**Kaynak :** [**https://www.bayramucuncu.com/asp-net-core-gelistiriciler-icin-yol-haritasi/**](https://www.bayramucuncu.com/asp-net-core-gelistiriciler-icin-yol-haritasi/)

[**ASP.NET Core geliştiriciler için yol haritası**](https://www.bayramucuncu.com/asp-net-core-gelistiriciler-icin-yol-haritasi/)

2019 yılında, bir grup geliştirici, [GitHub](https://github.com/)‘da ASP.NET Core geliştiriciler için bir yol haritası ortaya koymuşlar. Kaynağa [buradan](https://github.com/MoienTajik/AspNetCore-Developer-Roadmap) ulaşabilirsiniz.

Bu yazıda, her geliştirici için faydalı olabileceğini düşündüğüm bu yol haritasını sizlerle paylaşmak istedim.

**Yol Haritası Boyunca Bilinmesi Gerekenler**

1. **Öncelikle bilinmesi gerekenler.**
   * C#
   * Entity Framework
   * ASP.NET Core
   * SQL Fundamentals
2. **Bilinmesi gereken genel geliştirme becerileri.**
   * GIT sistemi öğrenilmeli, GitHub üzerinde yeni bir repository oluşturulabilmelidir, kod diğer geliştiricilerle paylaşılmalıdır.
   * HTTP(S) protokolü bilinmeli, http request metodları (GET, POST, PUT, PATCH, DELETE, OPTIONS) bilinmelidir.
   * Google kullanmaktan ve araştırmaktan korkmayın.
   * [Dotnet CLI](https://docs.microsoft.com/dotnet/core/tools/?tabs=netcore2x) öğrenmelisiniz.
   * Veri yapıları ve algoritmalar hakkında bir kaç kitap okumalısınız.
3. **Dependency Injection (DI – Bağımlılıkların dışarıdan alınması)**
   * **DI Containers**
     + [Microsoft.Extensions.DependencyInjection](https://docs.microsoft.com/aspnet/core/fundamentals/dependency-injection)
     + [AutoFac](https://autofaccn.readthedocs.io/en/latest/integration/aspnetcore.html)
     + [Ninject](http://www.ninject.org/)
     + [StructureMap](https://github.com/structuremap/structuremap)
     + [Castle Windsor](https://github.com/castleproject/Windsor)
   * [Life Cycles](https://docs.microsoft.com/aspnet/core/fundamentals/dependency-injection#service-lifetimes)
   * [Scrutor](https://github.com/khellang/Scrutor)
4. **Veritabanları**
   * **İlişkisel veritabanları**
     + [SQL Server](https://www.microsoft.com/sql-server/sql-server-2017)
     + [PostgreSQL](https://www.postgresql.org/)
     + [MariaDB](https://mariadb.org/)
     + [MySQL](https://www.mysql.com/)
   * **Bulut Veritabanları**
     + [CosmosDB](https://docs.microsoft.com/azure/cosmos-db)
     + [DynamoDB](https://aws.amazon.com/dynamodb/)
   * **Arama Motorları**
     + [ElasticSearch](https://www.elastic.co/)
     + [Solr](http://lucene.apache.org/solr/)
     + [Sphinx](http://sphinxsearch.com/)
   * **NoSQL Veri tabanları (Object database)**
     + [**MongoDB**](https://docs.microsoft.com/aspnet/core/tutorials/first-mongo-app)
     + [Redis](https://redis.io/)
     + [Apache Cassandra](http://cassandra.apache.org/)
     + [LiteDB](https://github.com/mbdavid/LiteDB)
     + [RavenDB](https://github.com/ravendb/ravendb)
     + [CouchDB](http://couchdb.apache.org/)
5. **Cache Mekanizmaları**
   * **Entity Framework 2nd Level Cache**
     + [EFSecondLevelCache.Core](https://github.com/VahidN/EFSecondLevelCache.Core)
     + [EntityFrameworkCore.Cacheable](https://github.com/SteffenMangold/EntityFrameworkCore.Cacheable)
   * [**Distributed Cache**](https://docs.microsoft.com/aspnet/core/performance/caching/distributed)
     + [Redis](https://redis.io/)
     + [Memcached](https://memcached.org/)
   * [**Memory Cache**](https://docs.microsoft.com/aspnet/core/performance/caching/memory)
6. **Log Mekanizmaları**
   * **Log kaydı yapabilmek için gerekli framework’ler**
     + [Serilog](https://github.com/serilog/serilog)
     + [NLog](https://github.com/NLog/NLog)
     + [Elmah](https://elmah.github.io/)
     + [log4net](https://github.com/huorswords/Microsoft.Extensions.Logging.Log4Net.AspNetCore)
   * **Log kayıtlarını yönetebilmek için gerekli sistemler**
     + [Sentry.io](http://sentry.io/)
     + [Loggly.com](https://loggly.com/)
     + [Elmah.io](http://elmah.io/)
7. **Web site geliştirme şablonları (Template Engines)**
   * [Razor](https://docs.microsoft.com/aspnet/core/mvc/views/razor)
   * [DotLiquid](https://github.com/dotliquid/dotliquid)
   * [Scriban](https://github.com/lunet-io/scriban)
   * [Fluid](https://github.com/sebastienros/fluid)
8. **Gerçek zamanlı iletişim araçları (Realtime Communication)**
   * [SignalR](https://docs.microsoft.com/aspnet/core/signalr)
9. **Nesne eşleştirme araçları (Object Mapping)**
   * [AutoMapper](https://github.com/AutoMapper/AutoMapper)
   * [Mapster](https://github.com/MapsterMapper/Mapster)
   * [AgileMapper](https://github.com/agileobjects/AgileMapper)
   * [ExpressMapper](http://expressmapper.org/)
10. **API istemcileri (Clients)**
    * **REST**
      + [OData](https://blogs.msdn.microsoft.com/odatateam/2018/07/03/asp-net-core-odata-now-available/)
      + [Sieve](https://github.com/Biarity/Sieve)
    * **GraphQL**
      + [GraphQL-dotnet](https://github.com/graphql-dotnet/graphql-dotnet)
11. **Bilinmesi faydalı olabilecek bilgiler**
    * [MediatR](https://github.com/jbogard/MediatR)
    * [Fluent Validation](https://github.com/JeremySkinner/FluentValidation)
    * [Swashbuckle](https://github.com/domaindrivendev/Swashbuckle.AspNetCore)
    * [Benchmark.NET](https://github.com/dotnet/BenchmarkDotNet)
    * [Polly](https://github.com/App-vNext/Polly)
    * [NodaTime](https://github.com/nodatime/nodatime)
    * [GenFu](https://github.com/MisterJames/GenFu)
12. **Testler**
    * **Birim testler (Unit testing)**
      + Test araçları
        - [MSTest](https://docs.microsoft.com/dotnet/core/testing/unit-testing-with-mstest)
        - [NUnit](https://docs.microsoft.com/dotnet/core/testing/unit-testing-with-nunit)
        - [xUnit](https://docs.microsoft.com/dotnet/core/testing/unit-testing-with-dotnet-test)
      + Mock araçları
        - [Moq](https://github.com/moq/moq4)
        - [NSubstitute](https://github.com/nsubstitute/NSubstitute)
        - [FakeItEasy](https://github.com/FakeItEasy/FakeItEasy)
      + Assertion araçları
        - [FluentAssertion](https://github.com/fluentassertions/fluentassertions)
        - [Shouldly](https://github.com/shouldly/shouldly)
    * **Davranış testleri (Behavior testing)**
      + [BDDfy](https://github.com/TestStack/TestStack.BDDfy)
      + [SpecFlow](https://github.com/techtalk/SpecFlow/tree/DotNetCore)
      + [LightBDD](https://github.com/LightBDD/LightBDD)
    * **Entegrasyon veya tümleştirme testleri (Integration testing)**
      + [WebApplicationFactory](https://docs.microsoft.com/aspnet/core/test/integration-tests)
      + [TestServer](https://koukia.ca/integration-testing-in-asp-net-core-2-0-51d14ede3968)
    * **Uçtan uca test (End to end – E2E testing)**
      + [Selenium](https://www.automatetheplanet.com/webdriver-dotnetcore2/)
      + [Puppeteer-Sharp](https://github.com/kblok/puppeteer-sharp)
13. **Görev zamanlayıcılar (Task scheduling)**
    * [HangFire](https://github.com/HangfireIO/Hangfire)
    * [Coravel](https://github.com/jamesmh/coravel)
    * [Fluent Scheduler](https://github.com/fluentscheduler/FluentScheduler)
14. **Mikro servisler (MicroServices)**
    * **Message-Broker**
      + [RabbitMQ](https://www.rabbitmq.com/tutorials/tutorial-one-dotnet.html)
      + [Apache Kafka](https://github.com/confluentinc/confluent-kafka-dotnet)
      + [ActiveMQ](https://github.com/apache/activemq)
      + [Azure Service Bus](https://docs.microsoft.com/azure/service-bus-messaging/service-bus-messaging-overview)
    * **Message-Bus**
      + [MassTransit](https://github.com/MassTransit/MassTransit)
      + [NServiceBus](https://github.com/Particular/NServiceBus)
      + [CAP](https://github.com/dotnetcore/CAP)
15. **SOLID prensipleri**
    * [Single Responsibility Principle (SRP)](https://www.dotnetcurry.com/software-gardening/1148/solid-single-responsibility-principle)
    * [Open-Closed Principle (OCP)](https://www.dotnetcurry.com/software-gardening/1176/solid-open-closed-principle)
    * [Liskov Substitution Principle (LSP)](https://www.dotnetcurry.com/software-gardening/1235/liskov-substitution-principle-lsp-solid-patterns)
    * [Interface Segregation Principle (ISP)](https://www.dotnetcurry.com/software-gardening/1257/interface-segregation-principle-isp-solid-principle)
    * [Dependency Inversion Principle (DIP)](https://www.dotnetcurry.com/software-gardening/1284/dependency-injection-solid-principles)
16. **Tasarım Kalıpları (Design-Patterns)**
    * [CQRS](https://docs.microsoft.com/azure/architecture/patterns/cqrs)
    * [Decorator](https://www.dofactory.com/net/decorator-design-pattern)
    * [Strategy](https://www.dofactory.com/net/strategy-design-pattern)
    * [Observer](https://www.dofactory.com/net/observer-design-pattern)
    * [Builder](https://www.dofactory.com/net/builder-design-pattern)
    * [Singleton](https://www.dofactory.com/net/singleton-design-pattern)
    * [Facade](https://www.dofactory.com/net/facade-design-pattern)
    * [Mediator](https://www.dofactory.com/net/mediator-design-pattern)

**Yol Haritası Görseli (**[**https://github.com/MoienTajik/AspNetCore-Developer-Roadmap**](https://github.com/MoienTajik/AspNetCore-Developer-Roadmap)**)**







