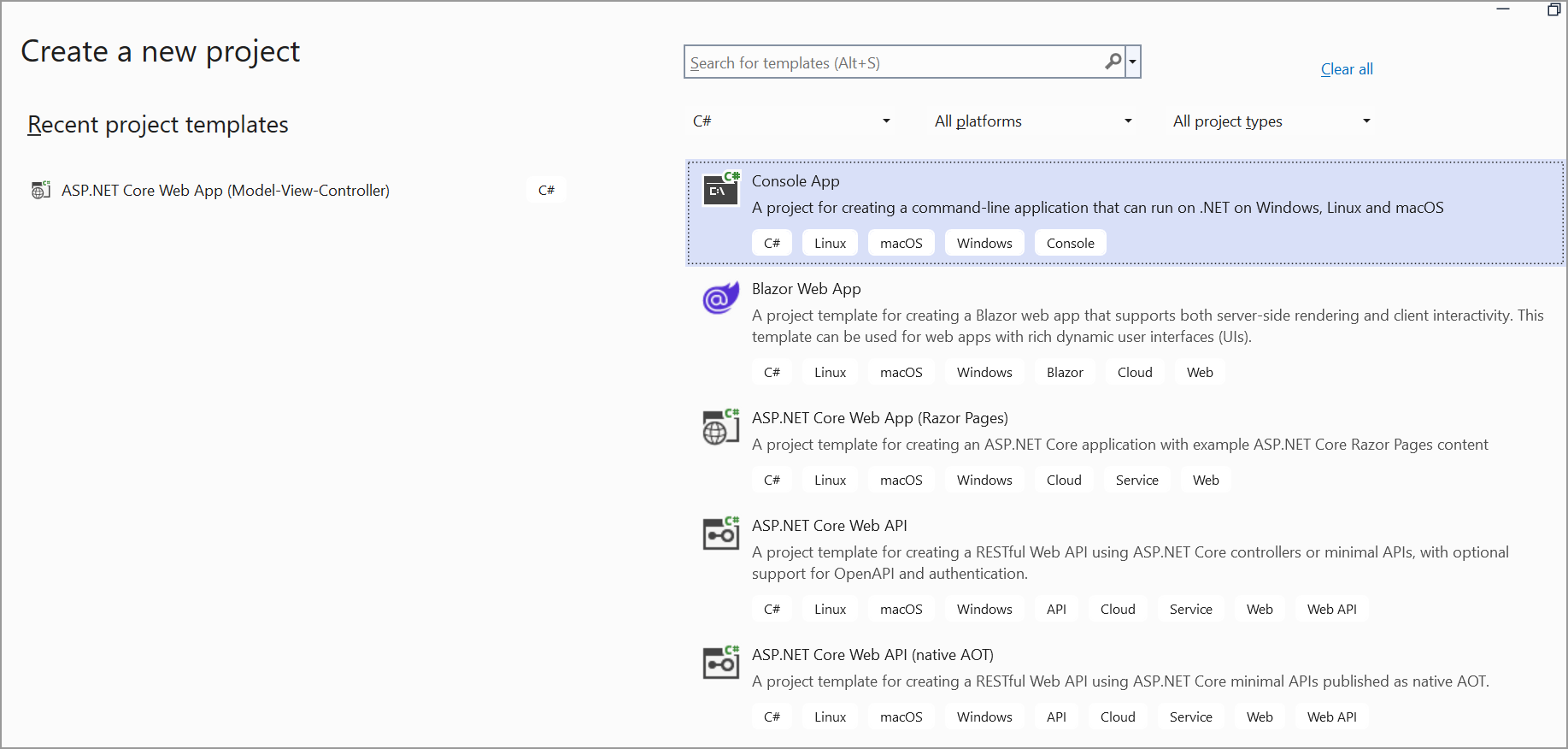
**How to Add RTL Languages When Converting HTML to PDF in .NET Core**

Our Syncfusion® [**HTML-to-PDF converter**](https://www.syncfusion.com/document-processing/pdf-framework/net/html-to-pdf) is a .NET PDF library for converting webpages, SVG MHTML and HTML files to PDF using C#. It uses the popular rendering engine Blink (Google Chrome). It is reliable and accurate. The result preserves all graphics, images, text, fonts, and the layout of the original HTML document or webpage.  
Using this library, you can add the Arabic text in the TOC of the PDF when performing HTML to PDF conversion. We used arial-unicode-ms font that supports Arabic text, you can specify such font under PdfTrueTypeFont class and assign it to HtmlToPdfTocStyle. However, if you want to use a different font that supports Arabic text then you can replace it.

**Steps to adding multi-language support, including Arabic, for TOC in HTML to PDF conversion**

1. Create a new console application project.
2. Install the **[Syncfusion.HtmlToPdfConverter.Net.Windows](https://www.nuget.org/packages?q=Syncfusion.HtmlToPdfConverter.Net.Windows" \t "_blank)** NuGet package as a reference to your console application from [**Nuget.org**](https://www.nuget.org/).

A screenshot of a computer

AI-generated content may be incorrect.

1. Include the following namespaces in the Program.cs file.

**C#**

**using** Syncfusion.Drawing;

**using** Syncfusion.HtmlConverter;

**using** Syncfusion.Pdf;

**using** Syncfusion.Pdf.Graphics;

**using** Syncfusion.Pdf.HtmlToPdf;

1. Use the following code sample in Program.cs to add multi-language support, including Arabic, for TOC in HTML to PDF conversion

**C#**

//Initialize the HTML to PDF converter.

HtmlToPdfConverter htmlConverter = **new** HtmlToPdfConverter();

//Initialize blink converter settings.

BlinkConverterSettings settings = **new** BlinkConverterSettings();

//Enable TOC.

settings.EnableToc = true;

//Set the style for level 1(H1) items in table of contents.

HtmlToPdfTocStyle tocstyleH1 = **new** HtmlToPdfTocStyle();

// Load the Arabic font from a file

PdfFont arabicFont = **new** PdfTrueTypeFont(@"Data/arial-unicode-ms.ttf", 10, PdfFontStyle.Regular);

// Set the TOC style with the Arabic font

tocstyleH1.Font = arabicFont;

// Set the background color

tocstyleH1.BackgroundColor = **new** PdfSolidBrush(**new** PdfColor(Color.FromArgb(68, 114, 196)));

// Set the foreground color

tocstyleH1.ForeColor = PdfBrushes.White;

// Set the padding

tocstyleH1.Padding = **new** PdfPaddings(5, 5, 3, 3);

settings.Toc.SetItemStyle(1, tocstyleH1);

settings.Toc.SetItemStyle(2, tocstyleH1);

settings.Toc.SetItemStyle(3, tocstyleH1);

settings.Toc.SetItemStyle(4, tocstyleH1);

//Assign Blink converter settings to HTML converter.

htmlConverter.ConverterSettings = settings;

//Convert HTML to PDF document.

PdfDocument document = htmlConverter.Convert(Path.GetFullPath(@"Data/input1.html"));

//Create a file stream to save the PDF document.

**using** (FileStream fileStream = **new** FileStream(Path.GetFullPath(@"Output/HTML-to-PDF.pdf"), FileMode.Create, FileAccess.ReadWrite))

{

//Save the PDF document

document.Save(fileStream);

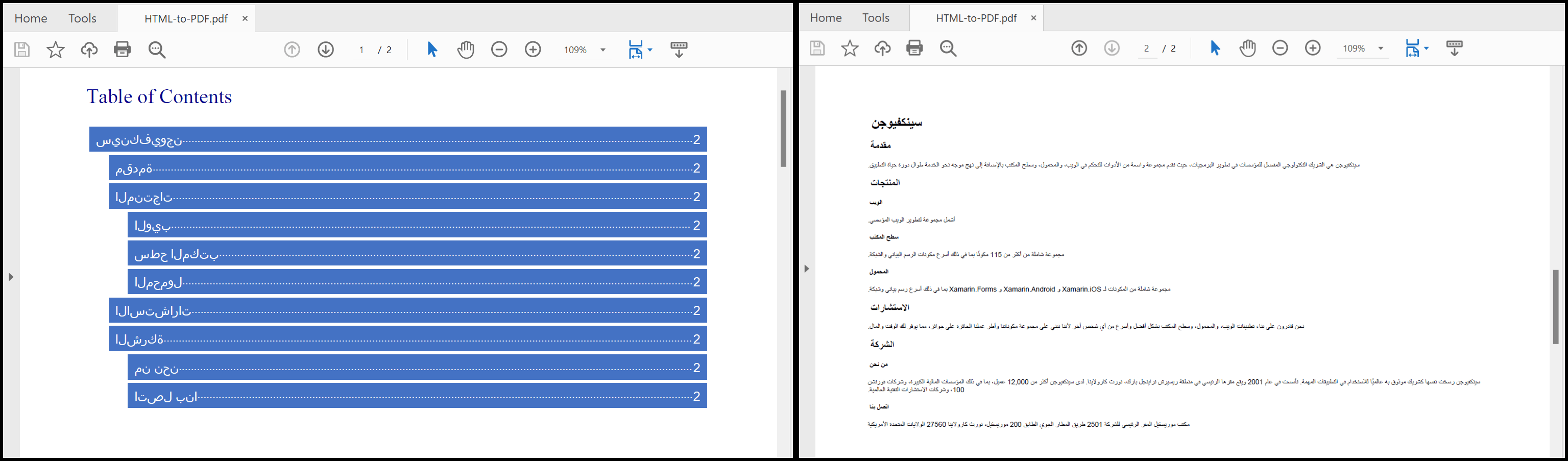
}

//Close the PDF document

document.Close(true);

A complete working sample can be downloaded from [**HTML-to-PDF-TOC-Arabic**](https://github.com/SyncfusionExamples/PDF-Examples/tree/master/HTML%20to%20PDF/Blink/HTML-to-PDF-TOC-Arabic-Text/.NET)

By executing the program, you will get the PDF document as follows.



Take a moment to peruse the [**documentation**](https://help.syncfusion.com/document-processing/pdf/conversions/html-to-pdf/net/features#table-of-contents-with-custom-style), where you can find converting HTML pages to PDF document with custom style of TOC.

You can use fonts that support other languages beyond Arabic for the Table of Contents (TOC) in your HTML to PDF conversion. Please ensure that the chosen font fully supports the required language characters to avoid issues in rendering.

**Conclusion**  
I hope you enjoyed learning about how to add RTL languages when converting HTML to PDF in .NET Core.

You can refer to our [**ASP.NET Core PDF 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 [**ASP.NET Core PDF 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!