServer.dll" /codebase /tlb");
ТекстКоманды.ДобавитьСтроку("pause");
ТекстКоманды.Записать(Каталог + "ExpoServer.bat", КодировкаТекста.ANSI);

ЗапуститьПриложение(Каталог + "ExpoServer.bat");
Предупреждение("Внешняя компонента ExpoServer была проинсталлирована в
каталог" + Символы.ПС + Каталог);
КонецЕсли;

Если ПодключитьВнешнююКомпоненту("AddIn.ExpoServer") Тогда
    Попытка
        Компонента = Новый("AddIn.ExpoServer");
    Исклучение
        Предупреждение(ОписаниеОшибки());
    КонецПопытки;
КонецЕсли;
КонецЕсли;

```

В этом примере компонента подключается на клиенте 1С и перед этим была помещена во временное хранилище методом ПолучитьКомпоненту() – он здесь не приведен.

Для успешной регистрации компоненты в реестре, 1С:Предприятие должно быть запущено **с правами администратора системы**. Так же с правами администратора должен производиться **первый вызов метода Start()** компоненты для регистрации номера прослушиваемого порта. Для изменения номера порта сервера последовательность действий следующая:

- запустить 1С с правами администратора;
- изменить номер порта;
- запустить сервер, вызвав метод Start();
- перезапустить 1С с обычными пользовательскими правами.

Выше приведенный алгоритм актуален для версий Windows 7, Windows Server 2008 и старше. Для предыдущих версий корректная работа компоненты возможна резервировании пространства имен сайта из командной строки. Подробности описаны в статье <http://msdn.microsoft.com/ru-ru/library/ms733768.aspx>.

Системные требования

1С:Предприятие 8.2.15-8.3;

Microsoft .Net Framework 4.

Регистрация и пробный период

Незарегистрированная версия компоненты может использоваться «как есть» без каких-либо гарантий. При такой эксплуатации она имеет ряд ограничений, а именно:

1. Обработка всех запросов будет происходить последовательно только одним потоком.
2. Отключается кэш компоненты и все запросы обрабатываются 1С.
3. Отключается возможность Windows и Basic аутентификации клиентов, остается только вариант Anonymous аутентификации.

Для получения регистрационного ключа необходимо приобрести компоненту и выслать на адрес support@exponenta.info имя компьютера (NetBIOS имя), на котором будет работать компонент. В ответном письме Вам будет выслан ключ регистрации.

Свойства

int AuthenType { get; set; }

Тип аутентификации, используемый при доступе к серверу. Допустимые значения:

0 – Anonymous (значение по умолчанию). Доступ к ресурсам сервера открыт всем. Аутентификацию необходимо реализовывать на стороне 1С средствами встроенного языка.

1 – Basic. При обращении к серверу браузер клиента автоматически запрашивает имя пользователя и пароль, передает их на сервер. Имя и пароль будут доступны в 1С в параметрах запроса User и Password, где могут быть обработаны.

2 – Integrated Windows Authentication. Проверка осуществляется с использованием учетных записей пользователей, зарегистрированных в ОС сервера. Пароль пользователя при этом недоступен, но в параметре User в 1С передается имя пользователя.

После изменения свойства понадобится перезапуск сервера для того, чтобы изменения вступили в силу.

int PortNum { get; set; }

Номер прослушиваемого порта. По умолчанию устанавливается значение «8080». После изменения свойства понадобится перезапуск сервера для того, чтобы изменения вступили в силу.

int HashCount { get; }

Количество записей в кэше. Свойство недоступно для записи.

int HashPeriod { get; set; }

Время жизни записи в кэше в секундах. Значение по умолчанию равно 1. Устаревшие записи из кэша удаляются автоматически. Для очистки кэша можно воспользоваться методом ClearHash().

Изменение значения возможно в процессе работы сервера и вступает в силу немедленно.

int WaitPeriod { get; set; }

Время ожидания ответа от 1С в миллисекундах. Значение по умолчанию равно 1. В случае, если за указанное время от 1С не пришли данные, клиенту будет отправлен ответ с кодом состояния сервера 500 и строкой описания «Timeout».

Изменение значения возможно в процессе работы сервера и вступает в силу немедленно.

int FromHash { get; set; }

Количество ответов отправленных клиентам из кэша компоненты. Попытка установки любого значения сбрасывает счетчик.

int FromBase { get; set; }

Количество ответов, для формирования которых потребовался вызов 1С. Попытка установки любого значения сбрасывает счетчик.

int SendError { get; set; }

Количество запросов, на которые не удалось сформировать ответ, и были отсланы сообщения об ошибках. Попытка установки значения сбрасывает счетчик.

int WathAll { get; set; }

Среднее время ответа на запрос в миллисекундах для всех запросов – и тех, которые обрабатывались 1С, и тех, которые обработаны кэшем компоненты. Время замеряется от начала обработки запроса до полной отсылки ответа клиенту. Попытка установки значения сбрасывает счетчик.

int WathBase { get; set; }

Среднее время получения данных из 1С от вызова события до получения данных. Не учитывается время обработки данных и отправки клиенту. Попытка установки значения сбрасывает счетчик.

string Log { get; set; }

Путь к лог-файлу, в котором регистрируется ход работы. Установка значения включает debug режим работы (**сильно снижает производительность!**), повторная установка значения – пустой строки отключает debug режим.

Методы

bool Start (string RegName, string RegCode)

Запустить сервер. В качестве параметров принимает регистрационное имя и код. В случае успешного запуска возвращает Истина. Оба параметра обязательны и должны быть переданы. В случае отсутствия регистрации – пустые строки.

bool Stop ()

Остановить сервер. Компонента перестает принимать запросы от клиентов. В случае успешной остановки возвращает Истина.

bool Ready ()

Состояние сервера. Если сервер запущен и готов принимать запросы, возвращает Истина.

void SetResponse (string Response, string RespCode, int Status, bool NoHash, string mime, string cookie)

Ответить на запрос. Метод вызывается в ответ на событие Request компоненты. Параметры:

Response – текст html страницы или двоичные данные, закодированные по алгоритму base64. Как текст компонента воспринимает полученное значение в том случае, если параметр **Mime** содержит подстроку «text», например «text/plain». Во всех остальных случаях перед передачей клиенту происходит обратное преобразование из base64 в поток байтов.

RespCode – код клиентского запроса, для которого формировался этот ответ. Код передается в 1С в данных внешнего события компоненты в атрибуте Number тега Request (см. событие Request).

Status – статус ответа (200 – ОК и проч.).

Отдельный случай – установка статуса равным 303. При этом браузер клиента не получает данные веб страницы, а должен быть перенаправлен на другую страницу. Новый адрес нужно передавать в параметре **Response**. Параметр **Mime** при этом должен содержать подстроку „text”, а **NoHash** = true.

NoHash – признак запрета кэширования данного ресурса. Значение Истина запрещает помещать ответ в кэш, а значит следующее обращение к этому ресурсу приведет к повторному вызову 1С:Предприятие.

Mime – тип данных ресурса.

Cookie – xml структура с cookie для клиента. Например:

```
<?xml version="1.0" encoding="utf-8"?>
```



```
<Cookie>
  <Cook Name="Session" Value="123456789" Path="/" Domain="mysite.com">
</Cookie>
```

Описание допустимых параметров (атрибутов) cookie доступно [по ссылке](#).

string GetHash()

Получить содержимое кэша компоненты. Возвращает xml представление содержимого кэша компоненты.

void AddHash(string Key, string Value, string Mime)

Добавить значение в кэш. Параметры:

Key – URL ресурса, по которому значение будет искать компонента. Например, если необходимо поместить в кэш картинку, которая находится «в корне» сайта, значение будет «/picture.png».

Value – значение, помещаемое в кэш: для текста – строка, двоичные данные необходимо закодировать по алгоритму base64.

Mime – строка, определяющая тип передаваемых данных. Если не содержит подстроки «text», тогда переданное **Value** компонента воспринимает как двоичные данные и будет проведено обратное base64 декодирование. Например, для картинки – «image/png».

void ClearHash()

Очистить кэш. Удаляет все записи из кэша компоненты.

string MD5(string input)

Рассчитывает для переданной строки хэш по алгоритму MD5. Для кодирования строки перед расчетом используется UTF-8. Возвращает его 16-ричное представление, т. е. строку из 32 символов.

string POST(string URL, string Data, string ContentType, int Timeout);

Отправка данных на указанный сервер POST методом. Возвращает строку – результат запроса.

Параметры:

URL – адрес запрашиваемого ресурса. Должен быть указан полностью, с префиксом «http://». Например «http://www.google.com/recaptcha/api/verify».

Data – строка - отправляемые данные.

ContentType – тип содержимого. Например «multipart/form-data».

Timeout – время ожидания ответа от сервера в миллисекундах.

В случае, если запрашиваемый ресурс не отвечает, метод вернет строку "No response from the server".

string GET(string URL, int Timeout);

Функция для получения ресурсов по протоколу http с использованием GET метода. Параметры соответствуют одноименным параметрам функции POST, описанной выше.

void ShowMainWindow();

Восстанавливает главное окно 1С из свернутого в трей состояния.

void HideMainWindow(string text);

Сворачивает главное окно 1С в трей. Текст, переданный в качестве параметра, будет отображаться как всплывающая подсказка при наведении указателя мыши на иконку в трее. Этот же текст отображается как название пункта контекстного меню, по которому можно развернуть окно 1С.

Функция не работает в демонстрационном режиме работы компоненты, а так же до вызова метода Start().

void AddTrayCommand(string text, string msg);

Добавляет команду в контекстное меню иконки трея. Параметры:

text – наименование пункта меню.

msg – строка, которая будет отправлена в 1С внешним событием компоненты, при выборе данного пункта меню.

События

В 1С события обрабатываются в predetermined процедурах ОбработкаВнешнегоСобытия(Источник, Событие, Данные). Все события, генерируемые внешней компонентой, имеют в поле Источник строку «ExproServer». В поле Событие передается имя события или краткое сообщение об ошибке. В поле Данные передаются связанные с текущим клиентским запросом данные или, если это событие-ошибка, подробное описание ошибки.

Событие «Request»

Событие возникает при получении запроса от клиента. Контекст запроса и параметры передаются в xml структуре. Пример xml пакета данных:

```
<?xml version="1.0" encoding="utf-8"?>
<Request Number="37489757" HttpMethod="POST" Host="127.0.0.1" Port="8080"
Url="/mod/find.1c" RemoteIP="127.0.0.1" RemotePort="63650" User="" Password="">
  <Headers>
    <Header Name="Content-Length" Value="35" />
    <Header Name="Accept" Value="text/html, */*; q=0.01" />
    <Header Name="Accept-Charset" Value="windows-1251,utf-8;q=0.7,*;q=0.3" />
    <Header Name="Accept-Language" Value="ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4" />
  </Headers>
  <QueryString />
  <PostData>
    <Post Name="find" Value="холод" />
  </PostData>
  <Cookie>
    <Cook Name="Session" Value="55bc0e90-4871-43d6-a52e-77d9a515a510" Domain=""
Path="" Port="" Comment="" Discard="False" Expired="False" HttpOnly="False"
Secure="False" />
  </Cookie>
</Request>
```

Начинается и заканчивается запрос элементом Request.

Атрибут **Number** содержит уникальный номер запроса, который должен быть возвращен вместе с ответом компоненте при вызове метода SetResponse().

Атрибут **HttpMethod** - метод HTTP, определенный клиентом. Как правило, методом HTTP является GET или POST в зависимости от действия, требуемого клиентом.

Атрибут **Url** соответствует [HttpListenerRequest.Url.AbsolutePath](#) и содержит сведения о пути, используемые сервером для разрешения информационных запросов. Сведения о пути не включают схему, имя узла или часть URI, представляющую запрос.

Атрибуты **RemoteIP** и **RemotePort** содержат клиентский IP-адрес и номер порта, которые являются источниками запроса.

В элементе **<Headers>...</Headers>** содержатся [заголовки запроса](#).

Далее, элемент **<QueryString>...</QueryString>** содержит вложенные элементы с данными, переданными GET методом («имя=значение»). Имя каждого элемента - имя в запросе.

В элементе **<PostData>...</PostData>** передаются данные полученные в теле запроса.