






Julio's .NET Backend Developer Roadmap

.NET Essentials

- What is .NET? [🔗](#)
- .NET CLR basics [🔗](#)
- C# Language [🔗](#)
- Essential .NET libraries and types [🔗](#)
- Basic .NET CLI commands [🔗](#)
- NuGet [🔗](#)

Internet Basics

- How does the Internet work? 
- What is HTTP? 
- What is DNS? 
- What is hosting? 
- How do browsers work? 

Web API

- What is a REST API? [🔗](#)
- Controllers [🔗](#)
- Minimal APIs [🔗](#)
- Postman [🔗](#)
- Swagger / Open API [🔗](#)
- JSON [🔗](#)
- DTO [🔗](#)
- Validation [🔗](#)

ASP.NET Core Fundamentals

- What is ASP.NET Core? [🔗](#)
- App startup [🔗](#)
- Dependency injection [🔗](#)
- Middleware [🔗](#)
- Host [🔗](#)
- Web server [🔗](#)
- Configuration [🔗](#)
- Routing [🔗](#)
- Error handling [🔗](#)

Databases

```
graph LR; Relational --> PostgreSQL; Relational --> AzureSQL[Azure SQL Database]; Relational --> MySQL; NoSQL --> MongoDB; NoSQL --> AzureCosmos[Azure Cosmos DB];
```

Security

- Frameworks
 - ASP.NET Core Authentication [↗](#)
 - ASP.NET Core Authorization [↗](#)
- Standards
 - JSON Web Tokens [↗](#)
 - OAuth 2.0 [↗](#)
 - OpenID Connect [↗](#)
- OpenID providers
 - Keycloak [↗](#)
 - Microsoft Entra ID [↗](#)
- Secrets management
 - .NET Secret Manager [↗](#)
 - Azure Key Vault [↗](#)
- Transport
 - HTTPS [↗](#)
 - TLS [↗](#)
- CORS [↗](#)

Deployment

```
graph LR; subgraph Containers; Docker[Docker]; DockerCompose[Docker Compose]; DockerHub[Docker Hub]; AzureContainerRegistry[Azure Container Registry]; end; subgraph NETAspire[.NET Aspire]; DevTimeOrchestration[Dev-time Orchestration]; Integrations[Integrations]; Dashboard[Dashboard]; end; subgraph Orchestration; Kubernetes[Kubernetes]; AKS[Azure Kubernetes Service \(AKS\)]; Helm[Helm]; end; subgraph APIGateway[API Gateway]; YARP[YARP]; NGINX[NGINX]; EmissaryIngress[Emissary-ingress]; end; subgraph Infrastructure; Bicep[Bicep]; Terraform[Terraform]; AzureCLI[Azure CLI]; end;
```

The diagram illustrates the following categories and their associated services:

- Containers**: Docker, Docker Compose, Docker Hub, Azure Container Registry
- .NET Aspire**: Dev-time Orchestration, Integrations, Dashboard
- Orchestration**: Kubernetes, Azure Kubernetes Service (AKS), Helm
- API Gateway**: YARP, NGINX, Emissary-ingress
- Infrastructure**: Bicep, Terraform, Azure CLI

Testing

```

graph LR
    subgraph Unit_testing [Unit testing]
        xUnit
        Moq
        TDD
    end
    WebApplicationFactory --- Integration_testing
  
```

Diagram illustrating the relationship between testing frameworks and testing types:

- Unit testing** (indicated by a bracket) includes:
 - xUnit
 - Moq
 - TDD
- Integration testing** (indicated by a line) includes:
 - WebApplicationFactory

Task Scheduling

Background service ☐

Observability


```
graph LR; Seq[Seq] --- CLT[Centralized logging tools]; ES[Elastic Stack] --- CLT; ILog[ILogger] --- Log[Logging]; Serilog[Serilog] --- Log; CLT --- Log; OT[OpenTelemetry] --- DT[Distributed Tracing]; Jaeger[Jaeger] --- DT; Prometheus[Prometheus] --- Mon[Monitoring]; Grafana[Grafana] --- Mon;
```

The diagram illustrates the Elastic Stack components grouped into three main categories:

- Centralized logging tools:** Includes Seq and Elastic Stack.
- Distributed Tracing:** Includes OpenTelemetry and Jaeger.
- Monitoring:** Includes Prometheus and Grafana.

Each tool name is accompanied by an external link icon (a square with a diagonal line).

Version control

- Git 
- GitHub

Communication

```
graph LR; S[Synchronous] --- H[HttpClient]; S --- G[gRPC]; A[Asynchronous] --- ASB[Azure Service Bus]; A --- R[RabbitMQ]; A --- AK[Apache Kafka]; RT[Real-time] --- WS[WebSockets]; RT --- SR[SignalR];
```

The diagram illustrates the classification of message brokers into three categories: Synchronous, Asynchronous, and Real-time. Each category is represented by a blue bracket on the left, with its name in white text. To the right of each category, the specific brokers are listed in white text, each followed by a small white icon of a document with a checkmark. The Synchronous category includes HttpClient and gRPC. The Asynchronous category includes Azure Service Bus, RabbitMQ, and Apache Kafka. The Real-time category includes WebSockets and SignalR.

- Synchronous
 - HttpClient
 - gRPC
- Asynchronous
 - Azure Service Bus
 - RabbitMQ
 - Apache Kafka
- Real-time
 - WebSockets
 - SignalR

Caching

Redis 

Design Principles

- SOLID
 - Single Responsibility Principle (SRP)
 - Open-Closed Principle (OCP)
 - Liskov Substitution Principle (LSP)
 - Interface Segregation Principle (ISP)
 - Dependency Inversion Principle (DIP)
- DRY (Don't Repeat Yourself)
- KISS (Keep It Simple, Stupid!)
- YAGNI (You Ain't Gonna Need It)
- DDD (Domain Driven Design)

Architectural Patterns

- Vertical Slice Architecture [↗](#)
- Microservices [↗](#)
- Monolithic Apps [↗](#)

Design Patterns

- Singleton [↗](#)
- Repository [↗](#)
- Retries with exponential backoff [↗](#)
- Circuit breaker [↗](#)
- Publisher-subscriber [↗](#)
- Saga [↗](#)
- Mediator [↗](#)
- Competing consumers [↗](#)
- Event sourcing [↗](#)

Essential Libraries

- FluentValidation [↗](#)
- Bogus [↗](#)
- Polly [↗](#)
- Refit [↗](#)

Keep learning at
<https://juliocasal.com>