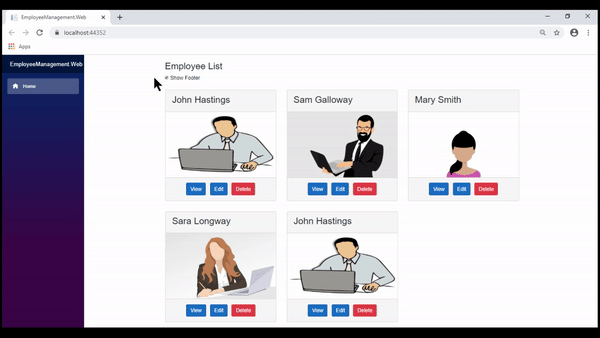
Blazor component parameters

In this video we will discuss **how to pass data from parent component to child component using parameters in Blazor**. Let's understand this with an example.

As you can see from the image below, when Show Footer checkbox is checked, the Card Footer should be displayed when it is unchecked it should be hidden.



On this page we have 2 blazor components

1. EmployeeList Component (Parent Component)
2. DisplayEmployee Component (Child Component)

Child Component Class (DisplayEmployeeBase.cs)

* This class has just 2 properties (Employee and ShowFooter)
* Both of them are decorated with the Parameter attribute
* This attribute allows the values to be passed from the parent component

using EmployeeManagement.Models;

using Microsoft.AspNetCore.Components;

namespace EmployeeManagement.Web.Pages

{

public class DisplayEmployeeBase : ComponentBase

{

[Parameter]

public Employee Employee { get; set; }

[Parameter]

public bool ShowFooter { get; set; }

}

}

Copy

Child Component View (DisplayEmployee.razor)

* This component does not have @page directive. We intentionally left it out, because we are going to use this component only as a child component. We don't want this to be a routable component i.e we don't want to be able to reach this component by using a URL in the browser.
* This component binds to the Employee property to display the employee details. It also uses the ShowFooter boolean property to show and hide the footer.
* The values for these 2 properties (Employee and ShowFooter) will be passed by the parent component (EmployeeList component)

@inherits DisplayEmployeeBase

<div class="card m-3" style="min-width: 18rem; max-width:30.5%;">

<div class="card-header">

<h3>@Employee.FirstName @Employee.LastName</h3>

</div>

<img class="card-img-top imageThumbnail" src="@Employee.PhotoPath" />

@if (ShowFooter)

{

<div class="card-footer text-center">

<a href="@($"employeedetails/{Employee.EmployeeId}")"

class="btn btn-primary m-1">View</a>

<a href="#" class="btn btn-primary m-1">Edit</a>

<a href="#" class="btn btn-danger m-1">Delete</a>

</div>

}

</div>

Copy

Parent Component Class (EmployeeListBase.cs)

* The main responsibility of this parent component (EmployeeList) is to retrieve the list of employees by calling the REST API.
* It also has the ShowFooter boolean property. The Show Footer checkbox on the view is bound to this property.

using EmployeeManagement.Models;

using EmployeeManagement.Web.Services;

using Microsoft.AspNetCore.Components;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace EmployeeManagement.Web.Pages

{

public class EmployeeListBase : ComponentBase

{

[Inject]

public IEmployeeService EmployeeService { get; set; }

public IEnumerable<Employee> Employees { get; set; }

public bool ShowFooter { get; set; } = true;

protected override async Task OnInitializedAsync()

{

Employees = (await EmployeeService.GetEmployees()).ToList();

}

}

}

Copy

Parent Component View (EmployeeList.razor)

* As this parent component loops through the list of employees, it passes the employee object to the child component using the Employee property.
* A value for ShowFooter is also passed from the parent component to the child component.
* Remember both these properties (Employee and ShowFooter) are decorated with the [Parameter] attribute in the child component. This allows the parent component to pass values to those 2 properties.

@page "/"

@inherits EmployeeListBase

<h3>Employee List</h3>

<input type="checkbox" id="showFooter" @bind="ShowFooter" />

<label for="showFooter">Show Footer</label>

@if (Employees == null)

{

<div class="spinner"></div>

}

else

{

<div class="card-deck">

@foreach (var employee in Employees)

{

<DisplayEmployee Employee="employee" ShowFooter="ShowFooter">

</DisplayEmployee>

}

</div>

}