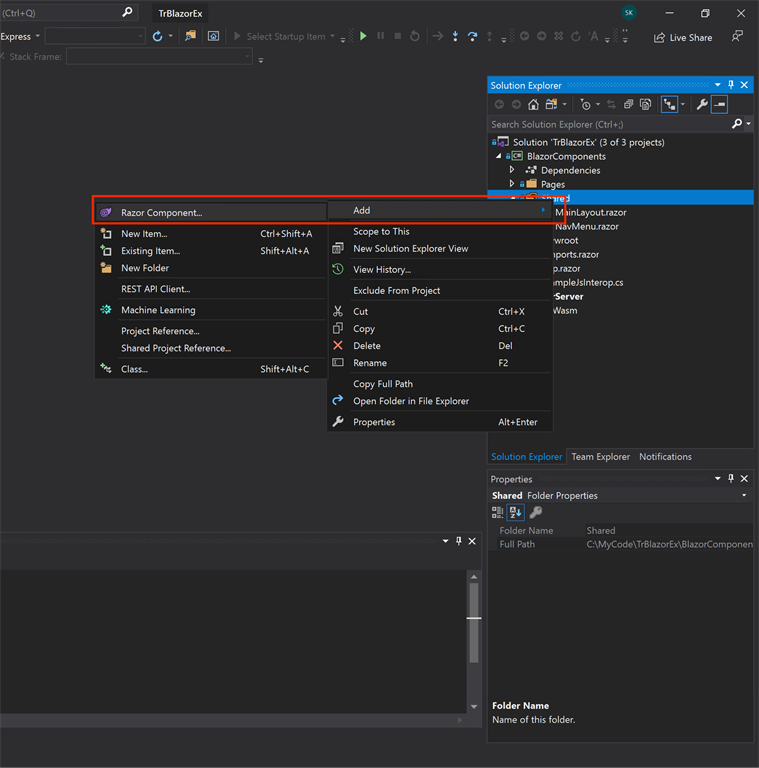
While I was developing a Blazor-based application I realized  that Blazor doesn't have an out-of-the-box checkbox list control, so I thought why not create one of my own.In this blog I will give  a step by step guide to create a Check Box List in Blazor.

**Step 1**

I will be using our Blazor Example project which I have created in my previous Blogs. This is about Futuristic Blazor application architecture. You can get the code from [here](https://github.com/techierathore/TrBlazorEx).

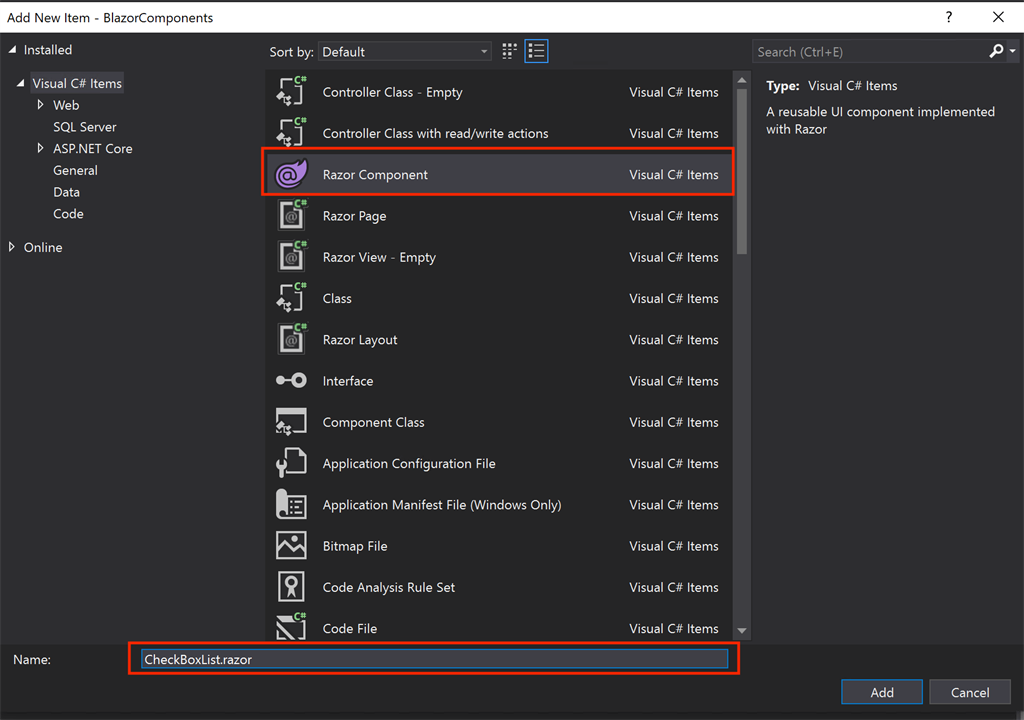
**Step 2**

Open the BlazorComponents Project, right click on the Shared folder and select Option Add => Razor Component like below screenshot.



**Step 3**

On selecting that option you will get the following screen, give the name of the component as CheckBoxList as shown in the screenshot.



**Step 4**

Copy and paste the  following code into the file,

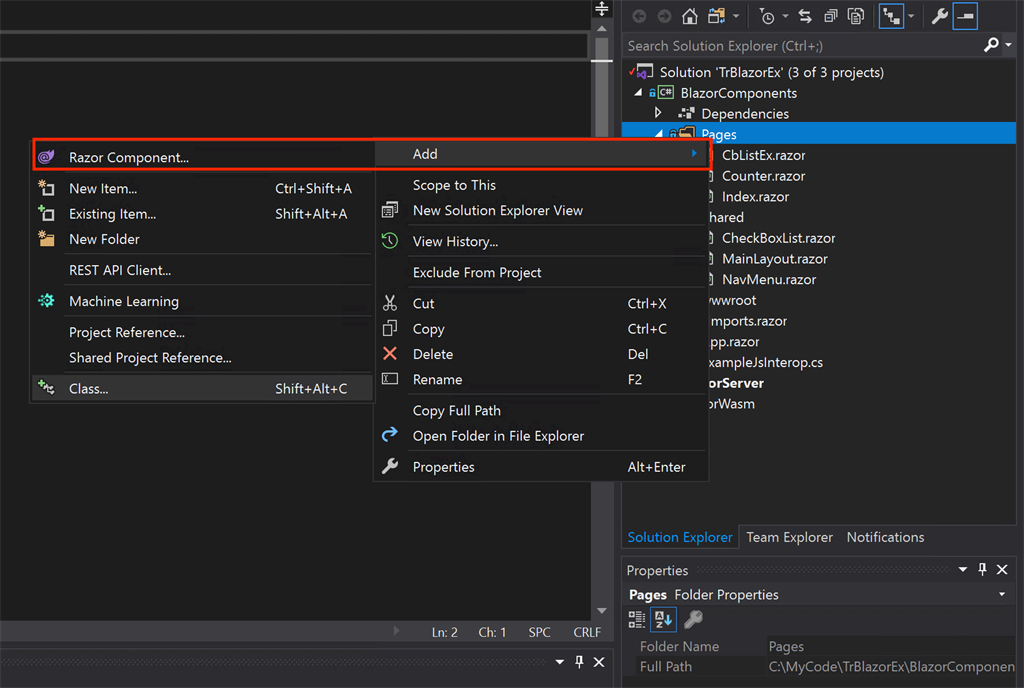
1. @typeparam TItem
2. <div style=@Style>
3. @**if** (Data != **null**)
4. {
5. foreach (**var** item **in** Data)
6. {
7. **var** Text = TextField?.Invoke(item);
8. **var** Value = ValueField?.Invoke(item).ToString();
9. bool Checked = **false**;
10. **if** (SelectedValues.Contains(Value))
11. {
12. Checked = **true**;
13. }
14. <input type="checkbox" checked=@Checked
15. @onchange="eventArgs => { CheckboxClicked(Value, eventArgs.Value); }" />
16. @Text <br />
17. }
18. }
19. @ChildContent
20. </div>
22. @code {
23. //Data for the Checkbox
24. [Parameter] **public** IEnumerable<TItem> Data { get; set; }
25. // The field to be shown adjacent to checkbox
26. [Parameter] **public** Func<TItem, string> TextField { get; set; }
27. // The Value which checkbox will return when checked
28. [Parameter] **public** Func<TItem, object> ValueField { get; set; }
29. // CSS Style for the Checkbox container
30. [Parameter] **public** string Style { get; set; }
31. // Any childd content for the control (if needed)
32. [Parameter] **public** RenderFragment ChildContent { get; set; }
33. // The array which contains the list of selected checkboxs
34. [Parameter] **public** List<string> SelectedValues { get; set; }
36. //Method to update the selected value on click on checkbox
37. **public** **void** CheckboxClicked(string aSelectedId, object aChecked)
38. {
39. **if** ((bool)aChecked)
40. {
41. **if** (!SelectedValues.Contains(aSelectedId))
42. {
43. SelectedValues.Add(aSelectedId);
44. }
45. }
46. **else**
47. {
48. **if** (SelectedValues.Contains(aSelectedId))
49. {
50. SelectedValues.Remove(aSelectedId);
51. }
52. }
53. StateHasChanged();
54. }
55. }

Let's understand the above code, the HTML part is basically code to generate the CheckBoxList on the basis of parameter Data which is a list of generic type TItem so that we can bind any object type list to the CheckBoxList. While creating the CheckBoxList the code also checks if the value is present in the SelectedValues list and marks the CheckBox Checked property based on that. This is basically for loading the checkbox list in edit mode.

I have taken these properties from [SelectList](https://blazorise.com/docs/extensions/selectlist/" \t "_blank) control present in [Blazorise UI library](https://github.com/stsrki/Blazorise" \t "_blank).The idea is to make this control part of that library some day.

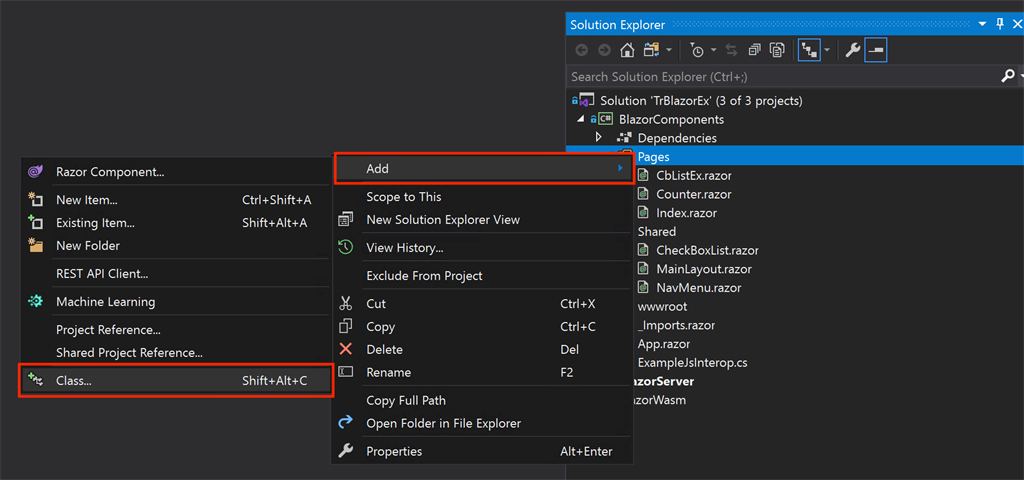
**Step 5**

To use this CheckboxList control, create a new page Named CbListEx.razor in Pages folder following Step-2 & 3.



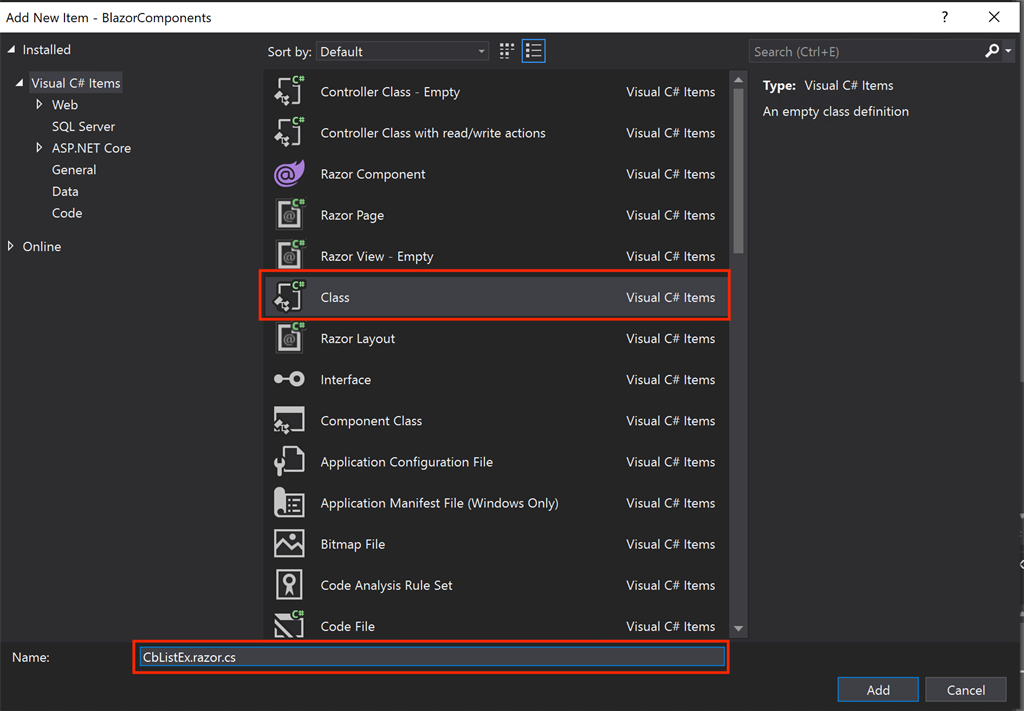
**Step 6**

Right click on the Pages folder again and select Option Add => Class like below screenshot.

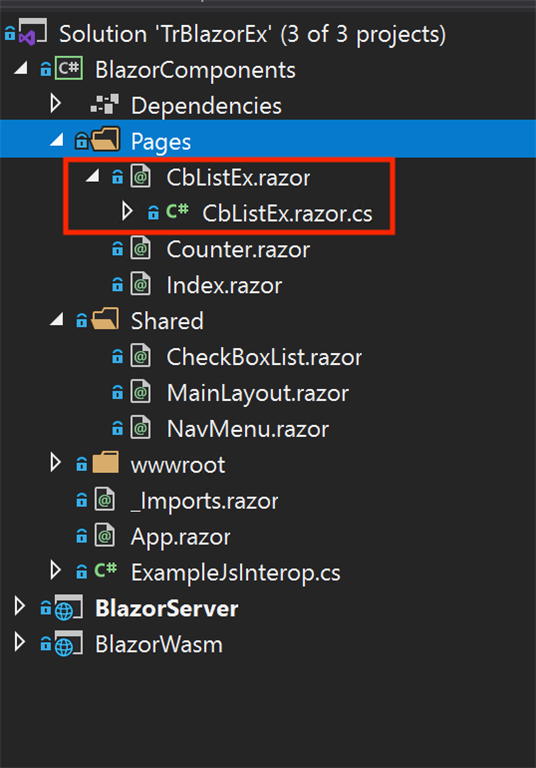


**Step 7**

On selecting that option you will get the following screen, give the name of the class as CbListEx.razor.cs as shown in screenshot.



The idea here is to create a code behind class for the razor page and when we give the same name with .cs extension Visual Studio automatically groups the files together as shown in the below screenshot,



**Step 8**

Copy the  following code in CbListEx.razor.cs file,

1. **using** Microsoft.AspNetCore.Components;
2. **using** System.Collections.Generic;
4. **namespace** BlazorComponents.Pages
5. {
6. **public** **class** CbListExBase : ComponentBase
7. {
8. **public** List<Employee> EmployeeList { **get**; **set**; }
9. **protected** List<**string**> SelectedIds = **new** List<**string**>();
10. **public** List<Employee> ObjectList { **get**; **set**; }
11. **public** **string** OutPutValue { **get**; **set**; }
12. **protected** **override** **void** OnInitialized()
13. {
14. EmployeeList = GetMockEmployees();
15. }
17. **protected** **void** ShowSelectedValues()
18. {
19. OutPutValue = **string**.Join(",", SelectedIds.ToArray());
20. StateHasChanged();
21. }
22. **private** List<Employee> GetMockEmployees()
23. {
25. var vSubOne = **new** Employee()
26. {
27. EmployeeId = 4,
28. FirstName = "S Ravi",
29. LastName = "Kumar",
30. MobileNo = "9901091975",
31. Email = "SKumar@gmail.com"
32. };
33. var vSubTwo = **new** Employee()
34. {
35. EmployeeId = 6,
36. FirstName = "Payal",
37. LastName = "Jain",
38. MobileNo = "9001091905",
39. Email = "PJain@gmail.com"
40. };
41. var vSubThree = **new** Employee()
42. {
43. EmployeeId = 7,
44. FirstName = "Alok",
45. LastName = "Ojha",
46. MobileNo = "900091905",
47. Email = "AOjha@gmail.com"
48. };
49. var vSubFour = **new** Employee()
50. {
51. EmployeeId = 9,
52. FirstName = "Divya",
53. LastName = "Bharti",
54. MobileNo = "9111767895",
55. Email = "DBharti@gmail.com"
56. };
57. var vSubFive = **new** Employee()
58. {
59. EmployeeId = 10,
60. FirstName = "Mayuri",
61. LastName = "Kango",
62. MobileNo = "9111000025",
63. Email = "MKango@gmail.com"
64. };
65. var vSubSix = **new** Employee()
66. {
67. EmployeeId = 11,
68. FirstName = "Tamraj",
69. LastName = "Kilwish",
70. MobileNo = "9251000025",
71. Email = "TKilwish@gmail.com"
72. };
73. var vSubSeven = **new** Employee()
74. {
75. EmployeeId = 12,
76. FirstName = "James",
77. LastName = "Bond",
78. MobileNo = "9521000025",
79. Email = "James007@gmail.com"
80. };
82. var vSubList = **new** List<Employee>
83. {
84. vSubOne, vSubTwo,
85. vSubThree, vSubFour,
86. vSubFive,vSubSix,
87. vSubSeven
88. };
90. **return** vSubList;
91. }
92. }
94. **public** **class** Employee
95. {
96. **public** **long** EmployeeId
97. { **get**; **set**; }
99. **public** **string** FirstName
100. { **get**; **set**; }
102. **public** **string** LastName
103. { **get**; **set**; }
105. **public** **string** FullName
106. {
107. **get**
108. {
109. **return** FirstName + " " + LastName;
110. }
111. }
112. **public** **string** MobileNo
113. { **get**; **set**; }
114. **public** **string** Email
115. { **get**; **set**; }
116. }
117. }

In the above code we are basically loading the CheckBoxList on page load with a list of Employee object which is a custom class. ShowSelectedValues method will be called on click of a button to show the selected values of the checkbox.

I have kept the Employee class in the same file just to save it; in the actual project it will be part of your Models Folder/project.

**Step 9**

Copy Paste the following code in CbListEx.razor file,

1. @page "/CbListEx"
2. @inherits CbListExBase
3. <div **class**="col-12">
4. <h3><b>CheckBox List Example </b></h3>
5. <hr />
6. <div **class**="col-12 row">
7. <label **class**="col-2 font-weight-bold">CheckBox List :</label>
8. <CheckBoxList Data="@EmployeeList"
9. TextField="@((item)=>item.FullName)"
10. ValueField="@((item)=>item.EmployeeId)"
11. SelectedValues="@SelectedIds" />
12. </div>
13. <div **class**="col-12 row">
14. <label **class**="col-2 font-weight-bold">Selected Ids :</label>
15. <b>@OutPutValue</b>
16. </div>
17. <br />
18. <div **class**="col-12 row">
19. <span **class**="col-2"></span>
20. <button **class**="form-control col-1 btn btn-primary" @onclick="ShowSelectedValues">Show Checked Values</button>
21. </div>
22. </div>

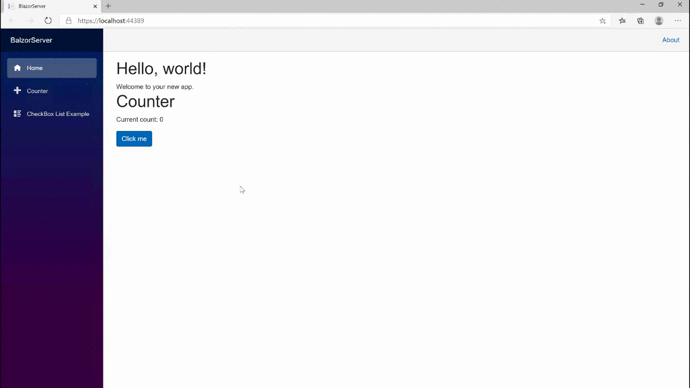
In the above code we are just showing the bound CheckBoxList, a button to show the selected values in a Label.

**Step 10**

Add the following code in NavMenu.razor inside the Shared folder to add a menu item for opening CbListEx.razor page.

1. <li **class**="nav-item px-3">
2. <NavLink **class**="nav-link" href="CbListEx">
3. <span **class**="oi oi-list-rich" aria-hidden="true"></span> CheckBox List Example
4. </NavLink>
5. </li>

Now when you execute the application you will get the following output,



* [Blazor](https://www.c-sharpcorner.com/topics/blazor)
* [Checkbox List](https://www.c-sharpcorner.com/topics/checkbox-list)
* [Checkbox List In Blazor](https://www.c-sharpcorner.com/topics/checkbox-list-in-blazor)
* [Create A Checkbox List In Blazor](https://www.c-sharpcorner.com/topics/create-a-checkbox-list-in-blazor)