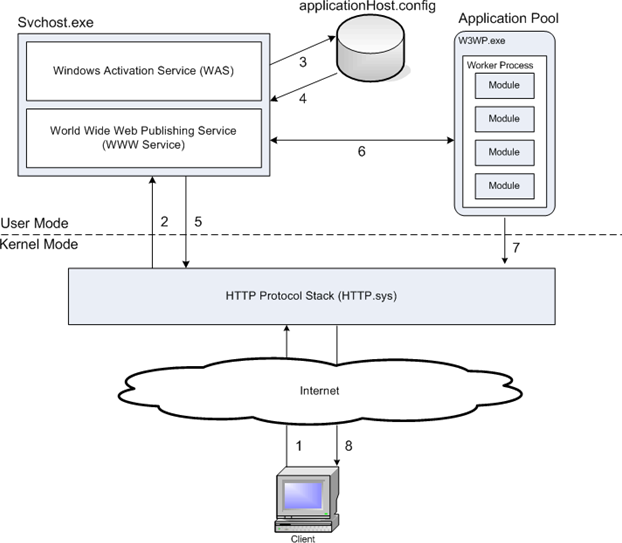
Общие сведения о жизненном цикле приложения ASP.NET для IIS 5.0 и 6.0

<https://msdn.microsoft.com/ru-ru/library/ms178473(v=vs.100).aspx>

Общие сведения о жизненном цикле приложения ASP.NET для служб IIS 7.0

<https://msdn.microsoft.com/ru-ru/library/bb470252(v=vs.100).aspx>

**Процесс обработки запроса в IIS называется «request processing pipeline» - «конвейер обработки запроса».**



* **HTTP.sys** – драйвер работающий в режиме ядра. Его задача прослушивать все входящие HTTP запросы и вносить их в очередь соответствующего пула приложения.

Преимущества:

* Кэширование в режиме ядра. Ответы хранимые в кэше отправляются без переключения в пользовательский режим.
* Очередь на уровне ядра. Запрос находиться в очереди, пока рабочий процесс не начнет его обработку.
* **WWW Service** - это адаптер для HTTP.sys. Этот компонент IIS ответственный за настройку HTTP.sys

В IIS 6.0 WWW Service был ответственным за управление рабочими процессами. Так как он был напрямую связан с HTTP.sys, IIS6 мог обрабатывать только HTTP и HTTPS запросы.

* **WAS** - Управляет пулом приложений и рабочими процессами.

При старте WAS считывает информацию из файла ApplicationHost.config и передает информацию адаптеру для того что бы наладить взаимодействие между WAS и слушателем протокола (например HTTP.sys).

* **Модули** – отдельные функциональные блоки, которые использует сервер для обработки запросов. Модули можно поделить на две категории – управляемые и не управляемые. Управляемые модули могут быть созданы разработчиком с использованием языка C#. Неуправляемые модули являются частью конвейера обработки запроса IIS сервера.
* Global.asax – Global Application Class – файл, который содержит определение класса производного от HttpApplication. Задача класса представлять объект, который обрабатывает входящие запросы. На каждый запрос, который поступает от клиента, создается новый экземпляр класса HttpApplication (или используется уже существующий экземпляр, который не занят обработкой запроса).

*Задание*

Создайте веб приложение и добавьте в него несколько страниц. Реализуйте логику в файле Global.asax, которая будет подсчитывать количество запросов, выполненных к веб приложению за время его работы. Информацию о количестве запросов выводите на одной из страниц.

* **HTTP обработчик** – «конечная точка», которая выполняет обработку запроса к ASP.NET приложению.

IsReusable – Возвращает значение, которое определяет может ли один и тот же экземпляр использоваться для обработки нескольких запросов.

ProcessRequest – Метод выполняется при обращении к обработчику.

* Для регистрации обработчика используется следующий код в файле web.config

<system.webServer>

<handlers>

<add name="FirstHandler" verb="GET" path="MyHandler.aspx" type="FirstHandler.MyFirstHandler"/>

</handlers>

</system.webServer>

## Модули

* **HTTP модуль** – объект, который выполняет действие при каждом обращении к ASP.NET приложению.

Dispose – для удаления ресурсов, которые использовал модуль.

Init – для определения ***обработчиков событий*** жизненного цикла ASP.NET приложения.

* Для регистрации модуля используется следующий код в файле web.config

<system.webServer>

<modules>

<add name="FirstHttpModule" type="FirstHttpModule.FirstHttpModule"/>

</modules>

</system.webServer>

Домашнее Задание

1. Создайте веб приложение, которое при возникновении необработанной ошибки будет отправлять сообщение на определенный email адрес. В сообщении должна храниться краткая информация о ошибке.