

Software

.Net Framework - supports websites, service, desktop apps & more on windows (fast build)

.Net Core - crossplatform, websites, services & desktop apps
(modular, light weight & flexible)
→ able to use on different computers.

.Net - free, cross platform open source for web, mobile, desktop, games etc

ASP.NET :- framework for web apps, with .Net & C#
web apps, APIs, Ratione, Microservices

developer platform made up of tools, prog. lang.
& lib to build web apps

- fast & scalable
- secure (XSS) - cross site script
- (CSRF) - controls xss, forgery

ASP.NET mvc - decouple UI, ^(view) data(model) & App. logic
(controller)

ASP.NET webapi - HTTP Service with easy way that returns
only data.

App with web API

App.net Web API (dotnet)

- startup.cs - startup & middleware configuration
 - ↳ requests & responses
- myWebApp/ Pages - contains web pages for app
- myWebApp.csproj - libraries referenced to

dotnet new webapp → myWebApp → no others
dotnet dotnet watch run

Pages

Software development framework

- founded for developer to build software applications
- It includes libraries, packages, compiler etc.
- It improves efficiency of working on the important details of app instead of working on the fundamental parts.

.Net

- is a software framework that runs primarily on Microsoft windows
- has a large library that supports interoperability (ability to exchange)
- access several languages., platform to build.

Entity

.Net → the ecosystem - collection of diff software projects

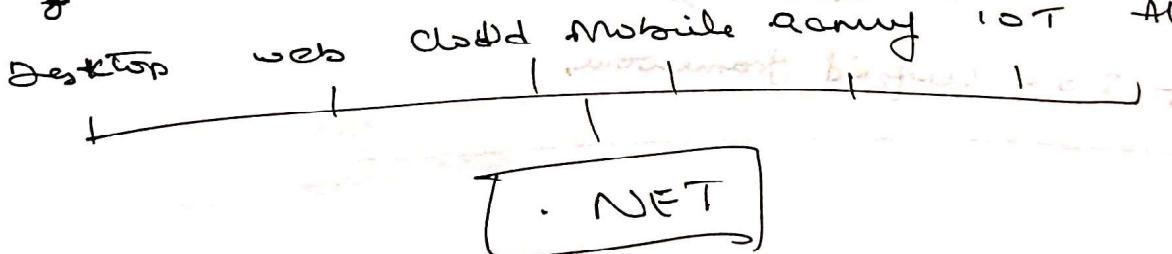
CLR → JVM

C#, F#, VB → language

Nuget → npm etc

dotnet cli → entry pt, SDK, Java etc

g



Net framework

Net core

Mono/Xamarin

.NET

Ecosystem — Languages - C#, F#, VB
 — C++
Libraries - Base Class Lib, etc

- .dll - dynamic link library
- .exe - executable

.Net 5.0 — follow up .Net Core 3.1 & added functionality,
— & removed core part

TPM - Target framework Manager

C# — 9.0

dotnet — package manager

.Net framework → windows apps.

→ WPF, WinForms,

.Net Core → crossplatform software framework
→ Modular,

.Net 5.0 — unified framework

~~Model View View Model~~

~~Startup.cs~~

Pages Pages - & for simple pages to receive & display data.

Startup.cs

Startup

↳ configuration

↳ configure (app, env)

- db development
- https redirect
- use static files
- use Poetry
- use Authorization
- use endpoints

C# HTML

@page

filename.cshtml

@page

@use newspace-name

@model modelname

↳

↳

model - .

↳

filename.cshtml.cs

namespace newspace-name {

 public class model-name : pageModel {

}

content - html markup & c# code with Razor Syntax
content.cs - C# code for page events

wwwroot - contains assets like HTML, JS, & CSS files

appsettings.json - config data & connection strings

program.cs - entry point for the system

Startup.cs - code for configure app behaviour.

→ app settings.json

→ Program.cs

→ Startup.cs

appsettings.json

→ logging

↳ log level

↳ default - ~~microsoft~~ information

↳ microsoft - warning

↳ .

- AllowedHosts

- ConnectionStrings - l

↳ → context: —

Program.cs

(ASP.NET Core, .Hosted, Extensions)

↳ CreateHostBuilder (args)

→ IHostBuilder

+ Hostflow ↳ ~~CreateDefault~~

Host.CreateDefaultBuilder (args)

ConfigureWebHostDefaults (webBuilder)

{

webBuilder.UseStartup < startup > () ;

} ;

}

Startup.cs

ε ASP.NET Core, .Build, .Hosting, .Extensions, .Configure
Dependency Injection, Hosting, Entity Framework Core,

→ Configure

→ Configure Services

→ AddDbContext

→ Configure (AppBuilder APP, IWebHostConfig)

Program

- `Host.CreateDefaultBuilder(args)`.
 • Configure WebHostDefaults (`webBuilder =>`)
- CreateDefaultBuilder
 - `WebHostConfigurationProvider` - source to app config.
[app-config] - custom proper & behaviour of app.
 - ~~appsettings.json~~ → configuration value
 - `Context Root` - sets current directory
 - ↳ env. variables & cli arguments
 - ↳ reads app config from `appsettings.json`.
 - ↳ logging provider
 - ↳ console
 - ↳ file
 - ↳ event source
 - ↳ event log

Configure WebHostDefaults

- ↳ host config
- ↳ kerstrel
 - ↳ adds middleware
- ↳ IIS Integration

Framework provided Services

- Host App's lifetime → logging
- Host lifetime
- Host Environment → appname, env vars, context root path

Host Configuration

Configure host configuration on HostBuilder

ASP.NETCORE - ENVIRONMENT \Rightarrow environment

Config host config file (copy Host \Rightarrow)

- Set Base Path
- add json file (HostSettings)
- Add Env variable
 - add command file

App Config

- App Name - key, type, default (entry point), envt var file
- Content root - key, type, set val, default - app assembly
- Env name - key, type, default - production, dev, test var

Web Apps

Capture startup events
Use Startup.cs (Startup.cs)

URLS

use URLs (

webroot

use Web Root ("public")

(with API)

→ use Development Page

→ use static

→ add Swagger - Author, license & claims

→ use https Redirect

→ use Routing

→ use Authorization

→ use Endpoints

Dot file Configuration

① Connected Services

- lets app. developers connect their apps to Service providers that runs in the cloud

② Dependencies

- ① Analyzers - find potential issues & fix them (Suggests)

② Microsoft. ASPNetCore.

→ Analyzers - NuGet package

- analyze the code generated in the code editor.

③ → Component Analyzers - Roslyn analyzers

- for quality, style, maintainability, design & other issues

→ MVC Analyzers.

- powerful, pattern based dynamic websites

④ Microsoft. RoslynAnalyzers. CodeAnalysis &

- inspect C# code

- security, performance, design & other issues

→ platform compatibility checks

⑤ Frameworks

- the dependency of code.

Ex) System. System. (Enumerable etc)

② Properties..

- provided with project configuration

launchSettings.json

- used only on local development machine
- not deployed
- contains profile settings

Default : IIS Express - webServer

↳ light weight, works well with PaaS & Azure

Project

③ JSON in ~~not~~ ASP.NET

- Global.json - applies to all projects in solution
 - (project list & SDK version)
 - src - actual app & test for testing
- appsettings.json - define app related settings like connection string & logging settings or key for web.config
- project.json - project setting & server side dependencies
- launchSettings.json - project specific setting associated with each profile. VS configures launch the application (IIS, Kestrel, weblistener)
- bower.json - maintain & manage package of HTML, CSS, JS
- package.json -

- (E) wwwroot - web root folder
- serves static files
 - ~~AppHost~~.`ASPNetCore.StaticFiles`
- (F) Pages - serve as path for URLs
- (\hookrightarrow) underscore instead of app name
 - they are not meant to be shown directly as a path.
 - not intended to be browsed.
 - @page prefix
- (G) Program
- \rightarrow Initialize the host
 - create `HostBuilder`(args).Build().Run();
 - \rightarrow Host.CreateDefaultBuilder() \Rightarrow configure WebHost DefaultHostBuilder \Rightarrow `WebHostBuilder.UseStartup<Startup>()`;
 - \Rightarrow `Y;`
- HostBuilder - properties
- control location for sharing static components like host sharey procs
 - \rightarrow `IDictionary<Object, Object> Cfgs;`

→ Create Default Builder.

- ① Set content root path
- ② Load IConfiguration from IHOSTED¹ current val
- ③ Load from args (command line)
- ④ Load app config from appsettings.json & app. env. json
- ⑤ Load secrets from .env - use secrets
- ⑥ Iam - env var
- ⑦ Load app - app from command line args
- ⑧ Logback config (Console, debug, eventSource, EventLog)
- ⑨ Validate or DI or when in dev mode

→ Configuration

(For [Song])

Configuration

represents a configuration file that is applicable to a particular computer, app or resource.

→ ConfigureWebHostDefaults

Configures with HostBuilder with defaults of HostBuilder

of HostBuilder

ConfigureApp()

Start()

Buid()

→ Extension methods - without creating derived type

Add methods

host HostConfig (ASPNETCORE), Kestrel,

HostFiltering middleware, ForwardedHeaders (middle)

encoder (IIS Integration)

~~IHost webbuilder~~ IWebHostBuilder - Build configura~~tion~~, server etc

~~webbuilder~~.Configure -> Builds for IWebHostBuilder

IWebHostBuilder IWebHost - represents a configured web host

↳ server features - collects of feature by server

↳ service - IService provider

useStartup - IWebHostBuilder & string for assembly name of startup)

- Specify the assembly containing the startup type

to be used by WebHost

IHostBuilder

~~Startup~~.Build() - actions to initialize the IHost - action

IHost

IHost - IDisposable (remove unmanaged resources)

↳ Service - program config & service

Run() - runs an application & block the calling thread until the IHost shutdown

② Startup.

configuration < configuration (host config, app config)

log ship provider

scope & dependency resolver

IServiceCollection - Count, IsReadOnly, Item

MVC Service Collection Extensions

Builder has services

→ Configuration

- IApplicationBuilder - Defined a class that provides the mechanism to configure an application pipeline.
- useHttpRedirection - adds middleware for redirecting HTTP to HTTPS
- useStaticFiles - Enables static file serving for current path request path
- useRouting - adds a routing endpoint middleware.
 - ↳ match endpoint
- useAuthorization - adds authorization middleware to specified app's builder & enables Authorization capabilities
 - btw → app. use routing() & ~~app. use authorization()~~
 - ↳ app. use endpoints()

- useEndpoints - Endpoint middleware - to specify app's endpoint instances built from configured EndpointRouteBuilder. The endpoint middleware executes the request

- End PointBuilder - Data sources - endpoint Data Sources copy in Endpoints

- Service Provider - get the set of (Service providers and endpoint servers) for routes

- Endpoint - Name, MetaData, RequestDelegate
 - ↳ Information / display name
 - ↳ Collector / metadate
 - ↳ Delay used & access

Endpoint Metadata Collector - list of metadata types of endpoints
types collected in order

- fields empty..

Endpoint

builder - to add middleware &

action \leftrightarrow - config to provide EndpointMetadata

Endpoint

map get method to define the endpoint.

→ selected by matching the url

→ executed by running the delegate