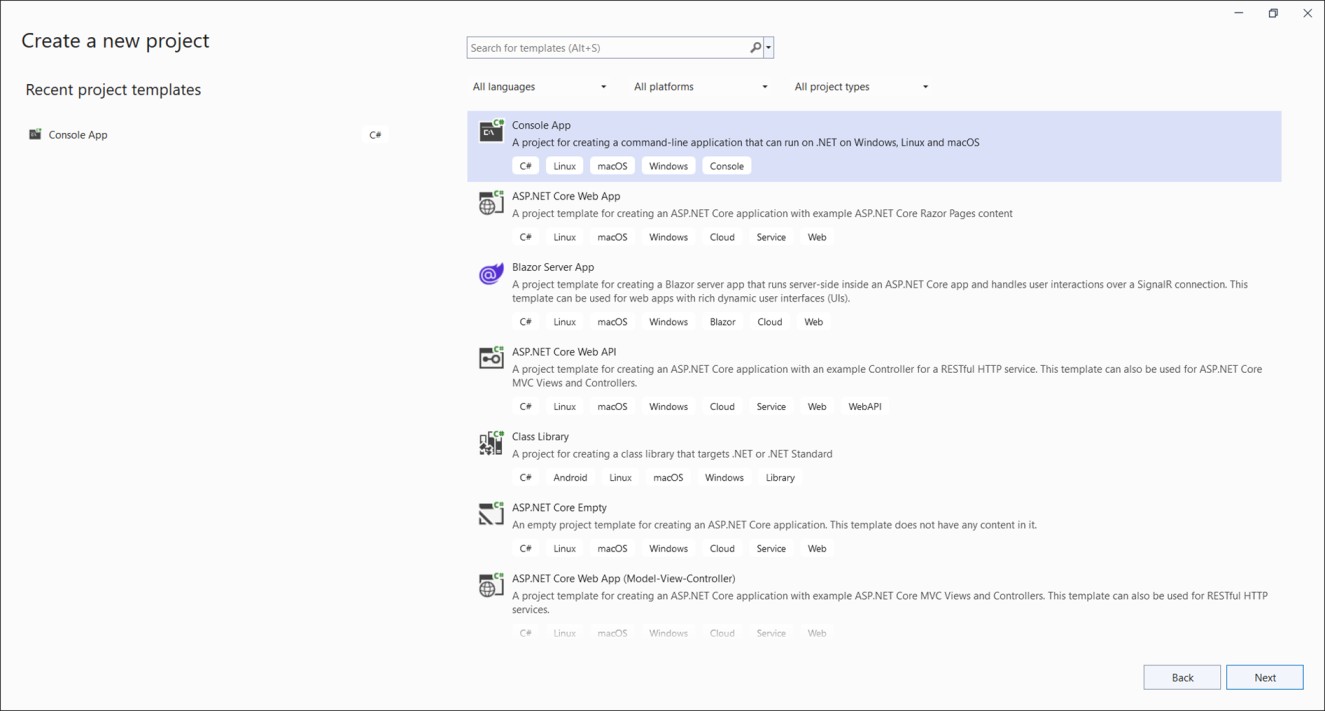
**Efficient HTML to PDF Conversion with Multithreading in .NET Core Using C#**

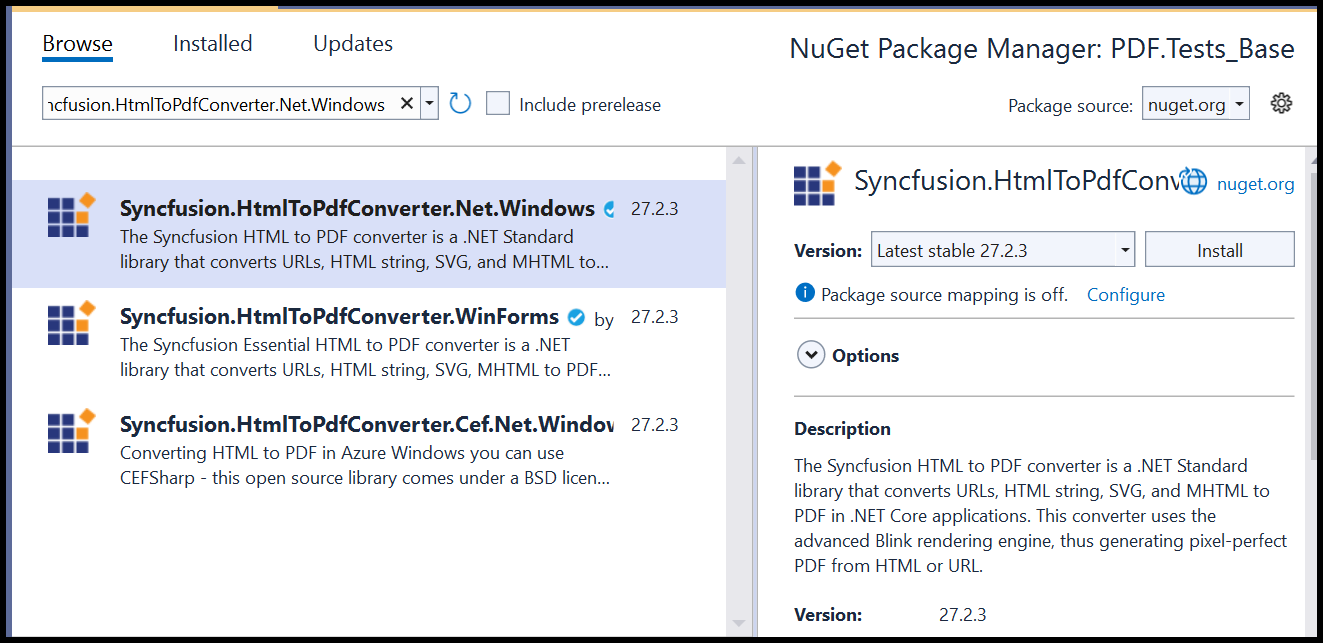
The Syncfusion® [HTML-to-PDF converter](https://www.syncfusion.com/document-processing/pdf-framework/net/html-to-pdf) is a .NET PDF library designed for converting webpages, SVG, MHTML, and HTML files to PDF using C#. It leverages the Blink rendering engine (used by Google Chrome) to ensure high-fidelity rendering. The converter delivers reliable and accurate results, preserving all graphics, images, text, fonts, and the original layout of the HTML document or webpage.

This guide outlines how to optimize performance and how to handle large-scale HTML conversion tasks using **Parallel.ForEach** method. It ensures efficient CPU utilization and significantly reduces conversion times.

**Steps for Implementing Multithreaded HTML to PDF Conversion:**

1. Create a New Project: Start a new Console application in .NET Core to facilitate the HTML-to-PDF conversion process. 

2. Install Required Packages: Add the [Syncfusion.HtmlToPdfConverter.Net.Windows](https://www.nuget.org/packages/Syncfusion.HtmlToPdfConverter.Net.Windows) NuGet package from [NuGet.org](https://www.nuget.org/) to your project.



3. **Set Up Your Environment**: In the Program.cs file, include these namespaces.  
**C#**

|  |
| --- |
| **using** Syncfusion.HtmlConverter;  **using** Syncfusion.Pdf;  **using** Syncfusion.Drawing;  **using** System.Threading.Tasks; |

4.**Implement the Conversion Logic with Multithreading**: Use the following code sample to perform parallel HTML-to-PDF conversions.

**C#**

|  |
| --- |
| class Program  {  static void Main()  {  string text = System.IO.File.ReadAllText(Path.GetFullPath("page1.html"));  text += System.IO.File.ReadAllText(Path.GetFullPath("page2.html"));  IEnumerable<int> works = Enumerable.Range(0, 100);  Parallel.ForEach(works, index => ConvertHTMLPDF(text));  Console.WriteLine("PDF Conversion completed.");  }  static byte[] ConvertHTMLPDF(string html)  {  BlinkConverterSettings blinkConverterSettings = new BlinkConverterSettings()  {  PdfPageSize = PdfPageSize.A4,  ViewPortSize = new Syncfusion.Drawing.Size(800, 1200),  };  HtmlToPdfConverter htmlConverter = new HtmlToPdfConverter()  {  ConverterSettings = blinkConverterSettings  };  using (var document = htmlConverter.Convert(html, string.Empty))  {  using (var stream = new MemoryStream())  {  document.Save(stream);  string outputName = Guid.NewGuid().ToString();  System.IO.File.WriteAllBytes("Output " + outputName + ".pdf", stream.ToArray());  return stream.ToArray();  }  }  }  } |

A complete working sample can be downloaded from [**HTML-to-PDF-Multithreading**](https://www.syncfusion.com/downloads/support/directtrac/general/ze/HTML-to-PDF-Multithreading_%282%292044108428).

**Conclusion**  
I hope you enjoyed learning on how to convert HTML to PDF with multithreading using C# in .NET Core.

You can refer to our [**ASP.NET Core PDF’s feature tour**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) page to know about its other groundbreaking feature representations. You can also explore our [**ASP.NET Core PDF example**](https://ej2.syncfusion.com/aspnetcore/PDF/Default#/material) to understand how to present and manipulate data.

For current customers, you can check out our [**ASP.NET**](http://asp.net/) Core components from the [**License and Downloads**](https://www.syncfusion.com/account/downloads) page. If you are new to Syncfusion®, you can try our 30-day [**free trial**](https://www.syncfusion.com/downloads/aspnetcore-js2) to check out our [**ASP.NET**](http://asp.net/) Core PDF and other [**ASP.NET**](http://asp.net/) Core components.

If you have any queries or require clarifications, please let us know in comments below. You can also contact us through our [**support forums**](https://www.syncfusion.com/forums), [**Direct-Trac**](https://www.syncfusion.com/support/directtrac/incidents/), or [**feedback portal**](https://www.syncfusion.com/feedback/aspnet-core?control=pdf). We are always happy to assist you!