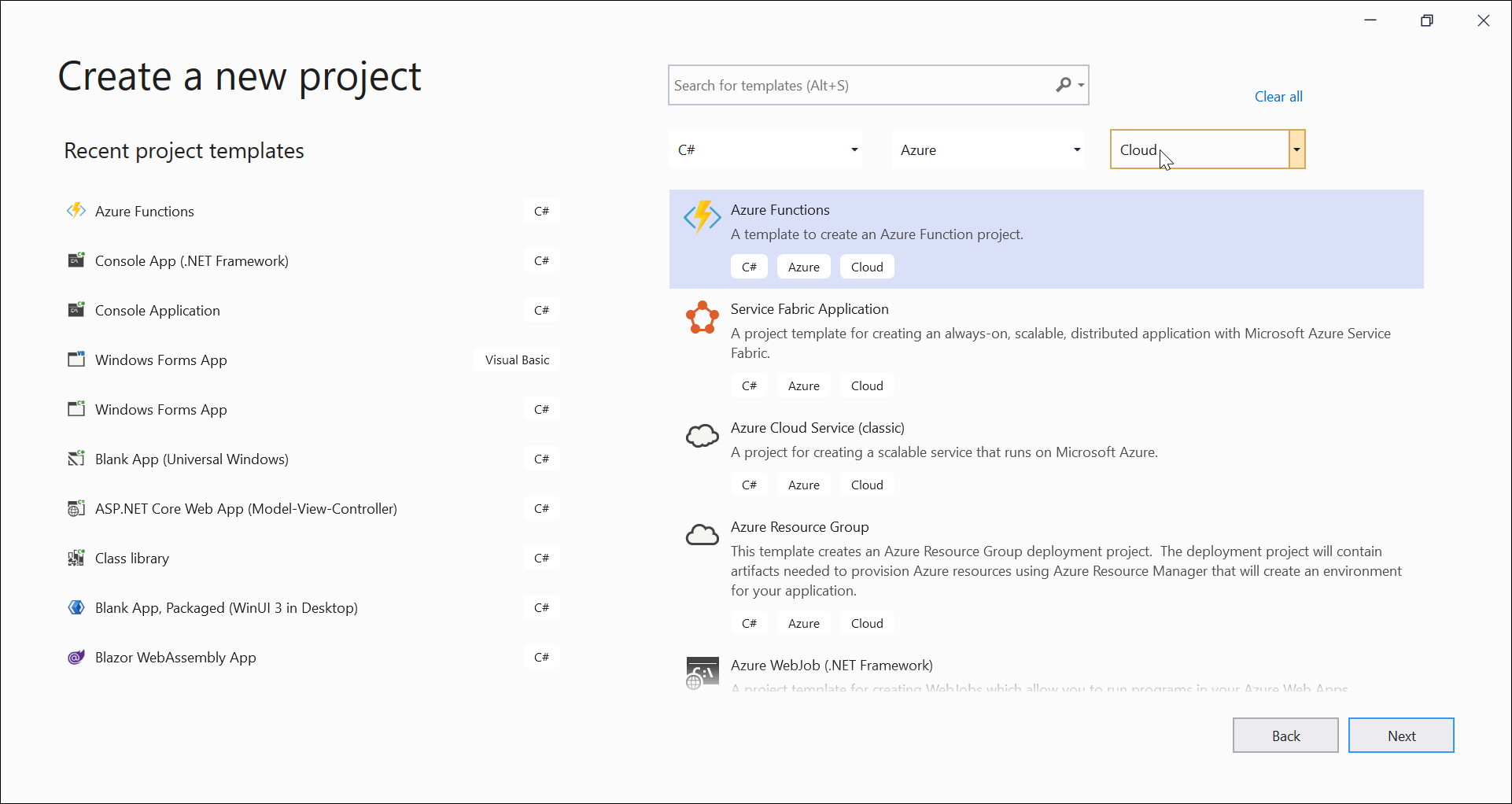
**Convert HTML to PDF in Azure Functions 4.0 Using ASP.NET Core**

The Syncfusion® [**HTML to PDF converter**](https://www.syncfusion.com/document-processing/pdf-framework/net/html-to-pdf) for [**ASP.NET Core**](https://www.syncfusion.com/document-processing/pdf-framework/net) to efficiently transform web pages, URLs, or HTML strings into PDFs using Azure Functions 4.0 with .NET 8.

Our tutorial demonstrates using the advanced WebKit rendering engine for robust and platform-agnostic PDF conversions, supporting environments like Azure Cloud, AWS, Docker, and more.

**Steps to convert HTML to PDF in Azure Functions 4.0**

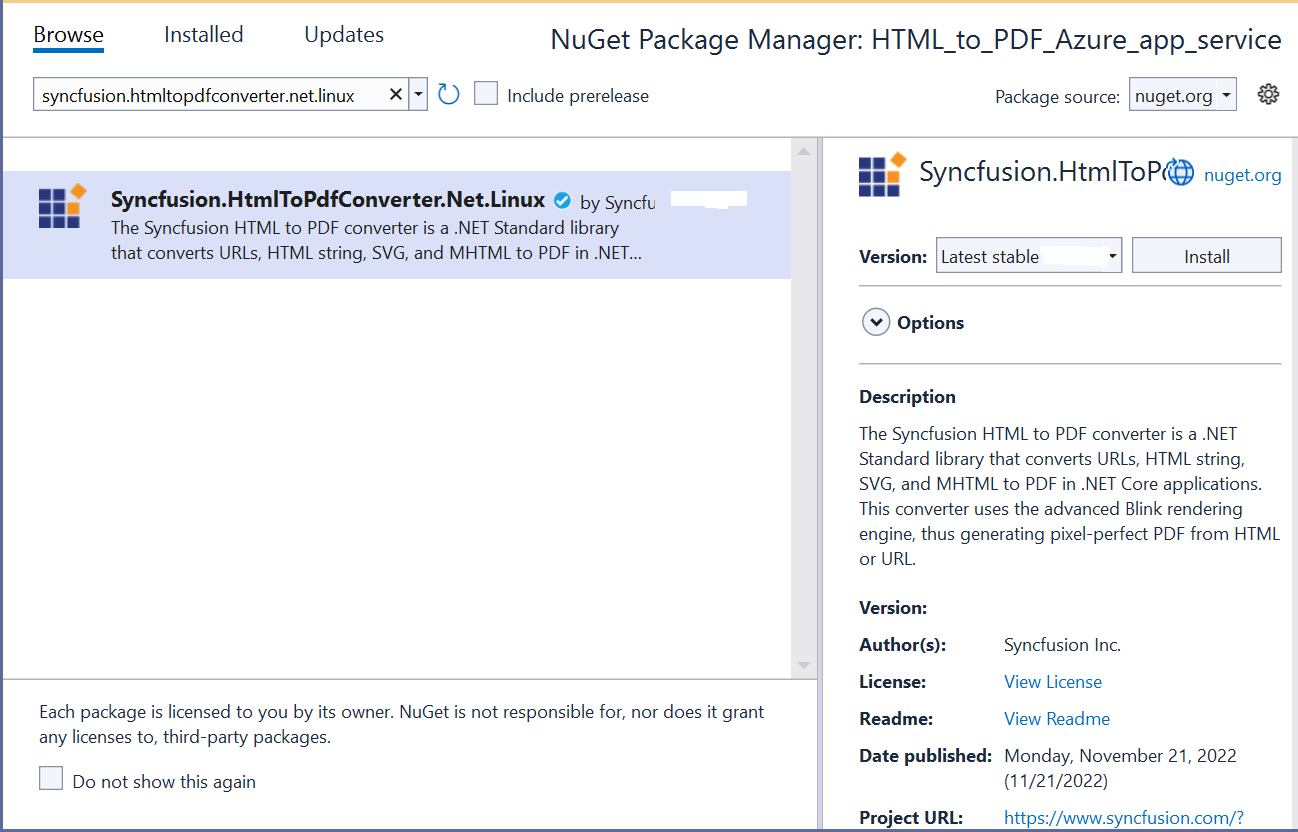
1. **Set Up Azure Functions Project**: Create an Azure Functions project.

2. Target Azure Functions V4 (.NET 8) and select HTTP triggers. A screenshot of a computer

AI-generated content may be incorrect.

3. **Install Required Package**: Add [Syncfusion.HtmlToPdfConverter.Net.Linux](https://www.nuget.org/packages/Syncfusion.HtmlToPdfConverter.Net.Linux/) from [NuGet](https://www.nuget.org/)

to your project.



3. **Include Necessary Namespaces**: Add the following in your **Function1.cs** file

**C#**

|  |
| --- |
| **using** Syncfusion.HtmlConverter;  **using** Syncfusion.Pdf;  **using** System.Runtime.InteropServices; |

4. **Implement HTML to PDF Conversion**: Add the code for conversion in the **Function1** class.

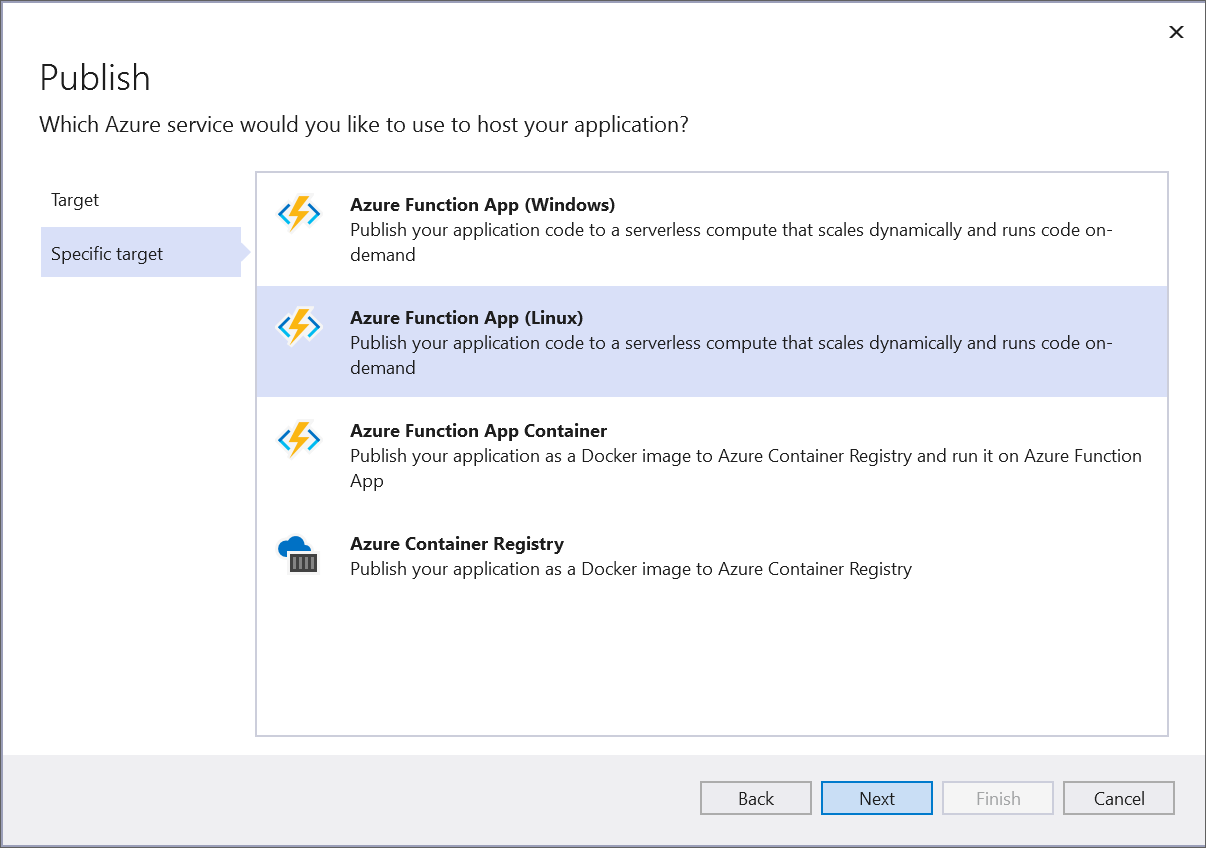
|  |
| --- |
| [FunctionName("Function1")]  public static async Task<IActionResult> Run([HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req, ILogger log, ExecutionContext executionContext)  {  // Initialize Blink binaries path  string blinkBinariesPath = SetupBlinkBinaries(executionContext);  // Retrieve URL from query parameters  string url = req.Query["url"];  // Set up HTML to PDF converter with Blink rendering engine  HtmlToPdfConverter htmlConverter = new HtmlToPdfConverter(HtmlRenderingEngine.Blink);  BlinkConverterSettings settings = new BlinkConverterSettings();  settings.CommandLineArguments.Add("--no-sandbox");  settings.CommandLineArguments.Add("--disable-setuid-sandbox");  settings.BlinkPath = blinkBinariesPath;  htmlConverter.ConverterSettings = settings;  // Convert the URL to PDF  PdfDocument document = htmlConverter.Convert(url);  MemoryStream ms = new MemoryStream();  document.Save(ms);  document.Close();  ms.Position = 0;  // Return the PDF as a FileStreamResult  return new FileStreamResult(ms, "application/pdf");  } |

**5.Helper Methods for Blink Binaries**: Set up Blink binaries and permissions

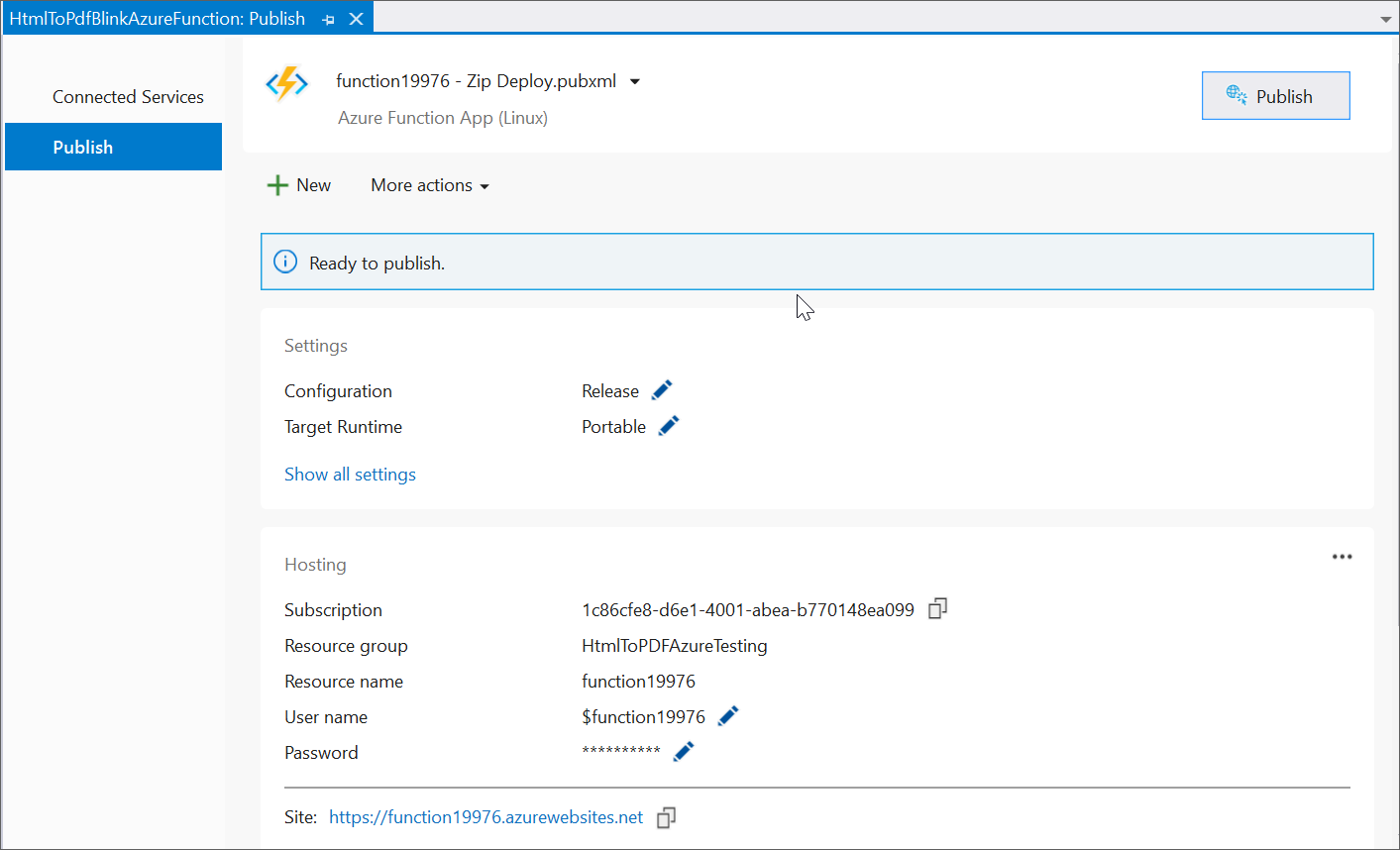
|  |
| --- |
| private static string SetupBlinkBinaries(ExecutionContext executionContext)  {  string blinkAppDir = Path.Combine(executionContext.FunctionAppDirectory, "BlinkBinariesLinux");  string tempBlinkDir = Path.GetTempPath();  string chromePath = Path.Combine(tempBlinkDir, "chrome");  if (!File.Exists(chromePath))  {  CopyFilesRecursively(blinkAppDir, tempBlinkDir);  SetExecutablePermission(tempBlinkDir);  }  return tempBlinkDir;  }  private static void CopyFilesRecursively(string sourcePath, string targetPath)  {  foreach (string dirPath in Directory.GetDirectories(sourcePath, "\*", SearchOption.AllDirectories))  {  Directory.CreateDirectory(dirPath.Replace(sourcePath, targetPath));  }  foreach (string newPath in Directory.GetFiles(sourcePath, "\*.\*", SearchOption.AllDirectories))  {  File.Copy(newPath, newPath.Replace(sourcePath, targetPath), true);  }  }  private static void SetExecutablePermission(string tempBlinkDir)  {  FileAccessPermissions ExecutableFilePermissions = FileAccessPermissions.UserRead | FileAccessPermissions.UserWrite | FileAccessPermissions.UserExecute |  FileAccessPermissions.GroupRead | FileAccessPermissions.GroupExecute | FileAccessPermissions.OtherRead | FileAccessPermissions.OtherExecute;  foreach (string executable in new string[] { "chrome", "chrome\_sandbox" })  {  var execPath = Path.Combine(tempBlinkDir, executable);  if (File.Exists(execPath))  {  var code = Function1.Chmod(execPath, ExecutableFilePermissions);  if (code != 0)  {  throw new Exception("Chmod operation failed");  }  }  }  }  [Flags]  internal enum FileAccessPermissions : uint  {  OtherExecute = 1,  OtherWrite = 2,  OtherRead = 4,  GroupExecute = 8,  GroupWrite = 16,  GroupRead = 32,  UserExecute = 64,  UserWrite = 128,  UserRead = 256  } |

**Public to Azure Functions Linux:**

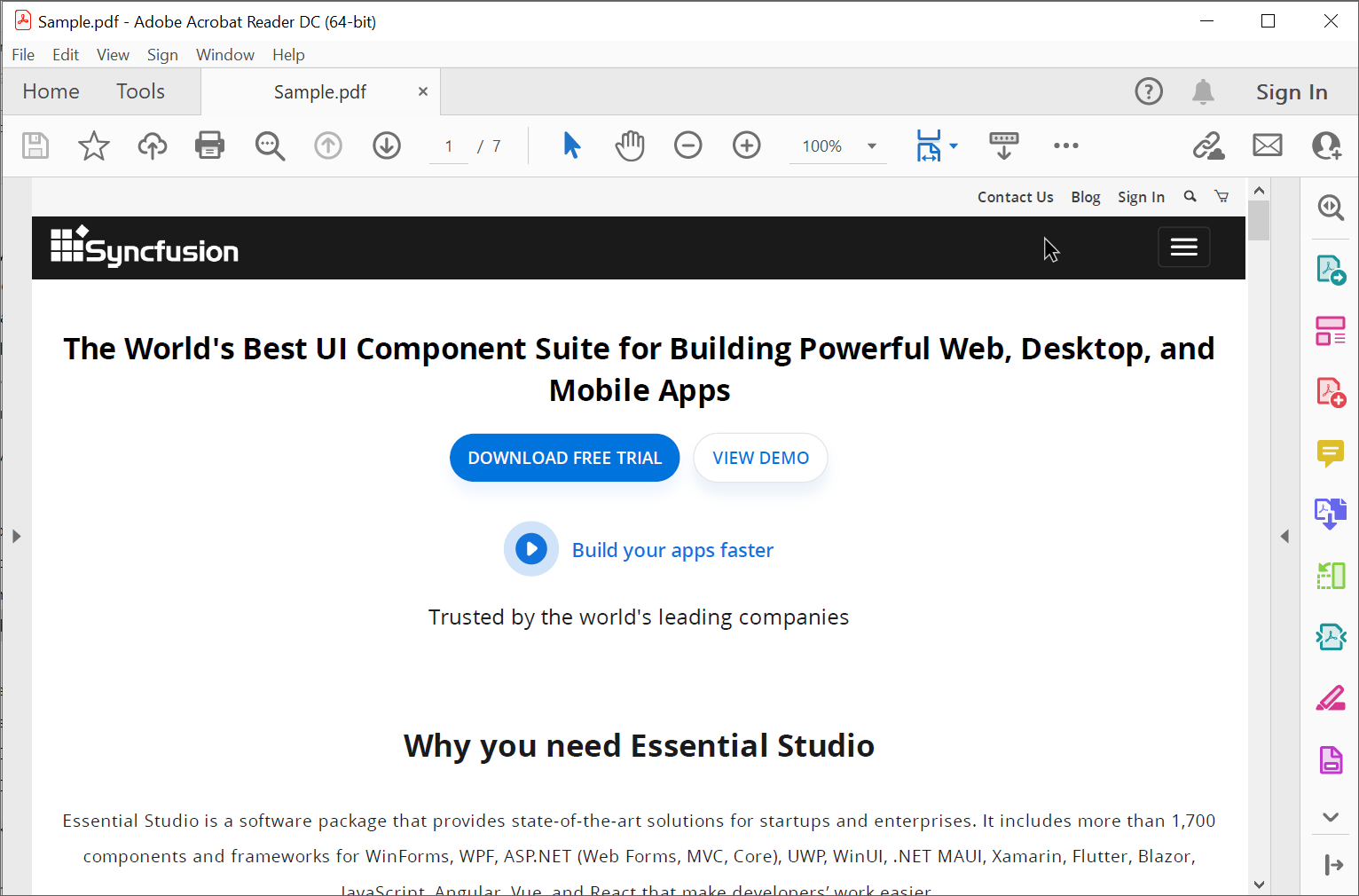
6. Right-click the project, select **Publish**, create a new profile in the **Publish Window**, and then create the **Azure Function App service** with a **consumption plan**, as the Blink rendering engine works in this plan.



7.After creating the profile, click the Publish button.



8. Go to the **Azure portal**, select **App Services**, click **Get function URL**, copy the URL, include it as a query string, and paste it into a new browser tab to get the PDF document as shown below.



A complete working sample is available for download:[HtmlToPdf\_AzureFunctions4.0.zip](https://www.syncfusion.com/downloads/support/directtrac/general/ze/HTMLToPDF_AzureFunctions4.01738776014).

Take a moment to review the[**documentation**](https://help.syncfusion.com/file-formats/pdf/converting-html-to-pdf?_ga=2.177406373.1281575720.1636574891-1699463946.1628662123) to learn more about converting HTML pages to a PDF document, along with the available customization options and features.  
Click [here](https://www.syncfusion.com/document-processing/pdf-framework/net/html-to-pdf) to explore the rich set of **Syncfusion Essential® PDF** features.  
You can also view an **online sample** for converting HTML to PDF [here](https://ej2.syncfusion.com/aspnetcore/PDF/HtmltoPDF#/bootstrap5).

**Conclusion**

I hope you enjoyed learning about how to convert HTML to PDF in Azure Functions 4.0.

You can refer to our [**ASP.NET Core PDF's**](https://www.syncfusion.com/document-processing/pdf-framework/net) feature tour page to know about its other groundbreaking feature representations and [**documentation**](https://help.syncfusion.com/aspnet-core/pdf/getting-started), and how to quickly get started for configuration specifications. You can also explore our [**ASP.NET PDF example**](https://ej2.syncfusion.com/net/PDF/Default?#/bootstrap5) to understand how to create and manipulate data.

For current customers, you can check out our [**Document Processing Libraries**](https://www.syncfusion.com/document-processing-libraries) 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 to check out our 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**](https://www.syncfusion.com/feedback) portal. We are always happy to assist you!