Threading

Parameterized and Non-Parameterized

Thread Safety

1. Lock, Monitor (InProc)
2. Mutex, Semaphore, Semaphore Slim (Out Proc)

WebApplication.CreateBuilder().Build() vs Host.CreateDefaultBuilder().Build()

Host: <https://learn.microsoft.com/en-us/dotnet/core/extensions/generic-host>

A *host* is an object that encapsulates an app's resources and lifetime functionality, such as:

* Dependency injection (DI)
* Logging
* Configuration
* App shutdown
* IHostedService implementations

Logging

Stream processing

File Download/ upload

Dependency Injection

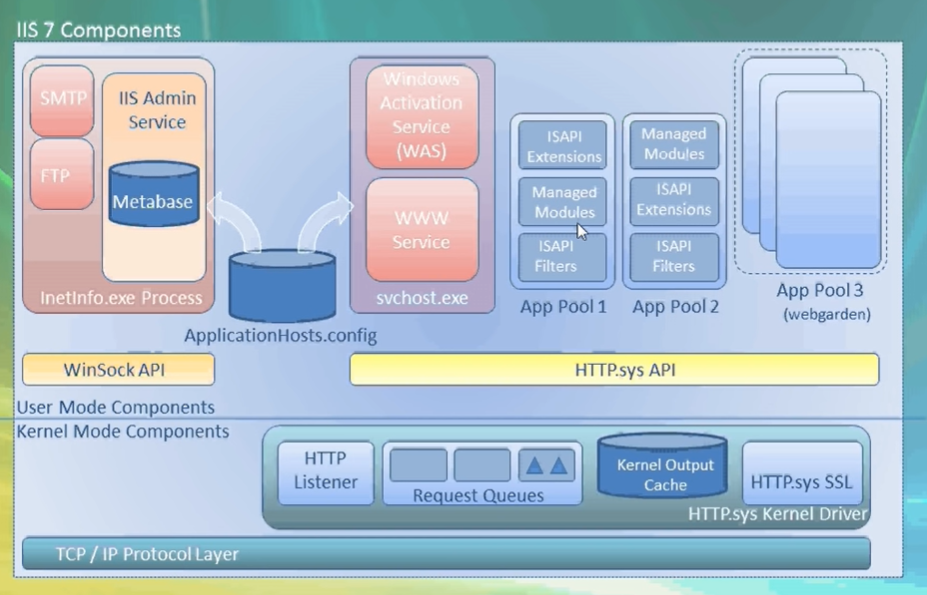
1. AddHostedService
2. AddTransient
3. AddScoped
4. AddSinglton

IIS Component

Diagram

Description automatically generated

1. An HTTP request first goes to HTTP.sys and now, HTTP.SYS is responsible for passing the request to a particular application pool.
2. HTTP.sys contacts to WAS and WAS requests configuration information from the xml file.
3. The configuration information is sent to WWW service receives.
4. The WWW service uses the configuration information to configure HTTP.sys.
5. Configured HTTP.sys contacts to WAS and now, WAS starts a worker process for the application pool to which the request was made.
6. The worker process processes the request and returns a response to HTTP.sys. The request is passed through an ordered series of module in the processing pipeline.



1. Kernel Mode
   1. Http.Sys
2. User Mode
   1. WWW Service
   2. WAS (Windows Activation Service)
   3. ApplicationHost.Config [C:\Windows\System32\inetsrv\config]
   4. App pool
   5. Worker process
   6. ISAPI\_Extensions i.e., Aspnet\_ISAPI.dll
   7. Asp.Net Runtime

Managed Pipeline : Integrated versus Classic

Accounts

1. Local Service
2. Local System
3. Network Service
4. Application Pool Identity
5. IUser