|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| IronPdf Guide  with .NET Core  Version 2019.8.9 |
| |  |  |  | | --- | --- | --- | | IronSoftware |  | Ahmed Aboelmagd | |

Contents

[Introduction 2](#_Toc29313251)

[Document Organization 2](#_Toc29313252)

[Chapter 1: Install IronPdf 2](#_Toc29313253)

[Install using NuGet 2](#_Toc29313254)

[Using NuGet Package Manager 3](#_Toc29313255)

[Using NuGet Package Console manager 5](#_Toc29313256)

[Sample 1: HelloWorldConsole Console Application 6](#_Toc29313257)

[Sample 2: HelloWorldCore .Net Core Web Application 6](#_Toc29313258)

[Chapter 2: Convert to Pdf 6](#_Toc29313259)

[Convert online website to Pdf 6](#_Toc29313260)

[Sample: ConvertUrlToPdf console application 6](#_Toc29313261)

[Convert HTML to Pdf 11](#_Toc29313262)

[Sample: ConvertHTMLToPdf Console application 11](#_Toc29313263)

[Convert MVC Partial view to Pdf 12](#_Toc29313264)

[Sample: TicketsApps .NET Core MVC Application 12](#_Toc29313265)

[Advanced options 12](#_Toc29313266)

[Chapter 3: Deployment with dockers 12](#_Toc29313267)

[Chapter 4: Working with Pdf Document 13](#_Toc29313268)

[Open Pdf 13](#_Toc29313269)

[Merge Pdf 13](#_Toc29313270)

[Add Header Or footer to Pdf 13](#_Toc29313271)

[Pdf security 13](#_Toc29313272)

[Pdf extraction and conversions 13](#_Toc29313273)

[Summary 13](#_Toc29313274)

[Appendix (A) 13](#_Toc29313275)

[References 13](#_Toc29313276)

[Author 13](#_Toc29313277)

# 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

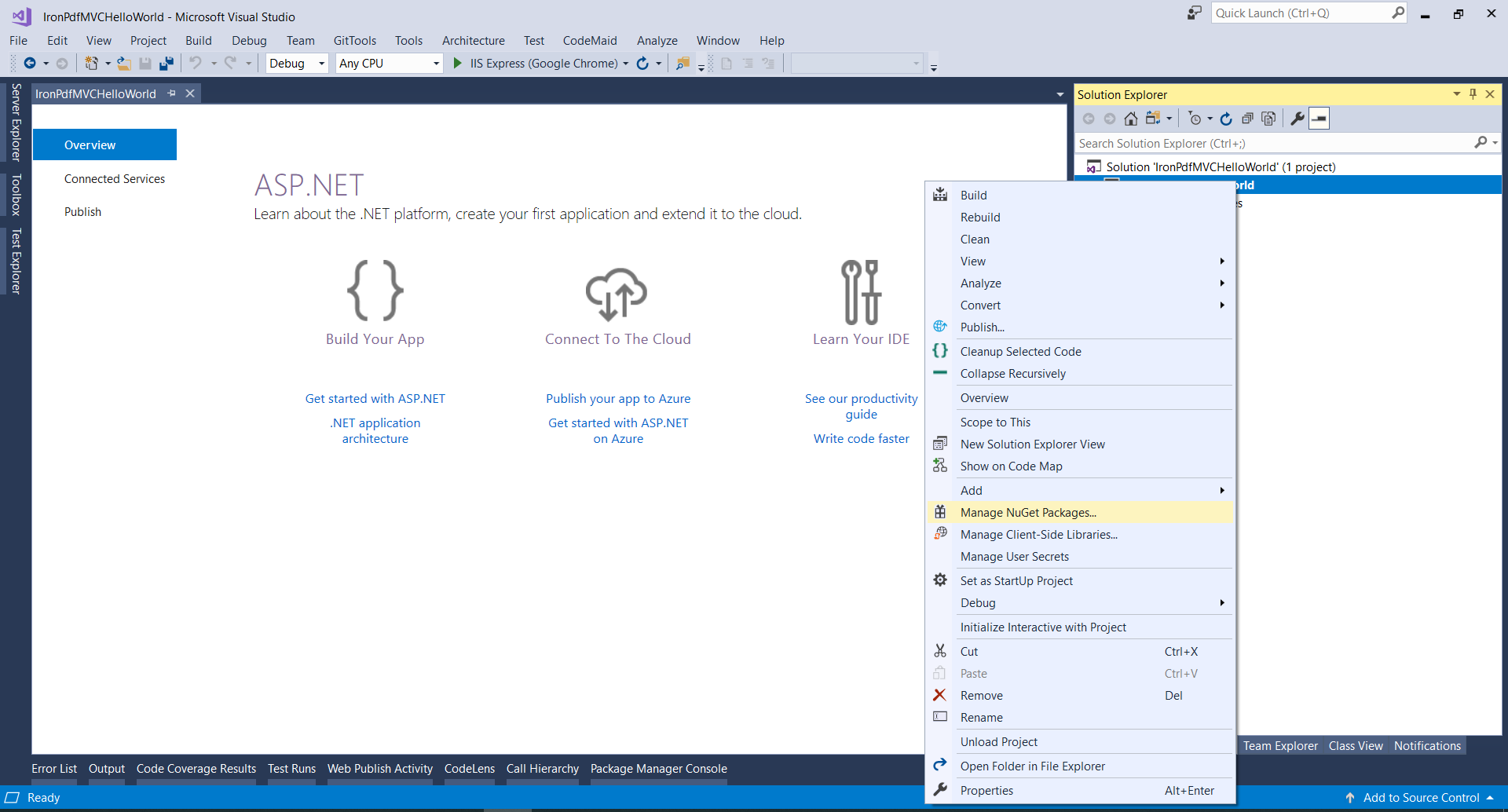
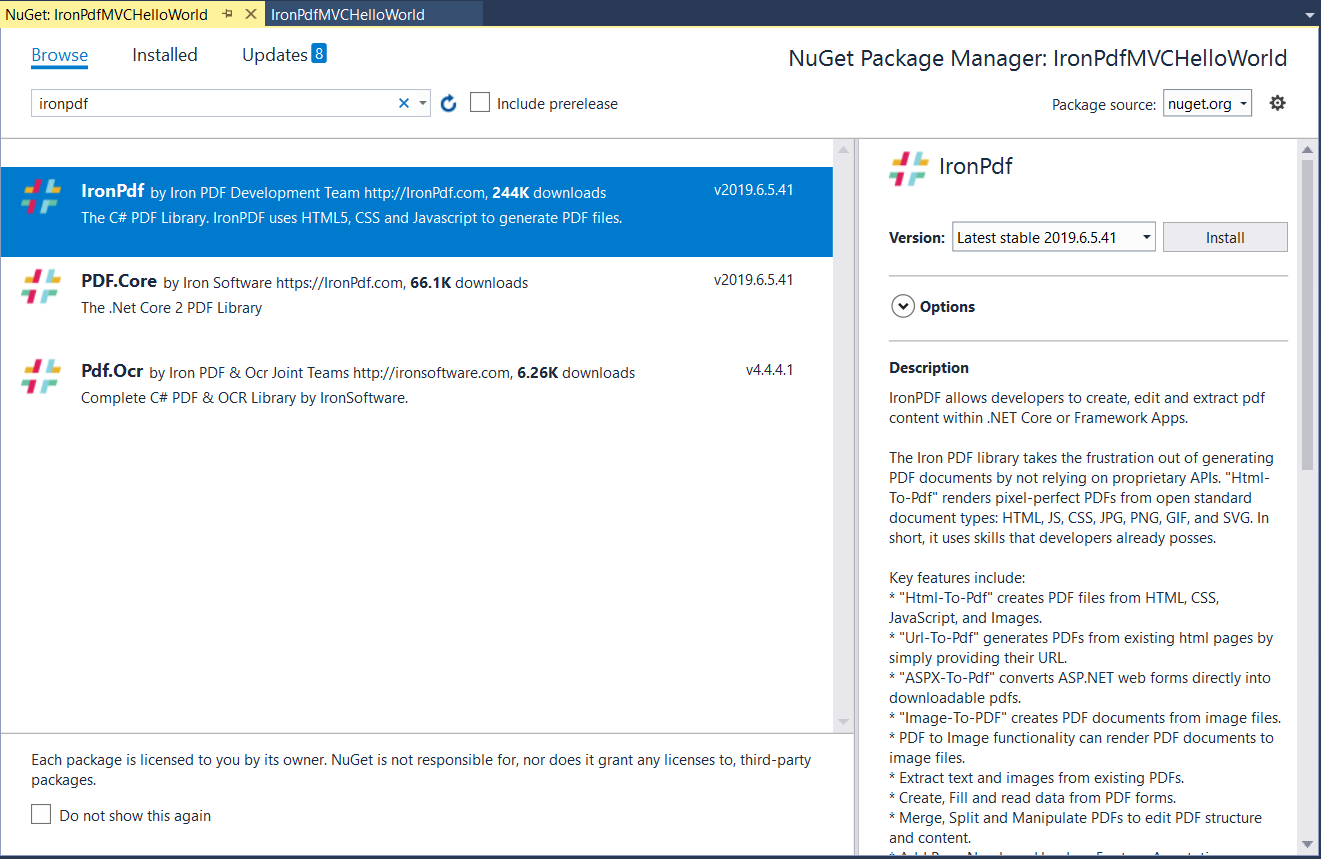
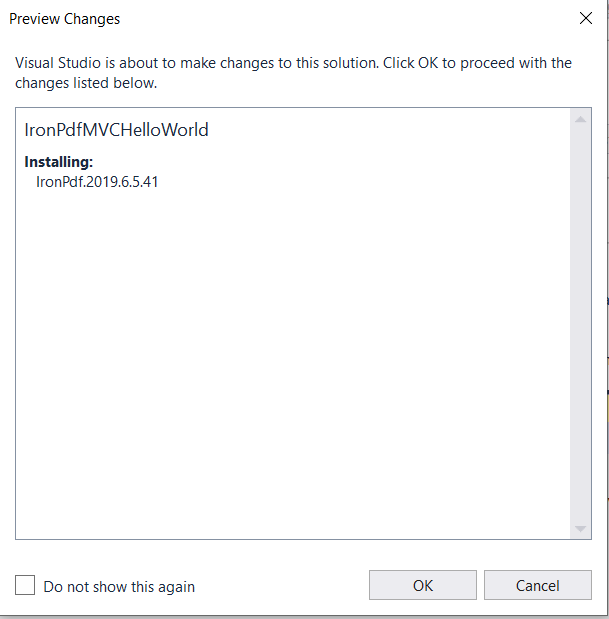
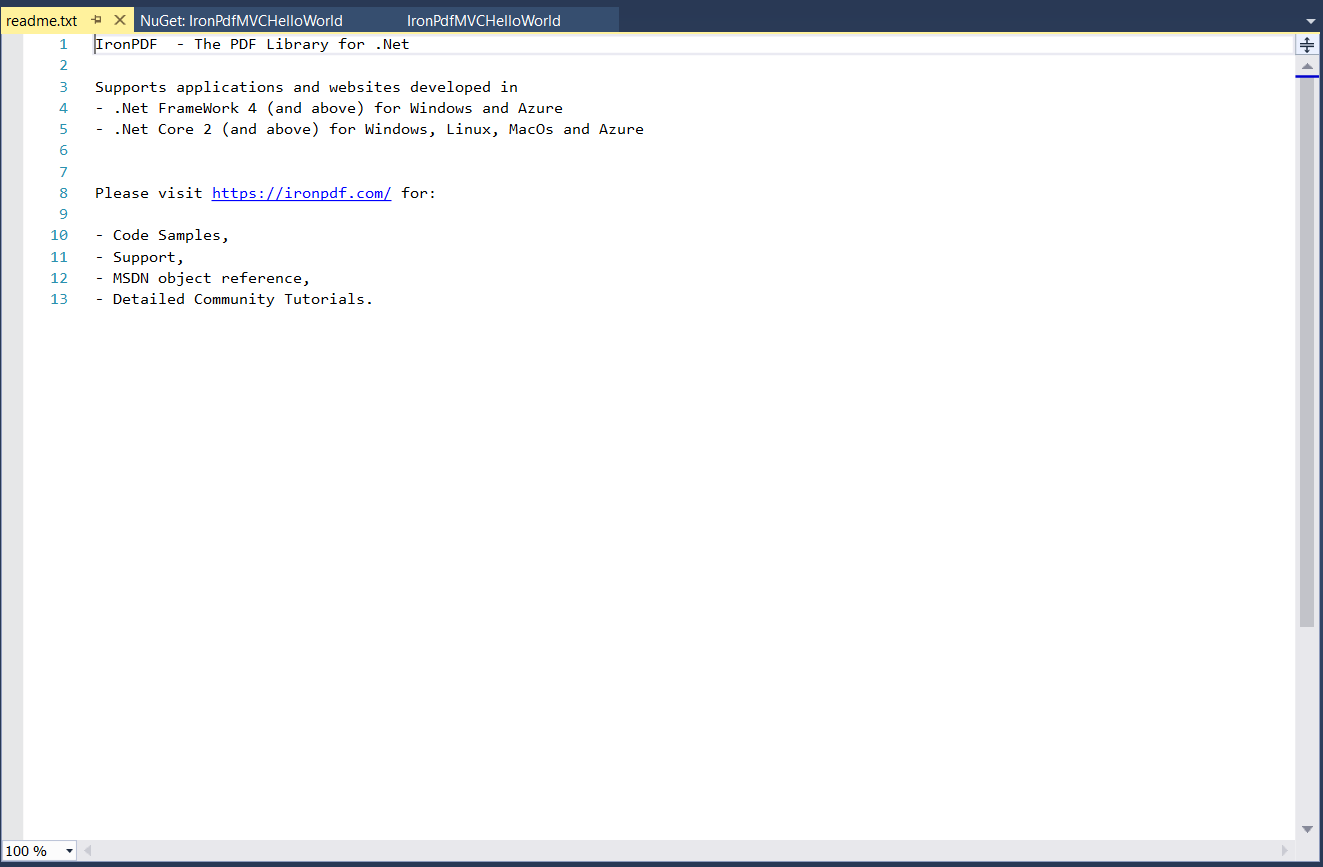
IronPdf can be installed and used on all of .NET projects type like windows application, ASP.NET MVC and .Net Core Application.

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: -

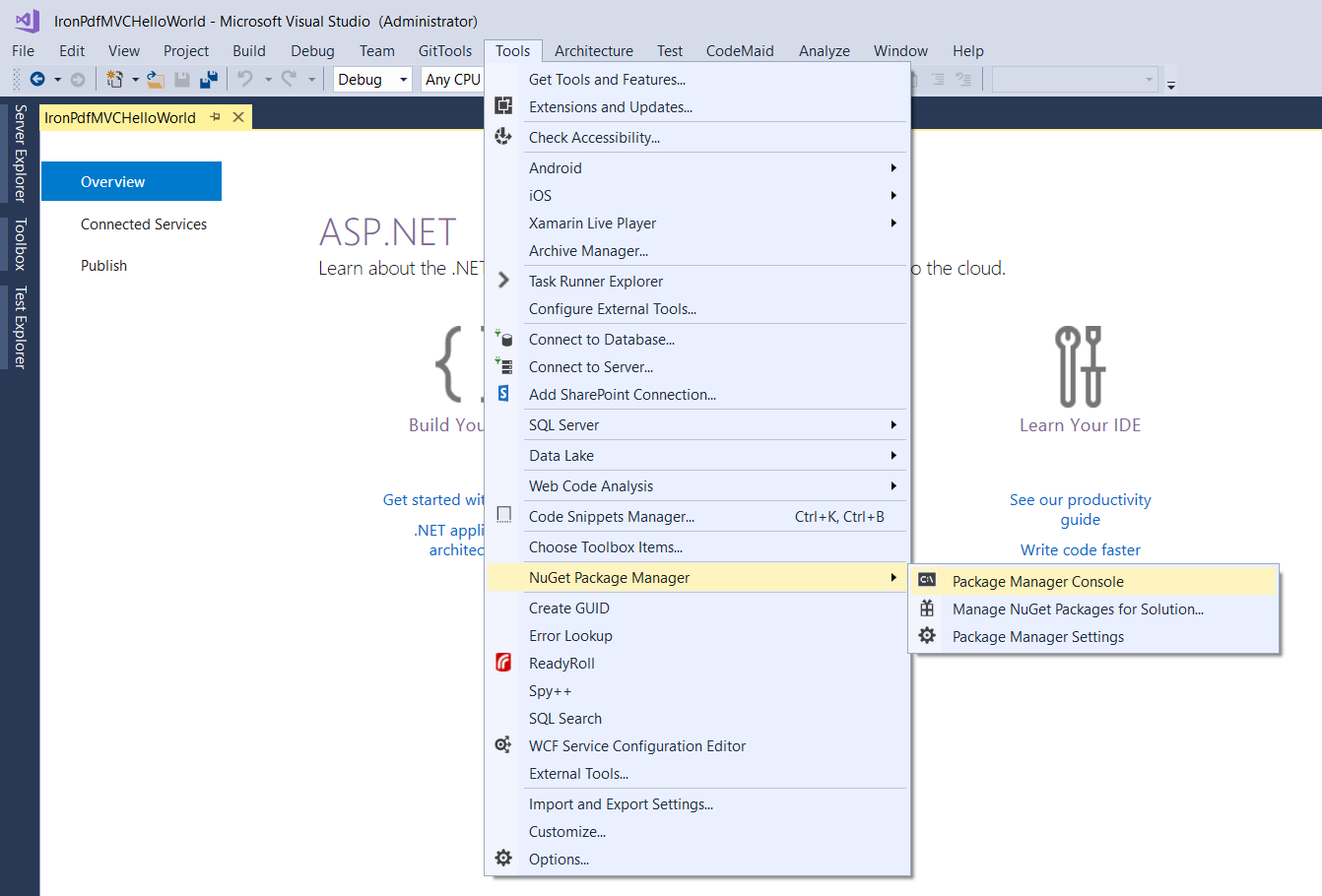
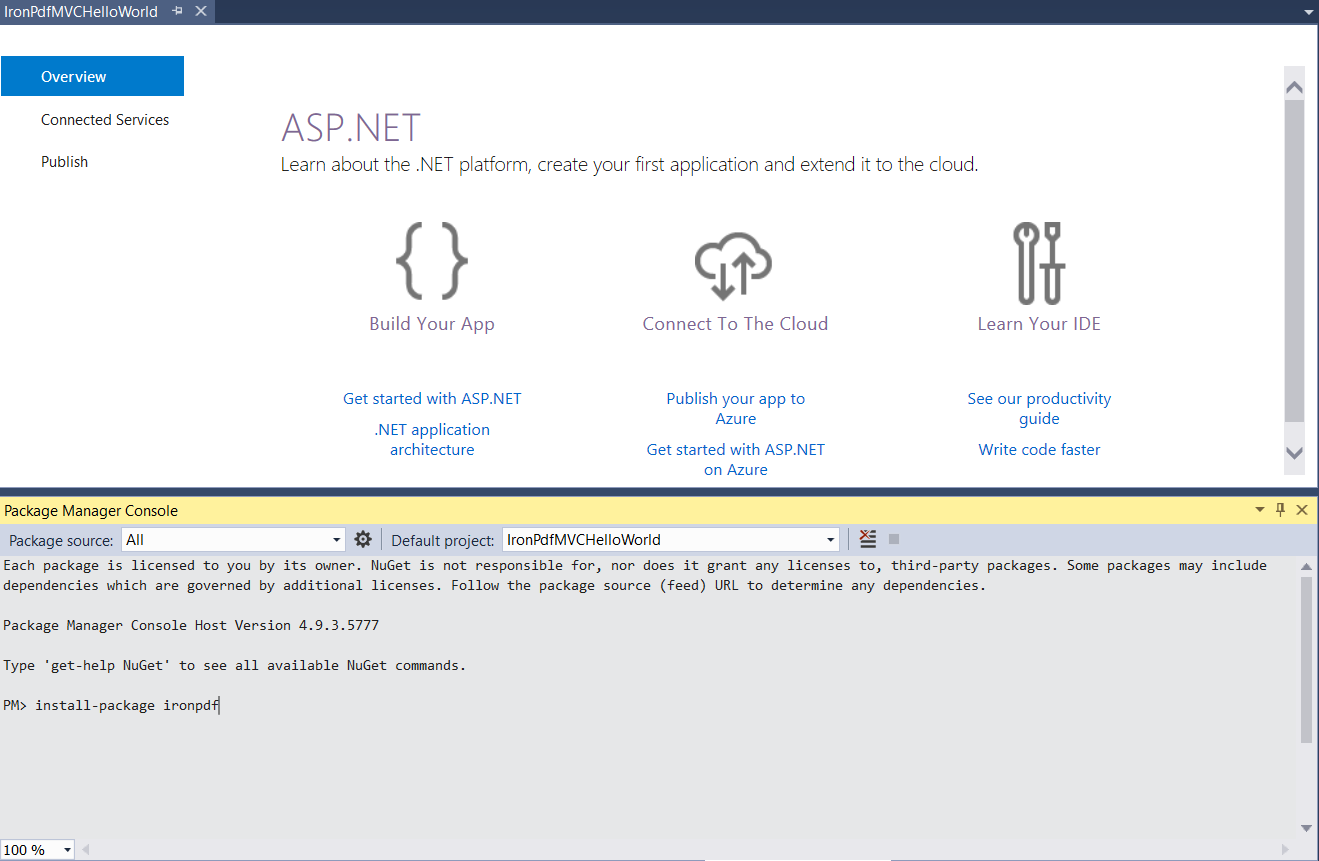
## Install using NuGet

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  
   
2. From brows tab -> search for IronPdf -> Install   
   
3. Click Ok  
   
4. And we are Done  
   

### Using NuGet Package Console manager

1. From tools -> NuGet Package Manager -> Package Manager Console  
   
2. Run command -> 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

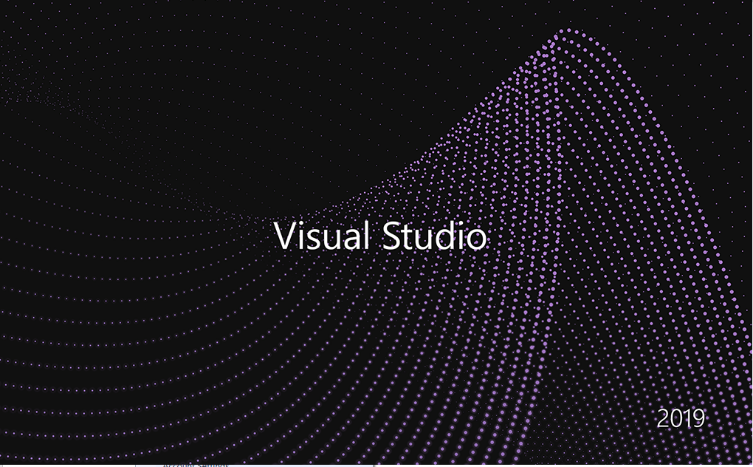
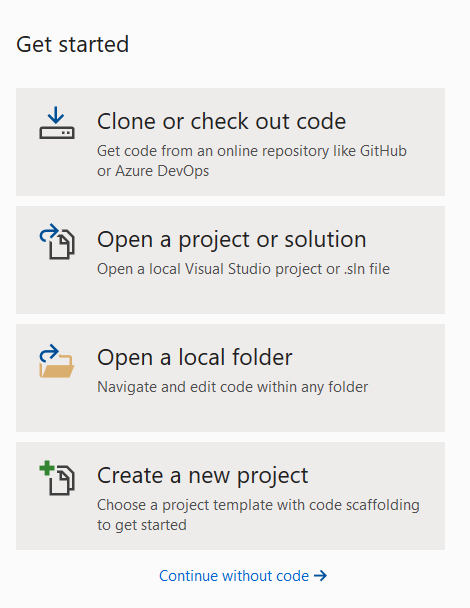
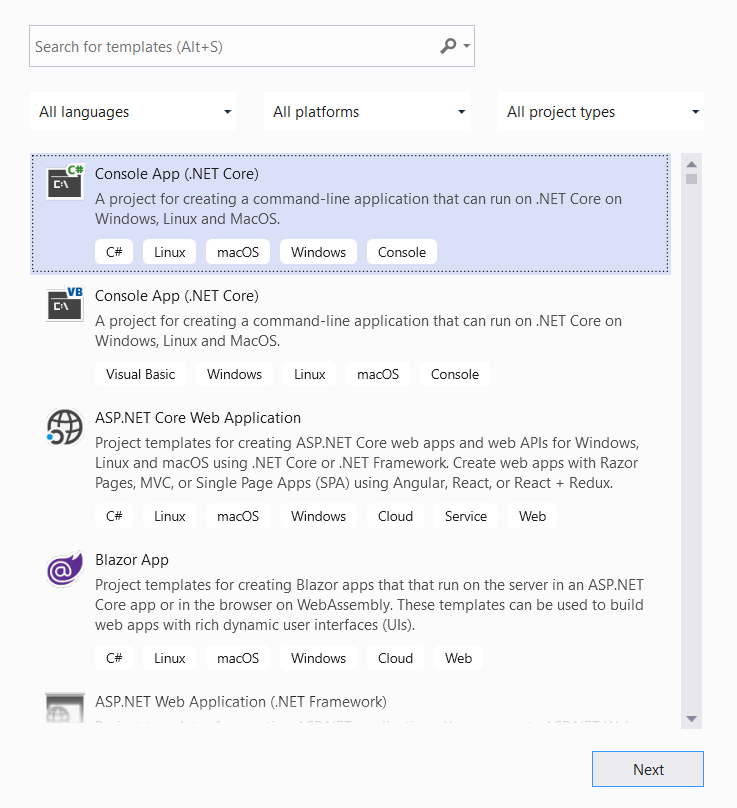
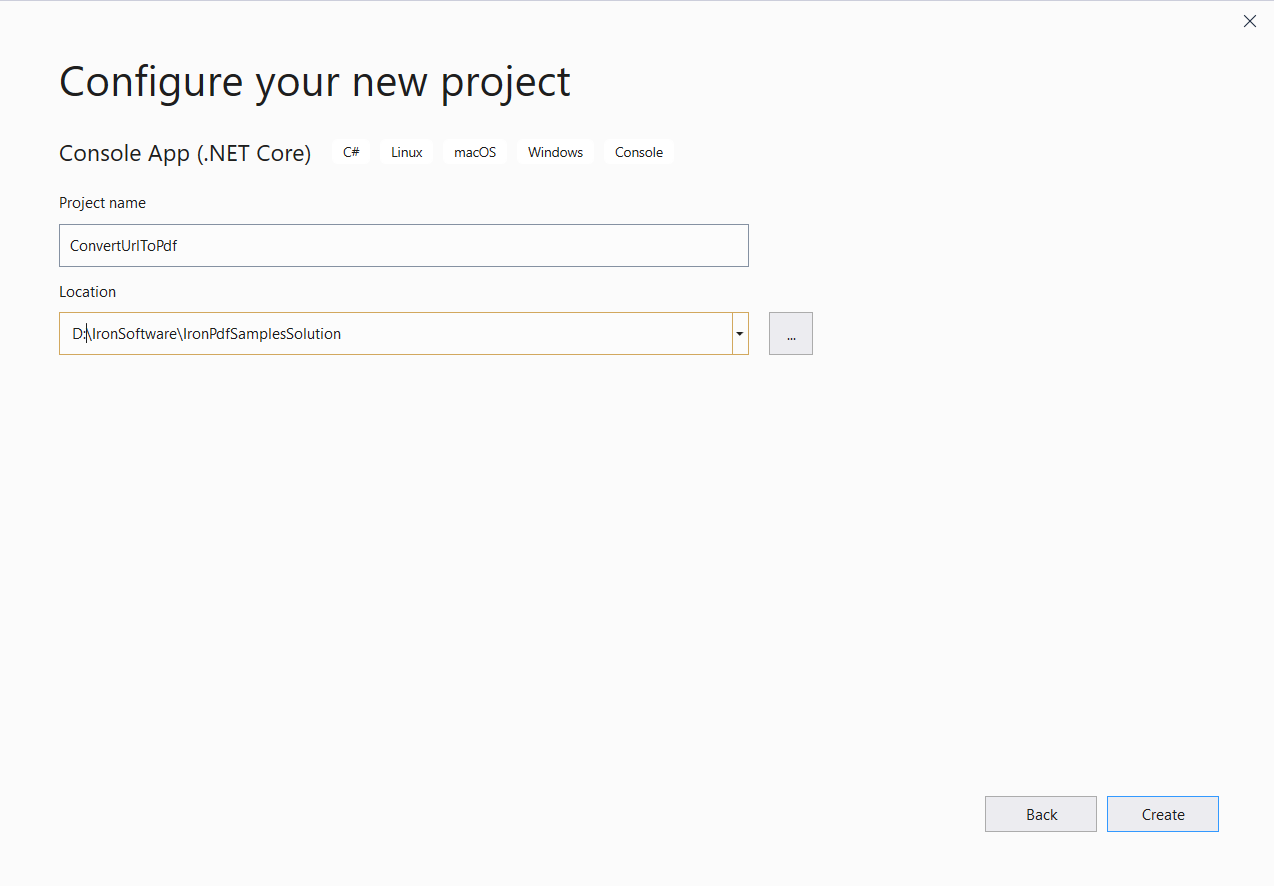
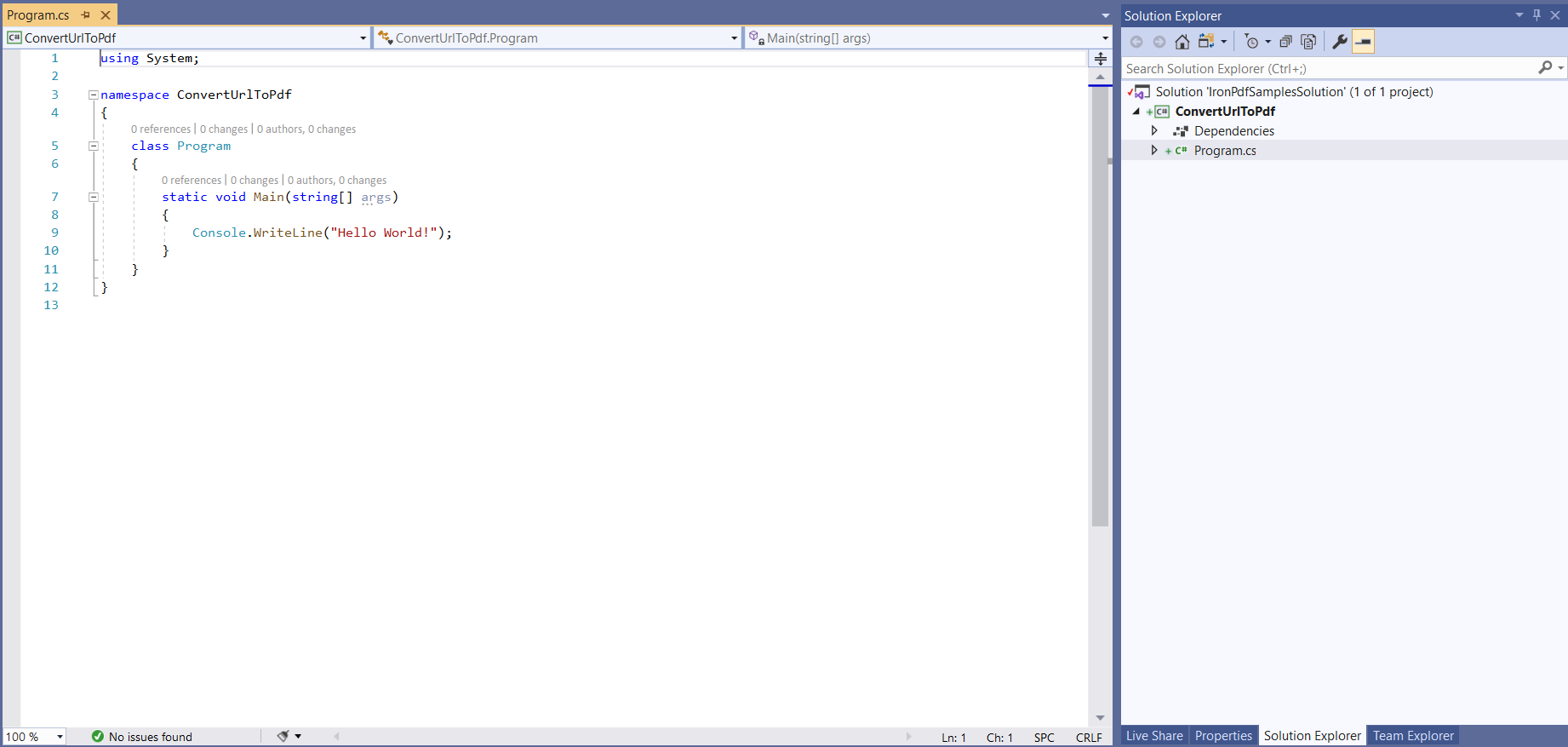
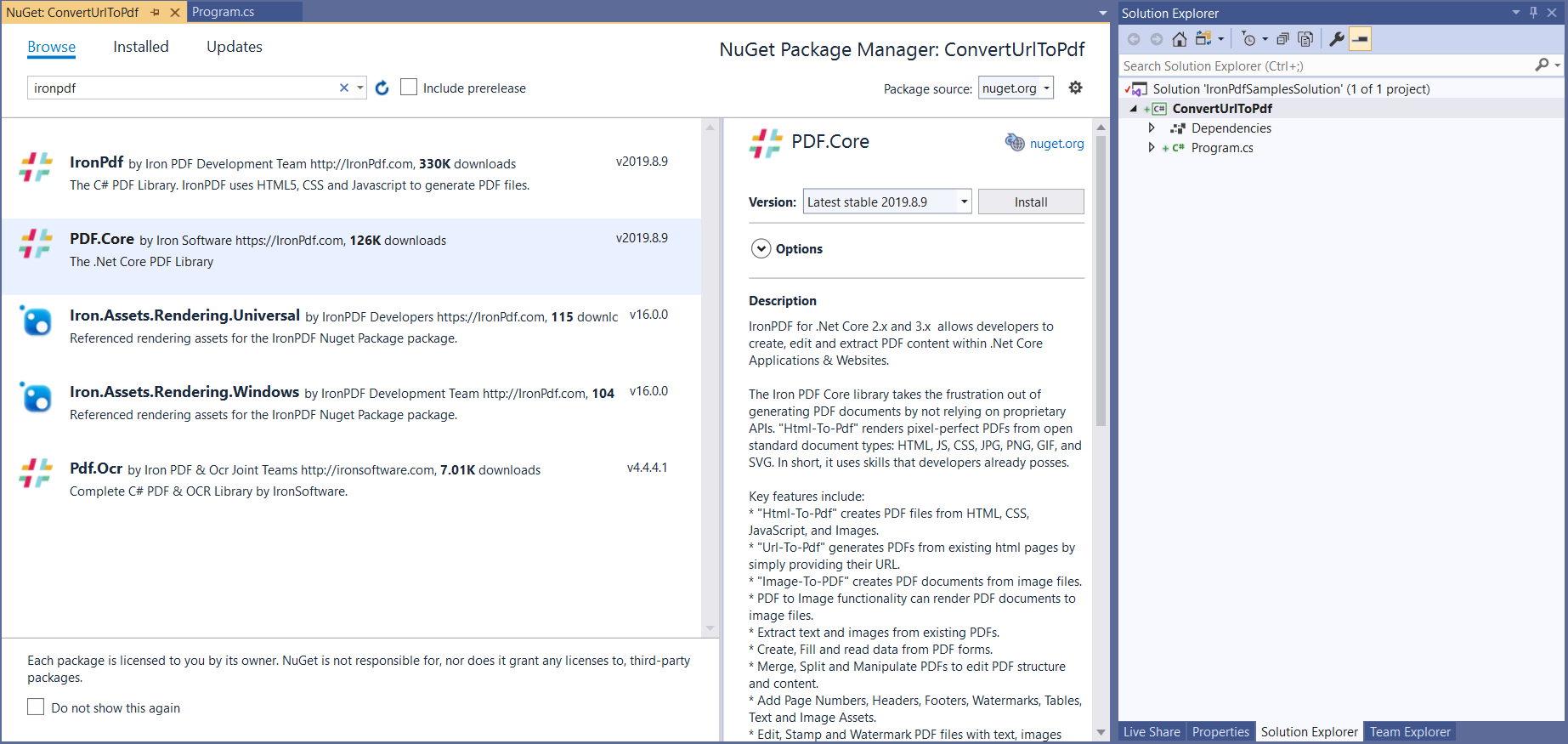
### Sample 2: HelloWorldCore .Net Core Web Application

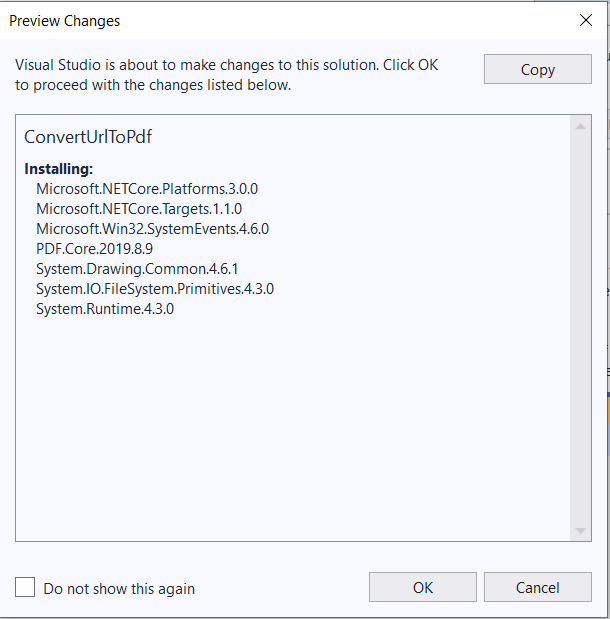
# Chapter 2: Convert to Pdf

## 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   
   
3. Choose Console App (.NET Core)  
   
4. Give our sample name “ConvertUrlToPdf” and click create   
   
5. Now we have console application created   
   
6. Add IronPdf => click install 



1. 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");  } |

1. Run and check created file wiki.pdf   
   

## Convert HTML to Pdf

### Sample: ConvertHTMLToPdf Console application

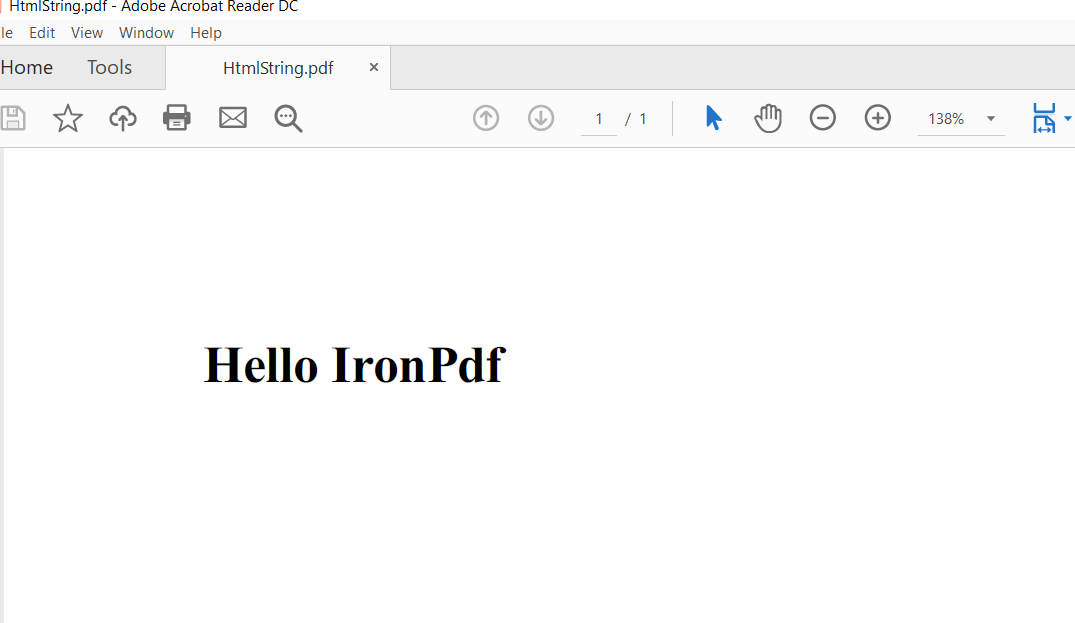
To render HTML to Pdf we have 2 way

1. Write html into string then render it
2. Write html into file and pass it path to ironPdf to render it

Rendering html string sample code like

|  |
| --- |
| static void Main(string[] args)  {  var render = new IronPdf.HtmlToPdf();  var doc = render.RenderHtmlAsPdf("<h1>Hello IronPdf</h1>");  doc.SaveAs($@"{AppDomain.CurrentDomain.BaseDirectory}\HtmlString.pdf");  } |

Result pdf like



## 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.)