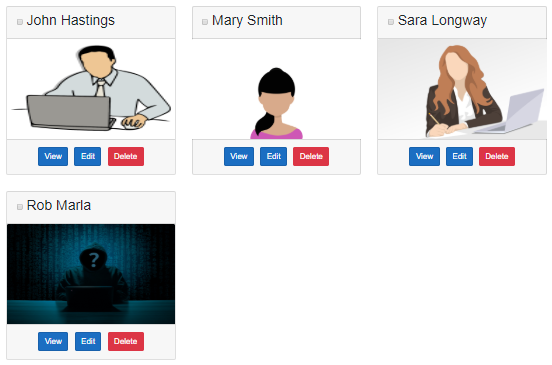
Pass data from child to parent component in blazor

In this video we will discuss implementing Delete CRUD operation on list component. Along the way we will understand, how to pass data from child component to parent component in Blazor.

On the following page we have 2 blazor components

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



* + When the Delete button is clicked, the respective employee must be deleted and the employee card should dissappear.
  + The Delete button is in the child component (DisplayEmployee Component)
* It is the parent component (EmployeeList Component) that loops through the list of employees and renders each employee using the child component (DisplayEmployee Component)
* When the Delete button in the child component (DisplayEmployee Component) is clicked, it is the child component that is responsible for deleting the respective employee from the database.

Child component to parent component communication

Child component (DisplayEmployee Component) should notify the parent component (EmployeeList Component) that a record is deleted so the parent component can remove the respective employee card from the list of employees. For this the child component exposes an event. The parent component assigns a callback method to the child component's event. In Blazor, to expose an event we use EventCallback.

Child Component Class (DisplayEmployeeBase.cs)

* OnEmployeeDeleted is the custom event.
* We use EventCallback to create a custom event. We discussed this in [Part 28](https://youtu.be/Kb5bWCuQ9bQ) of [Blazor tutorial](https://www.pragimtech.com/courses/blazor-tutorial-for-beginners/).
* Delete\_Click event handler deletes the employee record and raises the custom event - OnEmployeeDeleted.

[Parameter]

public EventCallback<int> OnEmployeeDeleted { get; set; }

[Inject]

public IEmployeeService EmployeeService { get; set; }

[Inject]

public NavigationManager NavigationManager { get; set; }

protected async Task Delete\_Click()

{

await EmployeeService.DeleteEmployee(Employee.EmployeeId);

await OnEmployeeDeleted.InvokeAsync(Employee.EmployeeId);

//NavigationManager.NavigateTo("/", true);

}

Copy

Child Component View (DisplayEmployee.razor)

* + Bind the Delete button to Delete\_Click event handler method.
* Everytime the delete button is clicked, the respective employee is deleted and the custom event OnEmployeeDeleted is raised.

<button type="button" class="btn btn-danger m-1"

@onclick="Delete\_Click">

Delete

</button>

Copy

Parent Component Class (EmployeeListBase.cs)

EmployeeDeleted is the callback method. This method will be called when the delete button in the child component is clicked.

protected async Task EmployeeDeleted()

{

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

}

Copy

Parent Component View (EmployeeList.razor)

In the view assign EmployeeDeleted() as the Callback method to the child component custom event OnEmployeeDeleted.

@foreach (var employee in Employees)

{

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

OnEmployeeSelection="EmployeeSelectionChanged"

OnEmployeeDeleted="EmployeeDeleted"></DisplayEmployee>

}