Real-World Web Development with .NET 9

Build production-ready web applications and APIs with modern C# and ASP.NET Core. For intermediate developers ready for professional projects.

Programming Instructor: David Clarke • 5 sections • 51 lessons • 17.50h total

View this course online Browse all courses

About this course

Most .NET tutorials stop at 'Hello World.' This course picks up where those leave off, teaching you the patterns, tools, and techniques used to build web applications that actually ship to production. You'll work with ASP.NET Core, Entity Framework Core, Blazor, and the full .NET 9 stack, applying each concept to realistic scenarios rather than toy examples.

Across 51 lessons, you'll build APIs, secure them with modern authentication, implement background processing, integrate message queues, and deploy to the cloud. The course finishes with a complete e-commerce application that ties everything together. Each section includes downloadable source code and practical exercises you can apply to your own projects.

Course curriculum

Foundations and Preparation

- Setting Up Your .NET 9 Development Environment (15m)
  Install the .NET 9 SDK, configure Visual Studio and VS Code for web work, and set up essential extensions. You'll have a working environment ready before we write a single line of code.
- Anatomy of a .NET 9 Web Project (18m)
  Walk through the project structure of an ASP.NET Core application file by file. We'll cover Program.cs, the wwwroot folder, appsettings.json, and what each configuration file does.
- C# 13 Features You'll Actually Use in Web Projects (22m)
  A focused look at collection expressions, primary constructors, interceptors, and other C# 13 features that show up constantly in modern .NET web code. No filler, just what matters.
- How the ASP.NET Core Request Pipeline Works (25m)
  Understand the middleware pipeline from the inside out. We'll trace a request from arrival to response, then build custom middleware for cross-cutting concerns like request timing.
- Dependency Injection Without the Confusion (20m)
  DI is built into ASP.NET Core, but knowing when to use transient, scoped, or singleton lifetimes trips people up. This lesson clarifies the patterns with practical examples.
- Configuration and the Options Pattern (15m)
  Manage application settings across environments using appsettings files, environment variables, and the strongly-typed Options pattern. Includes secret management with user secrets.
- Structured Logging with Serilog and Built-in Providers (18m)
  Set up structured logging that actually helps you debug production issues. We'll configure both the built-in ILogger and Serilog with sinks for console, files, and Seq.

Building APIs and Working with Data

- Your First Minimal API Endpoint (18m)
  Build a working REST endpoint in under 15 minutes using .NET 9 minimal APIs. Covers routing, parameter binding, and returning proper HTTP status codes.
- RESTful APIs with Controllers and ActionResult (22m)
  When your API grows beyond a few endpoints, controllers give you structure. We'll build a full CRUD controller with proper content negotiation and response types.
- Model Binding, Validation, and DTOs (20m)
  Separate your API contracts from your domain models. Learn FluentValidation, data annotations, and how to map between entities and DTOs with Mapster.
- Introduction to Entity Framework Core 9 (25m)
  Set up EF Core with SQL Server, create your first DbContext, and understand change tracking, lazy loading, and the query pipeline. This is the foundation for everything data-related.
- Code-First Migrations: From Schema to Database (18m)
  Generate, customize, and apply database migrations step by step. We'll handle a real scenario where the schema changes multiple times during development.

- Querying Data with LINQ and EF Core (22m)
  Write efficient queries using LINQ syntax, filtering, sorting, pagination, and projection. Includes a look at what SQL EF Core actually generates using query logging.
- Relationships and Complex Data Models (25m)
  Model one-to-one, one-to-many, and many-to-many relationships. We'll build a realistic data model for a product catalog with categories, tags, and reviews.
- Repository and Unit of Work Patterns (20m)
  Implement these patterns over EF Core for testability and cleaner service code. We'll discuss when they add value and when they just add unnecessary abstraction.
- Seeding Data and Handling Concurrency Conflicts (15m)
  Seed your database with realistic test data and handle optimistic concurrency using row versioning. Both are essential for real applications.
- Database Performance: Indexes and Query Optimization (22m)
  Identify slow queries, add composite indexes, and use compiled queries. We'll profile an API endpoint and cut response time in half with targeted changes.
- Working with NoSQL: Cosmos DB Integration (18m)
  Add Cosmos DB alongside your relational database for specific use cases like session storage and activity logs. Covers the EF Core provider and direct SDK usage.

Security, Performance, and Cross-Cutting Concerns

- Authentication with ASP.NET Core Identity (25m)
  Set up Identity with registration, login, password reset, and email confirmation. We'll customize the default Identity UI and integrate it with your API.
- Role-Based and Policy-Based Authorization (20m)
  Go beyond simple role checks. Build policy-based authorization with custom requirements, handlers, and resource-based access control for real scenarios.
- JWT Token Authentication for APIs (22m)
  Issue and validate JWT tokens for stateless API authentication. Covers access tokens, refresh tokens, token rotation, and secure storage strategies.
- OAuth 2.0 and External Login Providers (25m)
  Let users sign in with Google, GitHub, or Microsoft accounts. We'll configure external providers and merge external and local accounts safely.
- Caching Strategies: In-Memory, Distributed, and Output (22m)
  Apply the right caching approach for each scenario. Covers IDistributedCache with Redis, in-memory caching, response caching, and cache invalidation patterns.
- Rate Limiting and Request Throttling (15m)
  Protect your API from abuse using the built-in rate limiter middleware. Configure sliding window, token bucket, and fixed window policies per endpoint.
- Global Exception Handling Patterns (18m)
  Build a consistent error response model with ProblemDetails. Implement exception handler middleware, custom exception types, and RFC 9457 problem details.

- Health Checks and Application Monitoring (15m)
  Add health check endpoints for databases, external services, and custom conditions. Wire them into monitoring dashboards with the health check UI.
- API Documentation with OpenAPI and Scalar (12m)
  Generate interactive API documentation using the new built-in OpenAPI support in .NET 9 and Scalar as a modern alternative to Swagger UI.

Advanced Architecture and Patterns

- Clean Architecture for .NET Solutions (25m)
  Structure your solution with clear boundaries between domain, application, infrastructure, and presentation layers. We'll refactor a flat project into a layered one.
- CQRS Pattern with MediatR (22m)
  Separate read and write operations using commands and queries. Implement MediatR for in-process messaging and add pipeline behaviors for validation and logging.
- Domain-Driven Design Building Blocks (20m)
  Work with entities, value objects, aggregates, and domain events. We'll model a real business domain where DDD concepts solve actual design problems.
- Real-Time Communication with SignalR (25m)
  Build a live notification system and a real-time dashboard using SignalR hubs. Covers connection management, groups, and scaling with a Redis backplane.
- Background Processing with Hosted Services (18m)
  Run tasks outside the request pipeline: scheduled jobs, queue processors, and long-running workflows. We'll implement a background email queue processor.
- Message-Based Architecture with RabbitMQ (22m)
  Decouple services with asynchronous messaging. Set up RabbitMQ with MassTransit, implement publish/subscribe, and handle dead letter queues.
- gRPC Services for High-Performance APIs (20m)
  When HTTP/JSON isn't fast enough, gRPC gives you binary serialization and streaming. Build a gRPC service and consume it from both .NET and a Blazor client.
- Blazor Server: Building Interactive Server-Side UIs (25m)
  Build a data dashboard with Blazor Server components. Covers component lifecycle, parameter binding, event handling, and the SignalR circuit underneath.
- Blazor WebAssembly: Client-Side .NET in the Browser (25m)
  Run .NET directly in the browser with WebAssembly. Build a standalone Blazor WASM app that calls your API, handles routing, and manages offline state.
- Blazor Component Patterns and State Management (22m)
  Share state across components using cascading values, scoped services, and state containers. We'll also cover component libraries and JS interop when needed.
- Building Microservices with .NET Aspire (28m)
  Orchestrate multiple services in local development using .NET Aspire. Set up service discovery, shared resources, and the Aspire dashboard for distributed tracing.

Full-Stack Implementation and Deployment

- Project Blueprint: Planning the E-Commerce Application (18m)
  Before writing code, we'll define the requirements, user stories, and architecture for a real e-commerce platform. Includes database diagrams and API contract sketches.
- Setting Up the Solution Structure (15m)
  Create the solution with separate projects for domain, application, infrastructure, API, and Blazor frontend. Configure shared settings, CI build scripts, and editor configs.
- Building the Data Layer with EF Core (25m)
  Define entities for products, orders, customers, and payments. Set up the DbContext with proper configurations, value conversions, and seed data for testing.
- Implementing the Product Catalog API (22m)
  Build CRUD endpoints for products with filtering, sorting, pagination, and full-text search. Apply the CQRS patterns learned earlier with MediatR handlers.
- User Registration and Authentication Flow (20m)
  Implement registration with email verification, login with JWT, password reset, and profile management. Wire it into the Blazor frontend with auth state providers.
- Shopping Cart and Checkout Workflow (25m)
  Build the cart as a persistent entity tied to the user session. Implement address collection, shipping calculation, and an order placement workflow.
- Payment Integration with Stripe (22m)
  Integrate Stripe Checkout and Payment Intents for secure payment processing. Handle webhooks for payment confirmation, refunds, and failed payments.
- Order Processing and Email Notifications (20m)
  Build an order processing pipeline with status tracking, inventory updates, and background email notifications using hosted services and Razor email templates.
- Admin Dashboard with Blazor (25m)
  Create an admin area for managing products, viewing orders, and processing refunds. Use Blazor components with MudBlazor for a functional, clean interface.
- Writing Unit and Integration Tests (25m)
  Test your business logic with xUnit, verify API endpoints with WebApplicationFactory, and mock external dependencies. Aim for meaningful coverage, not just high percentages.
- Containerization with Docker (18m)
  Write Dockerfiles for your API and Blazor app, set up docker-compose with SQL Server and Redis, and configure multi-stage builds for smaller production images.
- Deploying to Azure App Service (20m)
  Deploy your application to Azure using deployment slots for zero-downtime releases. Configure environment variables, connection strings, and managed identities.
- Production Monitoring and Where to Go Next (18m)
  Set up Application Insights for telemetry, configure alerts, and review common production issues. We'll wrap up with guidance on continuing your .NET learning path.

Discover our blog

Learn more with free articles written by our professors and guest experts.

Latest article: **Flight Attendant Preparation: What to Study Before You Apply**

Visit the blog All courses This course