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

Клиент 1

Клиент 2

Клиент 3

Запрос

Веб-сервер с Windows и IIS 7.x – 8.x

Ответ

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.

Режим пользователя Windows (User mode)

Процесс Windows Svchost.exe

World Wide Web Publishing Service (W3SVC)

3. Шаг третий. W3SVC передаёт запрос WAS.

Windows Process Activation Service (WAS)

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

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.

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

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

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

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

applicationHost.config

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

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

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

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 упраляемую 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

HTTP Модули (Нативные)

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

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.

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 (Управляемые)

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

Microsoft.Owin.Host.SystemWeb.OwinHttpModule

InitializeBlueprint

Модули конвейера OWIN

Модуль SignalR
Microsoft.AspNet.SignalR.Owin.Handlers.HubDispatcherHandler

Собственный модуль
…….

Модуль Web API
System.Web.Http.Owin.HttpMessageHandlerAdapter

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.

- **DefaultDocumentModule**
- **StaticFileModule**
- **CGI**
- **ISAPI**

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

Сведения приведённые выше не являются официальными и не претендуют на полноту и абсолютную точность. Схема постепенно будет обновляться. Данная версия – 1.0 (10/08/2014).