

Ideas for Adding Your Own Code

1 CONTENTS

- 1 Introduction 2
- 2 Add a New Project 2
 - 2.1 Add a New MVC Project..... 2
 - 2.2 Add a New Class Library Project..... 4
- 3 Remove the Generated Web Application Project 6

Adding Projects

1 INTRODUCTION

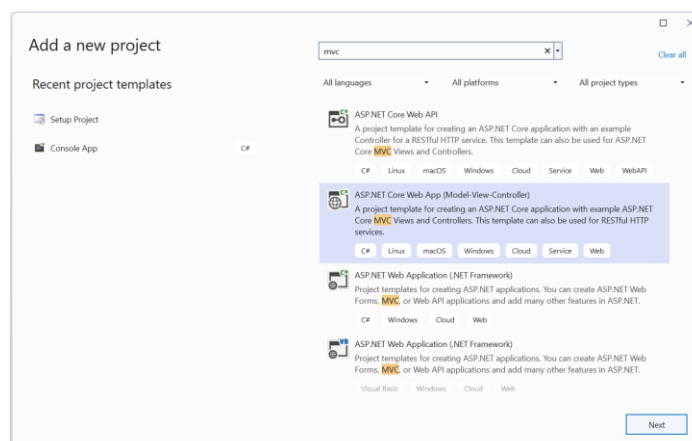
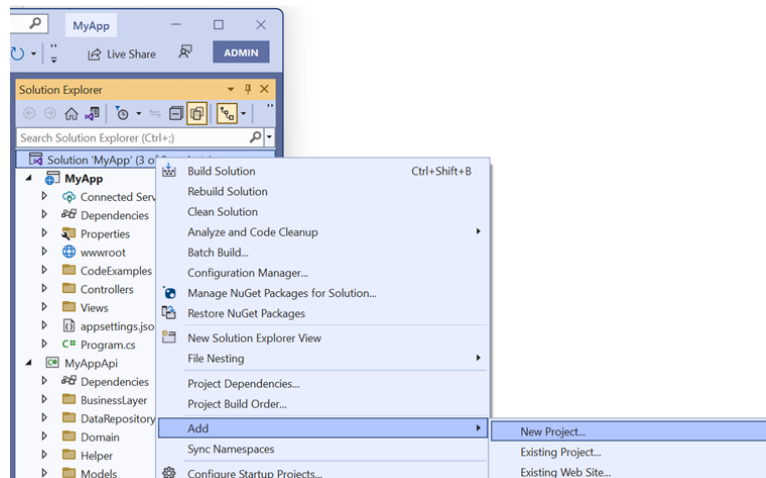
In addition to the *Customization by Adding Your Own Code* tutorial, this topic will show you some ideas on how to add your own code to the AspCoreGen 9.0 MVC's generated code.

2 ADD A NEW PROJECT

You can your own projects to the solution.

2.1 ADD A NEW MVC PROJECT

You can add a new web application and use it to add all your presentation layer code in it.



Configure your new project

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows Cloud Service Web

Project name

Location
 ...

Project will be created in "C:\inetpub\wwwroot\MyApp\MyOwnWebApp"

Back Next

Additional information

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows Cloud Service Web

Framework

Authentication type

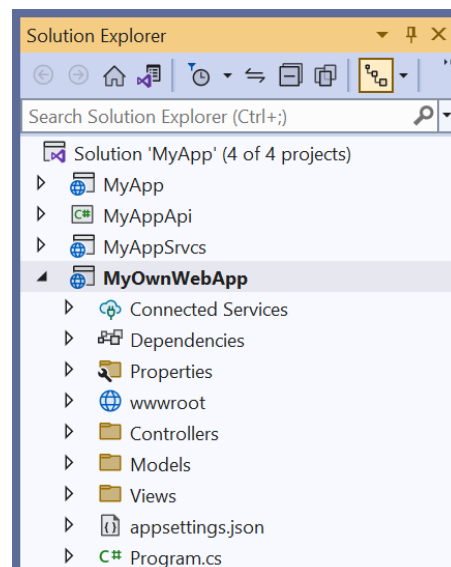
☒ Configure for HTTPS

☐ Enable Docker

Docker OS

☐ Do not use top-level statements

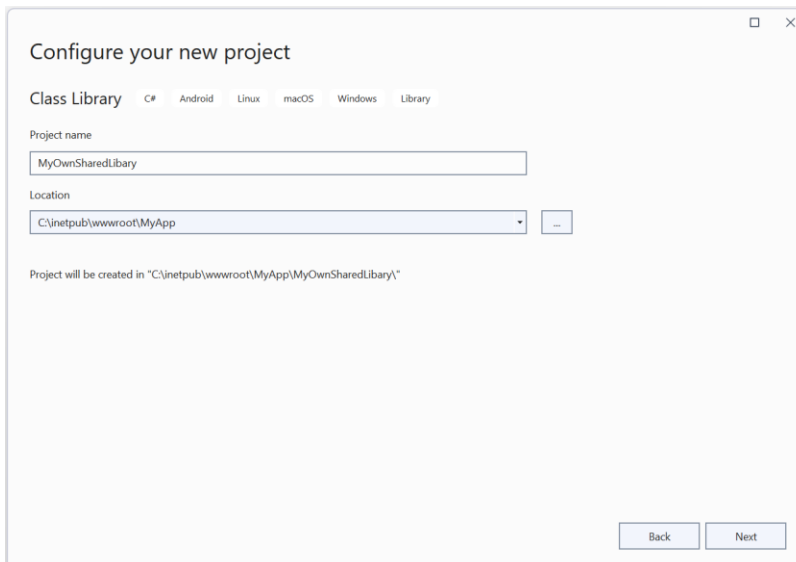
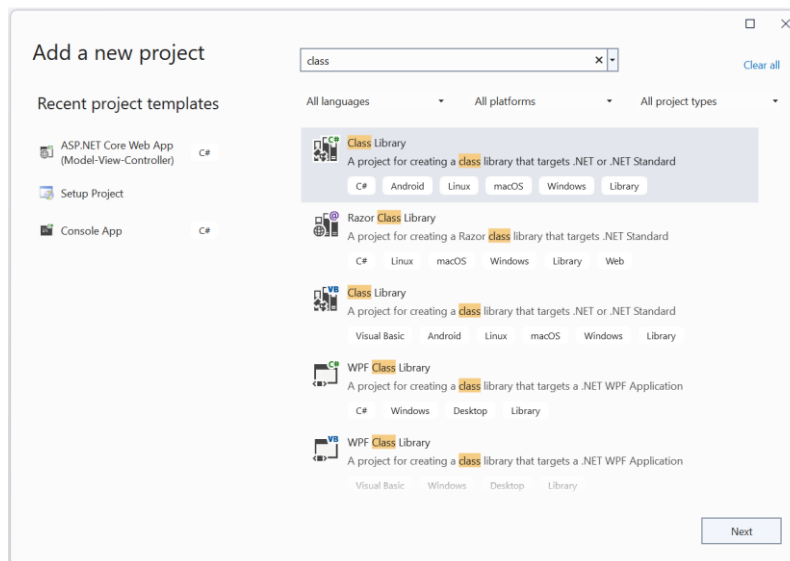
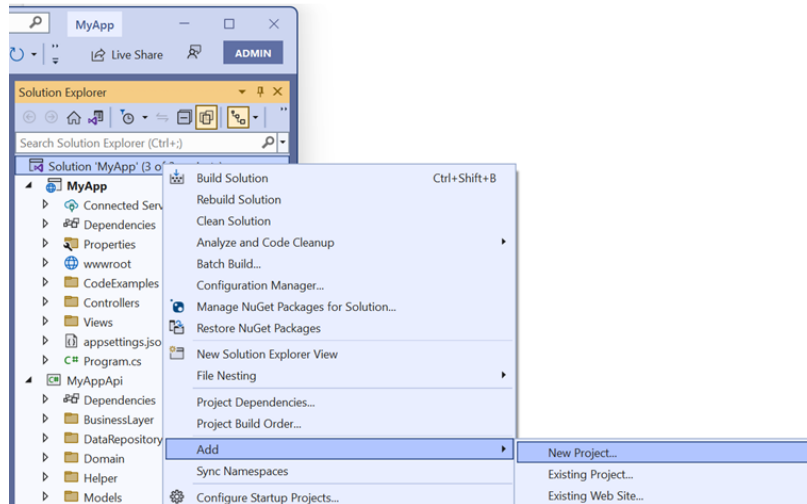
Back Create

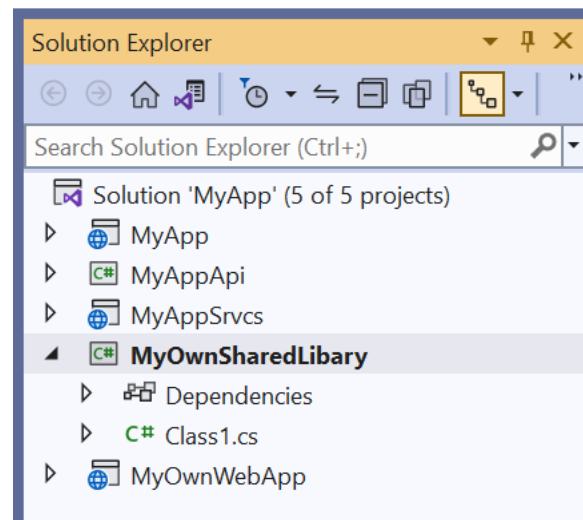
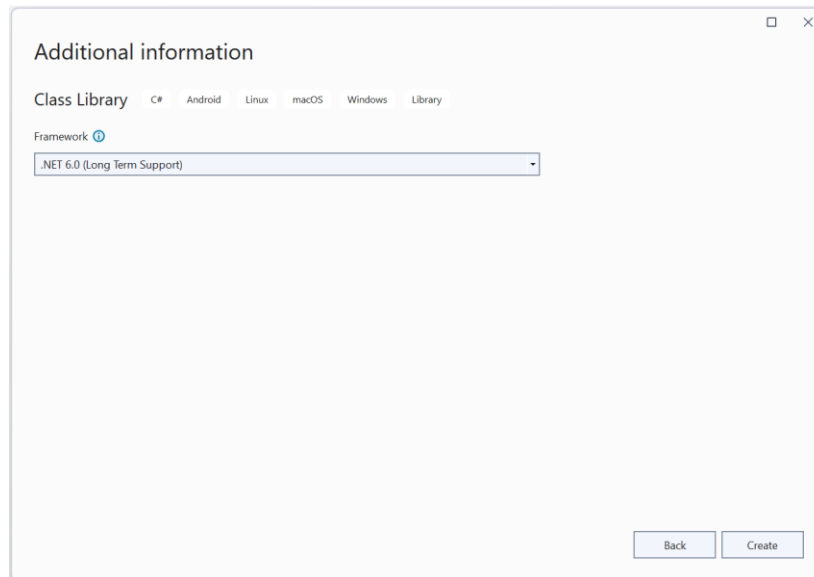


Note: In this example, even if you keep regenerating code for *MyApp* using *AspCoreGen 9.0 MVC*, your newly created web application (*MyOwnWebApp*) and everything in it will not get overwritten.

2.2 ADD A NEW CLASS LIBRARY PROJECT

You can add a new *Class Library Project* and use it to add all your presentation layer code in it.



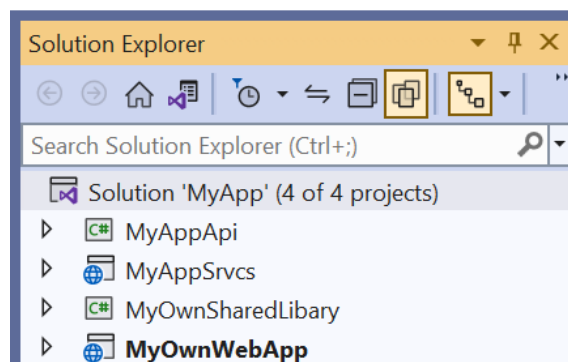
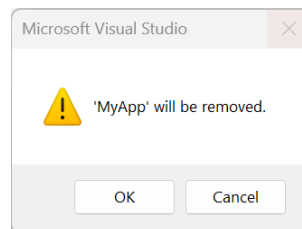
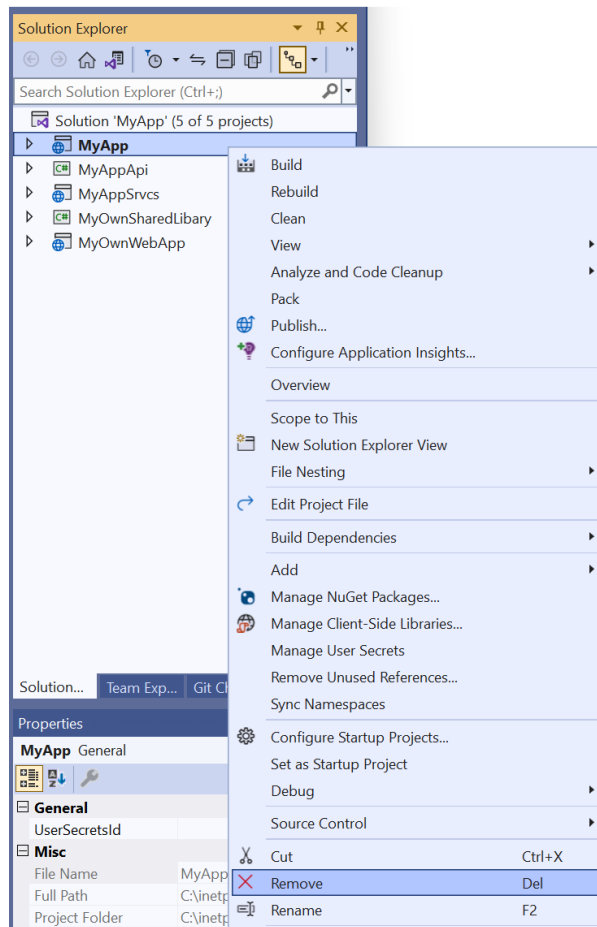


Note: In this example, even if you keep regenerating code for *MyApp* using ASP.NET Core 9.0 MVC, your newly created *Class Library Project* (*MyOwnSharedLibrary*) and everything in it will not get overwritten.

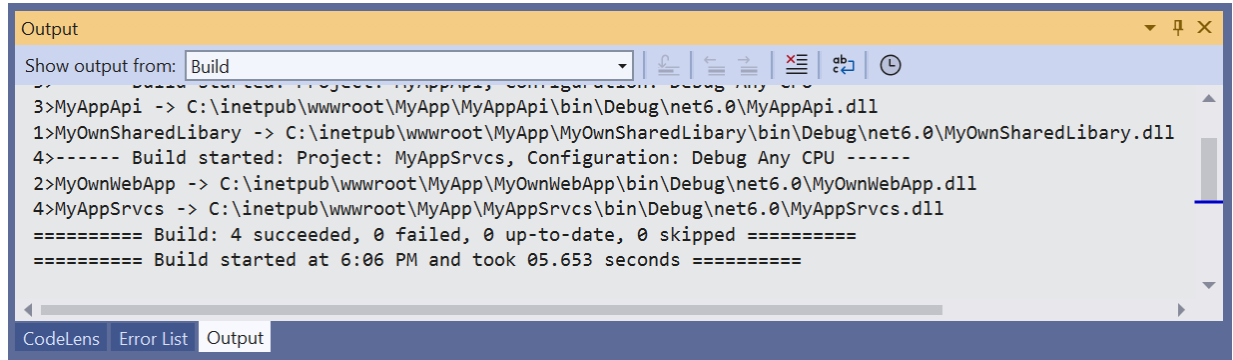
At this point you probably get what we mean. You can add new projects to the generated web application, in this example just make sure it's not named *MyApp*, *MyAppApi*, and *MyAppSrvs*.

3 REMOVE THE GENERATED WEB APPLICATION PROJECT

Yes you can remove the Web Application Project that was generated by ASP.NET Core 9.0 MVC. In this example, it is the *MyApp* Web Project.



When you rebuild the solution it builds just fine.

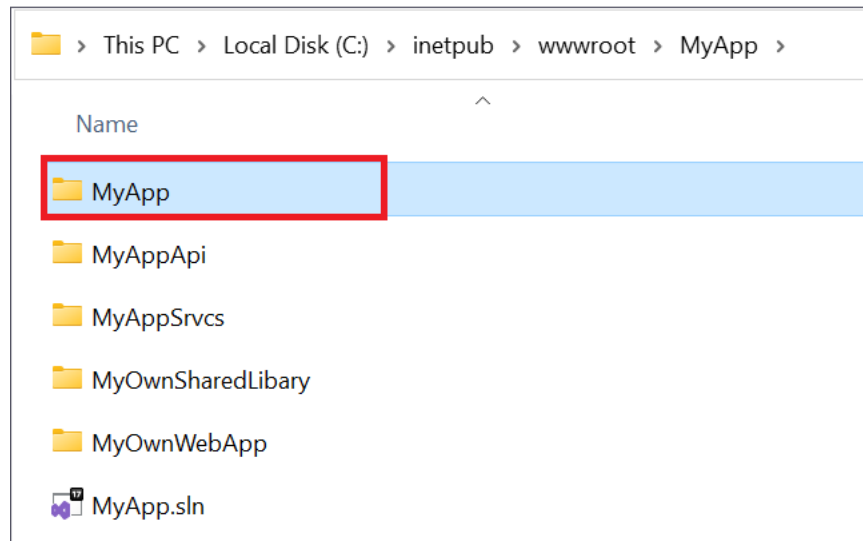


```

Output
Show output from: Build
3>MyAppApi -> C:\inetpub\wwwroot\MyApp\MyAppApi\bin\Debug\net6.0\MyAppApi.dll
1>MyOwnSharedLibrary -> C:\inetpub\wwwroot\MyApp\MyOwnSharedLibrary\bin\Debug\net6.0\MyOwnSharedLibrary.dll
4>----- Build started: Project: MyAppSrvcs, Configuration: Debug Any CPU -----
2>MyOwnWebApp -> C:\inetpub\wwwroot\MyApp\MyOwnWebApp\bin\Debug\net6.0\MyOwnWebApp.dll
4>MyAppSrvcs -> C:\inetpub\wwwroot\MyApp\MyAppSrvcs\bin\Debug\net6.0\MyAppSrvcs.dll
===== Build: 4 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
===== Build started at 6:06 PM and took 05.653 seconds =====
  
```

Note: In this example, when you regenerate code for *MyApp*, the web application is still continuously generated in the same folder, but it will be excluded from the *Solution*. So you can still use those generated MVC files as examples when you build your application.

Note: You can also open the *MyApp.csproj* on it's own.



You can read end-to-end tutorials on more subjects on using AspCoreGen 9.0 MVC Professional Plus that came with your purchase. These tutorials are available to customers and are included in a link on your invoice when you purchase AspCoreGen 9.0 MVC Professional. Download example shown here at: <https://junnark.com/CustomProjectSamples/acg6mvc/StoredProcWa.zip>

Note: Some features shown here are not available in the Express Edition. The code in this tutorial is available for download for paying customers only, please email us at Software Support for more information.

End of tutorial.