**Program.cs**

Site aspnet core = serveur web (Kestrel) embarqué dans une app console

Initialisation de Kestrel, en appelant une classe Startup

**Startup.cs**

Initialisation Kestrel

Configure()définit comment l’application va répondre aux requêtes - middlewares

ConfigureServices() : sert à configurer les services

Services.AddMvc()

**Middlewares**

Aspnet core est basé sur les middlewares pour traiter la requête – liste de composants qui vont traiter la requête un par un, pour envoyer un résultat

* App.UseExceptionHandler(« /Home/Error »)
* App.UseStaticFiles()
* App.UseIdentity
* App.UseMvc(routes=>{  
   routes.MapRoute(« Default », «{controller}/{action}/{id ?} ») ;  
  })

Middleware custom

app.Use(async (context, next) =>   
en milieu de pipeline

app.Run(async (context) =>  
en fin de pipeline (pas de next())

app.UseMiddleware<AuthMiddleware>();  
Classe séparée

**MVC**

Similaires à la version actuelle

Tag helpers

**Déploiement**

Derrière un reverse-proxy, pour raisons de sécurité

HTTPS/ Compression GZIP

Filtrage IP, url rewrite plus efficace