ASP.NET Core Web API Learning Roadmap

# 1. Setup & Project Structure

Install .NET SDK & Visual Studio / VS Code

Create your first Web API using: dotnet new webapi

Understand project structure:  
 - Program.cs  
 - Controllers/  
 - appsettings.json  
 - launchSettings.json

# 2. Controllers and Routing

What is a Controller?

Creating a new Controller

HTTP Verbs: [HttpGet], [HttpPost], [HttpPut], [HttpDelete]

Route attributes: [Route], [ApiController], [FromRoute], [FromQuery], [FromBody]

# 3. Models and Data Binding

Creating Model classes (POCOs)

Binding data from JSON to C# objects

Model Validation using Data Annotations ([Required], [StringLength], etc.)

# 4. Dependency Injection (DI)

What is DI?

Registering services in Program.cs (builder.Services.AddScoped())

Injecting services into Controllers

# 5. CRUD Operations with In-Memory Data

Create a simple in-memory list to store data

Implement full CRUD:  
 - Create (POST)  
 - Read (GET)  
 - Update (PUT)  
 - Delete (DELETE)

# 6. Using Entity Framework Core

Install EF Core packages

Create a DbContext and connect to a real database (SQL Server or SQLite)

Add Migrations & Update database

Use EF Core in your CRUD logic

# 7. API Response & Status Codes

Return types: IActionResult, ActionResult<T>, Ok(), NotFound(), BadRequest()

Custom response messages

# 8. Middleware & Pipeline

Understand the middleware pipeline in Program.cs

Add custom middleware (logging or header checker)

# 9. Exception Handling & Logging

Global exception handling using middleware

Built-in logging (ILogger)

# 10. appsettings.json & Configuration

Using configuration in code

Reading custom sections (e.g., connection strings)

# 11. Calling APIs with Swagger/Postman

Use Swagger UI for testing

Use Postman to test different endpoints with request bodies and headers

# 12. Advanced Concepts

Authentication & Authorization (JWT)

Filters (Action Filters, Exception Filters)

Versioning your API

Rate Limiting & Caching

Uploading files via API

API security best practices