IronPdf Guide with .NET Core

Version 2019.8.9

IronSoftware

Ahmed Aboelmagd

Contents

Introduction 2

Document Organization 2

Chapter 1: Install IronPdf 2

- Install using NuGet 2
  - Using NuGet Package Manager 3
  - Using NuGet Package Console manager 5
  - Sample 1: HelloWorldConsole Console Application 6
  - Sample 2: HelloWorldCore .Net Core Web Application 6

Chapter 2: Convert to Pdf 6

- Convert online website to Pdf 6
  - Sample: ConvertUrlToPdf console application 6
- Convert HTML to Pdf 11
  - Sample: ConvertHTMLToPdf Console application 11
- Convert MVC Partial view to Pdf 11
  - Sample: TicketsApps .NET Core MVC Application 11
- Advanced options 11

Chapter 3: Deployment with dockers 11

Chapter 4: Working with Pdf Document 12

- Open Pdf 12
- Merge Pdf 12
- Add Header Or footer to Pdf 12
- Pdf security 12
- Pdf extraction and conversions 12

Summary 12

Appendix (A) 12

References 12

Author 12

Introduction

Adding Pdf file generation in ASP.Net MVC project is a cumbersome task; also converting MVC views, HTML file, and online web pages to Pdf is a very hard and complex problem,

<-- Text in introduction part -->

And you can visit http://www.ironpdf.com for more information.

You can download sample project from GitHub (https://github.com/magedo93/IronSoftware.git)

Document Organization

- Chapter 1 Install IronPdf: this part describes How to install IronPdf to existing project.
- Chapter 2 Convert to pdf: this part describes different methods to create Pdf from different sources like (URL, HTML, MVC views) and different advanced options that we can use for different output pdf settings,
  Also how to deploy your project to different images (Linux, windows)
- Chapter 3 Working with Pdf Document: this part describes how to different manipulation capabilities on created pdf files like adding headers or footers, merge files, add STAMP and other features.
- Summery
  brief conclusion about what we have learned in this document
- Appendix (A) Dockers: this part describes what is dockers and how to use it
- About author
  brief about document author

Chapter 1: Install IronPdf

IronPdf can be installed and used on all of .NET projects type like windows application, ASP.NET MVC and .Net Core Application.

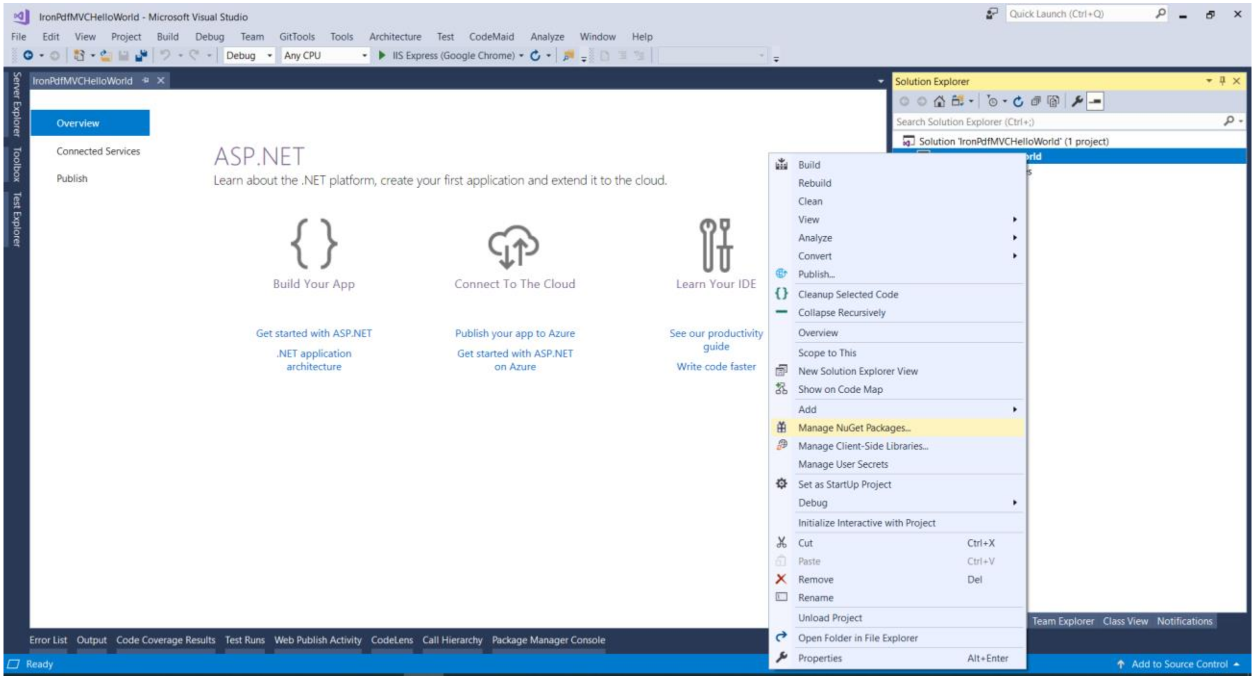
To add IronPdf library to the project we have two ways, from Visual studio editor install using NuGet or command line using package console manager as following: -

Install using NuGet

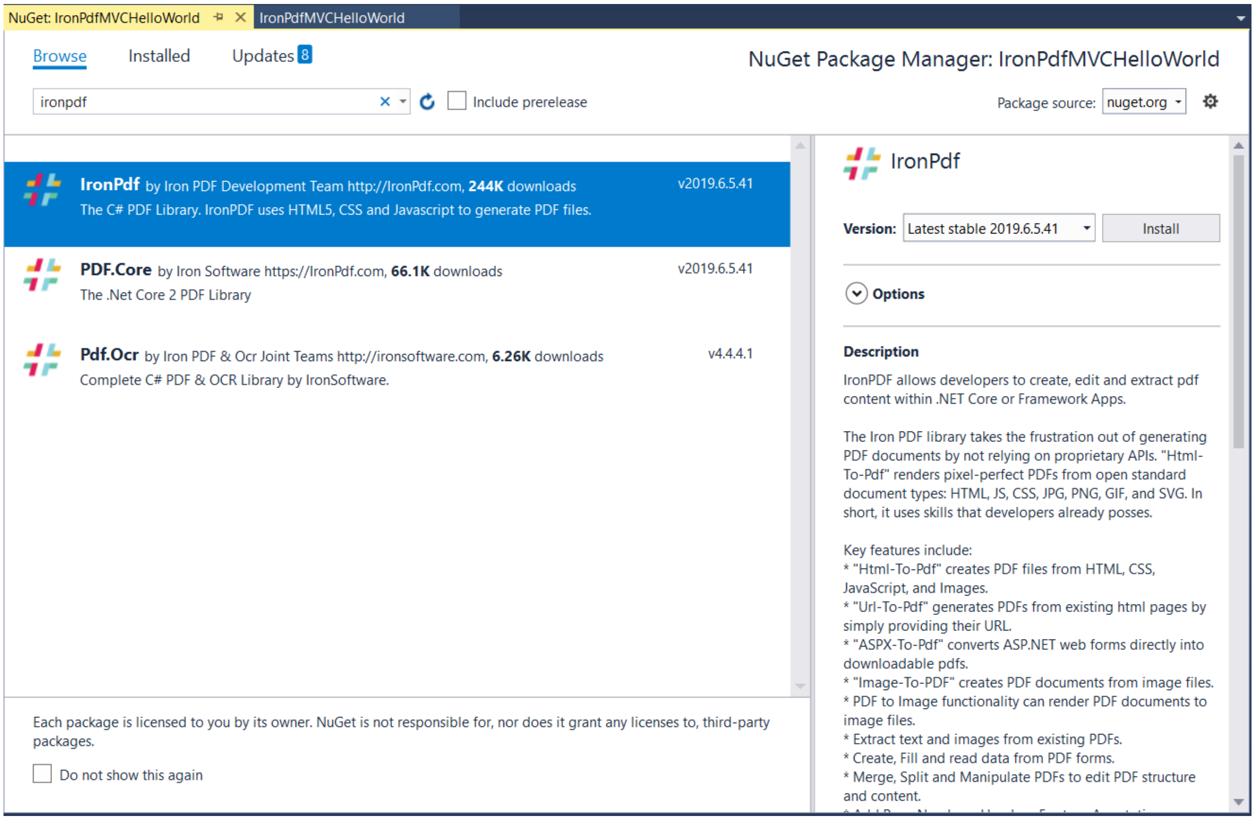
To add IronPdf library to our project using NuGet we can do it using visualized interface (NuGet Package Manager) or by command using Package Manager Console as following: -

Using NuGet Package Manager

1- Using mouse -> right click on project name -> Select manage NuGet Package


IronPdfMVCHelloWorld - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team GitTools Tools Architecture Test CodeMaid Analyze Window Help
Debug
Any CPU
IIS Express (Google Chrome)
Server Explorer
Toolbox
Test Explorer
IronPdfMVCHelloWorld
Overview
Connected Services
Publish
ASP.NET
Learn about the .NET platform, create your first application and extend it to the cloud.
Build Your App
Connect To The Cloud
Learn Your IDE
Get started with ASP.NET
.NET application architecture
Publish your app to Azure
Get started with ASP.NET on Azure
See our productivity guide
Write code faster
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'IronPdfMVCHelloWorld' (1 project)
Build
Rebuild
Clean
View
Analyze
Convert
Publish...
Cleanup Selected Code
Collapse Recursively
Overview
Scope to This
New Solution Explorer View
Show on Code Map
Add
Manage NuGet Packages...
Manage Client-Side Libraries...
Manage User Secrets
Set as StartUp Project
Debug
Initialize Interactive with Project
Cut Ctrl+X
Paste Ctrl+V
Remove Del
Rename
Unload Project
Open Folder in File Explorer
Properties Alt+Enter
Error List Output Code Coverage Results Test Runs Web Publish Activity CodeLens Call Hierarchy Package Manager Console
Team Explorer Class View Notifications
Ready
Add to Source Control

2- From brows tab -> search for IronPdf -> Install


NuGet: IronPdfMVCHelloWorld
IronPdfMVCHelloWorld
Browse
Installed
Updates 8
NuGet Package Manager: IronPdfMVCHelloWorld
ironpdf
□ Include prerelease
Package source: nuget.org
Install
Options

IronPdf by Iron PDF Development Team http://IronPdf.com, **244K** downloads v2019.6.5.41
The C# PDF Library. IronPDF uses HTML5, CSS and Javascript to generate PDF files.

PDF.Core by Iron Software https://IronPdf.com, **66.1K** downloads v2019.6.5.41
The .Net Core 2 PDF Library

Pdf.Ocr by Iron PDF & Ocr Joint Teams http://ironsoftware.com, **6.26K** downloads v4.4.4.1
Complete C# PDF & OCR Library by IronSoftware.

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

□ Do not show this again

IronPdf

Version: Latest stable 2019.6.5.41

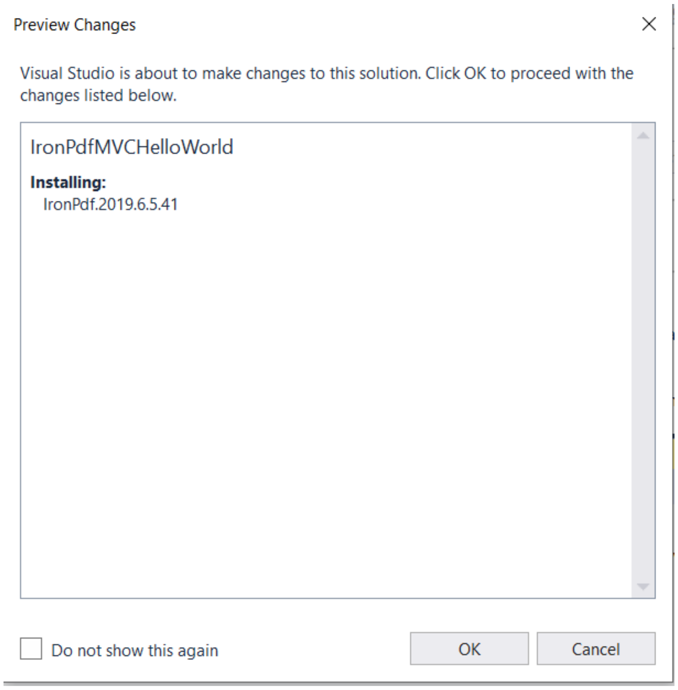
Description

IronPDF allows developers to create, edit and extract pdf content within .NET Core or Framework Apps.

The Iron PDF library takes the frustration out of generating PDF documents by not relying on proprietary APIs. "Html-To-Pdf" renders pixel-perfect PDFs from open standard document types: HTML, JS, CSS, JPG, PNG, GIF, and SVG. In short, it uses skills that developers already posses.

Key features include:
* "Html-To-Pdf" creates PDF files from HTML, CSS, JavaScript, and Images.
* "Url-To-Pdf" generates PDFs from existing html pages by simply providing their URL.
* "ASPX-To-Pdf" converts ASP.NET web forms directly into downloadable pdfs.
* "Image-To-PDF" creates PDF documents from image files.
* PDF to Image functionality can render PDF documents to image files.
* Extract text and images from existing PDFs.
* Create, Fill and read data from PDF forms.
* Merge, Split and Manipulate PDFs to edit PDF structure and content.

3- Click Ok



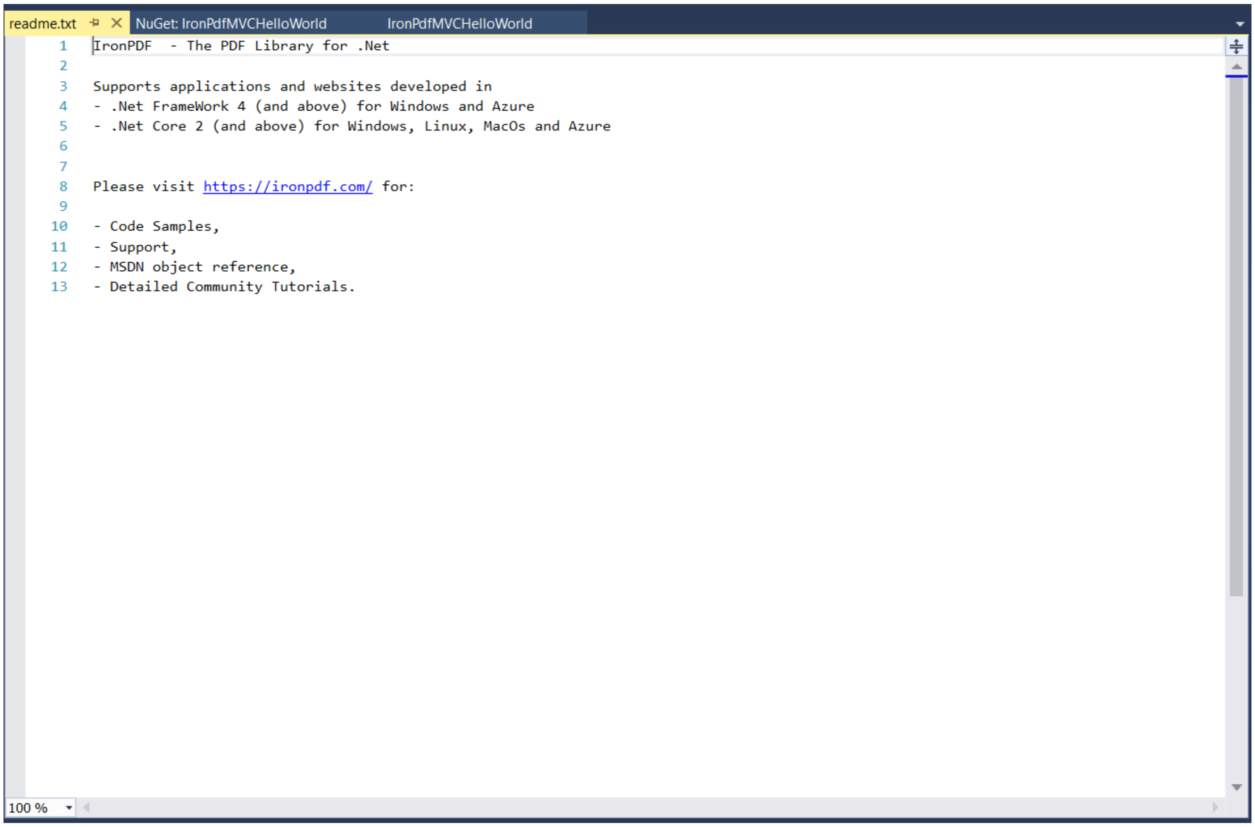
Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

IronPdfMVCHelloWorld

Installing:
IronPdf.2019.6.5.41

□ Do not show this again

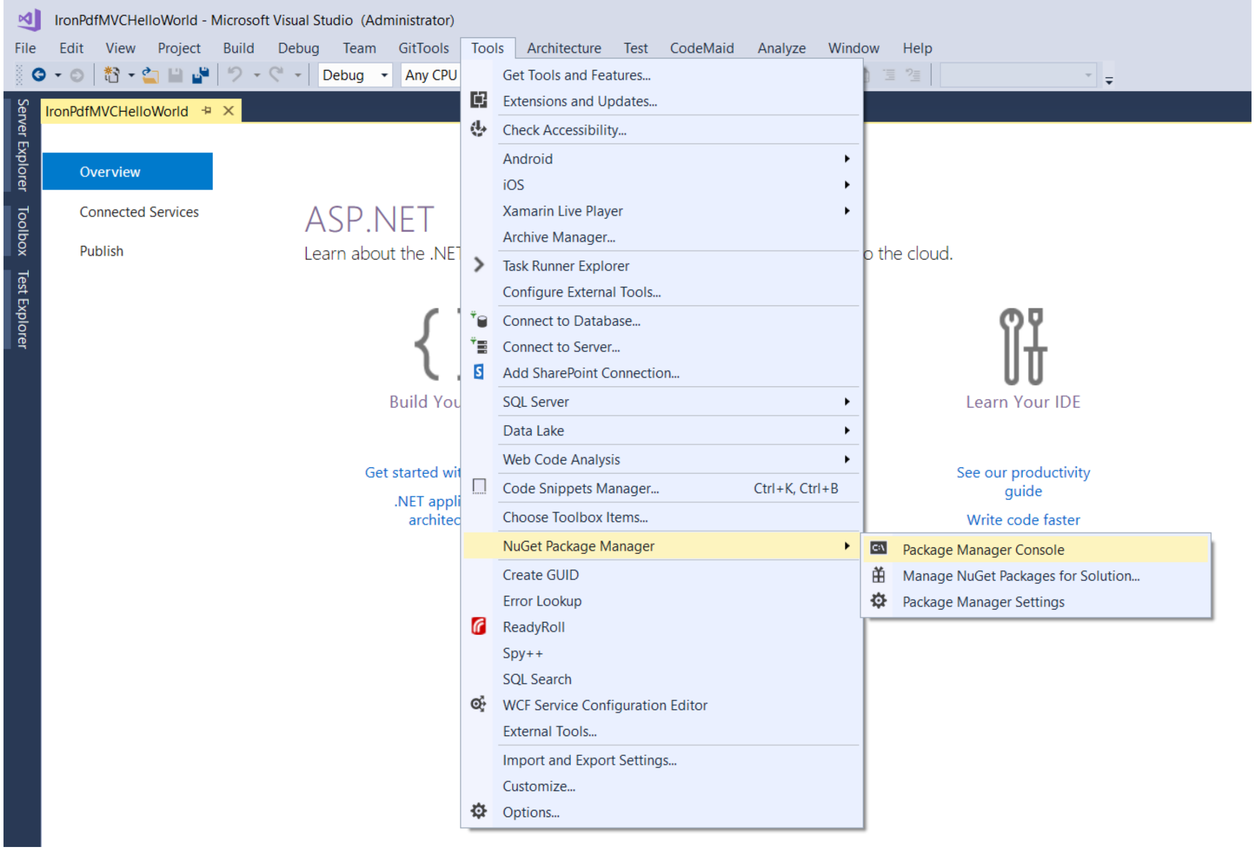
4- And we are Done



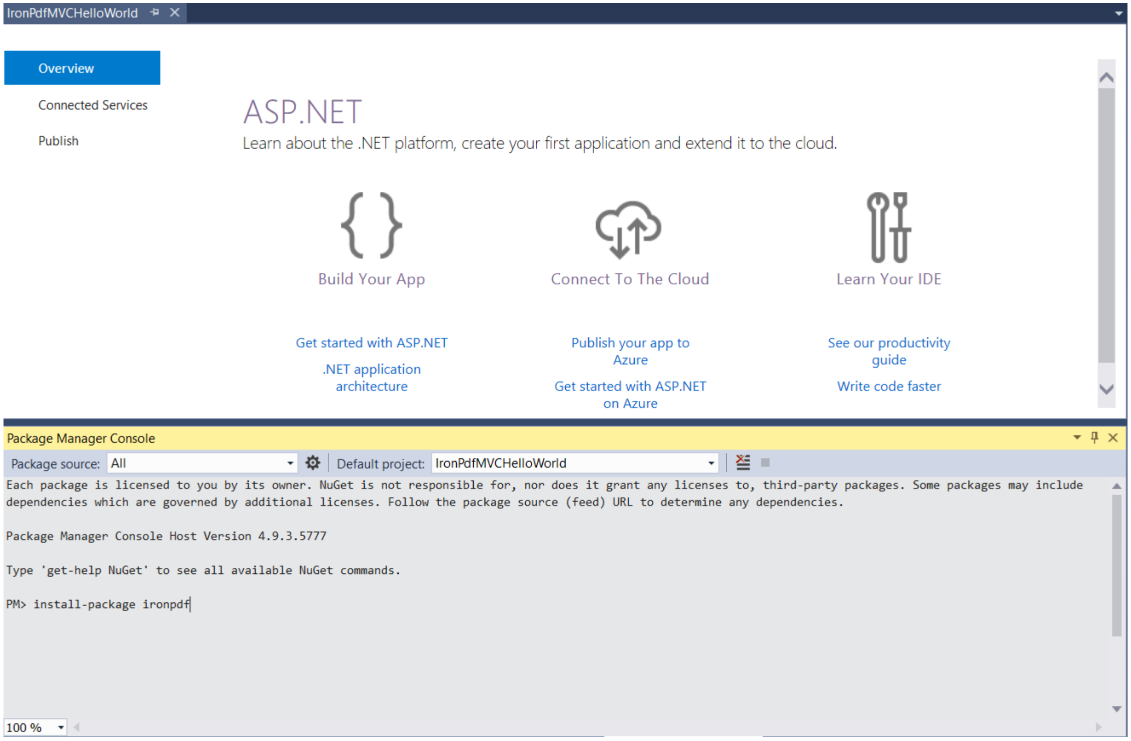
```
IronPDF  - The PDF Library for .Net

Supports applications and websites developed in
- .Net FrameWork 4 (and above) for Windows and Azure
- .Net Core 2 (and above) for Windows, Linux, MacOs and Azure


Please visit https://ironpdf.com/ for:

- Code Samples,
- Support,
- MSDN object reference,
- Detailed Community Tutorials.
```

Using NuGet Package Console manager

1- From tools -> NuGet Package Manager -> Package Manager Console



2- Run command -> Install-Package IronPdf


IronPdfMVCHelloWorld
Overview
Connected Services
Publish
ASP.NET
Learn about the .NET platform, create your first application and extend it to the cloud.
Build Your App
Connect To The Cloud
Learn Your IDE
Get started with ASP.NET
.NET application architecture
Publish your app to Azure
Get started with ASP.NET on Azure
See our productivity guide
Write code faster
Package Manager Console
100 %

Package source: All

Default project: IronPdfMVCHelloWorld

```
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages. Some packages may include
dependencies which are governed by additional licenses. Follow the package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 4.9.3.5777

Type 'get-help NuGet' to see all available NuGet commands.

PM> install-package ironpdf
```

We can now practice by implementing Hello World using .NET Core Console Application and another Hello world using .NET Core Web Application

Sample 1: HelloWorldConsole Console Application

Open visual studio => new => project

Sample 2: HelloWorldCore .Net Core Web Application

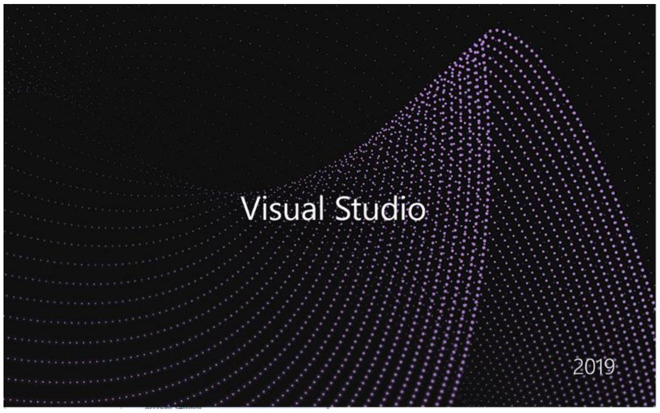
Chapter 2: Convert to Pdf

Convert online website to Pdf

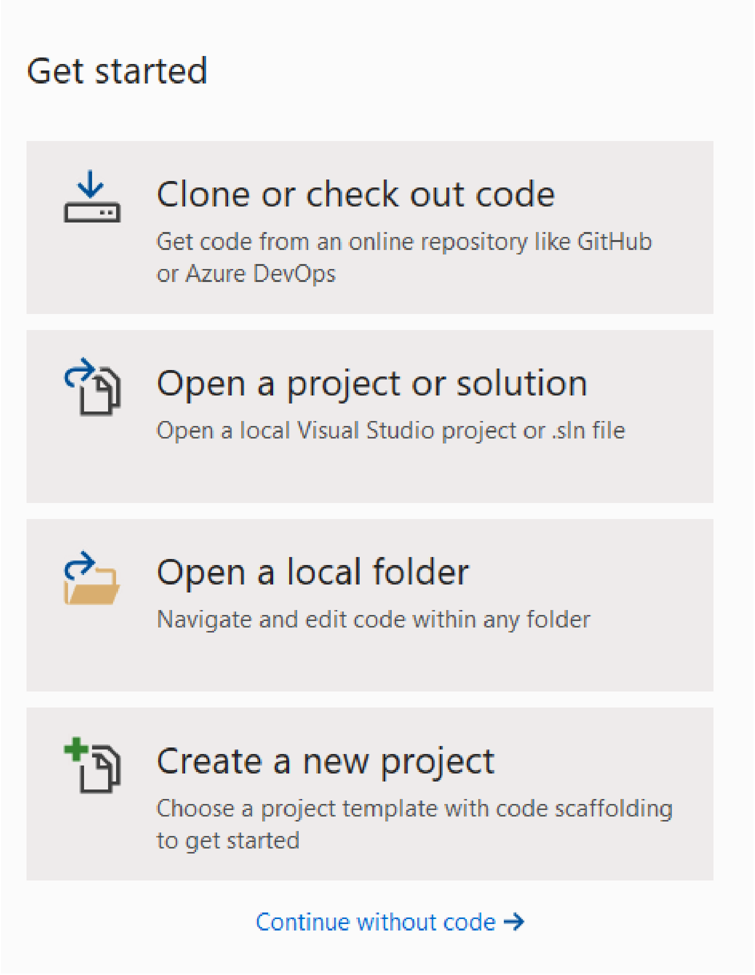
Sample: ConvertUrlToPdf console application

Follow coming steps to create new Asp.NET MVC Project

1- Open visual studio



2- Choose Create new project



Get started

Clone or check out code

Get code from an online repository like GitHub or Azure DevOps

Open a project or solution

Open a local Visual Studio project or .sln file

Open a local folder

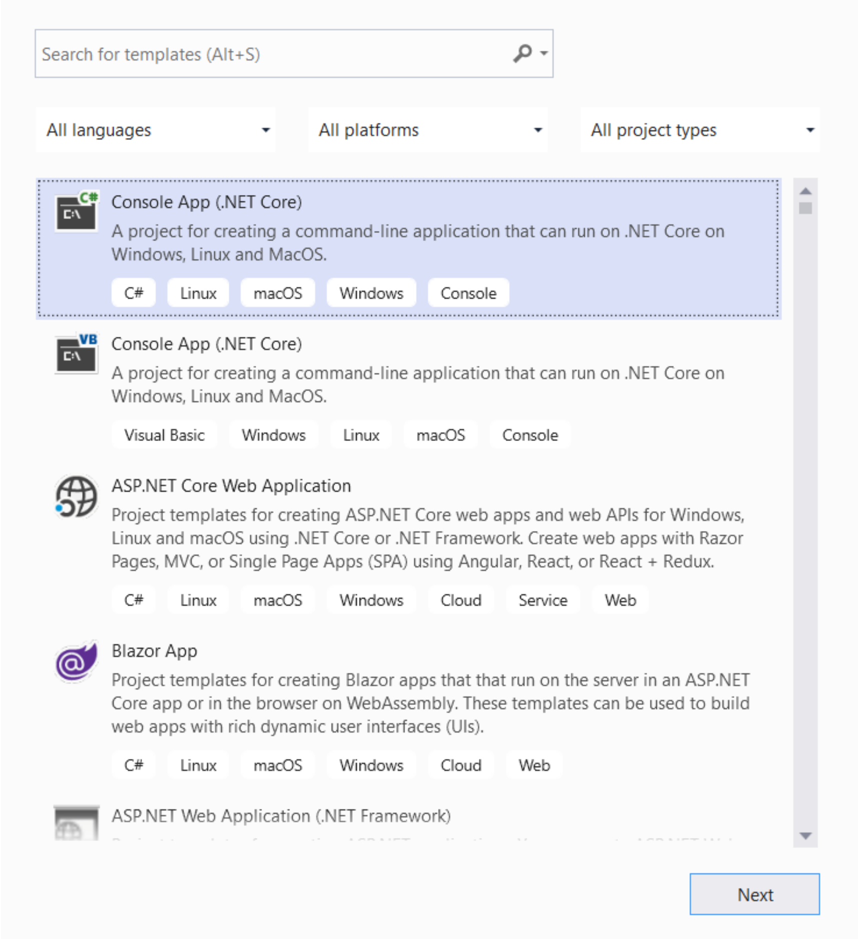
Navigate and edit code within any folder

Create a new project

Choose a project template with code scaffolding to get started

Continue without code →

3- Choose Console App (.NET Core)



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

ASP.NET Core Web Application

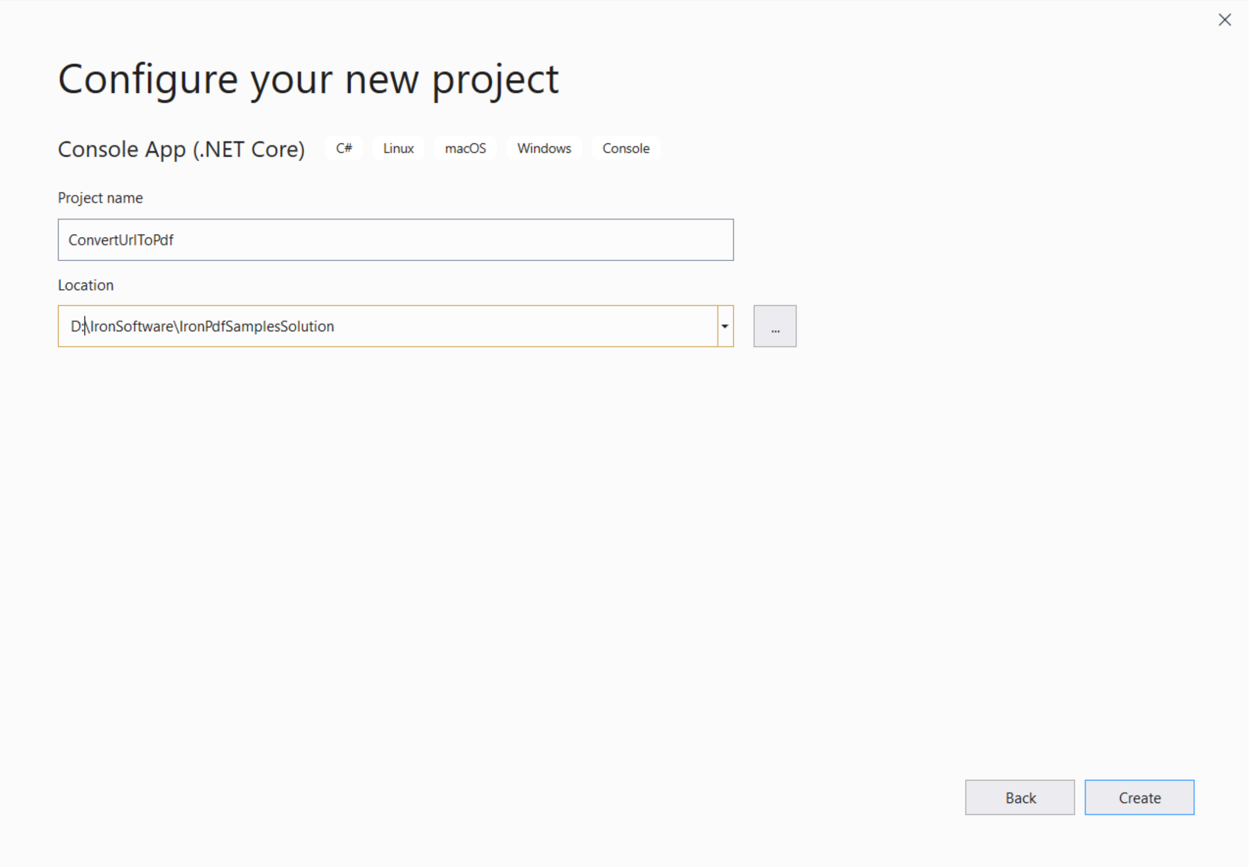
Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

Blazor App

Project templates for creating Blazor apps that that run on the server in an ASP.NET Core app or in the browser on WebAssembly. These templates can be used to build web apps with rich dynamic user interfaces (UIs).

ASP.NET Web Application (.NET Framework)

4- Give our sample name “ConvertUrlToPdf” and click create



Configure your new project

Console App (.NET Core)

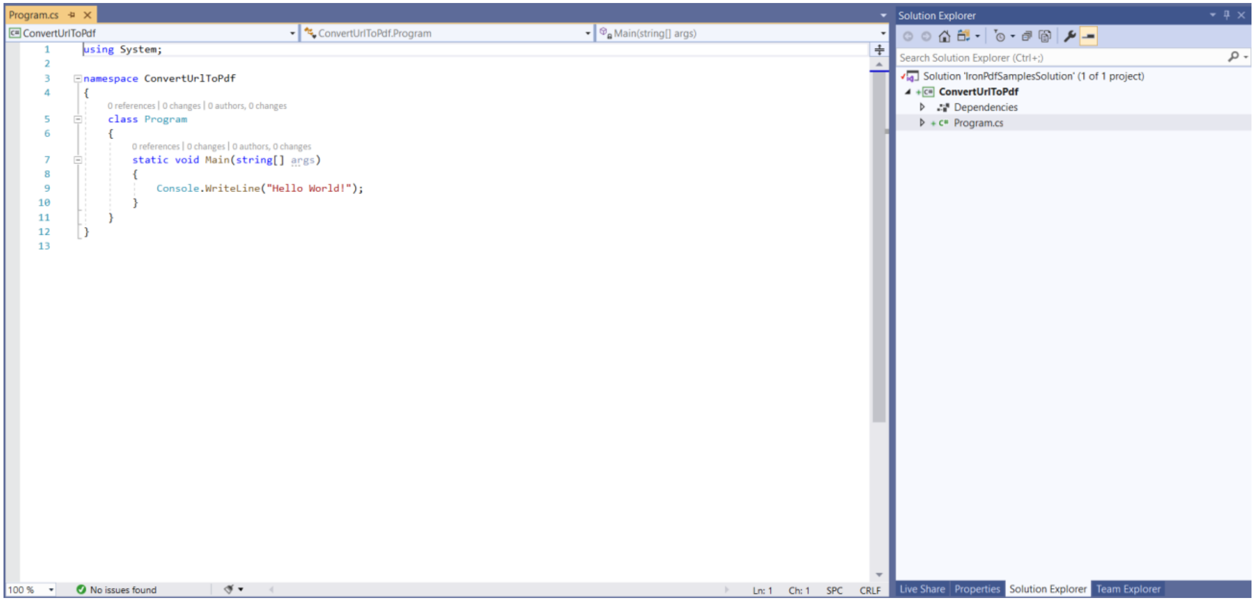
Project name

ConvertUrlToPdf

Location

D:\IronSoftware\IronPdfSamplesSolution

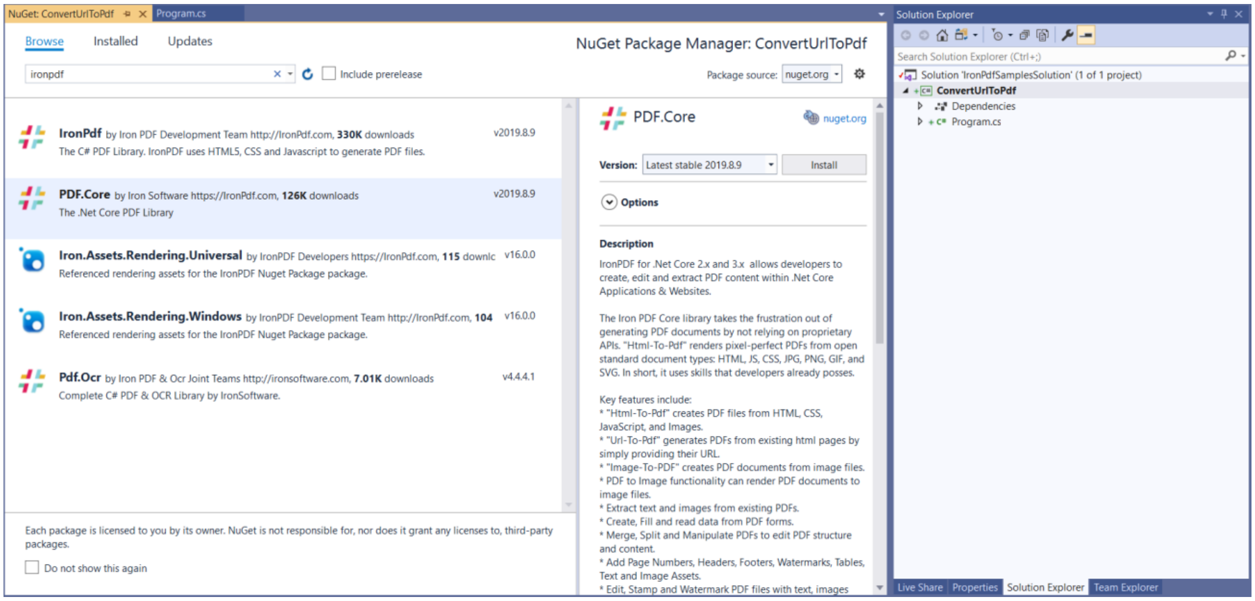
5- Now we have console application created



```
using System;

namespace ConvertUrlToPdf
{
    0 references | 0 changes | 0 authors, 0 changes
    class Program
    {
        0 references | 0 changes | 0 authors, 0 changes
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Solution 'IronPdfSamplesSolution' (1 of 1 project)
- ConvertUrlToPdf
  - Dependencies
  - Program.cs

6- Add IronPdf => click install



ironpdf

IronPdf by Iron PDF Development Team http://IronPdf.com, **330K** downloads v2019.8.9
The C# PDF Library. IronPDF uses HTML5, CSS and Javascript to generate PDF files.

PDF.Core by Iron Software https://IronPdf.com, **126K** downloads v2019.8.9
The .Net Core PDF Library

Iron.Assets.Rendering.Universal by IronPDF Developers https://IronPdf.com, **115** downlc v16.0.0
Referenced rendering assets for the IronPDF Nuget Package package.

Iron.Assets.Rendering.Windows by IronPDF Development Team http://IronPdf.com, **104** v16.0.0
Referenced rendering assets for the IronPDF Nuget Package package.

Pdf.Ocr by Iron PDF & Ocr Joint Teams http://ironsoftware.com, **7.01K** downloads v4.4.4.1
Complete C# PDF & OCR Library by IronSoftware.

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

□ Do not show this again

PDF.Core nuget.org

Version: Latest stable 2019.8.9 Install

Options

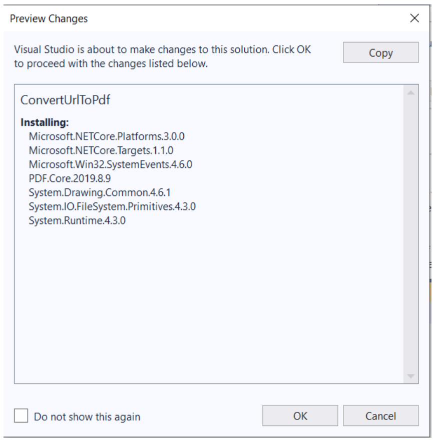
Description

IronPDF for .Net Core 2.x and 3.x allows developers to create, edit and extract PDF content within .Net Core Applications & Websites.

The Iron PDF Core library takes the frustration out of generating PDF documents by not relying on proprietary APIs. "Html-To-Pdf" renders pixel-perfect PDFs from open standard document types: HTML, JS, CSS, JPG, PNG, GIF, and SVG. In short, it uses skills that developers already posses.

Key features include:
* "Html-To-Pdf" creates PDF files from HTML, CSS, JavaScript, and Images.
* "Url-To-Pdf" generates PDFs from existing html pages by simply providing their URL.
* "Image-To-PDF" creates PDF documents from image files.
* PDF to Image functionality can render PDF documents to image files.
* Extract text and images from existing PDFs.
* Create, Fill and read data from PDF forms.
* Merge, Split and Manipulate PDFs to edit PDF structure and content.
* Add Page Numbers, Headers, Footers, Watermarks, Tables, Text and Image Assets.
* Edit, Stamp and Watermark PDF files with text, images

Solution 'IronPdfSamplesSolution' (1 of 1 project)
ConvertUrlToPdf
Dependencies
Program.cs



Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

ConvertUrlToPdf

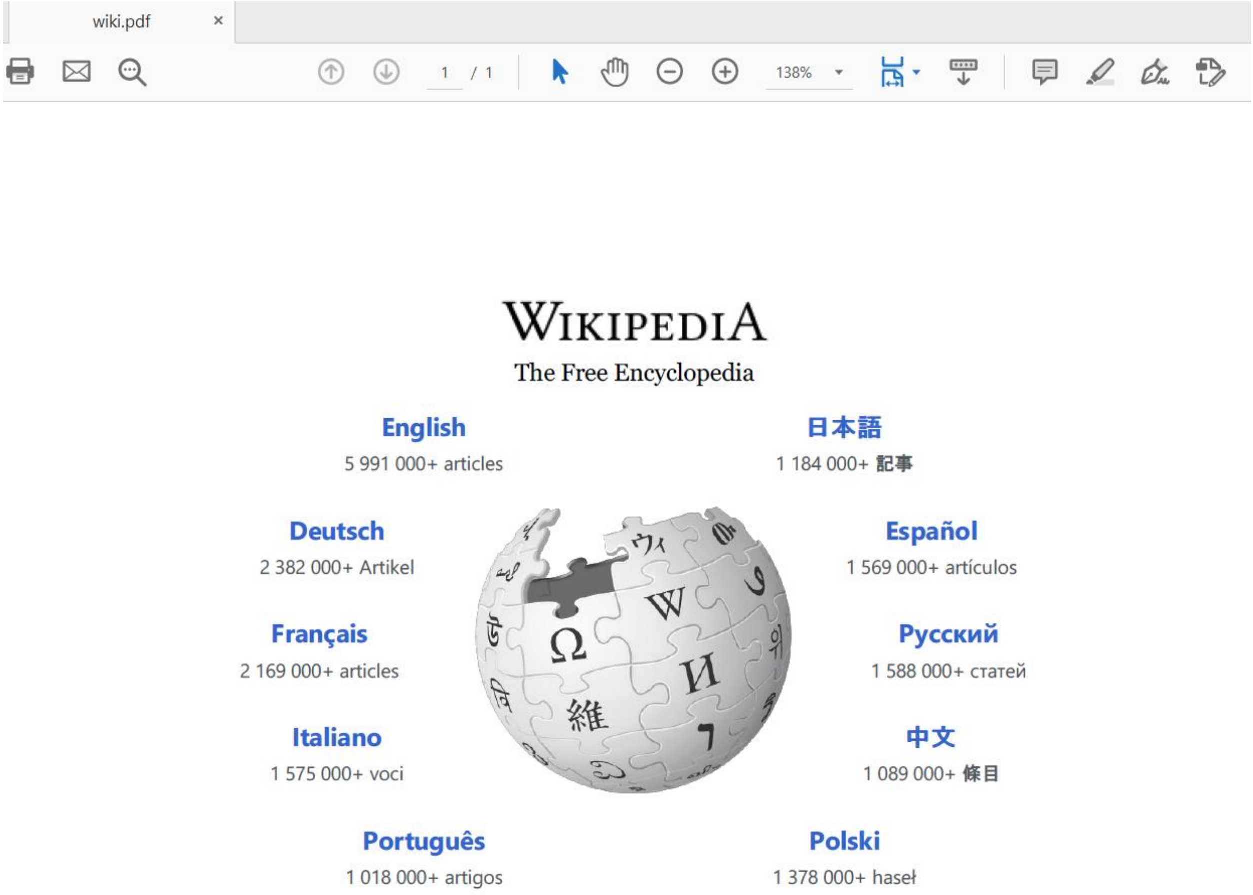
Installing:
Microsoft.NETCore.Platforms.3.0.0
Microsoft.NETCore.Targets.1.1.0
Microsoft.Win32.SystemEvents.4.6.0
PDF.Core.2019.8.9
System.Drawing.Common.4.6.1
System.IO.FileSystem.Primitives.4.3.0
System.Runtime.4.3.0

□ Do not show this again

7- Add our first few lines that render Wikipedia website main page to pdf

```
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            var render = new IronPdf.HtmlToPdf();
            var doc = render.RenderUrlAsPdf("https://www.wikipedia.org/");
            doc.SaveAs($@"{AppDomain.CurrentDomain.BaseDirectory}\wiki.pdf");
          }
```

8- Run and check created file wiki.pdf



Convert HTML to Pdf

Sample: ConvertHTMLToPdf Console application

Convert MVC Partial view to Pdf

Sample: TicketsApps .NET Core MVC Application

Advanced options

Chapter 3: Deployment with dockers

Deploy to Linux docker

Deploy to windows docker

Chapter 4: Working with Pdf Document

Open Pdf

Merge Pdf

Add Header Or footer to Pdf

Pdf security

Pdf extraction and conversions

Summary

Appendix (A)

References

1- Dockize and asp.net core application
https://docs.docker.com/engine/examples/dotnetcore/
2- Introduction to .NET and Docker
https://docs.microsoft.com/en-us/dotnet/core/docker/introduction
3- Containerize a .NET Core app
https://docs.microsoft.com/en-us/dotnet/core/docker/build-container

Author

Ahmed Aboelmagd is a Full-stack experienced and certified Microsoft technology specialist with 12+ year's experience in IT and Software development, delivered 15+ successful project in many size scales from small to an enterprise for industries like (tourism, educational, manufacturing, etc.)