

УДК 004

***ИССЛЕДОВАНИЕ ПРОЦЕССА ВНЕДРЕНИЯ КОМПОНЕНТА-
РАСШИРЯЕМОСТИ В ASP.NET ХОСТ ПРИЛОЖЕНИЕ***

Шакенов Д.С.

*Студент магистратуры, факультет программной инженерии и
компьютерной техники,*

*Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики,
Россия, г. Санкт-Петербург.*

Аннотация: в статье приводится исследование процесса внедрения компонента-расширения в ASP.NET хост приложение. Рассмотрены конвейеры обработки HTTP-запросов в IIS и ASP.NET с подробным описанием этапов. Описан способ перехвата HTTP-запроса с последующей его модификацией. Также приведен способ извлечения метаданных (классов, свойств, полей) хост приложения.

Ключевые слова: компонент-расширения, платформа .NET, ASP.NET, C#, HTTP, IIS, Middleware.

***STUDY OF THE PROCESS OF IMPLEMENTATION OF THE
EXTENSIBILITY COMPONENT IN THE APPLICATION HOST ASP.NET***

Shakenov D.S.

*Master's degree student, faculty of software engineering and computer
engineering,*

*St. Petersburg national research University of information technologies,
mechanics and optics,*

Russia, St. Petersburg.

Abstract: the article presents a study of the process of implementation of the component-expansion in the ASP.NET host application. The processing of pipelines and HTTP requests in IIS and ASP.NET with a detailed description of the stages. The method of interception of the HTTP request Protocol with its subsequent modification is described. It also provides a way to extract metadata (classes, properties, fields) of the host application.

Key words: component-extensions, platform. Clean, ASP.NET in C#, HTTP server, middleware.

В настоящее время, разработка приложений является процессом совмещения множества компонентов логики, часть из которых имеется возможность повторно использовать в других разработках. Наиболее популярным способом расширения приложения являются библиотеки, предоставляющие набор кода для решения прикладной задачи. Компонент-расширяемости обеспечивает схожий функционал, но ключевая особенность состоит в взаимодействии с исходным кодом хост-приложения через пользовательский интерфейс.

Любое веб-приложение размещается на веб-сервере, ASP.NET приложения в этих целях используют Microsoft Internet Information Services (IIS). IIS является серверным приложением, с которым можно связаться, используя различные интернет протоколы, такие как HTTP, FTP, SMTP и прочие [13].

Процесс внедрения компонента в хост-приложение осуществляется путём добавления сборки и вызова соответствующего метода-инициализатора. Это достигается за счет внедрения Middleware в конвейер обработки HTTP-запроса. В .NET за поступающие запросы отвечает IIS, он же является точкой входа в приложение и обеспечивает дальнейшую

отправку ответа. Ниже представлена схема жизненного цикла запроса внутри IIS сервера (Рисунок 1).

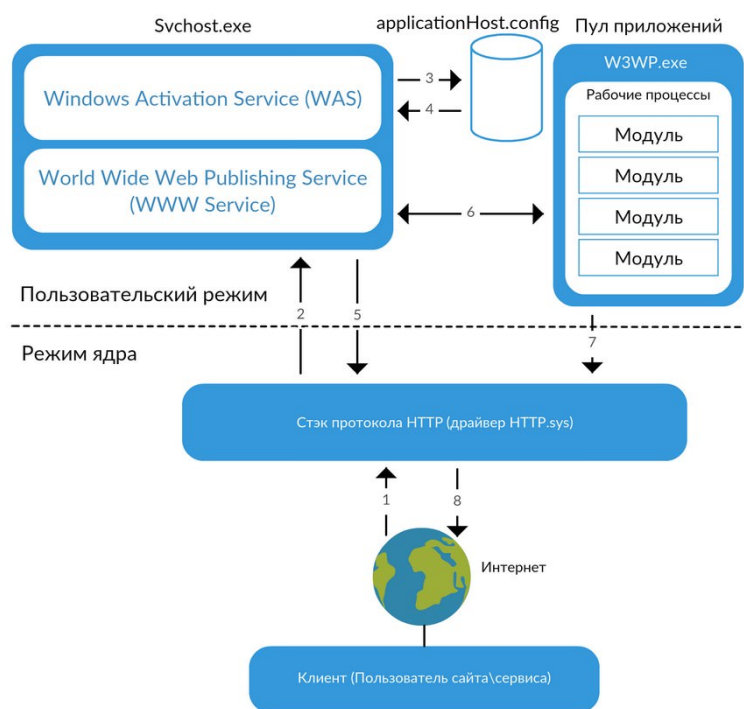


Рисунок 1. Конвейер обработки HTTP-запроса в IIS.

Этапы конвейера обработки HTTP-запроса в IIS [2]:

1. Пользователь обращается к веб-странице по указанному URL, чем инициирует запуск конвейера обработки;
2. Драйвер **HTTP.sys** перехватывает запрос и затем обращается в службу **WAS** для получения конфигурации;
3. **WAS** извлекает конфигурацию из хранилища – из **applicationHost.config**, который содержится в папке IIS;
4. Полученная конфигурация о имеющимся пуле приложений передается в службу **W3SVC**;
5. **W3SVC**, используя переданную информацию конфигурирует **HTTP.sys**;

6. **WAS** запускает **W3WP.exe**, если таковой процесс отсутствует в списке запущенных;

7. **W3WP.exe** получает запрос, формирует ответ на основе определенной логики и затем передает его в драйвер **HTTP.sys**;

8. **HTTP.sys** завершает обработку запроса и отправляет ответ.

Цепочка действий при обработке HTTP-запроса в ASP.NET располагается в этапах, связанных с процессом W3WP.exe, где обрабатывается веб-приложение, отвечающее за обработку запроса и формирование ответа. Оно так же имеет собственный конвейер обработки HTTP-запроса (Рисунок 2). Данный конвейер обработки запросов является унифицированным, поддерживает как модули с собственным кодом, так и модули управляемого кода [2]. ASP.NET позволяет внедрить собственную логику обработки запроса на один из уровней, путем его перехвата и передачи уже модифицированного запроса на уровень выше.

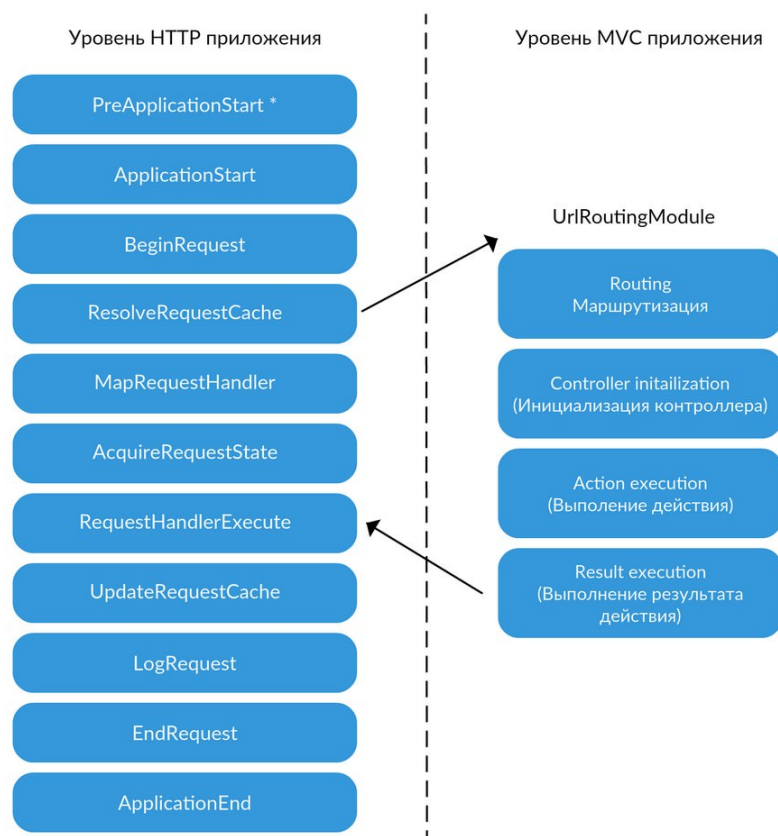


Рисунок 2. Конвейер обработки HTTP-запроса в ASP.NET

Middleware – это сквозной компонент, с помощью которого образуется конвейер между сервером и приложением, выполняя инспекцию, маршрутизацию и изменение запроса и ответа с определенной целью. Перехват HTTP-запроса происходит во время передачи его с уровня HTTP приложения на уровень MVC приложения, и после обработки логики, определенной в Middleware, возвращает на уровень MVC приложения. Допускается неограниченное количество внедрений различных Middleware компонентов в конвейер, образуя цепочку обработки HTTP-запросов. Наиболее популярным использованием являются логирование возникших ошибок во время выполнений приложения и использование статичных файлов, для отображения изображений, видео и прочего контента.

Ключевой особенностью компонента-расширения является использование исходного кода хост-приложения для предоставления нового функционала. System.Reflection является стандартной библиотекой, позволяющей извлекать метаданные об исходном коде приложения, тем самым позволяя оперировать с ними различными способами. Таким образом можно получить информацию о существующих классах, типах, свойствах и прочих составляющих [1].

ASP.NET допускает создание компонентов для расширения функционала хост-приложение, но выполняется это нетривиальным образом, а именно через использование Middleware. Он обеспечивает логику внедрения для модификации HTTP-запросов и формирования ответов. Стандартная библиотека System.Reflection предоставляет API для получения исходного кода хост-приложения и дальнейшее их использование. Эти составляющие в купе позволяют создавать компоненты-расширения и повторно их использовать в разработке программного обеспечения.

Библиографический список:

1. Рихтер Дж. [Richter J.] CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / пер. с англ – СПб. : Питер, 2013. – с. 896.
2. ASP.NET Application Life Cycle Overview for IIS 7.0 [Электронный ресурс]: URL: <https://docs.microsoft.com/en-us/previous-versions/aspnet/bb470252> (v=vs.100) (дата обращения: 23.03.2019).
3. Esposito D. Programming Microsoft ASP.NET 4 – Redmond : Microsoft Press, 2011. – с. 965.
4. Introduction to IIS Architectures [Электронный ресурс]: URL: <https://docs.microsoft.com/en-us/iis/get-started/introduction-to-iis/introduction-to-iis-architecture> (дата обращения: 22.03.2019).

Оригинальность 96%