

# Clean Architecture .NET Core training

3 days (21 hours)

## Presentation

Create a loosely coupled, reverse-dependent application!

Clean Architecture is a software architecture designed to keep code under control, without need tidy it up to prevent anyone from touching it after it has been released. The main concept of Clean Architecture is that the application's code/logic should be written without any direct dependencies.

If you modify the database or the user interface, the core system (business rules/domain) must remain unchanged. This means that external dependencies are completely replaceable.

Clean architecture makes an application independent of any framework, database or user interface. It is testable and well-organized. This architecture is composed of 4 distinct categories: Domain, Application, Infrastructure and Presentation. This architecture is often referred to as Onion architecture, hexagonal architecture or even port & adapter. The key word clean architecture comes from [Uncle Bob](#).

Thanks to our Clean Architecture in .NET Core course, you'll be able to create an architecture for developing testable, robust APIs that can be easily understood and adopted by other developers.

In this course, as in all our training courses, we'll be using the [latest stable version](#) (.NET 9 with the new [C# 13 language](#)).

## Objectives

- Help you keep your application easy to develop, understand and maintain
- Clean Architecture project structure
- Using CQRS (Common Query Responsibility Segregation)

- Implementation of unit and integration tests

## Target audience

- Web and application developers
- Architects

## Prerequisites

- Knowledge of object-oriented programming
- Knowledge of .NET environment
- [Test My Knowledge](#)

## Technical requirements

- VS Code installed
- Have a Github account

## Program of our Clean Architecture with .NET Core training course

### INTRODUCTION

- The SOLID principles
- The 3 main .Net application architectures
- Introduction to the principles of Clean Architecture
- Introduction to the principles of Test-Driven Development (TDD)

### ORGANIZING THE .NET SOLUTION according to Clean Architecture principles

- Domain
  - Define the entities handled by API
  - Enumerations
  - The exceptions

- Application
  - Handling domain data
  - Interfaces
  - The CQRS approach (MediatR)
  - Validators with FluentValidation
- Infrastructure
  - Service (API, EntityFramework, File management)
    - Dependency injections
    - Interface implementation
  - Interaction with a SQL database (DB context, migration, EF core)
- Presentation
  - API

## Introduction to TDD AND BDD

- Gherkin / Cucumber syntax
- Using SpecFlow
- Putting it into practice

## API ASP.NET CORE WEB APP

- Initiating the project
  - Swagger
  - MediatR
  - Dependency injections

## USER INTERFACE

- Introduction to Blazor
- Creating a Blazor application

## Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

## Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is confirmed, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the forthcoming course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or learning difficulties.

in-company security (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

## Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

## Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

## Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

## Sanction

A certificate will be issued to each trainee who completes the course.