Ideas for Adding Your Own Code

1 CONTENTS

1	I	ntro	oduction	. 2
2	A	٩dd	a New Project	. 2
	2.1		Add a New MVC Project	. 2
	2.2		Add a New Class Library Project	.4
3	F	Rem	ove 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 6.0 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	
roject name	
MyOwnWebApp	
ocation	
C\inetpub\wwwroot\MyApp •	
roject will be created in "C.\Inetpub\wwwroot\MyApp\MyOwnWebApp\"	
roject will be created in "C:\Inetpub\wwwroof\MyApp\MyOwnWebApp\"	
rojact will be created in *C:\inetpub\wwwroot\MyApp\MyOwnWebApp*	
roject will be created in "C.\inetpub\wwwroof\MyApp\MyOwnWebApp\"	
roject will be created in "C\inetpub\wwwroot\MyApp\MyOwnWebApp\"	

Iditional information P.NET Core Web App (Model-View-Controller) ce Linx macOs Windows Coud Service Web ework () T60 (Long Term Support) • entication type () te configure for HTTPS () inside Docker () er OS () co not use top-level statements ()					
Iditional information P.NET Core Web App (Model-View-Controller) C* Linux macOS Windows Cloud Service Web ework () T6.0 (Ing Term Support) • entication type () te configure for HTTPS () inable Docker () er OS () To to not use top-level statements ()					
P.NET Core Web App (Model-View-Controller) C* Linux macOS Windows Cloud Service Web ework () T6.0 (Ing Term Support) • emtication type () te configure for HTTPS () nable Docker () er OS () con out use top-level statements ()	Additional information				
ework () T 6.0 (Long Term Support) • entication type () te • Configure for HTIPS () inable Docker () err OS () to on tuse top-level statements ()	ASP.NET Core Web App (Model-View-Controller)	cOS Windows	Cloud Service	Web	
1 6.0 (Long Term Support) entication type ① ne Configure for HTTPS ① inable Docker ① err OS ① re Do not use top-level statements ①	Framework 🕦				
entication type ① ne Configure for HTTPS ① inable Docker ① err OS ① r Do not use top-level statements ①	.NET 6.0 (Long Term Support)	•			
ne Configure for HTTPS Configure for HTTPS Configure C	Authentication type 🚯				
Configure for HTTPS () inable Docker () er OS () III () Do not use top-level statements ()	None	•			
inable Docker ① err OS ① rr or on use top-level statements ①	Configure for HTTPS ()				
er OS 💿 IX Do not use top-level statements 🛈	Enable Docker 🚯				
∞ •	Docker OS 🚯				
Do not use top-level statements 🛈		-			
	Do not use top-level statements 🚯				
Back Create			Back	Cri	eate



Note: In this example, even if you keep regenerating code for *MyApp* using AspCoreGen 6.0 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.

MyApp MyApp The Live Share		
© ⊙ ☆ √ I o · ← Search Solution Explorer (Ctrl-		
Image: Solution MyApp'(3 of MyApp) Image: Solution MyApp'(3 of MyApp) Image: Image: Solution MyApp'(3 of MyApp) Image: Solution MyApp'(3 of MyApp) Image: Image: Solution MyApp'(3 of MyApp) Image: Solution MyApp'(3 of MyApp) Image: Image: Solution MyApp'(3 of MyApp) Image: Solution MyApp'(3 of MyApp) Image: Image: Solution MyApp'(3 of MyApp) Image: Solution MyApp'(3 of MyApp) Image: Image: Image: Solution MyApp'(3 of MyApp) Image: Solution MyApp'(3 of MyApp) Image: Image: Image: Solution MyApp'(3 of MyApp) Image: Solution MyApp'(3 of MyApp) Image: Image: Image: Image: Solution MyApp'(3 of MyApp) Image: Solution MyApp'(3 of MyApp) Image: Im	Build Solution Ctrl+Shift+B Rebuild Solution Clean Solution Clean Solution Analyze and Code Cleanup Batch Build Configuration Manager Manage NuGet Packages for Solution Restore NuGet Packages	
	 New Solution Explorer View File Nesting Project Dependencies Project Build Order Add 	New Orniget
Domain Domain Helper Models	Sync Namespaces Configure Startup Projects	Existing Project Existing Web Site



Onfigure your new project		
configure your new project		
Jacobianany Cit Andreid Davis marche Windows Dissay		
LIGSS LIDI di y Con Android Linux macos Windows Library		
roject name		
MyQwnSharedLibary		
ocation		
C:\inetpub\wwwroot\MyApp •		
roject will be created in "C:\inetpub\www.root(My/App)MyOwnSharedLibary\"		
oject will be created in "C:\inetpub\wwwroot\MyApp\MyCwnSharedLibary\"		
oject will be created in "C\inetpub\wwwroot\MyApp\MyCwnSharedLibary\"		
oject will be created in "C\inetpub\wwwroot\MyApp\MyCwnSharedLibary\"		
oject will be created in "C:\inetpub\wwwroot\MyApp\MyCwnSharedLibary\"		
oject will be created in "C\inetpub\wwwroot\MyApp\MyGwnSharedLibary\"		
oject will be created in "C\inetpub\wwwroot\MyApp\MyGwnSharedLibary\"		
roject will be created in "C^inetpub\wwwroot\MyApp\MyOwnSharedLibary\"		
roject will be created in "C:\inetpub\wwwroot\MyApp\MyOwnSharedLibary\"		
'roject will be created in "C\inetpub\wwwroot\MyApp\MyOwnSharedLibary\"	Bude	

					\times
Additional information					
Class Library C# Android Linux macOS Windows	Library				
Framework 🚯					
.NET 6.0 (Long Term Support)		•			
			Back	Create	



Note: In this example, even if you keep regenerating code for *MyApp* using AspCoreGen 6.0 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*.

Yes you can remove the Web Application Project that was generated by AspCoreGen 6.0 MVC. In this example, it is the *MyApp* Web Project.







When you rebuild the solution it builds just fine.



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 6.0 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 6.0 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.