

.NET 7 for Jobseekers

Elevate your coding journey with .NET 7

Filipe Vilhena



www.bpbonline.com

Copyright © 2023 BPB Online

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2023

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55518-224

www.bpbonline.com

Dedicated to

My beloved wife:

Sónia

&

My Son Afonso

About the Author

Filipe Vilhena is a software developer with nearly 17 years of experience, having worked in different companies and projects, in the banking and insurance area, in Portugal, and delivering projects using different Microsoft Technologies, such as ASP.net, C#, VB.net, T-SQL, Integration Services, Reporting Services, javascript, JQuery and others.

At the moment, he is working as a DevOps Engineer in Axa Partners Portugal, developing several in-house application for insurance, assistance and internal management. He is also working with Agile methodologies and helping to spread them throughout the company.

He finished his degree of studies in Escola Superior de Tecnologia de Setúbal, in 2007, and has since taken a few courses in Microsoft.

In his spare time, he loves to travel and spend quality time with his family (and his kitten!), whenever possible. Also, he's a fan of cinema, music, photography, and an eager reader, who loves a good thriller, being also a huge Nordic Noir fan!

About the Reviewer

Abhishek Jain is a passionate programmer with extensive professional experience in developing Large-Scale application with Microsoft technologies, Angular and other native language, Application fitment for cloud, Cloud Operating Model, Cloud Vendor Evaluation, Cloud Policy and Cloud Economics. He is an holder of multiple industry operation knowledge which enhance experience in understand workflow and in understand level of work to be done.

Abhishek specializes in Data Science and FinOps and loves working with customers on helping them in getting the most out of their cloud investment.

He is currently working as Senior Solution Architect in USA Company.

Acknowledgement

I would like to thank in the first place, to my family, Sónia and Afonso, for their strong support and encouragement and also for their patience when I was writing until late hours after work, and investigating for this book.

I would also like to thank my parents António and Otilia, and all my family for turning me into the people I am and for helping me to learn and study and evolve as a person.

I am also grateful to BPB Publications, it was a long journey of investigation, writing, and revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

I am also thankful to my teachers at Escola Secundária da Moita and Escola Superior de Tecnologia de Setúbal, who helped me and inspired me to follow this career, namely: my teacher Anabela from 12th-grade Physics and my Faculty teachers Adelaide Trabuco, Nuno Pina, Alpesh Ranchordas, Rui Madeira and Hugo Gamboa.

I would also like to thank the contributions of my colleagues and co-worker during many years working in the tech industry, who have taught me so much and guided me into learning what I know now and helped me to gain experience to write this book. Also, a special thanks to my colleague Edgar Elias, that also served as a reviewer for this book, and who gave me some insights on the matter.

Finally, I would like to thank all the readers who have interest in this book and in learning more.

Preface

Working with .net technologies is quite an interesting task and, since it is always evolving, it is a great challenge to learn more and more everyday.

This book is designed to provide a guide to what .Net is, how it is evolving and what is new in this .Net 7 version. It covers a wide range of topics, including knowledge about MVC, Razor, Blazor, .Net MAUI, Orleans and more.

Throughout the book, you will learn the basics and a bit of history on .Net framework, and will get to know more about different functionalities and improvements that .Net 7 brings.

This book is intended for the ones who want to learn about .Net and mainly its latest version of .Net 7, so it will help them in their always challenging task of Job seeking.

With this book, you will gain the knowledge and skills to learn how .Net 7 works and what it brings as new. I hope you will find this book informative and helpful.

Chapter 1: Introduction to .Net 7- will make a quick introduction to the environment and framework, in order to understand a bit of the story behind it and how did it get to this version 7. It will contain examples of the existing framework and what it covers now.

Chapter 2: New Features and Libraries - will review this framework version and all the new features and libraries implemented, regarding the last stable framework (.net 6). It will be illustrated with a few samples of the features.

Chapter 3: Writing Your First .Net 7 Program – covers, in a small walkthrough, all the steps to create a simple program with the Hello World typical example, using this new .net 7 framework. It will be created in a simple way using Razor pages.

Chapter 4: Designing the Views - will explain in a simple way how to create some views of a .net 7 application.

Chapter 5: Creating Your Controllers - will explain in a simple way how to create the controllers for a .net 7 application, giving some examples of an application. It will show how the controller can connect with the Views.

Chapter 6: Testing Your Views and Controllers- will go through the way to test the connection and use of the Views and controllers.

Chapter 7: Working with .Net MAUI - explains what is .Net MAUI and how to use this new App UI. It will show some examples of possible integrations. It will also explain what are the Operating Systems and platforms on which it will be implemented.

Chapter 8: Blazor in .Net7 - is dedicated to learn what is Blazor and the improvements it has in .Net 7. It will show some examples and some different ways to use it.

Chapter 9: Creating a Desktop UI – after learning about MAUI it is dedicated to learn how to create a desktop UI and to use a bit of WinForms.

Chapter 10: Communication with the Views - is dedicated to explain how to communicate between UI and the Views created in chapter 4.

Chapter 11: Use SignalR - explains what is SignalR, what's the improvements on .Net 7 and how to use it.

Chapter 12: Adding a Database - explains how to add a database and how to connect it to this model using .net 7.

Chapter 13: Orleans - explains what is Orleans, how to use it and what it has new in .Net 7.

Chapter 14: Adding Specific Code Using System.Devices - will explain how to add specific code with System.Devices and this new framework.

Chapter 15: Possible Questions and Answers - will go through some possible questions when landing a .Net role and provide some answers.

Code Bundle and Coloured Images

Please follow the link to download the *Code Bundle* and the *Coloured Images* of the book:

<https://rebrand.ly/dpljvfh>

The code bundle for the book is also hosted on GitHub at **<https://github.com/bpbpublications/.NET-7-for-Jobseekers>**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **<https://github.com/bpbpublications>**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At **www.bpbonline.com**, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

<https://discord.bpbonline.com>



Table of Contents

| | |
|--|----------|
| 1. Introduction to .Net 7..... | 1 |
| Introduction..... | 1 |
| Structure..... | 2 |
| A small piece of .Net history | 2 |
| Examples of .Net 6..... | 4 |
| <i>Middleware integration</i> | 4 |
| <i>Routing</i> | 5 |
| <i>Add Services</i> | 6 |
| Conclusion..... | 7 |
| 2. New Features and Libraries..... | 9 |
| Introduction..... | 9 |
| New features of this framework..... | 10 |
| <i>Blazor</i> | 10 |
| <i>New loading page</i> | 11 |
| <i>Data binding modifiers</i> | 11 |
| <i>Virtualization improvements</i> | 12 |
| <i>Navigation manager</i> | 12 |
| <i>Additional cryptography support</i> | 13 |
| <i>New problem details service</i> | 13 |
| <i>Diagnostics middleware updates</i> | 13 |
| <i>Minimal API</i> | 13 |
| <i>Typed results</i> | 14 |
| <i>OpenAPI improvements</i> | 15 |
| <i>Self-describing minimal APIs</i> | 15 |
| <i>Return multiple result types</i> | 16 |
| <i>MVC</i> | 17 |
| <i>Links</i> | 17 |

| | |
|---|-----------|
| New libraries | 18 |
| <i>Nullable annotations for Microsoft.Extensions.....</i> | 18 |
| <i>System.Text.RegularExpressions improvements</i> | |
| <i>and new APIs.....</i> | 19 |
| <i>LibraryImportGenerator</i> | 19 |
| .Net 6 vs .Net 7 | 20 |
| <i>Blazor and .Net MAUI.....</i> | 20 |
| <i>Cloud native and containers.....</i> | 21 |
| <i>Json.....</i> | 22 |
| <i>JWT authentication configuration.....</i> | 23 |
| <i>C#11.....</i> | 24 |
| Conclusion | 26 |
| 3. Writing Your First .Net 7 Program..... | 27 |
| Introduction | 27 |
| Structure..... | 27 |
| Creating a .Net 7 project \solution | 28 |
| Adding and configuring pages..... | 33 |
| <i>Changing view Index.cshtml</i> | 34 |
| <i>Navigate through the Solution Explorer</i> | 35 |
| <i>Use minimal APIs</i> | 36 |
| <i>Add pages and interact between pages</i> | 37 |
| Implementing the project | 41 |
| Conclusion..... | 43 |
| 4. Designing the Views..... | 45 |
| Introduction | 45 |
| Structure..... | 45 |
| Objectives..... | 46 |
| Creating the Views..... | 46 |
| MVC..... | 46 |
| Razor pages..... | 50 |

| | |
|--|-----------|
| <i>Creating a partial view</i> | 52 |
| MVC | 53 |
| <i>Razor pages</i> | 58 |
| Conclusion..... | 61 |
| 5. Creating Your Controllers..... | 63 |
| Introduction..... | 63 |
| Structure..... | 63 |
| Objectives..... | 64 |
| Creating a simple Controller..... | 64 |
| Integrating the Controller with the View | 67 |
| Creating and using a Model..... | 71 |
| <i>Adding a Model Class</i> | 72 |
| Conclusion..... | 76 |
| 6. Testing Your Views and Controllers | 77 |
| Introduction..... | 77 |
| Structure..... | 77 |
| Objectives..... | 78 |
| Unit tests | 78 |
| Performing tests and debug..... | 80 |
| Advantages of unit tests | 81 |
| Conclusion..... | 82 |
| 7. Working with .NET MAUI | 83 |
| Introduction..... | 83 |
| Structure..... | 84 |
| Objectives..... | 84 |
| .Net MAUI in a nutshell | 85 |
| New functionalities in .Net MAUI..... | 87 |
| <i>Maps</i> | 87 |
| <i>Desktop improvements</i> | 88 |

| | |
|---|------------|
| Using .Net MAUI in a .Net 7 application..... | 91 |
| Different libraries and platform specific frameworks..... | 96 |
| Conclusion..... | 98 |
| 8. Blazor in .NET 7..... | 99 |
| Introduction..... | 99 |
| Structure..... | 100 |
| Objectives..... | 100 |
| An introduction to Blazor in .Net 7..... | 100 |
| New in Blazor for .Net 7..... | 103 |
| A Bit on Blazor Hybrid..... | 105 |
| Sharing code and libraries..... | 106 |
| JavaScript Interop | 106 |
| Conclusion..... | 110 |
| 9. Creating a Desktop UI..... | 113 |
| Introduction..... | 113 |
| Structure..... | 114 |
| Objectives..... | 114 |
| Desktop UI..... | 114 |
| <i>WinUI on .Net MAUI.....</i> | <i>115</i> |
| <i>Other desktop applications - Win32</i> | <i>116</i> |
| <i>Other desktop applications – Windows Forms.....</i> | <i>117</i> |
| Creating a WinUI 3 project | 117 |
| Windows forms improvements in .Net 7 | 123 |
| Conclusion..... | 126 |
| 10. Communication with the Views..... | 127 |
| Introduction..... | 127 |
| Structure..... | 128 |
| Objectives..... | 128 |
| Ways of communicating with the views | 128 |

| | |
|---|------------|
| <i>How a controller calls a view</i> | 129 |
| <i>Pass data to views</i> | 131 |
| Model-View-Controller..... | 136 |
| Model-View-ViewModel (MVVM) | 138 |
| Conclusion..... | 139 |
| 11. Use SignalR | 141 |
| Introduction..... | 141 |
| Structure..... | 142 |
| Objectives..... | 142 |
| What is SignalR and how to use it | 142 |
| <i>How SignalR works in ASP.Net Core applications</i> | 143 |
| <i>The HubConnection class</i> | 144 |
| Advantages of SignalR..... | 146 |
| What can you do with ASP.Net and SignalR..... | 147 |
| Improvements in .Net 7 | 152 |
| Conclusion..... | 154 |
| 12. Adding a Database | 155 |
| Introduction..... | 155 |
| Structure..... | 156 |
| Objectives..... | 156 |
| <i>Adding the database</i> | 156 |
| Ways of using databases..... | 160 |
| Entity framework 7..... | 161 |
| New interceptors and events in EF | 163 |
| JSON columns | 164 |
| Custom database first templates | 166 |
| Conclusion..... | 166 |
| 13. Orleans | 167 |
| Introduction..... | 167 |

| | |
|--|----------------|
| Structure..... | 168 |
| Objectives..... | 168 |
| What is Orleans..... | 168 |
| How and where to use Orleans | 169 |
| What is new in Orleans in .Net 7..... | 179 |
| Conclusion..... | 182 |
| 14. Adding Specific Code Using System.Devices | 183 |
| Introduction..... | 183 |
| Structure..... | 183 |
| Objectives..... | 184 |
| System.Devices..... | 184 |
| Conclusion..... | 196 |
| 15. Possible Questions and Answers..... | 197 |
| Introduction..... | 197 |
| Structure..... | 198 |
| Objectives..... | 198 |
| Possible questions and answers | 198 |
| Conclusion..... | 204 |
| Index | 205-209 |

CHAPTER 1

Introduction to .Net 7

Introduction

Programming today is something that can be done in more simpler ways. The days where it is a huge mess and lots of trouble to make a simple program are now gone.

Microsoft created the .Net programming environment and the .Net Framework, which already contains several class libraries and provides language interoperability between several programming languages (C#, VB.net, F# and others, like C++, J#, Jscript.Net, IronPython or IronRuby).

This framework, or abstraction, contains generic functionalities that are executed on a CLR, providing user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. All of this can be combined in the source code that is created by programmers and software engineers, also with other libraries that are available.

It allows developers to create cloud apps, console apps, web apps, APIs, desktop apps, games, mobile apps, windows apps, machine learning, and IoT applications.

.Net has a large set of class libraries which facilitate these developments and can be used in this different project types. For example, we can use **System.Object** from which every .Net type derives, primitive types like **System.Boolean**, some collections such as **System.Collections.Generic.List<T>**, file and stream IO (**System.IO.FileStream**), JSON or XML serialization (**System.Text.Json.JsonSerializer** and **System.IO.TextWriter**). Also, if you need other different packages, you can use NuGet Package Manager to download and add them to your project.

Structure

In this chapter we will discuss following topics:

- A small piece of history on .Net
- Show some examples of the current release (.Net 6)
- Share a bit of the evolution of this platform

A small piece of .Net history

.Net Framework began its journey in late 1990s, under the name of Next Generation Windows Services. The first beta version of .Net 1.0 was released in early 2000. C# and the CLI were standardized in August 2000.

Two years later (in 2002), was the celebration of the first official .Net Framework release (which turns 20 years this year).

Later, in October 2007, the source code for .Net Framework 3.5 libraries was launched and became available online in January next year, including, BCL, ASP.Net, ADO.Net, Windows Forms, WPF and XML. *(This was one of the first frameworks on which I've worked!)*.

But things changed in 2014, with the introduction of .Net Core as a cross-platform and successor to .Net Framework. The name .Net Core was maintained until .Net Core 3.1 being replaced with only .Net upon the release of version 5, which followed 3.1 and skipped directly to 5 to avoid confusion with framework versions.

So, this .Net version is a core version and does not replace the framework, but includes several improvements and updates in C#, F# and Visual Basic, new JSON features, single file apps, annotation

on 80% of nullable reference types and improvements in JSON, Garbage Collection, RegEx, Async and many more areas.

Afterwards, in November 2021, another .Net Version came out: release 6. This was the final part of the unification plan that started with the earlier version. .Net 6 unifies SDK, base libraries, and runtimes across mobile, desktop, IoT and cloud apps. It also includes simplified development, better performance, and ultimate productivity.

Yet another very important news in .Net 6 is the introduction of a C# source generator to improve performance of applications that use **System.Text.Json**. It will help and simplify Json Serialization – and can be seen in detail over at Microsoft DevBlogs <https://devblogs.microsoft.com/dotnet/try-the-new-system-text-json-source-generator/>.

Other improvements of version 6 include: date, time and time zone improvements, new LINQ

APIs, Microsoft Extensions API, Reflection APIs, Generic Math and much more (all these functionalities and information are available on Microsoft documentation site <https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-6>).

One of the biggest improvements was a rewrite of **System.IO.FileStream** that now simplifies asynchronous I/O on Windows. This version also showed, as a preview, the new .NET MAUI (Multi-Platform App UI) – which will be covered ahead as it is included in .Net 7 – that makes possible to build native client apps for desktop and mobile OS with a single codebase (kind of a cooler upgraded version of and MVC simplifying multi-platform development).

And finally in February 2022 it was released Preview 1 of .Net 7

(<https://devblogs.microsoft.com/dotnet/asp-net-core-updates-in-net-7-preview-1/>), which is to be launched in November this year. In the next chapter it will be covered more thoroughly but the planned areas of focus on .Net 7 are: Performance, HTTP/3, Minimal APIs, SignalR, Razor, Blazor, gRPC, MVC and Orleans. Since then, much

more information was released and we will go through it in the following chapters.



Figure 1.1: .Net 7

Image downloaded from .Net blog

Examples of .Net 6

As seen earlier, .Net 6 implemented a few changes and updates.

Below will be some code samples from earlier versions *versus* .Net 6 (although I will be introducing .Net 7, it is still important to know some upgrades made on .Net 6).

I will give some examples of code made on earlier versions and the code that is now created and done in .Net 6. It will cover only some topics as this is just to understand what Microsoft is been working on and to show that everything is getting simpler and easier to work on these new versions.

Middleware integration

Middleware is something that sits between the requestor and the target and directly modify the response, log information, or modify the behaviour of code that generates the response. It can be used, for example, for logging requests.

In earlier versions it was used a Startup File, like the following code sample:

```
1. public class Startup
2. {
```

```

3.     public void Configure(IApplicationBuilder app=
4.     {
5.         app.UseStaticFiles();
6.     }
7. }

```

Since .Net 6 there is no need of this file and class, being all inserted directly without classes or namespaces in the program file:

```

1. var builder = WebApplication.CreateBuilder(args);
2. var app = builder.Build();
3.
4. app.UseStaticFiles();
5.
6. app.Run();

```

Routing

Routing is responsible for matching incoming HTTP requests and dispatching those requests to the endpoints. An application can have routing configured using Controllers, Razor Pages, SignalR, gRPC Services, Endpoint enabled middleware or delegates and lambdas that are registered with routing.

In earlier versions there was the need to call **UseEndpoints** or **UseRouting**:

```

1.     public class Startup
2.     {
3.         public void Configure(IApplicationBuilder app)
4.         {
5.             app.UseRouting();
6.             app.UseEndpoints(endpoints =>
7.             {
8.                 endpoints.MapRazorPages();
9.                 endpoints.MapDefaultControllerRoute();
10.            });
11.
12.        }
13.    }

```

Now, it is not used anymore, and can be done directly:

```
1. var builder = WebApplication.CreateBuilder(args);
2. var app = builder.Build();
3.
4. app.MapRazorPages();
5. app.MapDefaultControllerRoute();
6.
7. app.Run();
```

Add Services

To add memory cache services, custom scoped services, or other type of services you have to use Dependency Injection which allows an Inversion of Control between classes and their dependencies.

As in the previous examples, it was needed to use Startup and create a Configuration Method and call the services on it:

```
1. public class Startup
2. {
3.     public void ConfigureServices(IServiceCollection services)
4.     {
5.         //Add Memory Cache Services
6.         services.AddMemoryCache();
7.
8.         //Add Custom Scoped Services
9.         services.AddScoped<ITestService, TestService>();
10.        services.
            AddScoped<ITestRepository, TestRepository>();
11.    }
12. }
```