**Create PDF Files in Blazor Using C# with Syncfusion**

The [Syncfusion .NET Core library](https://www.syncfusion.com/document-processing/pdf-framework/net-core/pdf-library), generating PDF documents in the Blazor framework becomes straightforward and efficient. This guide walks you through the process of creating PDF files from scratch in a Blazor application using C#. Follow these steps to integrate PDF functionalities into your Blazor projects.

**Prerequisites:**

* Visual Studio 2022
* .NET Core SDK 8.0: Download from [here](https://dotnet.microsoft.com/en-us/download/dotnet/8.0)

**Creating a Blazor PDF Project**:

1.Enable Visual Studio Preview SDKs:

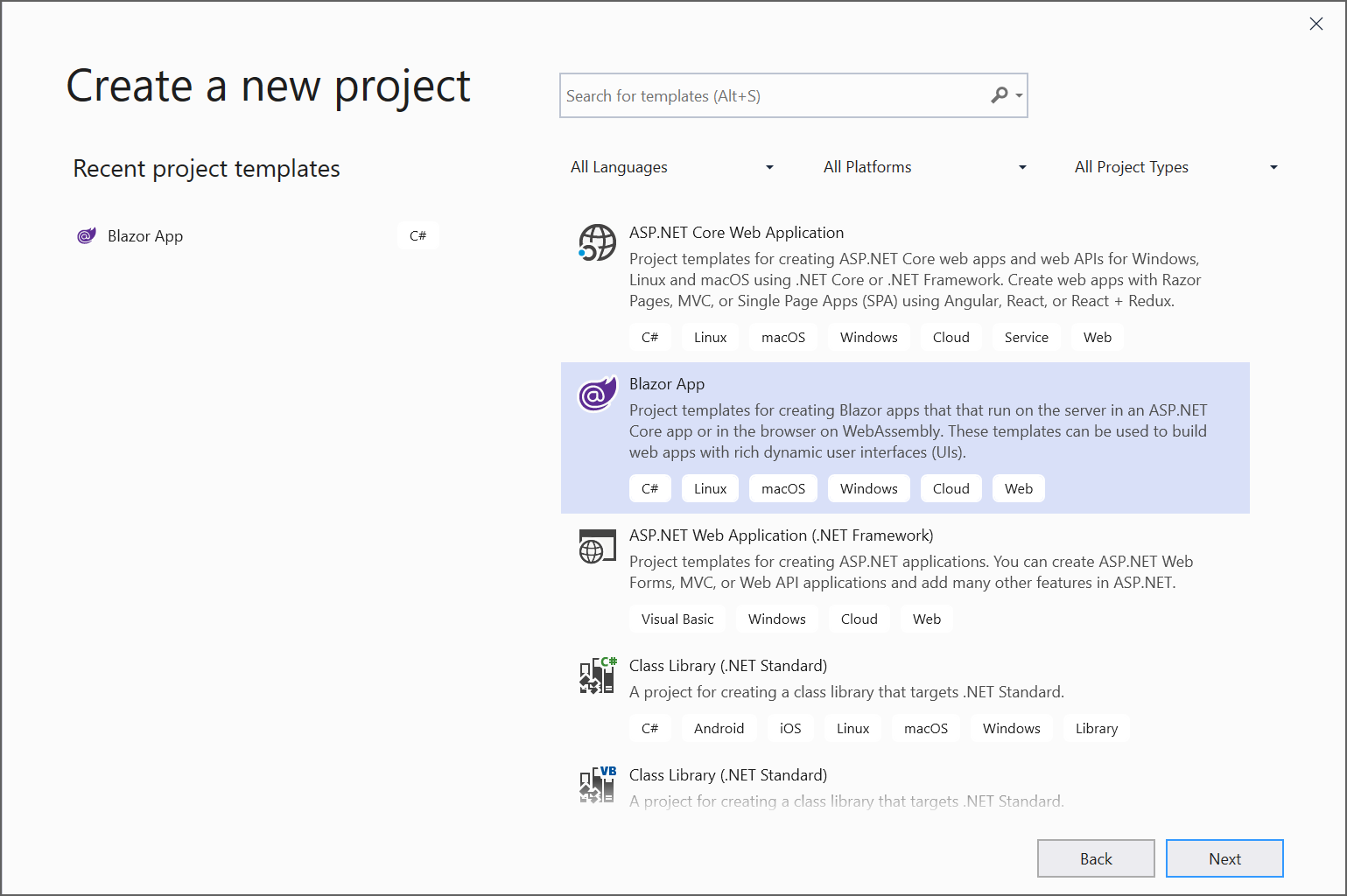
\* Navigate to **Tools > Options**.

\* Go to **Projects and Solutions > .NET** Core tab.

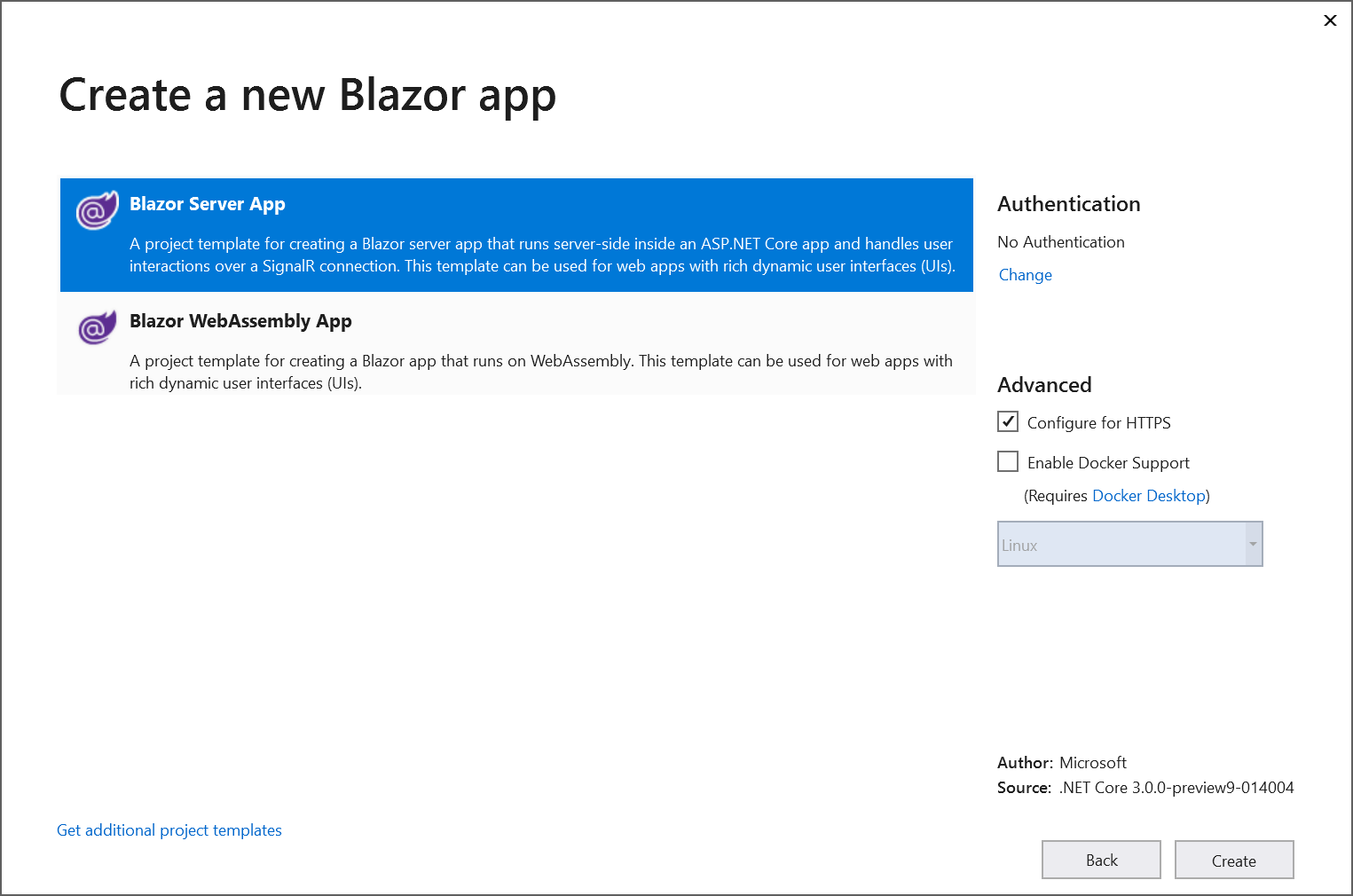
\* Check Use previews of the **.NET Core SDK** and click **OK**.

\* Restart Visual Studio 2022.

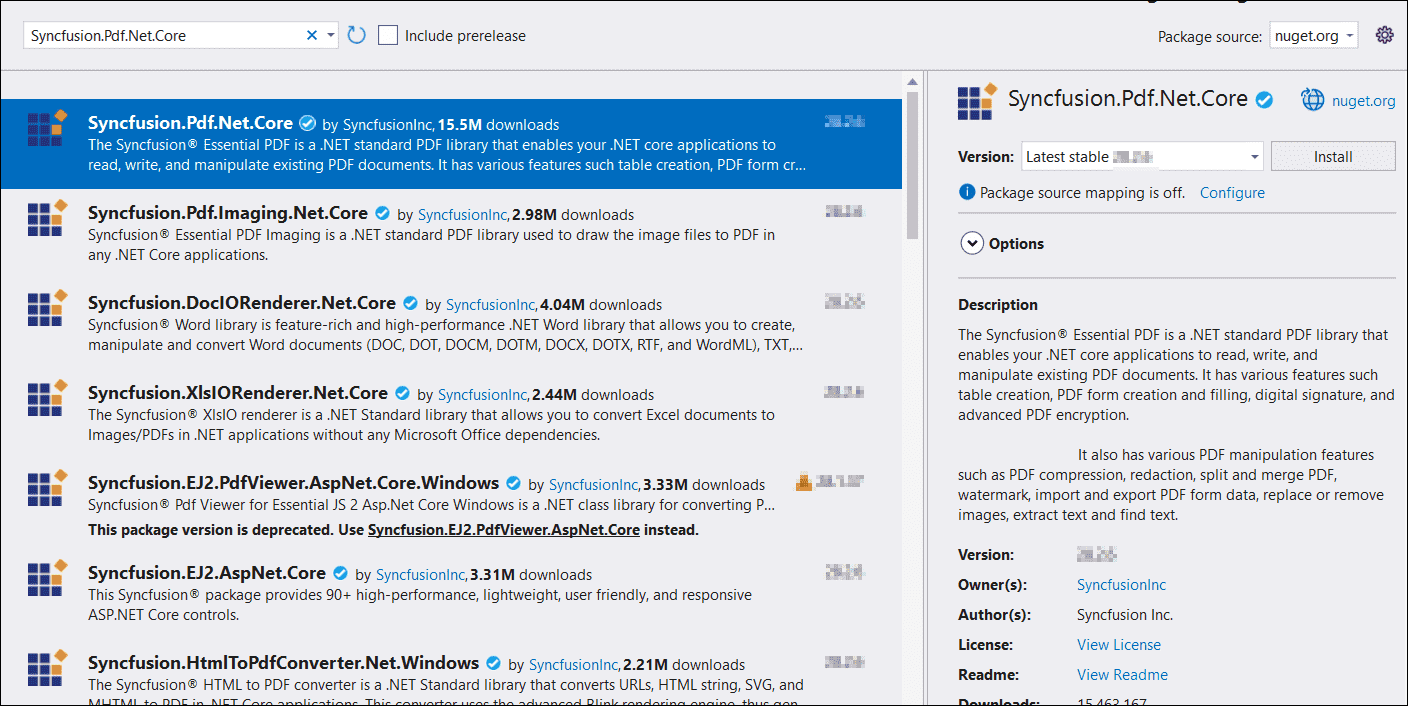
2. **Initialize a Blazor Project**: Select Blazor App, then click Next.



\*Select **Blazor Server** **APP**.



2. **Install Package**: Add the [Syncfusion.PDF.Net.Core](https://www.nuget.org/packages/Syncfusion.Pdf.Net.Core/) package to your Blazor project.



3.Add the following namespace in the **Index.razor** to create a PDF document .

|  |
| --- |
| @using Syncfusion.Pdf;  @using Syncfusion.Pdf.Graphics;  @using System.IO;  @inject Microsoft.JSInterop.IJSRuntime JS |

4.Add a button and hook the click event function.

|  |
| --- |
| <button class="btn btn-primary" @onclick="@CreatePDF">Create PDF</button> |

5.Add the following code to create a PDF file in Blazor.

|  |
| --- |
| @code {  void CreatePDF()  {  // Create a new PDF document  PdfDocument document = new PdfDocument();  // Add a page to the document  PdfPage page = document.Pages.Add();  // Create PDF graphics for the page  PdfGraphics graphics = page.Graphics;  // Set the standard font  PdfFont font = new PdfStandardFont(PdfFontFamily.Helvetica, 20);  // Draw the text  graphics.DrawString("Hello World!", font, PdfBrushes.Black, new Syncfusion.Drawing.PointF(0, 0));  // Save the PDF to a MemoryStream  MemoryStream stream = new MemoryStream();  document.Save(stream);  document.Close(true);  // Download the PDF in the browser  JS.SaveAs("Sample.pdf", stream.ToArray());  }  } |

6.**Save PDF to Browser**: In **FileUtil** Class

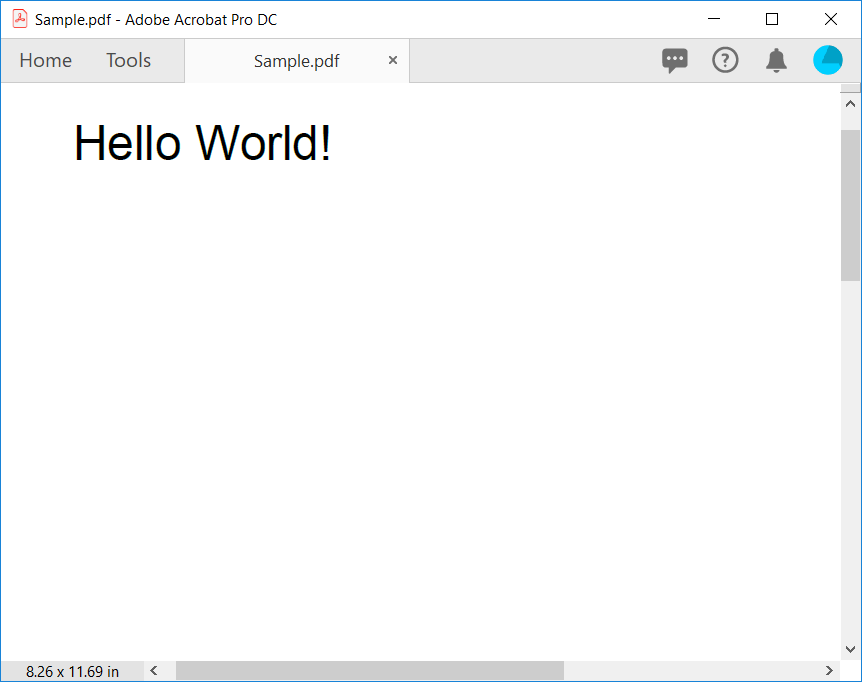
|  |
| --- |
| public static class FileUtil  {  public static ValueTask<object> SaveAs(this IJSRuntime js, string filename, byte[] data)  => js.InvokeAsync<object>("saveAsFile", filename, Convert.ToBase64String(data));  } |

7. Add JavaScript in **\_Host.cshtml**

|  |
| --- |
| <script type="text/javascript">  function saveAsFile(filename, bytesBase64) {  var link = document.createElement('a');  link.download = filename;  link.href = "data:application/octet-stream;base64," + bytesBase64;  document.body.appendChild(link);  link.click();  document.body.removeChild(link);  }  </script> |

A complete working sample is available for [**BlazorPDFGettingStarted.zip**](https://www.syncfusion.com/downloads/support/directtrac/general/ze/BlazorPDFGettingStarted-1007429454)

By executing the program, the output PDF document will be generated as shown below.



Take a moment to peruse the [**documentation**](https://help.syncfusion.com/file-formats/pdf/working-with-text), where you can find other options like drawing right-to-left text and multi-column text, consuming TrueType fonts, Standard fonts, and CJK fonts. Also, the features like [**PDF form filling**](https://help.syncfusion.com/file-formats/pdf/working-with-forms), convert [**HTML to PDF**](https://help.syncfusion.com/file-formats/pdf/convert-html-to-pdf/webkit) , and [**protect PDF documents**](https://help.syncfusion.com/file-formats/pdf/working-with-security) with code examples.

Take a moment to explore the [documentation](https://help.syncfusion.com/file-formats/pdf/working-with-text), where you’ll find additional options such as drawing right-to-left and multi-column text, using TrueType, Standard, and CJK fonts. You’ll also discover features like [PDF form filling](https://help.syncfusion.com/file-formats/pdf/working-with-forms), [HTML to PDF](https://help.syncfusion.com/document-processing/pdf/conversions/html-to-pdf/net/features) conversion, and [PDF document protection](https://help.syncfusion.com/file-formats/pdf/working-with-security).

**Conclusion**

I hope you enjoyed learning about how to create a PDF file in Blazor using C#.

You can refer to our [**.NET Core PDF**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) [**feature tour**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) page to know about its other groundbreaking feature representations and [**documentation**](https://help.syncfusion.com/file-formats/pdf/create-pdf-file-in-asp-net-core), and how to quickly get started for configuration specifications. You can also explore our [**.NET Core PDF**](https://ej2.syncfusion.com/aspnetcore/PDF/Default?_gl=1*oxrtb5*_ga*MjkzODA3NDIuMTY4MjQwOTYyOA..*_ga_WC4JKKPHH0*MTY4ODQ0ODEzOS4xODQuMS4xNjg4NDUwNjc2LjI5LjAuMA..*_ga_2QTHE2Y2YX*MTY4ODQ0ODEzOS44LjEuMTY4ODQ1MDY3Ni4yOS4wLjA.&_ga=2.48642854.1072980492.1688360769-29380742.1682409628#/bootstrap5) [**example**](https://ej2.syncfusion.com/aspnetcore/PDF/Default#/bootstrap5) to understand how to create and manipulate data.

For current customers, you can check out our components from the [**License and Downloads**](https://www.syncfusion.com/sales/teamlicense) 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 other controls.

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