Конвейер обработки запросов IIS 7.x – 8.x и ASP.NET в интегрированном режиме

Запрос

Ответ

Веб-сервер с Windows и IIS 7.x – 8.x

Клиент 1

Клиент 2

Клиент 3

10. Шаг десятый.
HTTP ответ посылается клиенту.

1. Шаг первый.
Приходит HTTP запрос от клиента на сервер.

Режим ядра Windows (Kernel mode)

HTTP.SYS

Драйвер устройств режима ядра – http.sys, слушает приходящие запросы по протоколам HTTP И HTTPS.

2. Шаг второй.
Проверка запроса. Если HTTP.SYS уже меет информацию необходимую для обработки запроса, то управление передаётся на *шаг 8* т.е. рабочему процессу W3WP.EXE. Если нужного приложения нет, запрос передаётся W3SVC.

7. Шаг седьмой.
W3SVC обновляет конфигурационные данные HTTP.SYS о приложениях.

Режим пользователя Windows (User mode)

Процесс Windows Svchost.exe

World Wide Web Publishing Service (W3SVC)

3. Шаг третий.
W3SVC передаёт запрос WAS.

6. Шаг шестой.
WAS передаёт W3SVC информацию о приложении, которое будет обрабатывать запросы .

Глобальный файл конфигурации IIS
(%windir%\system32\inetsrv)

applicationHost.config

Windows Process Activation Service (WAS)

WAS управляет рабочими процессами. Запускает, перезапускает и останавливает пулы приложений.

4. Шаг четвёртый.
WAS получает данные из глобального файла конфигурации

5. Шаг пятый.
WAS проверяет рабочие процессы пулов приложений. Если нужного нет, то он создаётся

Пулы приложений (Application pools)

W3WP.EXE

W3WP.EXE

……..

W3WP.EXE

W3WP.EXE

Процессы Windows W3WP служат хостами для работающих приложений. Один процесс обыно представляет однин пул приложений, в котором могут работать множество отдельных веб-приложений в различных доменах приложений. Вся основная работа связанная с обработкой запроса выполняется в них. Весь конвейер обработки запросов ASP.NET выполняется именно в таком процессе.

9. Шаг девятый.
W3WP.EXE обработал запрос, передаёт ответ.

8. Шаг восьмой.
HTTP.SYS передаёт запрос на обработку рабочему процессу W3WP.EXE.

Класс **System.Web.Hosting.ApplicationManager** создаёт домены приложений и управляет ими для каждого приложения.

Пул приложений в IIS

Рабочий процесс Windows (W3WP.EXE)

- Домен приложения (App Domain) Веб-приложение 1
- Домен приложения (App Domain) Веб-приложение 2
- Домен приложения (App Domain) Веб-приложение 3
- Домен приложения (App Domain) Веб-приложение 4
- Домен приложения (App Domain) Веб-приложение 5

Объекты типа **HostingEnvironment** хранит данные окружения выполняющегося приложения.

Объекты типа **HttpApplication** представляют выполняющееся приложение. Пул объектов HttpApplication обслуживает запросы. Каждому запросу назначается свой объект. Объекты приложения используются повторно.

Объект класса **HttpRuntime** – точка входа в управляемую среду обработки запросов. Один из важнейших объектов среды выполнения управляемого кода конвейера. Является сердцем выполняемого приложения.

W3WP.EXE

Этапы прохождения запроса по конвейеру обработки запросов IIS. На каждом этапе вызываются соответствующие неуправляемые (нативные, модули машинного кода) и управляемые модули.

Модель событий IIS

- BeginRequest
- Authenticate Request
- AuthorizeRequest
- ResolveRequestCache
- MapRequestHandler
- AcquireRequestState
- PreExecuteRequestHandler
- ExecuteRequestHandler
- ReleaseRequestState
- UpdateRequestCache
- LogRequest
- EndRequest

CLR

Домен приложения (App Domain) Веб-приложение 4

HttpRuntime

HostingEnvironment

HttpApplication
- HttpContext
  - HttpRequest
  - HttpResponse
- HttpModuleCollection
- Application Events

HttpApplication
- HttpContext
  - HttpRequest
  - HttpResponse
- HttpModuleCollection
- Application Events

HttpApplication
- HttpContext
  - HttpRequest
  - HttpResponse
- HttpModuleCollection
- Application Events

События приложения ASP.NET

- BeginRequest
- Authenticate Request
- PostAuthenticateRequest
- AuthorizeRequest
- PostAuthorizeRequest
- ResolveRequestCache
- PostResolveRequestCache
- PostAuthenticateRequest
- MapRequestHandler
- PostMapRequestHandler
- AcquireRequestState
- PostAcquireRequestState
- PreRequestHandlerExecute
- PostRequestHandlerExecute
- ReleaseRequestState
- PostReleaseRequestState
- UpdateRequestCache
- PostUpdateRequestCache
- LogRequest
- PostLogRequest
- EndRequest
- PreSendRequestHeaders
- PreSendRequestContent

HTTP Модули (Нативные)

- AnonymousAuthenticationModule
- HttpLoggingModule
- DirectoryListingModule
- ……..
- ManagedEngine
- WebSocketModule
- CustomErrorModule
- HttpCacheModule

HTTP Модули (Управляемые)

- UrlMappingsModule
- Custom IHttpModule
- FormsAuthentication
- ……..
- UrlRoutingModule
- Session
- Custom IHttpModule
- OwinHttpModule

Управляемый модуль **Microsoft.Owin.Host.SystemWeb.OwinHttpModule** подключает *сервер OWIN* (Компоненты конвейера OWIN) на ранней стадии инициализации модулей в объекте HttpApplication. Модули инициализируются вызовом внутреннего метода InitModulesCommon() класса HttpApplication. Пространство имен **Microsoft.Owin.Host.SystemWeb.IntegratedPipeline** содержит самые важные типы для подключения конвейера OWIN.

Модуль **ManagedEngine** служит посредником (т.е своего рода мостом) для связи с управляемым кодом. Уведомляет управляемый код о событиях и синхронизирует состояние запроса между управляемым и неуправляемым кодом.

Вызов синхронного или асинхронного метода **ProcessRequest** обработчика

Управляемые HTTP-обработчики данных или говоря по другому – запросов (реализующие тип IHttpHandler). Назначаются напрямую или посредством модуля маршрутизации. Такие обработчики как MvcHandler или HttpControllerHandler назначаются в качестве обработчика запроса только модулем маршрутизации.

- DefaultDocumentModule
- StaticFileModule
- CGI
- ISAPI

Нативные обработчики запроса. Некоторые нативные модули IIS сами являются обработчиками. Например StaticFileModule.

- CustomHandler (IHttpHandler)
- System.Web.UI.Page (IHttpHandler)
- System.Web.Http.WebHost.HttpControllerHandler (IHttpHandler)
- System.Web.Mvc.MvcHandler (IHttpHandler)

Управляемые обработчики запросов. Каждый из них инициализирует свой конвейер обработки запроса. В случае с Page – Запросы обрабатываются веб-формами (Web Forms). Если обработчиком назначен MvcHandler, то подключается конвейер ASP.NET MVC. При HttpControllerHandler подключается конвейер ASP.NET Web API.

Сведения приведённые выше не являются официальными и не претендуют на полноту и абсолютную точность. Схема постепенно будет обновляться. Данная версия – 1.0 (01/02/2014).