

Spike Projects Information.

For all projects in the Solution:-

- This is based on the Asp.net core v5 MVC template in VS.
- Vue.js Pack 2019 extension is also installed in VS to help with intellisense etc.

Spike One Project (Vue 2 & Vuex 3)

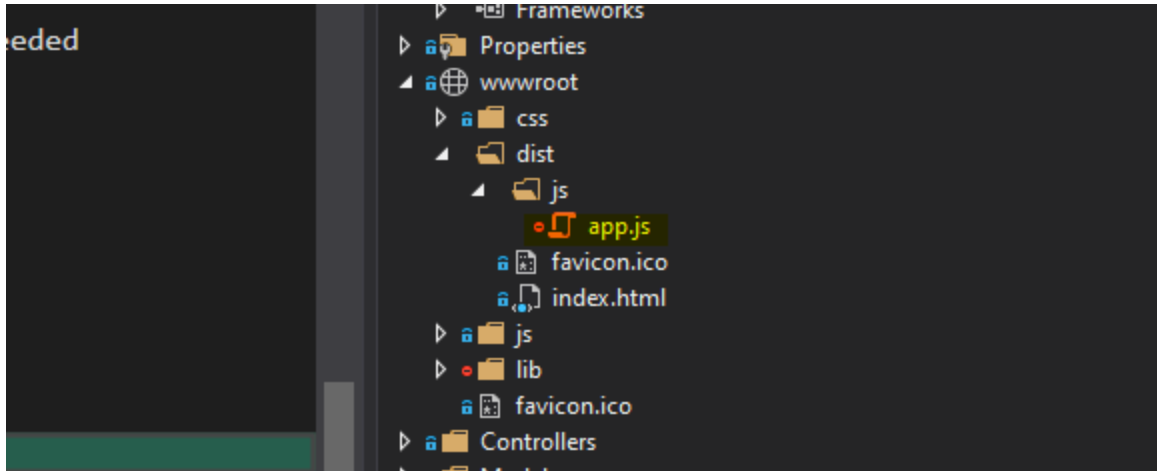
General Information

Has the following functionality:-

- Proper integrated Vue components into Razor pages in the MVC.
- Use of Vuetify style package.
- Use of Axios package for basic API calls.
- Use of Vuex for state management ****Removed now - all file refs have either been excluded or commented out**** as was causing error and components not to load into cshtml files.
- Nested components structures demo.
- Can do rudimentary debugging in chrome and breakpoints under the webpack folder in the source window.
- Vue.js dev tools chrome extension installed and works ok for component analysis.
- No webpack dependencies at all.
- Controlled by vue.config.js authored by the vue cli at initial set up.
- Vue is detected by the debug chrome tools
- <http://localhost:8084/Home/> (npm run watch in developer powershell)
- HMR not working with the watch flag.
- No webpack.config.js overrides are done in vue.config.js

But not ideal Issues:-

- Does not have separate bundling files output at mo just one output, need to look at this:-



- Does not have hot module loading working also.

To debug:

- Choose either project run or dotnet option from debugging in VS.
- Then open a terminal window off the project right click menu.
- Enter - npm run watch
- In chrome developer tools disable cache to see edit changes from IDE reload on page refresh as HMR not working yet.
- Then I should be able to navigate around the website, edit components in IDE and save file changes, the npm watch will recompile the output, then need to hit refresh to see changes made visible.

Other Concerns

Introduced vuex now gets errors in the console that can't seem to find suitable solutions to. Components on cshtml files will not load as a consequence now, although this was working before trying to introduce vuex state management. Tried to back out the install of vuex package dependency now, but won't get npm built in the terminal window, get dependency errors inside node_modules. Tried to nuke node_modules, to reinstall from scratch, but still get errors. Really frustrated with this.

Issues & Errors

```
21%  No issues found  mike-p, 4 days ago | 1 author, 3 changes
C:\Users\Mike\Source\Repos\VueCoreSpikeOne\package.json
Buffer Graph: JSON (TextBuffer)

Output
Show output from: Debug

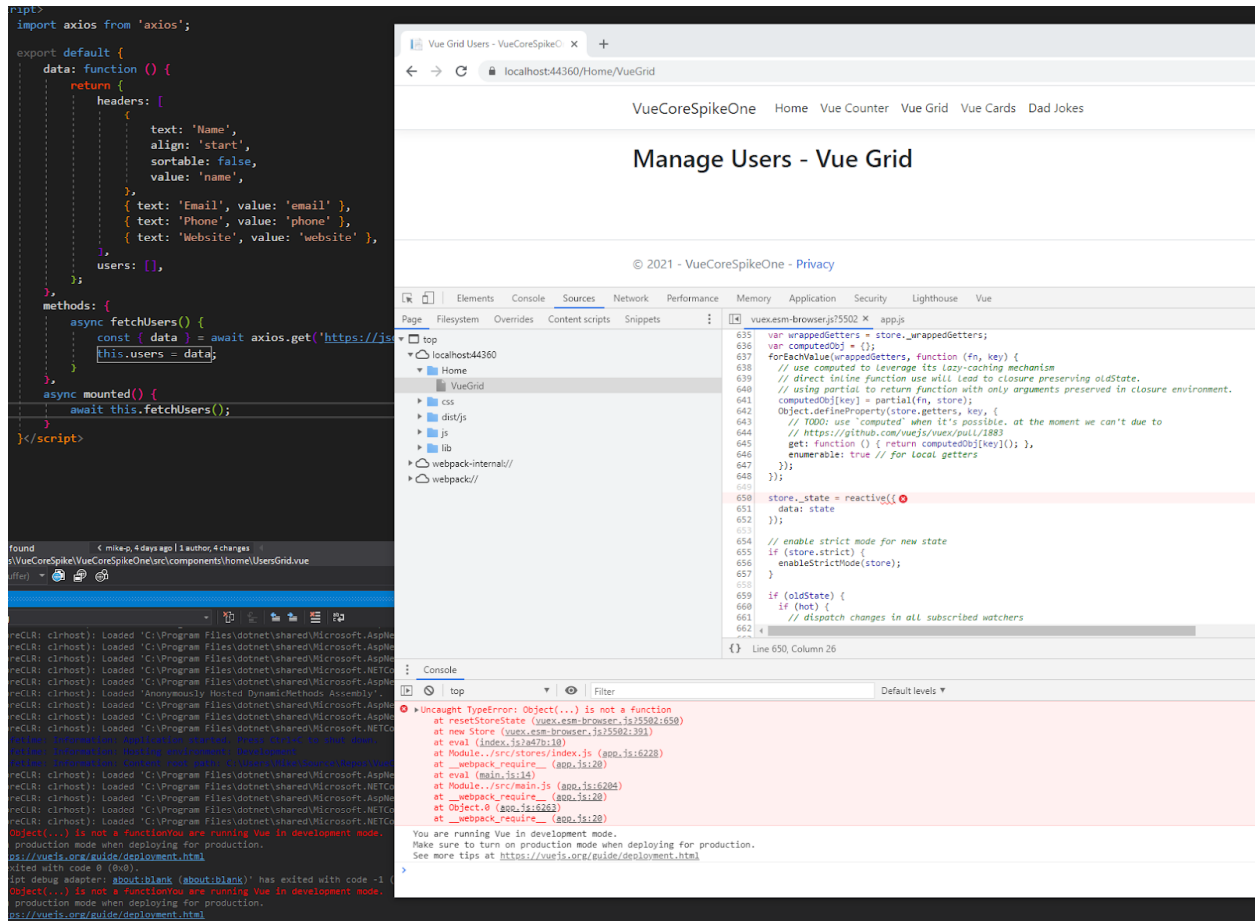
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.Net.Http.Headers.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.Html.Abstractions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.ComponentModel.Annotations.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.Metadata.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.Diagnostics.Abstractions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.IO.Pipelines.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Security.Cryptography.X509Certificates.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
Microsoft.Hosting.Lifetime: Information: Application started. Press Ctrl+C to shut down.
Microsoft.Hosting.Lifetime: Information: Hosting environment: Development
Microsoft.Hosting.Lifetime: Information: Content root path: C:\Users\Mike\Source\Repos\VueCoreSpikeOne\
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.WebUtilities.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Security.Cryptography.Primitives.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.Http.Extensions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Net.WebSockets.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Security.Principal.Windows.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
Uncaught TypeError: Object(...) is not a functionYou are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html
The thread 0x1575c has exited with code 0 (0x0).
The thread 0x16644 has exited with code 0 (0x0).
The thread 0x11488 has exited with code 0 (0x0).
The thread 0xc9c has exited with code 0 (0x0).
The thread 0xd528 has exited with code 0 (0x0).
The thread 0x128bc has exited with code 0 (0x0).
|
```

```
C:\Users\Mike\Source\Repos\VueCoreSpikeOne\package.json
Buffer Graph: HTML (TextBuffer)

Output
Show output from: Debug

'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.StaticFiles.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.Net.Http.Headers.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.Html.Abstractions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.ComponentModel.Annotations.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.Metadata.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.Diagnostics.Abstractions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.IO.Pipelines.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Security.Cryptography.X509Certificates.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
Microsoft.Hosting.Lifetime: Information: Application started. Press Ctrl+C to shut down.
Microsoft.Hosting.Lifetime: Information: Hosting environment: Development
Microsoft.Hosting.Lifetime: Information: Content root path: C:\Users\Mike\Source\Repos\VueCoreSpikeOne\
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.WebUtilities.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Security.Cryptography.Primitives.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\Microsoft.AspNetCore.Http.Extensions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Net.WebSockets.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Security.Principal.Windows.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
Uncaught TypeError: Object(...) is not a functionYou are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html
The thread 0x5 has exited with code 0 (0x0).
The program 'JavaScript debug adapter: about:blank (about:blank)' has exited with code -1 (0xffffffff).
Uncaught TypeError: Object(...) is not a functionYou are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.5\System.Buffers.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is disabled.
The thread 0x14298 has exited with code 0 (0x0).
The thread 0x14b74 has exited with code 0 (0x0).
The thread 0x13b4c has exited with code 0 (0x0).
|

Call Stack Exception Settings Immediate Window Error List Output Breakpoints Locals Watch 1
Ready
```



Fully excluded the index.js for the store from the project. Now we get components loading again.

Top tips if issues when running.

- In developer powershell window in VS, navigate to the project in question (spike one in this instance)
- type - npm run watch
- Then hit F5
- Then in the output window monitor if any feedback for errors.

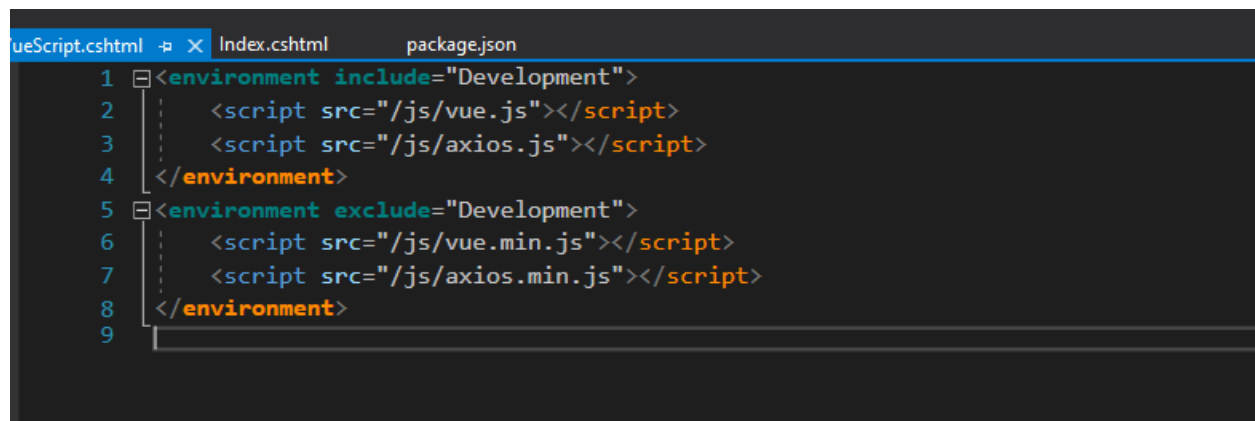
Spike Two Project (Vue 2)

General Information

- Simplest strategy. Vue is added as a pure javascript dependency. Kind of old school approach using vanilla js files. Also same for axios.
- No packages at all. Installs were done by Nuget then manually copying js files into wwwroot lib.
- Basically allows each razor to host its own Vue instance.
- Vue developer tools works and detects vue.js as referenced in the browser.
- No use of '.vue' components.
- Currently no state management usage with Vuex, possibly have to rely on the underlying session state.

Setup Information

This was also using a strategy of a partial view dropping the vue library dependencies into the razor files that needed them.

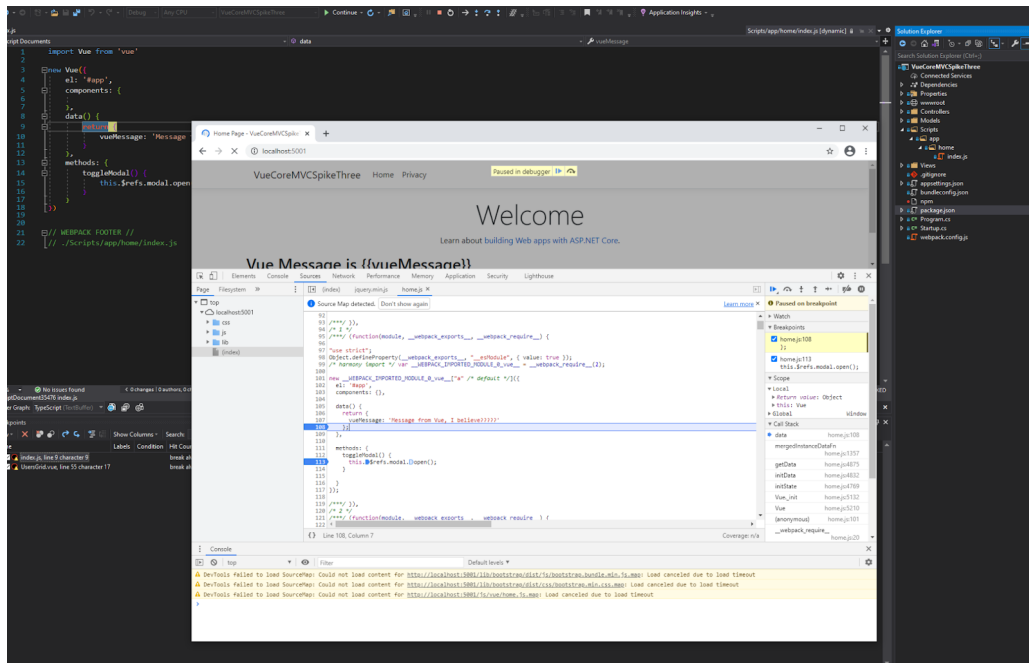


```
1 <environment include="Development">
2   <script src="/js/vue.js"></script>
3   <script src="/js/axios.js"></script>
4 </environment>
5 <environment exclude="Development">
6   <script src="/js/vue.min.js"></script>
7   <script src="/js/axios.min.js"></script>
8 </environment>
9
```


Spike Three Project (Vue 2)

General Information

- This project was taken from a walk through setup, quite old resource 2017.
<https://medium.com/corebuild-software/vue-js-and-net-mvc-b5cede228626>
- Primary purpose was to get debugging working, hitting breakpoints in VS IDE and chrome developer tools.
- Downside using old packages for webpack, vue, vuetify etc.
- No state management Vuex package in use.
- Most complex to get up and running.
- Kinda need lots of webpack knowledge to work with.
- Need to create an index.js file for each razor page in the relevant script output path. Webpack will then bundle out a named equivalent js file into the wwwroot folder.
- Tried to upgrade some of the packages then hit countless errors (see other doc - MVC & Vue Project setup) with dependency issues. Something to bare in mind in future.
- Project is run by selecting the dropdown for the project itself, rather than using IISExpress.
- Vue is detected on page but, cant load information in chrome dev tooling.



Spike Four Project (Vue 3 & Vuex 4)

General Information

- Using vue cli and vue 3 version
- In chrome Vue developer tools doesn't work
- Can't get components to show in cshtml with a new way to createApp in vue 3.
- If using npm server all works ok in the browser (see link below).

The mechanism for creating a Vue instance has changed in Vue 3, this maybe why we can't get components to load. Further investigation on a file by file basis between the Spike Project One and this one, the only difference of consideration is the way we bootstrap a Vue instance in the main.js. This new approach is a change in Vue 3, which is still deemed to be in preview change and some dependent packages like Vetify are not yet updated against it. But we could be hampered with issues when we go from Vue 2 to Vue 3 in a future upgrade.

Tried to implement Bootstrap-Vue into this project and dot dependency errors, so had to back out changes.

Operation

In VS Code:-

- Open in vs code
- Type - npm run serve
- Vue loads nicely in the browser, no errors.

In VS:-

In terminal window
Type npm run build
Then F5 it

Or

Type npm run build
Then dotnet run dev

```
npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\Mike\AppData\Roaming\npm-cache\_logs\2021-05-05T11_07_27_295Z-debug.log
PS C:\Users\Mike\Source\Repos\VueCoreSpike\VueCoreMVCSpikeFour> npm run serve

> vuecoremvcspikefour@0.1.0 serve C:\Users\Mike\Source\Repos\VueCoreSpike\VueCoreMVCSpikeFour
> vue-cli-service serve

INFO Starting development server...
98% after emitting CopyPlugin

DONE Compiled successfully in 1840ms

App running at:
- Local: http://localhost:8084/
- Network: http://192.168.1.155:8084/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

<http://localhost:8084>

Issues & Errors

So tried to import some styling eg Bootstrap Vue
Then get get errors and warnings:-

```
C:\Users\Mike\Source\Repos\VueCoreSpike\VueCoreMVCSpikeFour>npm run serve

> vuecoremvcspikefour@0.1.0 serve C:\Users\Mike\Source\Repos\VueCoreSpike\VueCoreMVCSpikeFour
> vue-cli-service serve

INFO Starting development server...
98% after emitting CopyPlugin

WARNING Compiled with 1 warning

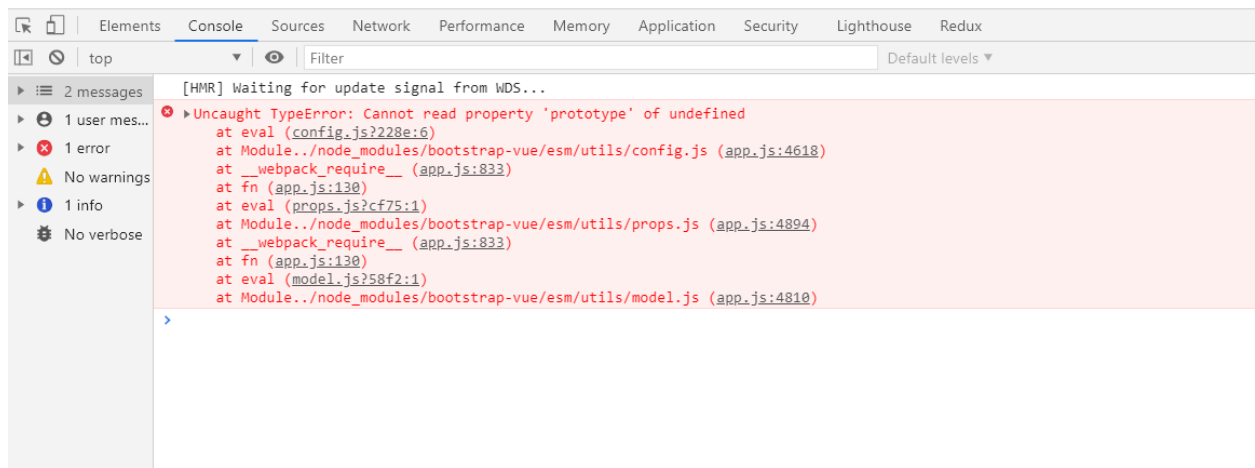
warning in ./node_modules/bootstrap-vue/esm/vue.js

"export 'default' (reexported as 'Vue') was not found in 'vue'

App running at:
- Local: http://localhost:8084/
- Network: http://192.168.1.155:8084/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

And in browser error:



Had to back it out.

Spike Five Project (Vue 2 & Vuex 3)

General Information

- Taken from a DotNet show.
- Package ref's out of date.
- Uses Vue 2 (and packages dependencies bit dated therefore).
- Try to get Vuex 3 going.
- Use of complex webpack for build output.
- Watch for vue file changes when running "npm run dev" via the webpack.config file, but still has to refresh the page for changes to be seen.
- Vue router intercepts also to then push the browser to the vue app, this can be shown in Vue developer chrome extension.
- Vue chrome extension detects and works ok.
- Debugging works ok in the browser.

Operation

1. In terminal window type - npm run dev
2. Then hit F5 in VS.
3. Changes to any vue apps are rendered into the wwwroot folder, allowing a page refresh to see them.

To Do

Get Vuex going inside it.

Issues & Errors

Upgrading packages could break build and razor integration as experienced in Spike 4.

Spike Six Project (Vue 3 & Vuex 4)

General Information

- Build with vue cli integration into MVC core template.
- Attempt of a multi-page variation of a Vue app.
- Use of 'vue' components to build up singular instances of a vue app in the MVC project.
- Managed to get shared state between the instances going.
- Watch works on editing a vue component, just then need to do a page refresh.
- Use of partial view option for vue js files.

To Do

Get Routing working - done in 'Blogs Vue App'

Issues & Errors

Sometimes Vue devtools doesn't seem to work in the browser. Maybe due to compatibility issues with Vue 3 as still in preview mode.

Debugging for some reason is not as reliable as in Vue 2 setup.

Some inbuilt components seem a bit dodgy at the moment as not a full blown release of Vue 3 is stable yet. So get warnings, which lead to errors, eg datepicker in the console and nested component issues.

Spike Seven Project (Vue 2 & Vuex 3)

- This is going to be a replica for spike 6 but hopefully built upon Vue 2 and alot more stable package integration.
- This is a combination of all the stuff where possible from previous spike work.
- Has its own contained folder for the Vue application, for everything vue related, does not pollute the .Net Core structure, apart from output to the wwwroot.

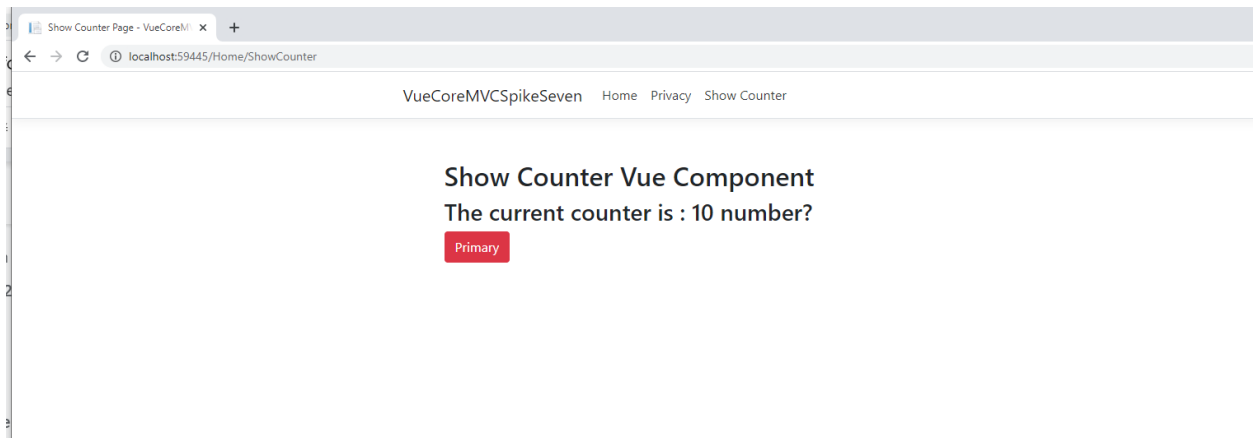
Deving, HMR & Debugging this solution.

You operate in two browsers if you want to take advantage of hot module reloading functionality (HMR):

Browser One.

Hit F5 in visual studio

Then navigate to the appropriate page with the embedded Vue app.

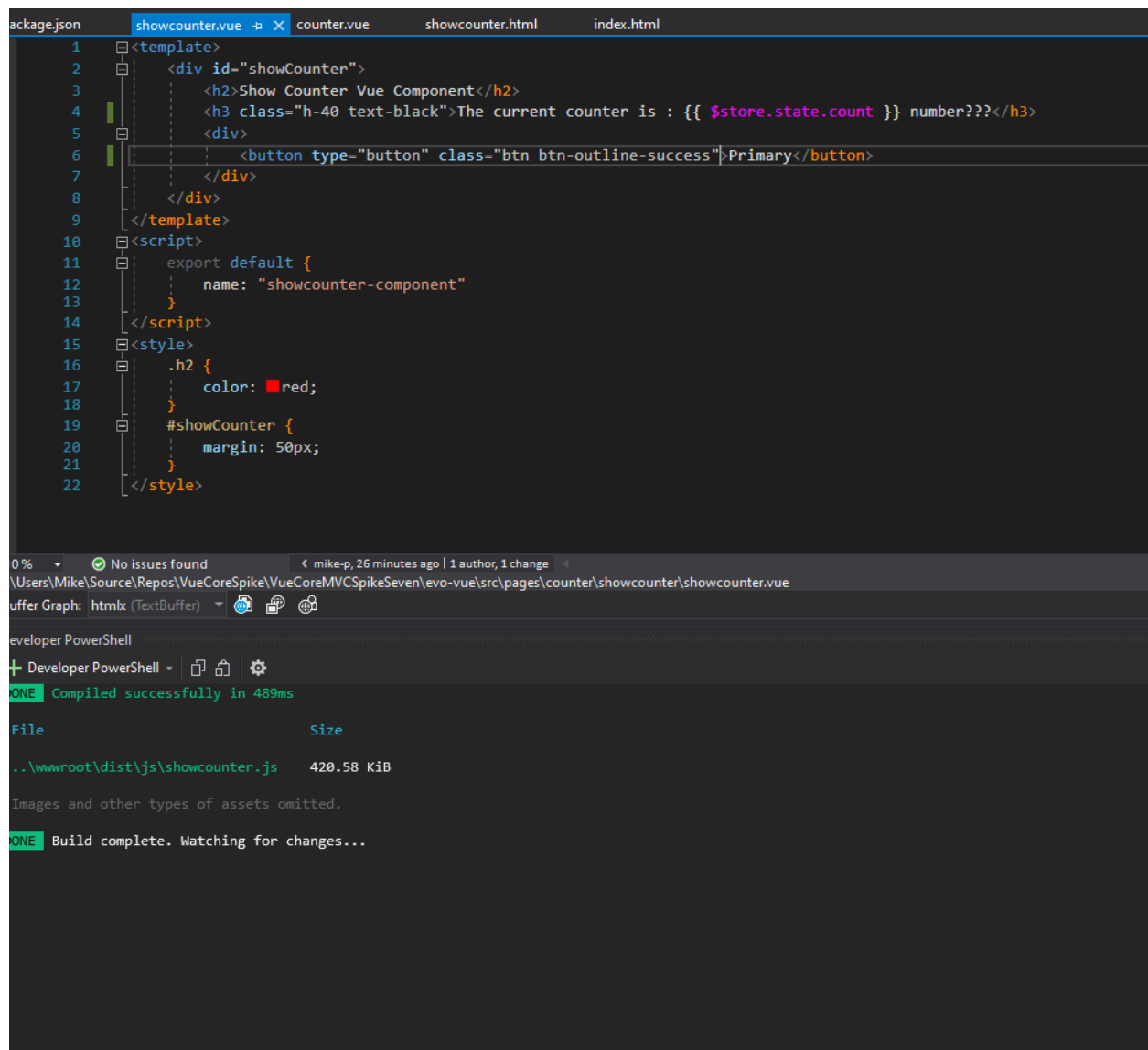


Open a terminal window from VS and navigate to the Vue folder.

Type - npm run watch (depending on what you named this script label)

```
"watch": "vue-cli-service build --mode development --watch --hot"
```

This produces a production version of the js output, by using the 'build' service of the vue-cli-service.



The screenshot displays the Visual Studio Code editor with a project open. The top pane shows the `showcounter.vue` file, which contains a Vue component with a template, script, and style section. The template includes a `div` with an `id` of `showCounter`, containing an `h2` tag, an `h3` tag with a class, and a `button` tag. The script section exports a default object with a `name` property. The style section uses `scoped` CSS classes for the `h2` and `#showCounter` elements.

The bottom pane shows the output of the build process. It indicates that the build was successful, compiling the component in 489ms. A table lists the output files and their sizes:

File	Size
..\wwwroot\dist\js\showcounter.js	420.58 KiB

Below the table, it states "Images and other types of assets omitted." and "Build complete. Watching for changes..."

Any changes to the vue files are monitored and a new output is instantly recompiled.

To then see the Vue app changes in the browser window need to do a full page refresh.

Browser Two.

Best to open a project in VS Code.

Open a terminal window from VS Code and navigate to the Vue folder.

Type - npm run dev (depending on what you named this script label)

This is the script setting for this combination of node server actions:-

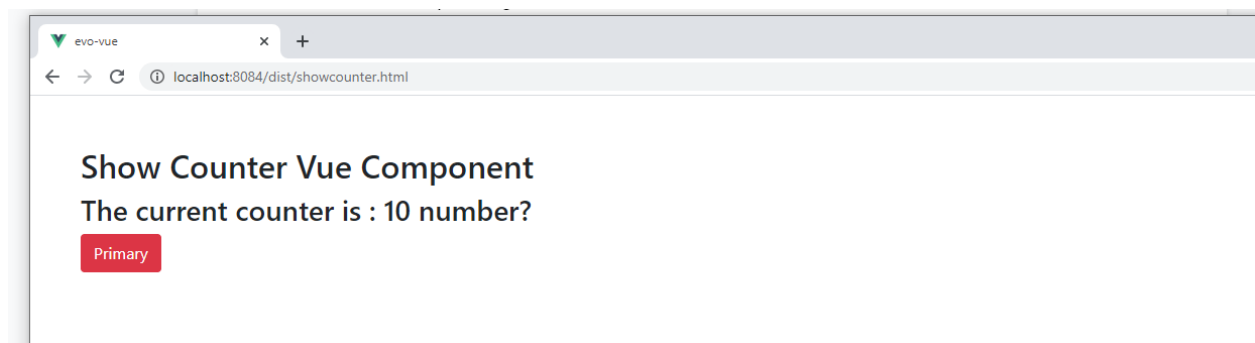
“vue-cli-service **serve** --mode development --watch --hot”

```
  "lint": "vue-cli-service lint",  
  "watch": "vue-cli-service build --mode development --watch --hot",  
  "dev": "vue-cli-service serve --mode development --watch --hot",  
},  
"dependencies": {  
  "axios": "^0.21.1",  
}
```

This produces a monitored version of the js output, by using the ‘serve’ service of the vue-cli-service.

You should then be able to go to the actual output html page in the wwwroot output directory.

```
> evo-vue@0.1.0 serve C:\Users\Mike\Source\Repos\VueCoreSpike\VueCoreMVCSpokeSeven\evo-vue  
> vue-cli-service serve  
  
INFO Starting development server...  
98% after emitting CopyPlugin  
  
DONE Compiled successfully in 1444ms  
  
App running at:  
- Local: http://localhost:8084/dist/  
- Network: http://10.5.0.2:8084/dist/  
  
Note that the development build is not optimized.  
To create a production build, run npm run build.
```



You will not get the full ASP.net core presented here just the vue app instance rendered on the html output page.

Any changes you make to the vue components will be presented instantly on saving the vue files. This will force an instant recompile and the app will be refreshed in the html page. No need to do a full page reload at all.

See 'tips' later for further setup information.

Conclusions.

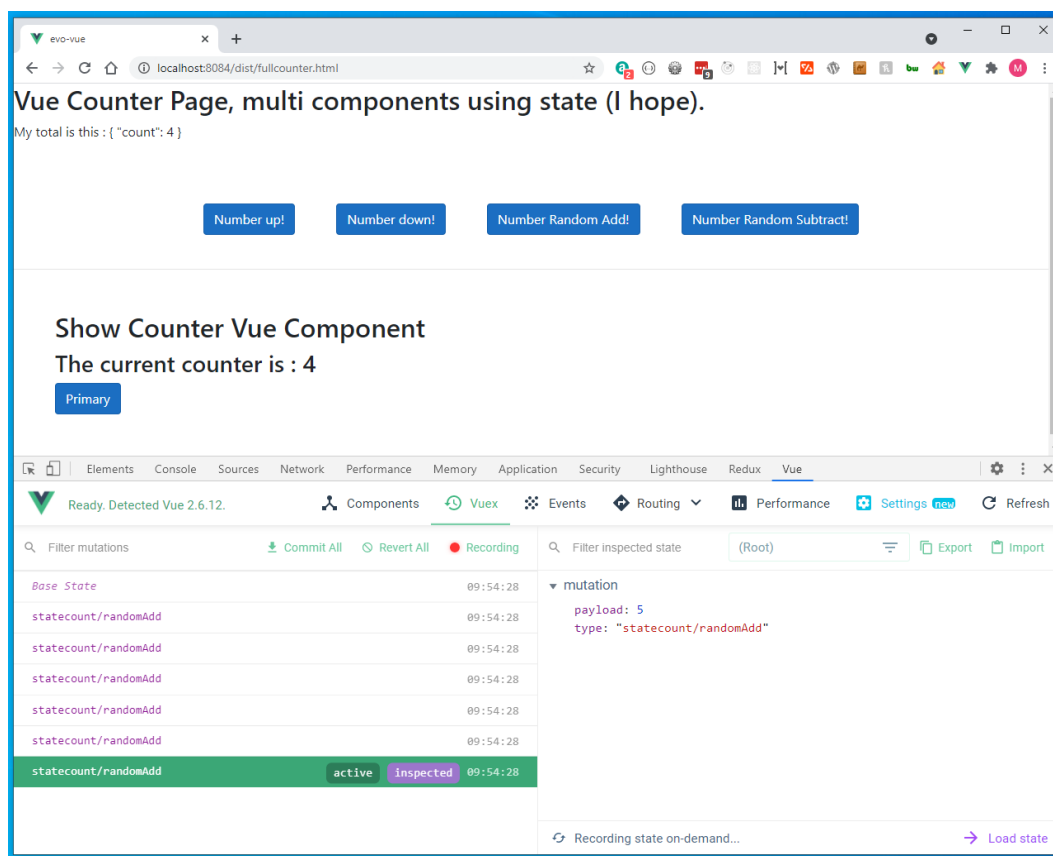
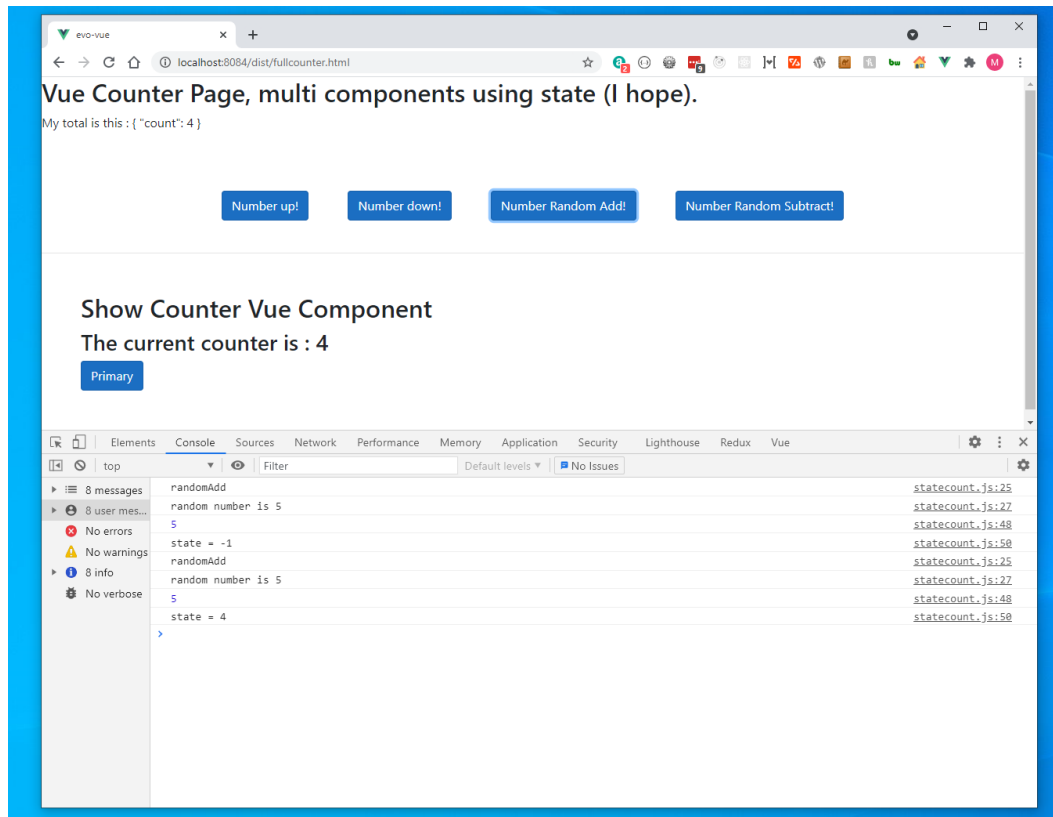
You can keep these browser windows open and do changes in the code base. In the conventional IISExpress server you will have to do a page reload to see vue changes however.

Downside is you need to have 2 IDE's going and running in sort of debug mode.

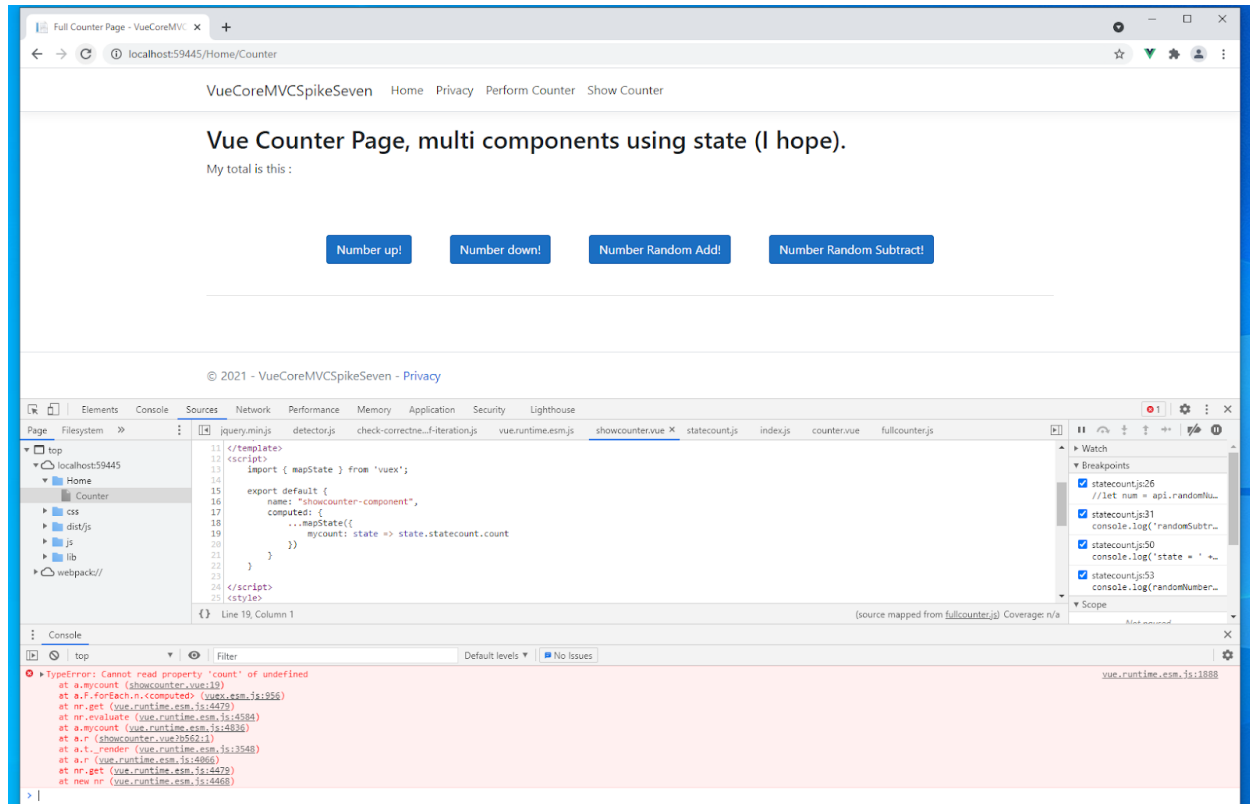
Vue chrome developer tools works in both options also.

Strange behaviour - Modularisation of Vuex state.

Have tried to implement best practice, of delegating state to specific modules, for then reuse and access in vuex state. This was implemented following the official github Vuex repository. The behaviour here is a bit unexplainable in that, I can access the modular state properties when running a normal vue instance spun up from a node server (see below):-



However, as soon as the vue is embedded in a razor page the code to access the nested modular state property breaks for some reason. In the example below the second component to show the state counter value when loading.

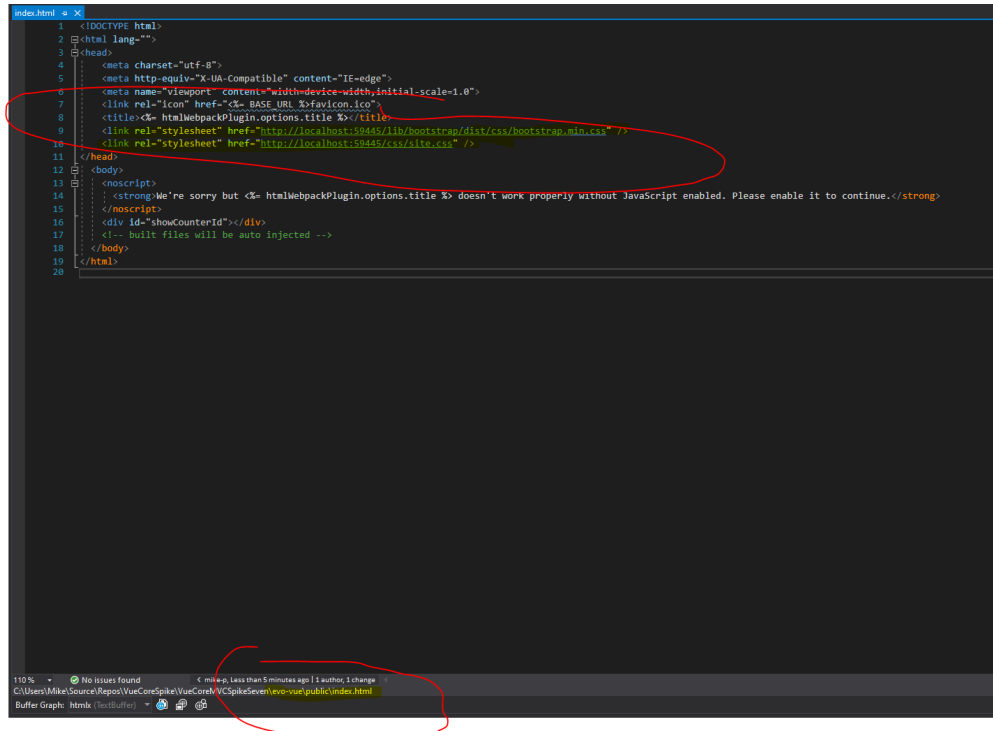


Its an if the statecount module has not been instantiated in the Vue app correct from the store references,

They are using the exact same code. I can't really explain why it works in one scenario and not the other to be honest. The Vue app is a compiled instance in both scenarios of the same code.

Tips for the Vue app in the HMR browser window.

If you want to see style changes etc from third party libraries like bootstrap, you can add links to the node server landing page in the vue app public folder like so:-



```
1 <!DOCTYPE html>
2 <html lang="">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,initial-scale=1.0">
7   <link rel="icon" href="%BASE_URL%favicon.ico">
8   <title>%htmlWebpackPlugin.options.title %</title>
9   <link rel="stylesheet" href="http://localhost:59449/lib/bootstrap/dist/css/bootstrap.min.css">
10  <link rel="stylesheet" href="http://localhost:59449/css/app.css" />
11 </head>
12 <body>
13   <noscript>
14     <strong>We're sorry but %htmlWebpackPlugin.options.title % doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>
15   </noscript>
16   <div id="showCounterId"></div>
17   <!-- built files will be auto injected -->
18 </body>
19 </html>
20
```

110% No issues found < 5 min ago, Less than 5 minutes ago | 1 author, 1 change
C:\Users\Mike\Source\Repos\VueCoreSpike\VueCoreSpike\Seven\evio-vue-public\index.html
Buffer Graph: HTML (text/html) < > < > < >

Note:- This is a hard coded link to your dev url including port number needed.

You need to also set the div id to be the correctly mounted id from the .js page like so:-

```
index.js  X  index.html
7-VueCoreMVCSpikeSeven JavaScript Content Files  {} "index"

1 import App from './showcounter.vue'
2 import store from '.././.././store'
3 import Vue from 'vue'
4
5 new Vue({
6   store,
7   render: h => h(App)
8 }).$mount('#showCounterId');
9

110 %  No issues found  mike-p, 8 minutes ago | 1 author, 1 change
C:\Users\Mike\Source\Repos\VueCoreSpike\VueCoreMVCSpikeSeven\evo-vue\src\pages\counter\showcounter\index.js
Buffer Graph: TypeScript (TextBuffer)

1 </head>
2 <body>
3   <noscript>
4     <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work proper
5   </noscript>
6   <div id="showCounterId"></div>
7   <!-- built files will be auto injected -->
8 </body>
9 </html>
```

Also to keep IISExpress running after stopping debugging you can try this:-

<https://stackoverflow.com/questions/60287963/keep-iis-express-running-when-vs2019-debugging-stop>

If getting issues, then run “npm run build” this should show any compile problems in the Vue app.

Spike Eight Project - Quasar

General Information.

Installation fires off a ps script, so need to allow correct powershell permissions.

Set-ExecutionPolicy -ExecutionPolicy Unrestricted

Conclusions.

Have managed to integrate the Quasar product and display a component on a razor page.

Took some trial and error to get the Quasar component to display on the page, and its integration is less intuitive than native Vue alone integration.

Recommendation is though because of the more rigid integration you get with Quasar as opposed to the more flexibility with just using Vue alone, it would be better to create silo'd quasar app per razor page, in its own holding folder within the solution.

I think Quasar has a lot of benefits for it if its usage in a conventional sense e.g. in a correct SPA application.

State management is now working between two separate and distinct razor pages, with 2 new Vue instances with quasar integrated. Quasar components are shown on both pages as confirmation also.

I still recommend silo'd folders for apps, the downside to this would be the none sharing of common state between the two app boundaries.

To Do.

Need to look into the 'development' process with HMR and debugging.

Issues & Errors.

The styling seems to be a bit off if we have a mixture of bootstrap and vue and quasar on the quasar components used.

Not wholly convinced about its Quasar's usage in this hybrid context in a ASP.NET core MVC application.

On further investigation, removal of bootstrap fixed the style conflict issues as I thought was happening. It is therefore recommended a dedicated razor layout be used in conjunction with the quasar landing page views.

Conclusions

Project spike One is using Vue 2.6.11 not vue 3 (which is still classed as in preview mode). Project spike Four uses Vue 3, but cannot get the components to show in razor pages, even using the same strategy as in project One, and get no errors to tell me why embedded Vue components won't show in cshtml files.

Upgrading packages can prove tricky with dependency errors occurring in console and rendering, leading to components not working. But this is a general problem always regarding working with npm packages and their inter dependencies.

I generally don't think there is an ideal option here. Even with vanilla js approach which is most straightforward and simple to orchestrate, you lack state management functionality, as each razor is using its own Vue instance.

Some of the official Vue docs are out of date for Vue 3, so there is that mismatch at the moment, and probably will be for a while. Also not much assistance on the web for issues, seen quite a few questions posted on stackoverflow with no answers at all. In contrast to this React's web support is vast.

Vue 3 or Vue 2?

Personally my leanings are to Vue 2. There is a lot more stable and compatible functionality around the packages and dependencies. There is also a lot richer ecosystem regards issue or error support. We can stand up state between disparate razor pages hosting independent Vue instances. And the chrome Vue js developer tools works.

Vue 3 is still in preview mode, and no date as to when it will go to final release. A lot of packages are not compatible and are currently being worked upon. Vuex 4 (the state management component) is working ok with Vue 3 though, and there is some supporting documentation.

Debugging can be orchestrated better in Vue 2 also, or seems more stable with regard to webpack sourcemaps it outputs to help you.