

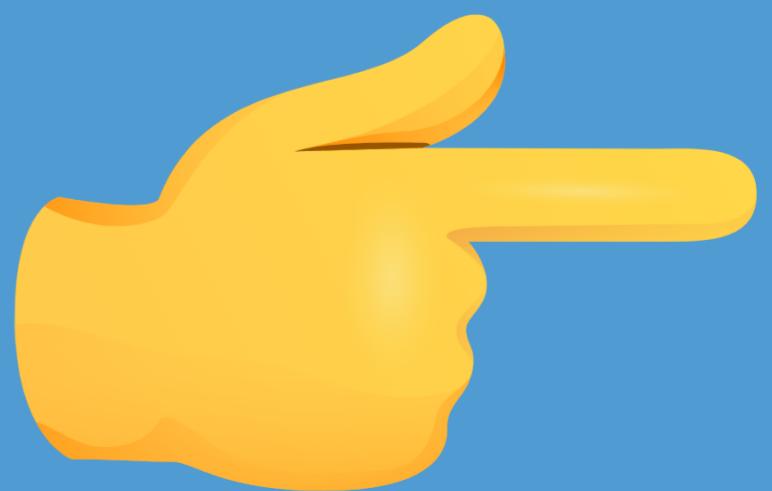
## For Junior, Middle, Senior, Lead developers

This roadmap is designed to help developers progress from foundational knowledge to advanced expertise in ASP.NET Core for .NET 9.

It is structured to guide you through essential concepts, practical applications, and advanced techniques, useful for developers at all levels - from junior to senior and lead.



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)

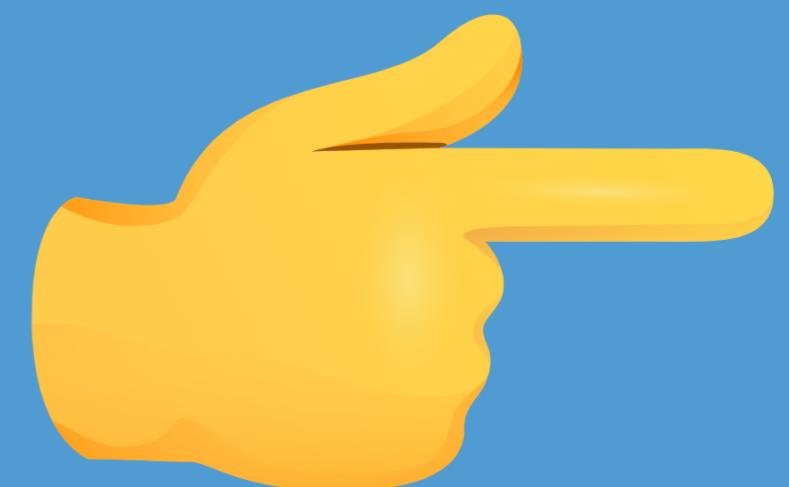


# Phase 1: Getting Started with ASP.NET Core

1. Create a New ASP.NET Core Project
2. Understand Core Concepts
  - Middleware: Learn how HTTP requests are handled via middleware components.
  - Routing: Understand how URL paths are mapped to controllers and actions.
  - Dependency Injection (DI)
    - AddSingleton, AddScoped, AddTransient



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)

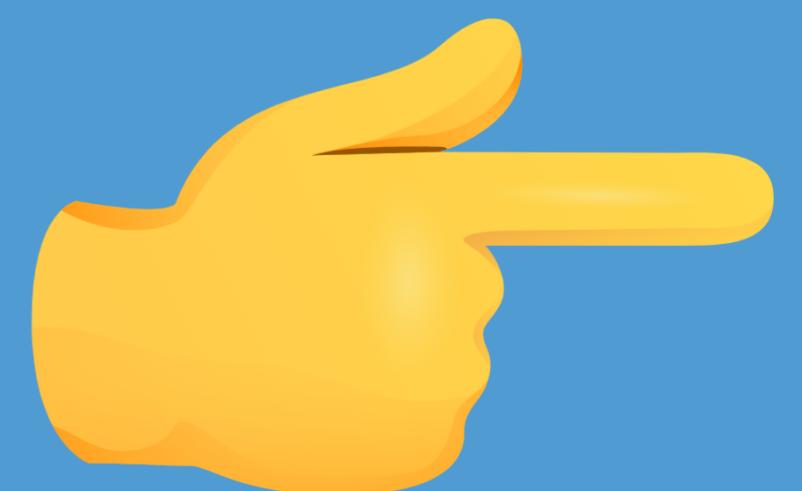


### Phase 2: REST Essentials

1. What is REST ?
2. Resource Naming and Routing
3. HTTP Methods: GET, POST, PUT, DELETE
4. HTTP Response Codes
5. Idempotency of REST Requests



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



### Phase 3: Building APIs

#### 1. Web API Controllers

- **Create API controllers**
- **Define GET, POST, PUT, DELETE actions**
- **Understand content negotiation and returning data in JSON format.**

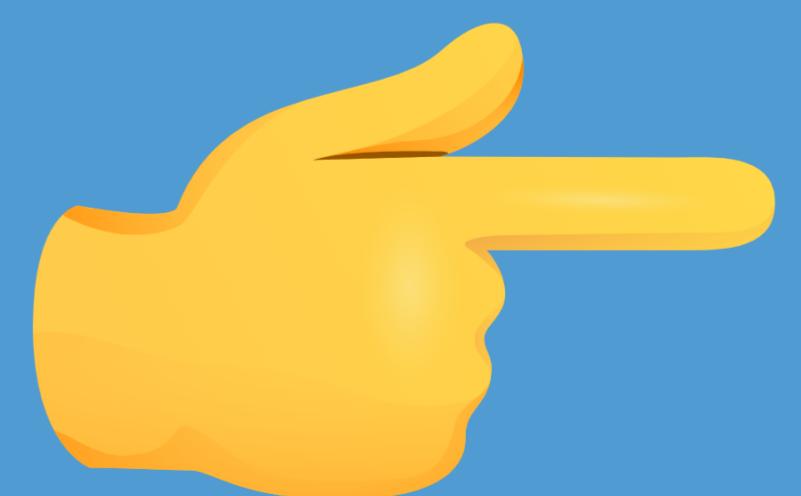
#### 2. Minimal APIs

- **Explore minimal APIs for building modern HTTP endpoints**

#### 3. Understand when to choose minimal APIs over traditional controllers.



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



### Phase 4: Middlewares and Action Filters

#### 1. Custom Middleware

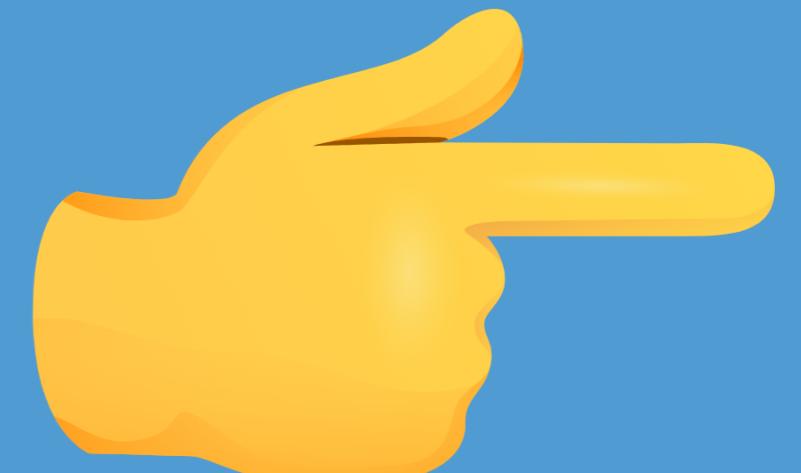
- Learn how HTTP requests are processed through a sequence of middleware components.
- Understand the importance of middleware ordering
- Create custom middlewares using "Use" delegate, convention and factory based approaches

#### 2. Action Filters

- Understand how filters can run code before or after action methods
- Learn about different filter types: Authorization, Resource, Action, Exception, and Result filters



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



### Phase 5: Cross-Cutting Concerns

#### 1. Configuration with Options Pattern

- Understand the `IOptions`,  
`IOptionsSnapshot`, and `IOptionsMonitor`  
interfaces

#### 2. Logging

- Utilize the ASP.NET Core logging  
framework, explore logging levels and  
categories
- Implement structured logging with  
logging libraries like Serilog or NLog

#### 3. Validation

- Learn how to use Data Annotations for  
model validation
- Implement validation with  
FluentValidation library



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



# Phase 6: Errors and Exception Handling

## 1. Errors

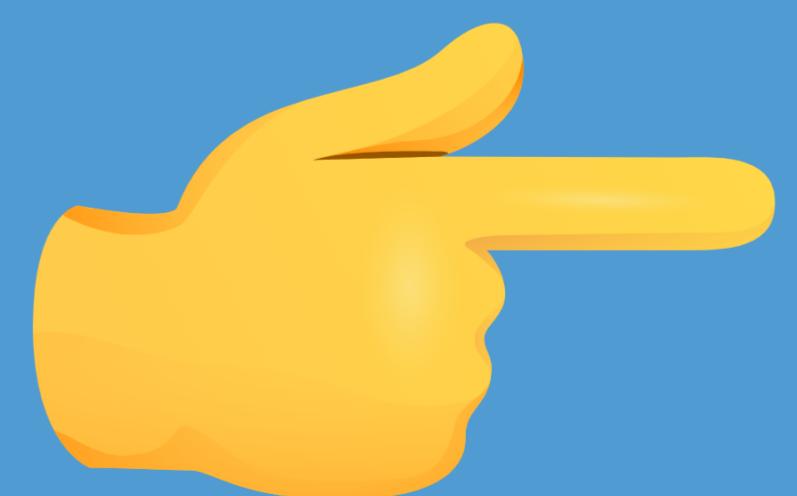
- Learn what are 4xx and 5xx codes used for
- Learn how to return errors in a standardized way with `ProblemDetails`

## 2. Exception handling

- Learn how to configure a middleware to handle exceptions
- Implement `IExceptionHandler` to handle exceptions



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)

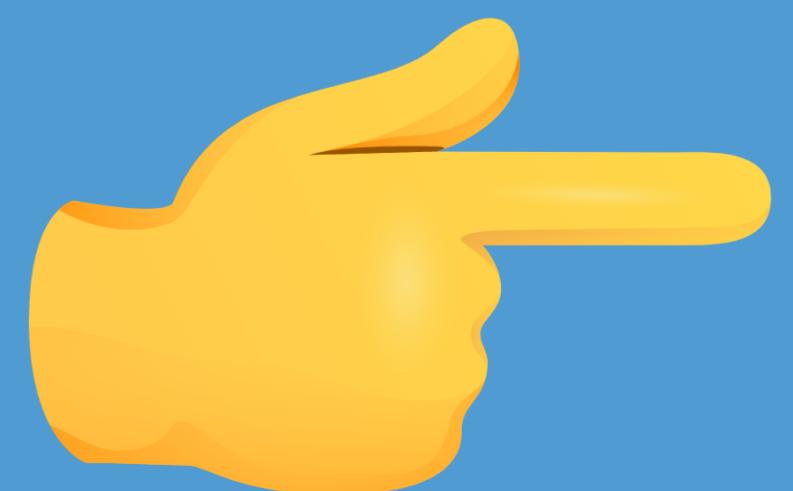


### Phase 7: Data Access with Entity Framework Core

1. Integrate EF Core with ASP.NET Core
  - Set up `DbContext` and connect to a database
  - Implement Create, Read, Update, Delete functionalities
2. Configure EF Core in DI
  - Learn how and when to use `DbContext`, `DbContextFactory` and their Pooled variations
3. Implement UnitOfWork Pattern



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



# Phase 8: Authentication and Authorization

## 1. Identity and Security

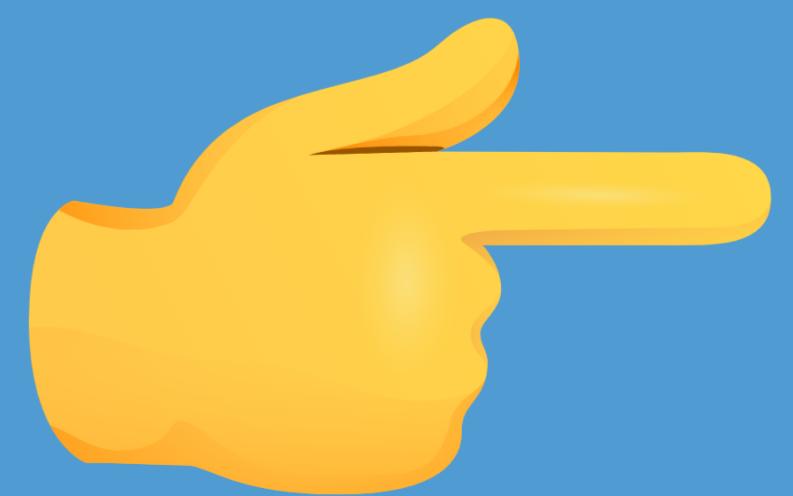
- Implement ASP.NET Core Identity for user management.
- Understand authentication mechanisms like cookies and JWT
- Implement token-based authentication with JWT
- Integrate third-party authentication providers

## 2. Authorization Policies

- Define and apply authorization policies.
- Use roles and claims-based authorization.



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



### Phase 9: API Maintenance

#### 1. API Versioning

- Implement versioning using URL, query string parameter and header value

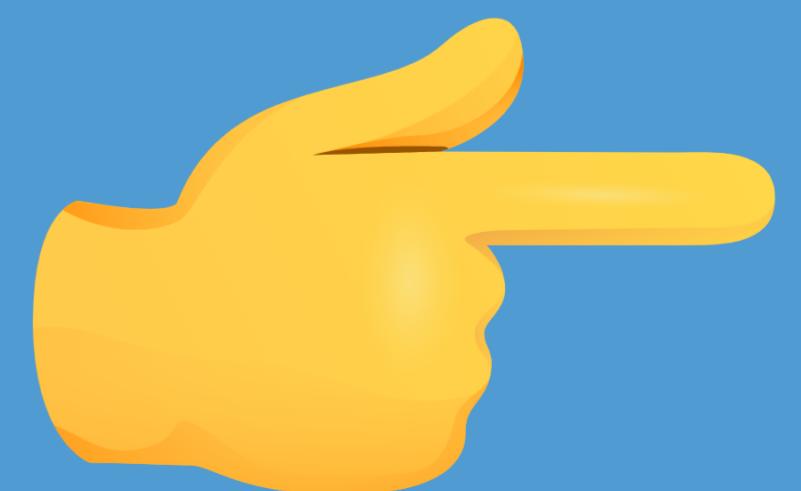
#### 2. Swagger

- Configure Swagger UI to visualize and test your APIs
- Configure auth in Swagger

#### 3. HATEOAS



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



# Phase 10: Performance Optimization

## 1. Caching

- Use Hybrid cache (.NET 9)
- Use Redis cache
- Use Output cache

## 2. Compression

- Use compression to decrease size of your payloads

## 3. Database Optimization

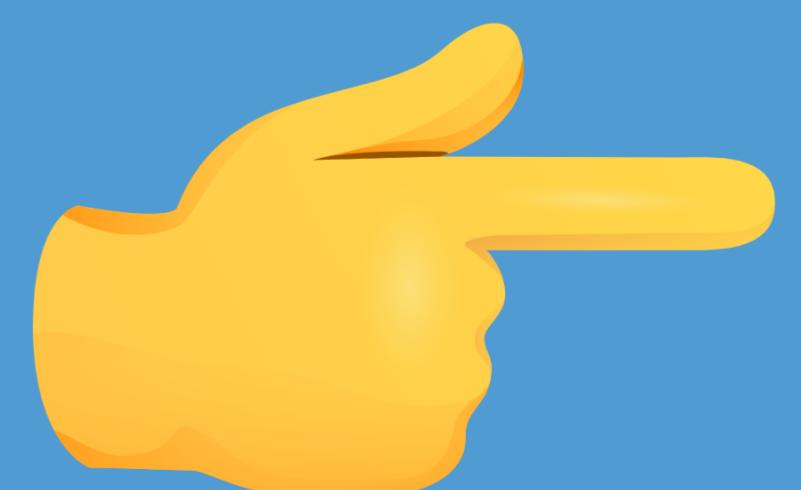
- Optimize queries and use efficient data retrieval techniques

## 4. Profiling and Diagnostics

- Use profiling tools to identify performance bottlenecks.
- Analyze application metrics and logs.



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



### Phase 11: Observability

#### 1. Health Checks

- Configure health checks for critical application components like databases, caches, and external services.

#### 2. Open Telemetry

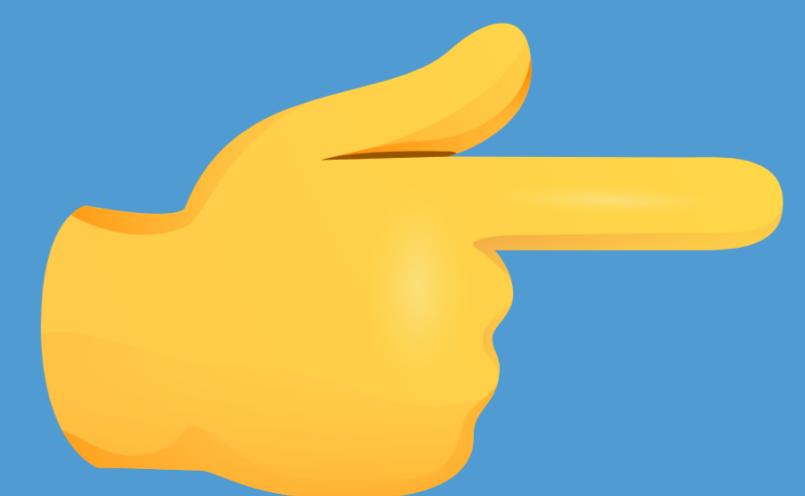
- Distributed Tracing
- Metrics Collection
- Logging Integration

#### 3. Monitoring Tools

- Prometheus + Grafana
- Jaeger
- Seq
- Azure Monitor



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)



### Phase 12: Advanced Topics

#### 1. Rate Limiting

- **Implement rate limiting for your APIs**

#### 2. Multiple Auth

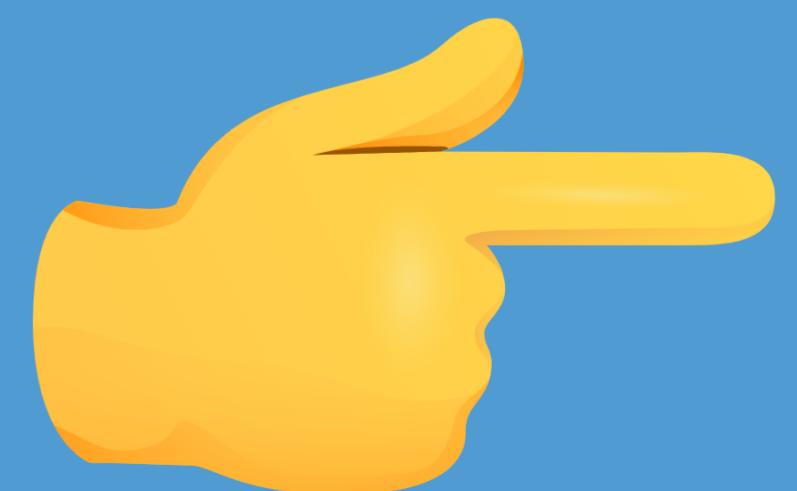
- **Implement multiple Auth types**

#### 3. Create SDK for your APIs

- **Create and publish a Nuget package with models, interfaces to communicate with an API**
- **Use Refit to communicate with an API**



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)





THANKS FO READING

LETS CONNECT ON LINKED-IN

REPOST TO HELP ME SHARE  
USEFUL CONTENT FOR OTHERS

SUBSCRIBE TO ANTONDEVTIPS  
TO GET MORE .NET TIPS



Anton Martyniuk  
[antondevtips.com](http://antondevtips.com)

