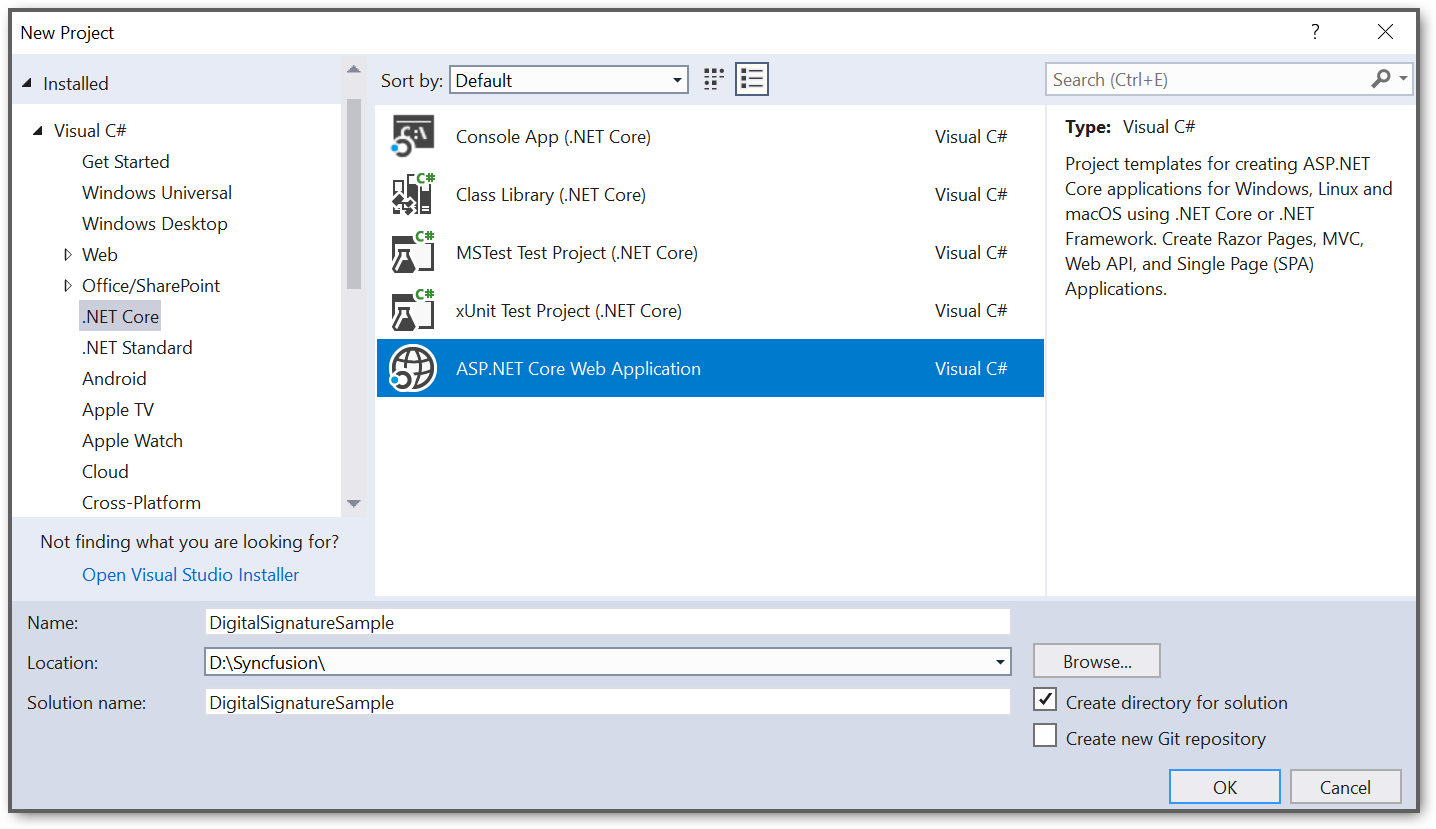
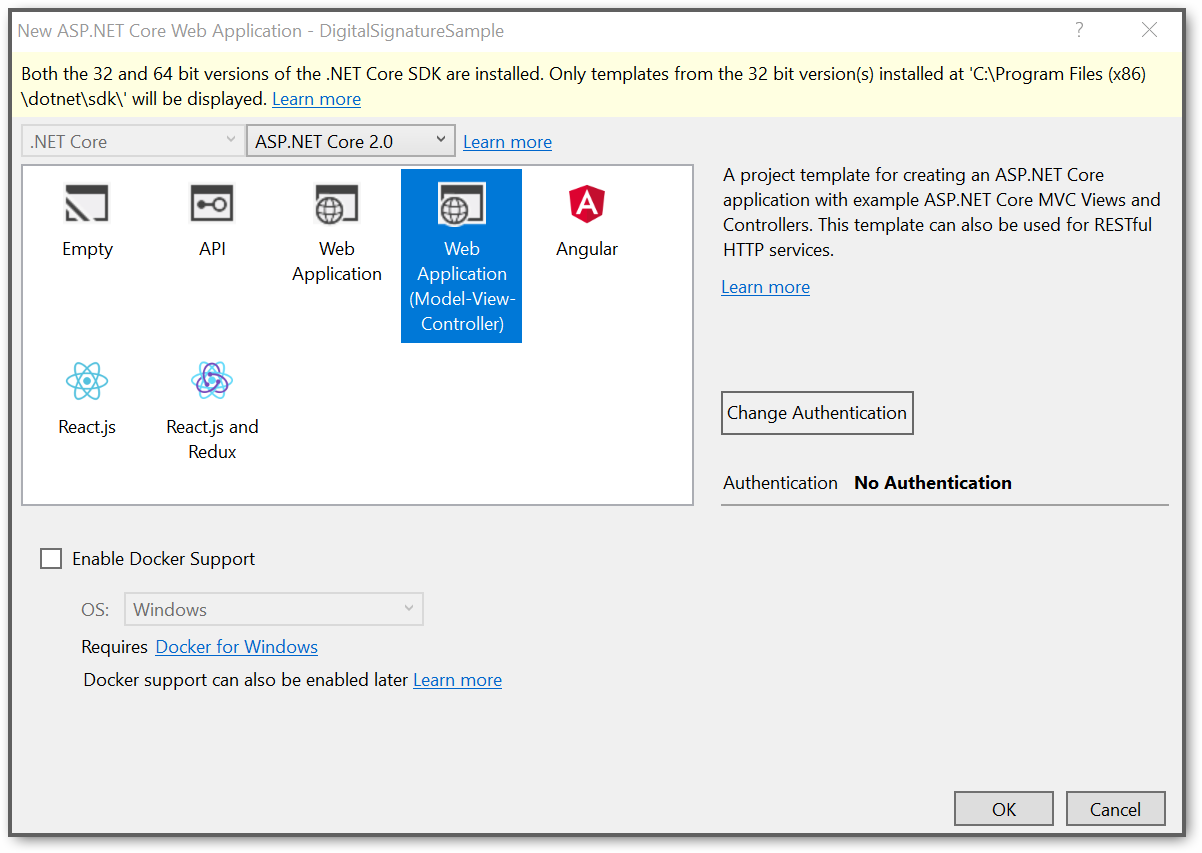
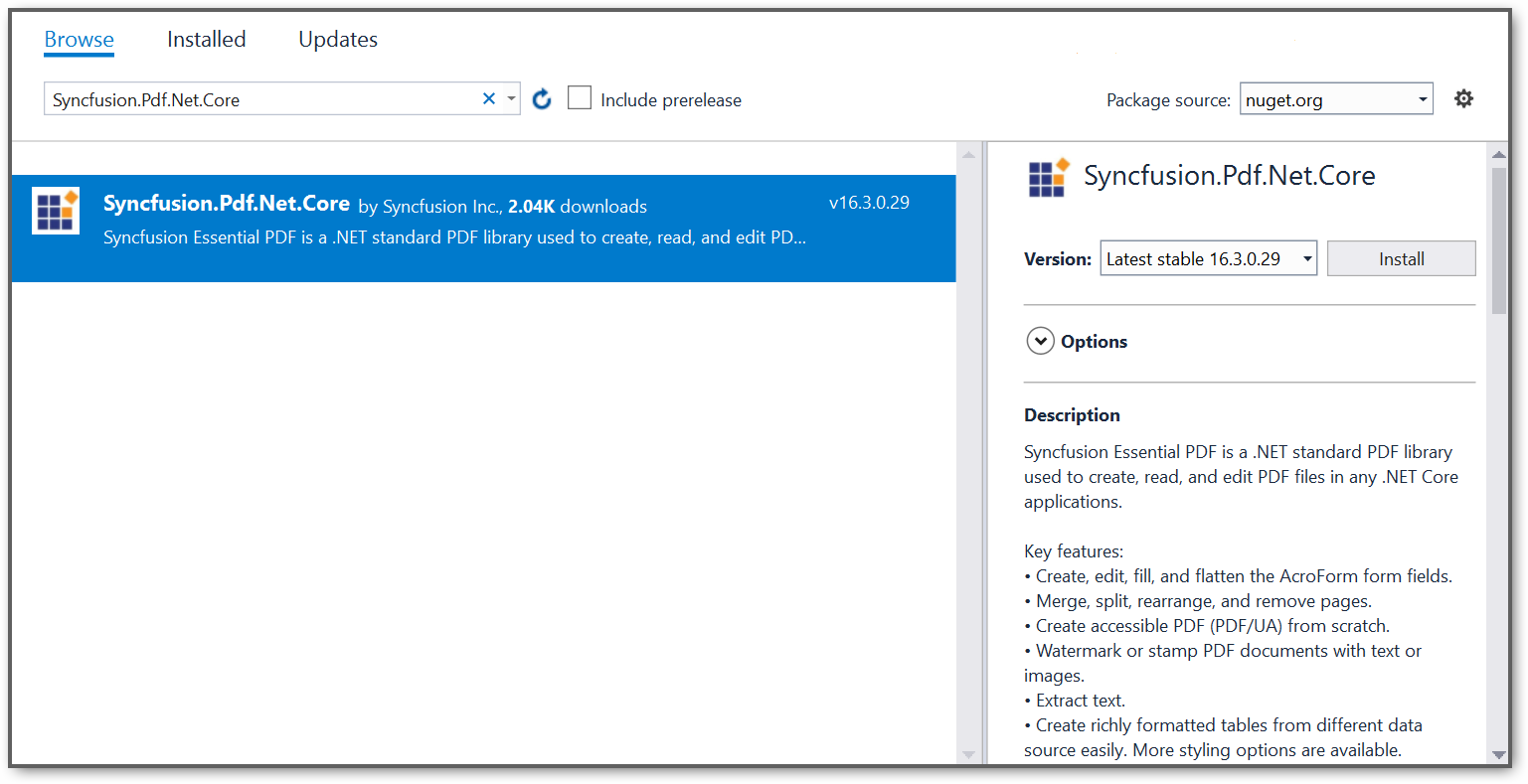
**Apply one or more digital signatures to a PDF in ASP.NET Core**

Syncfusion® Essential PDF is a [**ASP.NET Core PDF library**](https://www.syncfusion.com/document-processing/pdf-framework/net-core/pdf-library) used to create, read, and edit PDF documents. Using this library, you can apply one or more digital signature to a PDF document using C# and VB.NET.

Steps to apply one or more digital signature to PDF document programmatically:

1. Create a new C# ASP.NET Core Web application project. 
2. Select Web application pattern (Model-View-Controller) for the project. 
3. Install the [Syncfusion.Pdf.Net.Core](https://www.nuget.org/packages/Syncfusion.Pdf.Net.Core/" \t "_blank) NuGet package as reference to your .NET Standard application from [NuGet.org](https://www.nuget.org/). 
4. A default controller with name **HomeController.cs** gets added on creation of ASP.NET Core project. Include the following namespaces in that HomeController.cs file.

**C#**

**using** Syncfusion.Pdf;

**using** Syncfusion.Pdf.Parsing;

**using** Syncfusion.Pdf.Security;

**using** Syncfusion.Pdf.Graphics;

**VB.NET**

**Imports** Syncfusion.Pdf

**Imports** Syncfusion.Pdf.Parsing

**Imports** Syncfusion.Pdf.Security

**Imports** Syncfusion.Pdf.Graphics

1. A default action method named Index will be present in HomeController.cs. Right-click the Index method and select **Go To View** where you will be directed to its associated view page***Index.cshtml***.
2. Add a new button in the Index.cshtml as follows.
3. <h2>Click the button to generate PDF</h2>
4. @using (Html.BeginForm("GeneratePDF", "Home", FormMethod.Post))
5. {
6. <input type="submit" value="GeneratePDF" />
7. }

1. Add a new action method **GeneratePDF** in HomeController.cs and include the following code snippet to create a PDF file and download it.

**C#**

//Load the PDF document

FileStream docStream = **new** FileStream("SignatureFields.pdf", FileMode.Open, FileAccess.Read);

PdfLoadedDocument loadedDocument = **new** PdfLoadedDocument(docStream);

//Gets the first page of the document

PdfLoadedPage page = loadedDocument.Pages[0] **as** PdfLoadedPage;

//Gets the first signature field of the PDF document

PdfLoadedSignatureField signatureField1 = loadedDocument.Form.Fields[0] **as** PdfLoadedSignatureField;

//Creates a certificate

FileStream certificateStream1 = **new** FileStream("PDF.pfx", FileMode.Open, FileAccess.Read);

PdfCertificate certificate1 = **new** PdfCertificate(certificateStream1, "syncfusion");

signatureField1.Signature = **new** PdfSignature(loadedDocument, page, certificate1, "Signature", signatureField1);

FileStream imageStream = **new** FileStream("Student Signature.jpg", FileMode.Open, FileAccess.Read);

//Draw image

PdfBitmap signatureImage = **new** PdfBitmap(imageStream);

signatureField1.Signature.Appearance.Normal.Graphics.DrawImage(signatureImage, 0, 0, 90, 20);

//Save the document into stream

MemoryStream stream = **new** MemoryStream();

loadedDocument.Save(stream);

//Load the signed PDF document

PdfLoadedDocument signedDocument = **new** PdfLoadedDocument(stream);

//Load the PDF page

PdfLoadedPage loadedPage = signedDocument.Pages[0] **as** PdfLoadedPage;

//Gets the first signature field of the PDF document

PdfLoadedSignatureField signatureField2 = signedDocument.Form.Fields[1] **as** PdfLoadedSignatureField;

signatureField2.Signature = **new** PdfSignature(signedDocument, loadedPage, certificate1, "Signature", signatureField2);

FileStream imageStream1 = **new** FileStream("Teacher Signature.png", FileMode.Open, FileAccess.Read);

PdfBitmap signatureImage1 = **new** PdfBitmap(imageStream1);

//Draw image

signatureField2.Signature.Appearance.Normal.Graphics.DrawImage(signatureImage1, 0, 0, 90, 20);

//Saving the PDF to the MemoryStream

MemoryStream signedStream = **new** MemoryStream();

signedDocument.Save(signedStream);

//Set the position as '0'.

signedStream.Position = 0;

//Download the PDF document in the browser

FileStreamResult fileStreamResult = **new** FileStreamResult(signedStream, "application/pdf");

fileStreamResult.FileDownloadName = "DigitalSignatureSample.pdf";

return fileStreamResult;

**VB.NET**

'Load the PDF document

**Dim** docStream **As** FileStream = New FileStream("SignatureFields.pdf", FileMode.Open, FileAccess.Read)

**Dim** loadedDocument **As** PdfLoadedDocument = New PdfLoadedDocument(docStream)

 'Gets the first page of the document

**Dim** page **As** PdfLoadedPage = TryCast(loadedDocument.Pages(0), PdfLoadedPage)

 'Gets the first page of the document

**Dim** signatureField1 **As** PdfLoadedSignatureField = TryCast(loadedDocument.Form.Fields(0), PdfLoadedSignatureField)

 'Creates a certificate

**Dim** certificateStream1 **As** FileStream = New FileStream("PDF.pfx", FileMode.Open, FileAccess.Read)

**Dim** certificate1 **As** PdfCertificate = New PdfCertificate(certificateStream1, "syncfusion")

 signatureField1.Signature = New PdfSignature(loadedDocument, page, certificate1, "Signature", signatureField1)

**Dim** imageStream **As** FileStream = New FileStream("Student Signature.jpg", FileMode.Open, FileAccess.Read)

 'Draw image

**Dim** signatureImage **As** PdfBitmap = New PdfBitmap(imageStream)

 signatureField1.Signature.Appearance.Normal.Graphics.DrawImage(signatureImage, 0, 0, 90, 20)

 'Save the document into stream

**Dim** stream **As** MemoryStream = New MemoryStream()

 loadedDocument.Save(stream)

 'Load the signed PDF document

**Dim** signedDocument **As** PdfLoadedDocument = New PdfLoadedDocument(stream)

 'Load the PDF page

**Dim** loadedPage **As** PdfLoadedPage = TryCast(signedDocument.Pages(0), PdfLoadedPage)

**Dim** signatureField2 **As** PdfLoadedSignatureField = TryCast(signedDocument.Form.Fields(1), PdfLoadedSignatureField)

 signatureField2.Signature = New PdfSignature(signedDocument, loadedPage, certificate1, "Signature", signatureField2)

**Dim** imageStream1 **As** FileStream = New FileStream("Teacher Signature.png", FileMode.Open, FileAccess.Read)

 'Draw image

**Dim** signatureImage1 **As** PdfBitmap = New PdfBitmap(imageStream1)

 signatureField2.Signature.Appearance.Normal.Graphics.DrawImage(signatureImage1, 0, 0, 90, 20)

 'Saving the PDF to the MemoryStream

**Dim** signedStream **As** MemoryStream = New MemoryStream()

 signedDocument.Save(signedStream)

 'Set the position as '0'.

 signedStream.Position = 0

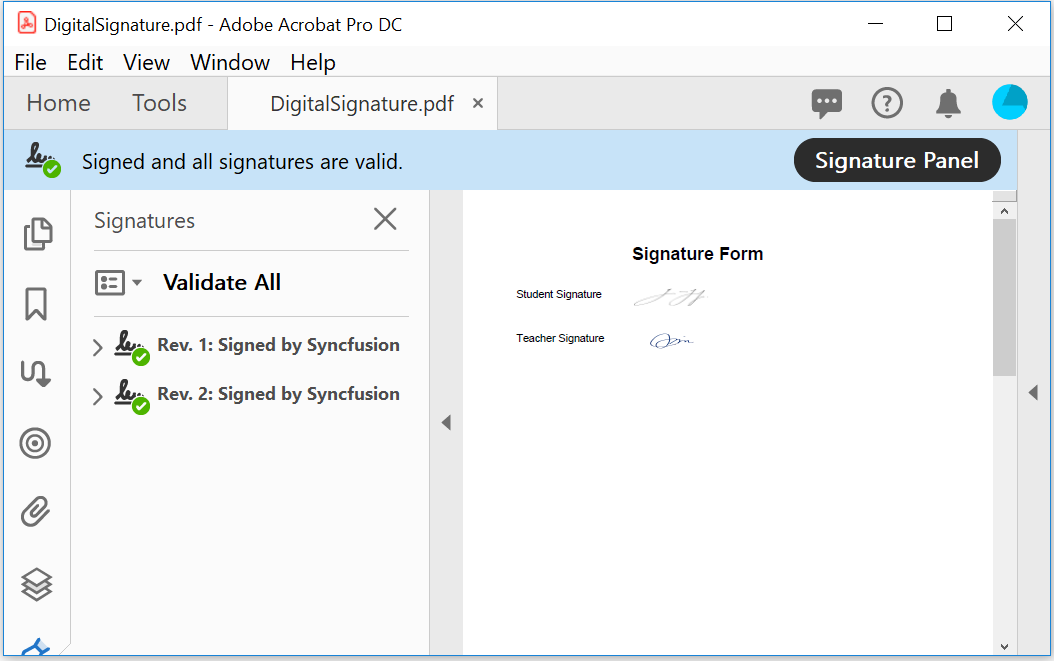
 'Download the PDF document in the browser

**Dim** fileStreamResult **As** FileStreamResult = New FileStreamResult(signedStream, "application/pdf")

 fileStreamResult.FileDownloadName = "DigitalSignatureSample.pdf"

**Return** fileStreamResult

A complete working sample can be downloaded from [DigitalSignatureSample.zip](https://www.syncfusion.com/downloads/support/directtrac/general/ze/DigitalSignatureSample-2027118639).

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

Note:

Starting with v16.2.0.x, if you reference Syncfusion® assemblies from trial setup or from the NuGet feed, include a license key in your projects. Refer to [link](https://help.syncfusion.com/common/essential-studio/licensing/license-key) to learn about generating and registering Syncfusion® license key in your application to use the components without trail message.

**Conclusion**

I hope you enjoyed learning about how to apply one or more digital signatures to a PDF in ASP.NET Core.

You can refer to our [**ASP.NET Core PDF**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) 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 Core PDF example**](https://www.syncfusion.com/demos/fileformats/pdf-library) to understand how to create and manipulate data in the .NET PDF.

For current customers, you can check out our 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**](https://www.syncfusion.com/downloads/aspnetcore-js2) to check out our ASP.NET Core PDF and other .NET Core 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/169347/response-exported-pdf), [**Direct-Trac**](https://support.syncfusion.com/create), or [**feedback portal**](https://www.syncfusion.com/feedback). We are always happy to assist you!