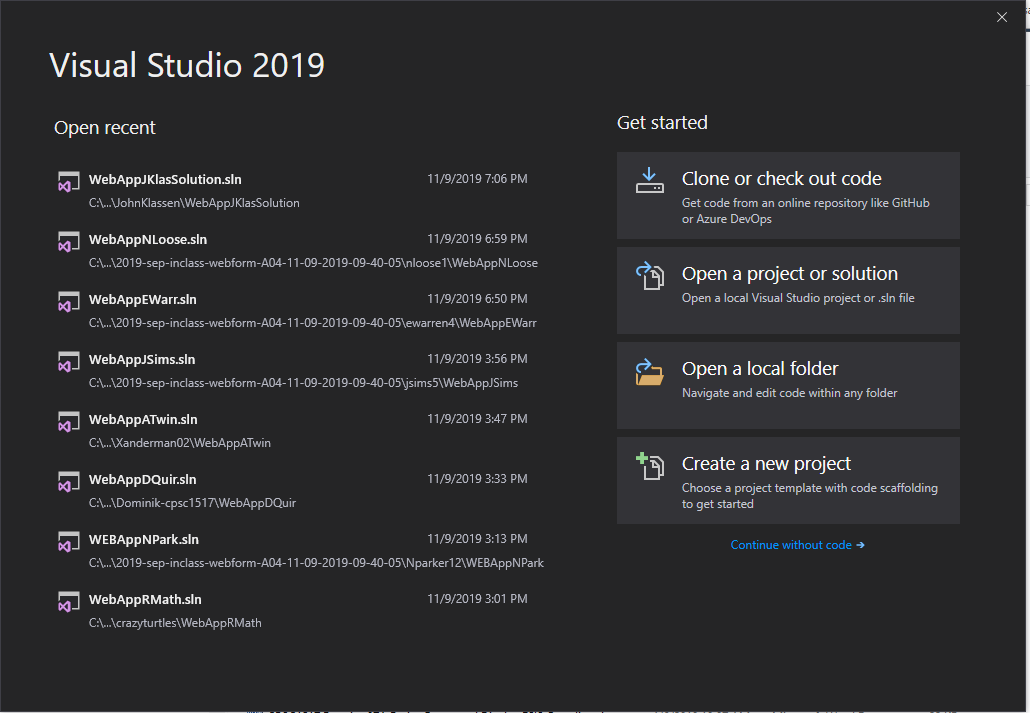
**HOW TO: Create a Visual Studio Razor Web Application from scratch**.

Open Visual Studio 2019. If this is your first time opening the software, you will be prompted for your preferred IDE style and coding. Choose C#. Choose your preferred interface color from one of the 4 choices.

On the main window, select Create a new project for a new project.

Previous projects you have opened and worked with



There are many project templates within Visual Studio. You can limit your choices by filtering items under the Language, Platform and Project Type lists.

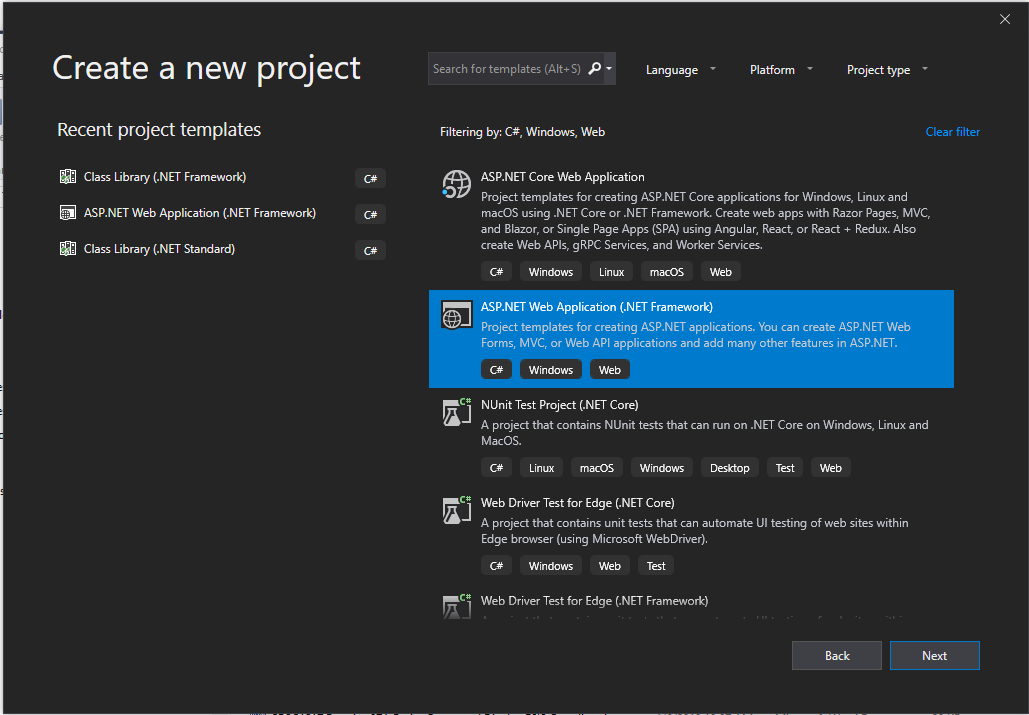
Under Language select C#, under Platform select Windows and under Project type select Web.

Now you have a short list of projects from which to choose.

Select ASP.Net Web Application (.Net Framework). **VERY IMPORTANT:** Do NOT select (.NET Core)

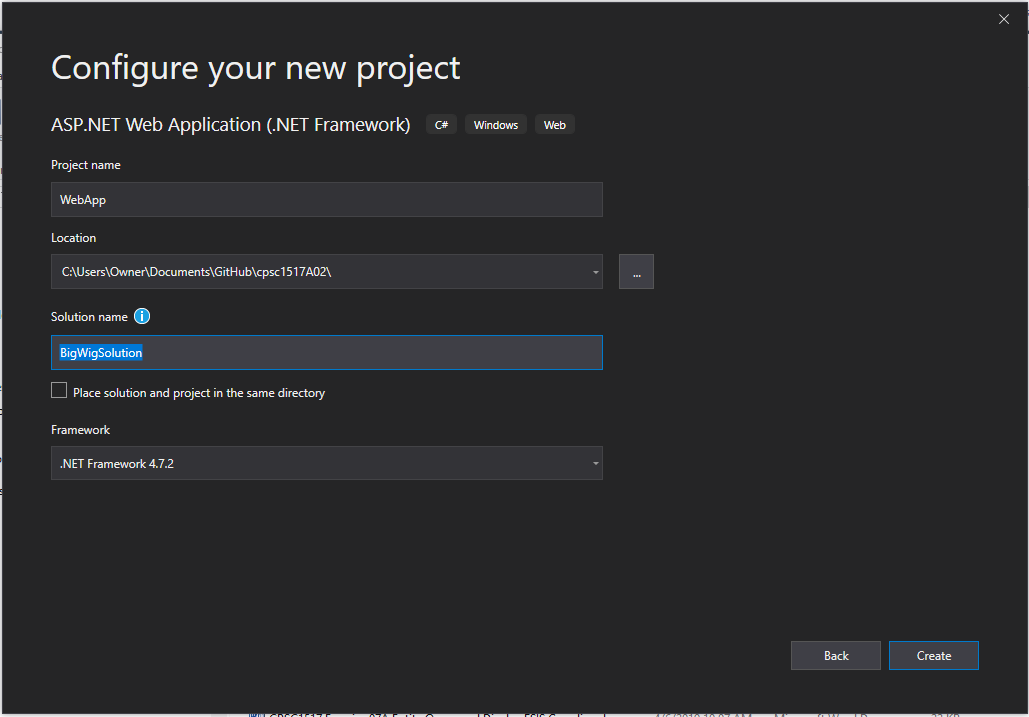
Press Next.

Previous project templates that have been used



The next step is to name and select your storage location for your work. The project name and solution name can be anything you desire. Here we have used “WebApp” for the web application project and “BigWigSolution” for the solution name. One solution can have a collection of projects. The Location is where you want to store your work.

Press Create.



There are several Web Application forms from which to choose.

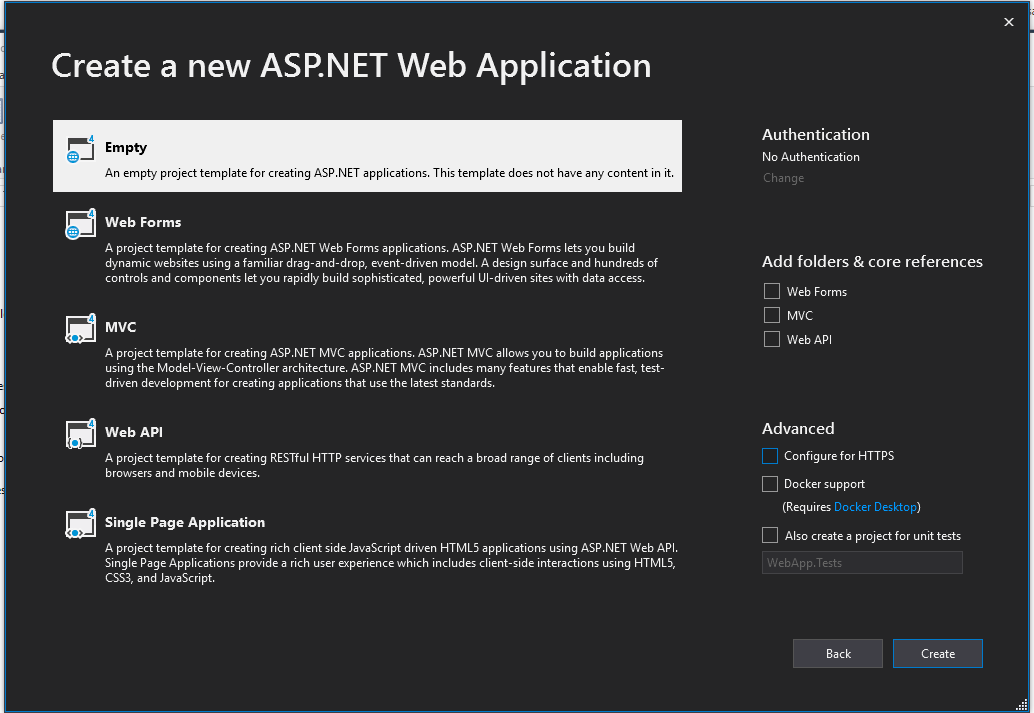
Select “Empty” as this will create a basic project with no additional files to help create your application. We will use this when we examine Razor in the first part of the course.

Use of security can be set by changing the Authentication. Starting with vs2019, projects are created with a default setting for allowing Certifications. This is under Advanced.

**VERY IMPORTANT:** Turn off this default by **unchecking** the Configure for HTTPS check box.

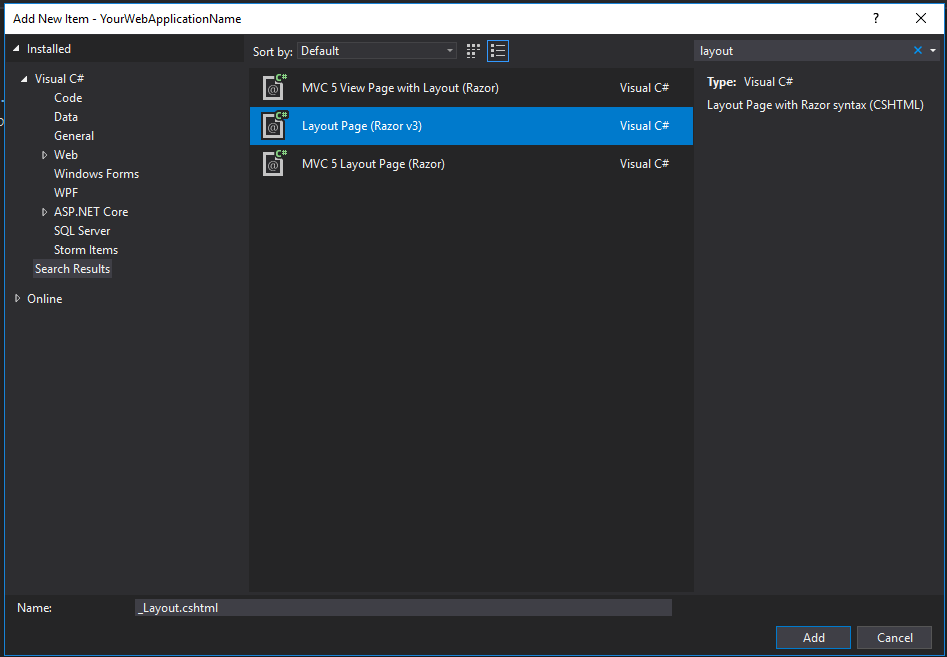
Press Create.

Razor Applications

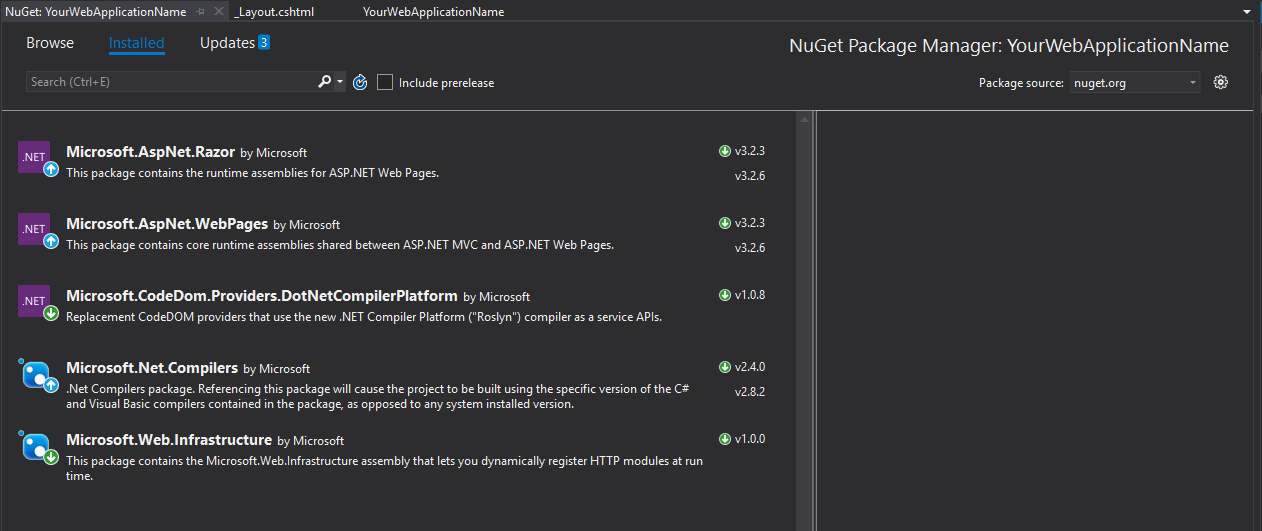


Right click your empty project name, “WebApp” in this case 🡪 Add 🡪 New Item. Click “Razor” on the left-hand side under “Web”.

Select Layout Page (Razor v3). Change name to be \_Layout.cshtml. Press “Add”. This will add several NuGet packages to your empty project. You can see the added packages by viewing the “packages.config” file in the project folder.

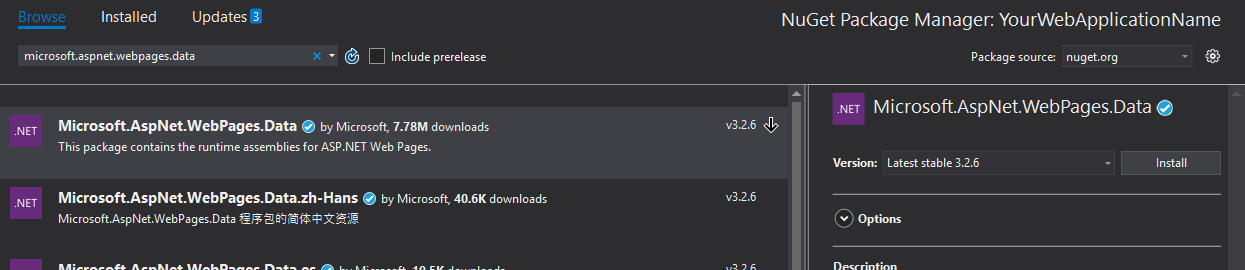


Right click project name, “WebApp” in this case 🡪 Manage NuGet Packages.

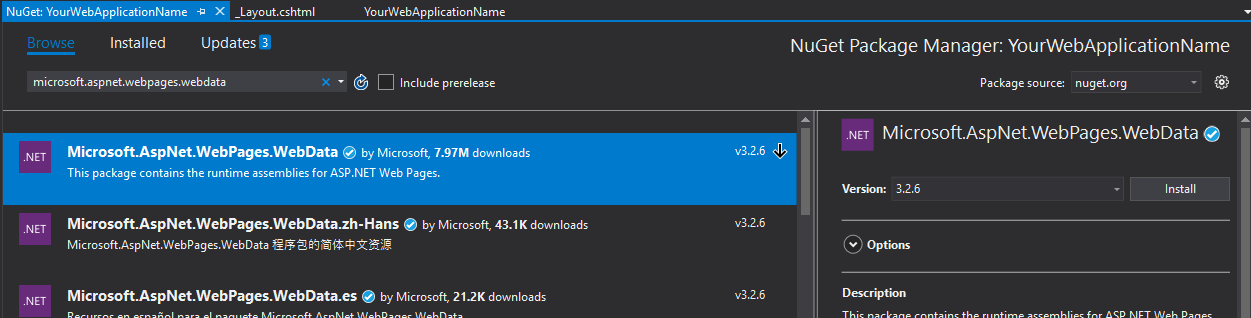


Add four additional packages that will be needed for Database handling.

Select “Browse” and enter Microsoft.aspnet.webpages.data. Select the first package and install. Accept the license.

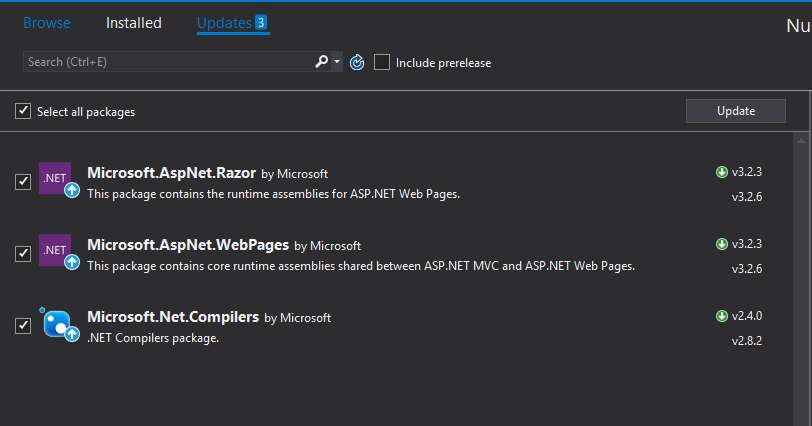


Enter Microsoft.aspnet.webpages.webdata. Select the first package and install. Accept the license.

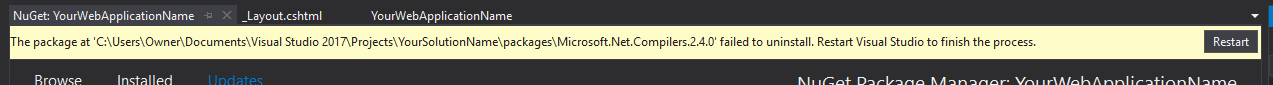


Enter Bootstrap. Select the first package. Then go to the version dropdown and select version 3.3.7. Press Install. Note: jQuery will also be installed at this time.

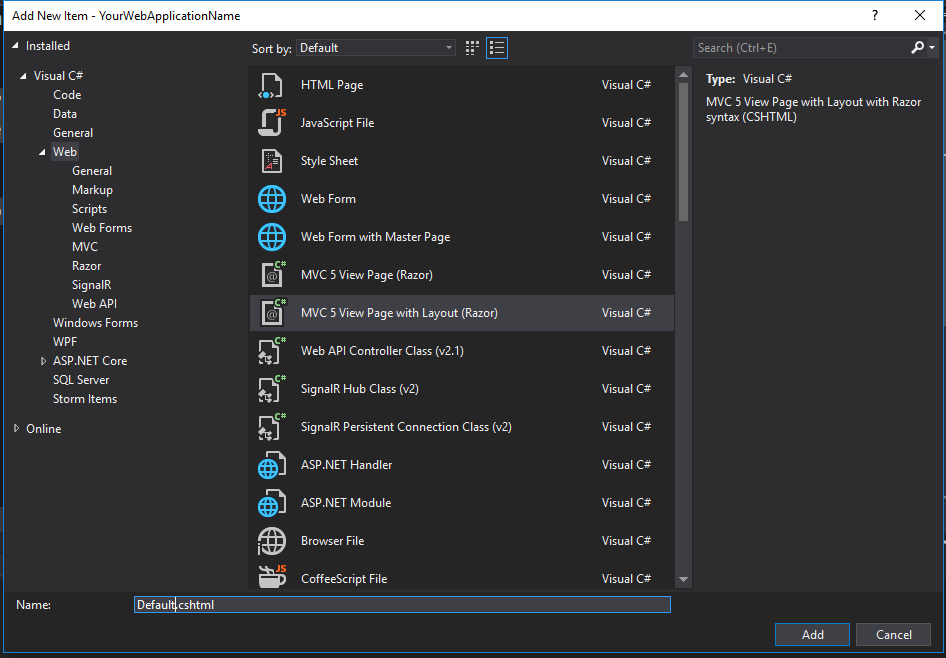
Select Updates if there are any packages to be updated. Empty the search textbox. Check everything but the Bootstrap and jQuery. Press Update. Then update the jQuery but only to version 3.3.1.

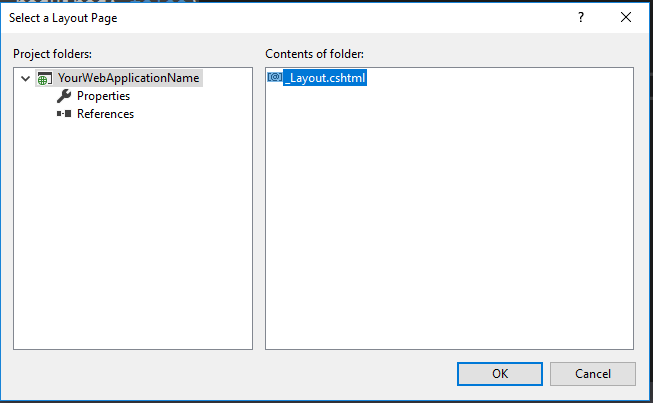


You may get the following yellow banner warning. DO NOT ignore this. Press Restart.

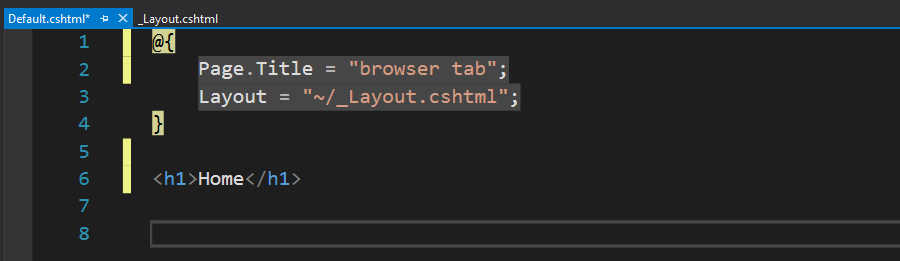


Add your first content page. Right click project name, “WebApp” in this case 🡪 Add 🡪 New Item. Click “MVC” on the left-hand side under “Web”. Select “MVC 5 View Page with Layout (Razor)” and change the name to Default.cshtml. Click “Add”. Select the \_Layout.cshtml page and then click “OK”.





Add Page.Title and an h1 tag.



Press Ctrl + F5 to run. You should see “browser tab” on the browser tab of the page and your h1 text.