BOARD INFINITY

BEGINNER LEVEL · BOARD INFINITY ON COURSERA

Introduction to .NET Core

Full course syllabus - all three modules and every learning objective, from what .NET Core is to your own middleware.



Bhawna Gunwani
Backend development specialist · your instructor

3	40	10	11h
Modules	Video lessons	Assignments	Total time

INSIDE THE COURSE

01	Introduction to .Net Development	approx. 4 hours
02	Building Web Applications using ASP.NET Core	approx. 3 hours
03	.NET Core Advance Concepts	approx. 4 hours

Enroll on Coursera

https://www.coursera.org/learn/intro-to-dotnet-core

01 ABOUT THE COURSE

One course that goes all the way to middleware.

Most .NET introductions stop once a project runs. This one keeps going - through ASP.NET Core and MVC into dependency injection, the middleware pipeline, configuration, debugging and testing.

It starts with what .NET Core actually is, how its version history got here, how it differs from the .NET Framework, and what the CLR, CoreCLR and CoreFX are doing underneath. Every module follows the same shape: a short screen-recorded lesson, then a quiz on the same idea - 40 lessons and 10 assignments, seven practice quizzes and three graded quizzes that close each module.

It assumes no prior .NET background. Module one sets up your environment in Visual Studio and VS Code on screen and gets a console app and a web app running before it introduces a single controller.

WHO IT'S FOR

Beginners

New to .NET entirely. Module one covers features, versions and the differences between them before any code.

Self-learners

You can scaffold a project but cannot say what the pipeline is doing. The 10 assignments show what you actually know.

Developers

Coming from another stack: the CLI, the MVC pattern, the IoC container and how a request travels through middleware.

Students

Full-stack .NET, backend and API courses all assume ASP.NET Core. This is that prerequisite, taught properly.

WHAT YOU'LL LEARN

01 **Understand .NET Core's features, versions and setup** Module 1
Version history, .NET Core vs the .NET Framework, and an environment set up in Visual Studio and VS Code.

02 **Master the architecture, CLI and core components** Module 1
Language compilers, CoreCLR and CoreFX, and the CLI commands to develop, build, test and deploy.

03 **Build web applications with ASP.NET Core and MVC** Module 2
The MVC pattern, requests, static files, session and state, routing and hosting environments.

04 **Work with DI, middleware, configuration and testing** Module 3
IoC containers, the middleware pipeline, custom middleware, exception handling and multi-targeting.

02 SYLLABUS

Three modules, from what .NET Core is to your own middleware.

Eleven hours end to end, fully self-paced. Deadlines flex around your schedule.

01 Introduction to .Net Development

Features, versions, architecture and the .NET CLI

approx. 4 hours

16 videos (47 min)

BY THE END OF THIS MODULE YOU CAN

- ✓ Explain what .NET Core is, the features it introduced, and its version history
- ✓ Describe .NET Framework vs .NET Core, and ASP.NET vs ASP.NET Core applications
- ✓ Set up Visual Studio and VS Code, and read an ASP.NET Core project folder structure
- ✓ Use .NET CLI commands, and name the language compilers, CoreCLR and CoreFX

CONTENTS 16 videos (47 min) · 3 readings (70 min) · 3 assignments (90 min) · 1 plugin (5 min)

TOPICS .NET Core features · Version history · .NET Framework vs .NET Core · ASP.NET Core · Project folder structure · Visual Studio & VS Code · .NET CLI

You leave with: A working .NET development environment, and your first console and ASP.NET Core web applications running.

02 Building Web Applications using ASP.NET Core

MVC, requests, static files, routing and hosting

approx. 3 hours

10 videos (41 min)

BY THE END OF THIS MODULE YOU CAN

- ✓ Explain the MVC pattern and how the model, the view and the controller communicate
- ✓ Build a web application with ASP.NET Core and handle requests in .NET Core MVC
- ✓ Serve static files, and manage session and state across requests
- ✓ Configure routing, and choose between the hosting environments .NET Core provides

CONTENTS 10 videos (41 min) · 2 readings (60 min) · 3 assignments (90 min)

TOPICS MVC pattern · Model-view-controller communication · Handling requests · Static files · Session & state management · Routing configuration · Hosting environments

You leave with: An ASP.NET Core web application you routed and configured yourself, built the way the framework expects.

03

.NET Core Advance Concepts

Dependency injection, middleware, debugging and testing

approx. 4 hours

14 videos (47 min)

BY THE END OF THIS MODULE YOU CAN

- Use ASP.NET Core environment variables to configure an application per environment
- Apply dependency injection with the IoC container; loosely vs tightly coupled design
- Trace the middleware request pipeline, and write custom middleware
- Handle exceptions, target multiple frameworks and share code across projects

CONTENTS 14 videos (47 min) · 4 readings (100 min) · 4 assignments (105 min)

TOPICS

Environment variables | Dependency injection | IoC containers | Middleware pipeline | Custom middleware | Exception handling | Multi-targeting & code sharing

You leave with: A .NET Core application built from scratch, with dependency injection and your own middleware in the pipeline.

03 TOOLS & SKILLS

The tools you work in.

.NET Core — The framework

Architecture, the platform, ASP.NET Core, dependency injection, middleware and deployment.

In this course: The whole course is built on it, from the first console app to custom middleware.

Visual Studio — Where you write

A full IDE, set up in a demo lesson in module one.

In this course: You write in a real IDE rather than a toy console from the first module.

VS Code — The lighter option

Covered alongside Visual Studio in the same demo.

In this course: Lets you follow the course on Windows, macOS or Linux.

.NET CLI — Where you build

The commands you use to create, build, test and deploy from a terminal.

In this course: Its own lessons in module one.

SKILLS YOU LEAVE WITH

Building applications

- Application development
- Web development
- Cross platform development
- Open source technology

Designing it

- Software architecture
- Model view controller
- Development environment
- Microsoft development tools

Shipping it

- Software testing
- Debugging
- Application deployment

04 CERTIFICATE & CAREER

Finish with proof you can put to work.

One-click LinkedIn add
Filed under Licenses & Certifications on your profile, where recruiters look.

Independently verifiable
Its own verification link, so a hiring manager confirms it in seconds.

Earned, not attended
Awarded on the graded quizzes that close each of the three modules.

Two names on it
Board Infinity as the provider, Coursera as the platform.

WHERE THESE SKILLS ARE IN DEMAND

.NET Developer | ASP.NET Developer | Backend Developer | Web Developer | Software Developer | Career switchers



Bhawna Gunwani
Backend development specialist · Works across C# .NET, Python, MEAN & MERN · instructor for this course with Board Infinity

Ready to build your first ASP.NET Core app?

Enrollment, lessons, grading and the certificate all happen on Coursera.

Enroll on Coursera

Course page: https://www.boardinfinity.com/courses/intro-to-dotnet-core/

Want a coach alongside the course?

Board Infinity 1:1 coaching is an optional add-on: a practitioner who writes .NET for a living reviews the code you build. https://www.boardinfinity.com/career-coaching/

Board Infinity is an independent education provider. This course is hosted on Coursera; Coursera and the Coursera logo are trademarks of Coursera, Inc. This course is not affiliated with, endorsed by or sponsored by Microsoft. Syllabus details are accurate at the time of publication and may be updated by the instructor.