

asp net core 2 and angular 5 ebook now just 5

Published: September 19, 2026 | Index ID: #-

In today's fast moving digital landscape, building modern web applications that are scalable, maintainable, and deliver a rich user experience is more important than ever. Two of the most powerful frameworks that enable developers to accomplish this are **ASP.NET Core 2** for the server side and **Angular 5** for the client side. Together, they form a cohesive full stack solution that blends the robustness of .NET with the flexibility of a modern JavaScript framework.

Whether you're a seasoned developer looking to refresh your toolkit, or a newcomer eager to dive into full stack development, this article will walk you through everything you need to know. From setting up the development environment to deploying a production ready application, we'll cover it all. And if you're looking to deepen your knowledge, the **ASP.NET Core 2 and Angular 5 ebook now just 5** offers a comprehensive, step by step guide that's perfect for self study.

Why ASP.NET Core 2 and Angular 5? The Power of a Modern Stack

Before we dive into code, it's useful to understand why this particular combination is so popular:

- **Cross Platform and Open Source** – ASP.NET Core 2 runs on Windows, macOS, and Linux, and Angular is built on TypeScript, a strongly typed superset of JavaScript. This means you can develop and deploy on a wide range of environments.
- **Performance and Scalability** – ASP.NET Core 2 is engineered for high throughput, while Angular 5's component based architecture promotes efficient rendering and state management.
- **Rich Ecosystem** – Both frameworks boast extensive libraries, tooling, and community support. Whether you need authentication, real time communication, or advanced UI components, you'll find ready made solutions.
- **Seamless Integration** – The two frameworks integrate naturally through RESTful APIs, WebSockets, or gRPC, allowing a clean separation of concerns between front end and back end.

With these advantages, it's no wonder that enterprises and indie teams alike choose ASP.NET Core 2 and Angular 5 for building SaaS platforms, e commerce sites, internal dashboards, and more.

Getting Started – Setting Up Your Development Environment

Before we can write any code, we need to install the necessary tooling. Below is a step by step guide for Windows, macOS, and Linux.

1. Install the .NET Core 2 SDK

1. Navigate to the official .NET Core 2.2 download page. The 2.2 LTS version is still widely used and fully supported.
2. Choose the installer that matches your operating system and architecture.
3. Run the installer and follow the prompts. Verify the installation with:

```
dotnet --version
```

It should return 2.2.x.

2. Install Node.js and npm

Angular relies on Node.js for its build tools and package management.

1. Visit nodejs.org and download the LTS version.
2. Install it, ensuring that npm (Node Package Manager) is included.
3. Verify with:

```
node -v npm -v
```

3. Install the Angular CLI

The Angular CLI streamlines project creation, scaffolding, and building.

1. Run:

```
npm install -g @angular/cli@5
```

This installs the CLI for Angular 5, ensuring compatibility with the rest of the stack.

4. Create a New Solution

Let's create a single solution that houses both the API and the Angular front end.

1. Open a terminal or command prompt and run:

```
dotnet new sln -n FullStackDemo
```

This creates a new solution file.

5. Scaffold the ASP.NET Core API

1. Navigate to a folder where you want the API:
2. Run:

```
dotnet new webapi -n Api
```

Then add it to the solution:

```
dotnet sln FullStackDemo.sln add Api/Api.csproj
```

6. Scaffold the Angular Application

1. Navigate to the solution root and run:

```
ng new Client --routing --style=scss
```

When prompted, choose **No** for *Would you like to add Angular routing?* (Yes/No)(we'll add it manually). Accept the default settings for SCSS.

After creation, add the Angular project to the solution:

```
dotnet sln FullStackDemo.sln add Client/Client.csproj
```

Note: The Client.csproj file is a placeholder; the Angular project does not need a .NET project file, but adding it keeps the solution organized.

Project Structure – A Clear Separation of Concerns

Understanding the folder layout helps you maintain and scale your application. Below is a typical structure for a combined ASP.NET Core 2 and Angular 5 project:

- Api/ – Contains all server side logic: Controllers, Services, Models, and Data Access.
- Client/ – The Angular workspace, containing src/app for components, services, and modules.
- FullStackDemo.sln – The solution file that ties everything together.
- Dockerfile – Optional, for containerizing the application.

Building the API – ASP.NET Core 2 Fundamentals

1. Defining Models

Start by defining a simple Product model. Create Models/Product.cs inside the Api project:

```
namespace Api.Models { public class Product { public int Id { get; set; } public string Name { get; set; } public decimal Price { get; set; } }}
```

2. Creating a Repository

To keep data access logic separate, create a repository interface and implementation:

Interfaces/IProductRepository.cs

```
using System.Collections.Generic; using Api.Models; namespace Api.Interfaces { public interface IProductRepository { IEnumerable GetAll(); Product GetById(int id); void Add(Product product); void Update(Product product); void Delete(int id); }}
```

Repositories/ProductRepository.cs

```
using System.Collections.Generic; using System.Linq; using Api.Interfaces; using Api.Models; namespace Api.Repositories { public class ProductRepository : IProductRepository { private readonly List _products = new List { new Product { Id = 1, Name = "Laptop", Price = 999.99m }, new Product { Id = 2, Name = "Smartphone", Price = 499.99m } }; public IEnumerable GetAll() => _products; public Product GetById(int id) => _products.FirstOrDefault(p => p.Id == id); public void Add(Product product) { product.Id = _products.Max(p => p.Id) + 1; _products.Add(product); } public void Update(Product product) { var existing = GetById(product.Id); if (existing != null) { existing.Name = product.Name; existing.Price = product.Price; } } public void Delete(int id) { var product = GetById(id); if (product != null) { _products.Remove(product); } } }}
```

3. Registering Services in Startup.cs

Open Startup.cs and add the repository to the dependency injection container.

```
public void ConfigureServices(IServiceCollection services) { services.AddMvc(); services.AddSingleton<IProductRepository, ProductRepository>(); services.AddCors(options => { options.AddPolicy("AllowAll", builder => builder.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader()); }); }
```

4. Implementing the Controller

Create Controllers/ProductController.cs:

```
using System.Collections.Generic; using Microsoft.AspNetCore.Mvc; using Api.Interfaces; using Api.Models; namespace Api.Controllers { [Route("api/[controller]")] [ApiController] public class ProductController : ControllerBase { private readonly IProductRepository _repo; public ProductController(IProductRepository repo) { _repo = repo; } [HttpGet] public ActionResult> Get() => Ok(_repo.GetAll()); }}
```

```
[HttpGet("{id}")] public ActionResult Get(int id) { var product = _repo.GetById(id); if (product == null) return NotFound(); return Ok(product); } [HttpPost] public ActionResult Post([FromBody] Product product) { _repo.Add(product); return CreatedAtAction(nameof(Get), new { id = product.Id }, product); } [HttpPut("{id}")] public IActionResult Put(int id, [FromBody] Product product) { if (id != product.Id) return BadRequest(); _repo.Update(product); return NoContent(); } [HttpDelete("{id}")] public IActionResult Delete(int id) { _repo.Delete(id); }
```

Baca Juga (Artikel Terkait)

- [american crime story season 2 release date cast and](http://adulvibex.com/american-crime-story-season-2-release-date-cast-and.pdf)

URL: <http://adulvibex.com/american-crime-story-season-2-release-date-cast-and.pdf>

- [american crew product guide](http://adulvibex.com/american-crew-product-guide.pdf)

URL: <http://adulvibex.com/american-crew-product-guide.pdf>

- [american corrections 10th edition](http://adulvibex.com/american-corrections-10th-edition.pdf)

URL: <http://adulvibex.com/american-corrections-10th-edition.pdf>

- [american constitutional law essays and cases](http://adulvibex.com/american-constitutional-law-essays-and-cases.pdf)

URL: <http://adulvibex.com/american-constitutional-law-essays-and-cases.pdf>

Tags: [#core](#) [#angular](#) [#ebook](#) [#just](#)

