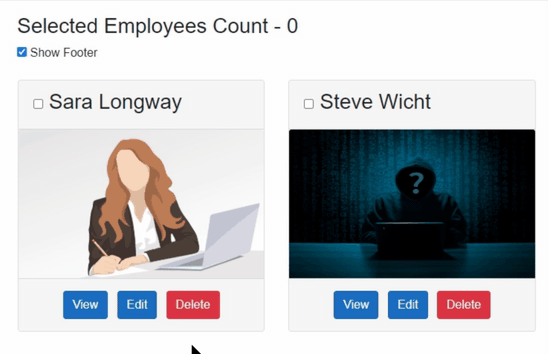
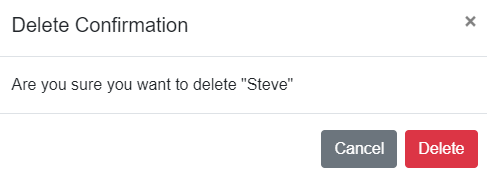
Delete confirmation dialog in Blazor

Once a record is deleted from the database, there is no easy way to recover it. So it's always a good practice to display a confirmation on Delete. In this video we will discuss, how to implement delete confirmation in Blazor.



Reusable Blazor Confirmation Dialog Component

We will create a reusable confirmation dialog component so it can be used for displaying a confirmation dialog for any requirement, not just delete confirmation.



Confirmation Component Class (ConfirmBase.cs)

* We do not want to hard-code confirmation title and message.
* We want the calling component to be able to pass confirmation title and message.
* This makes this confirmation dialog component more reusable.
* This is the reason we created ConfirmationTitle and ConfirmationMessage properties.
* To make this component even more reusable, we can create additional properties that can be used by the calling component to pass the display text for both DELETE and CANCEL buttons.
* The custom event ConfirmationChanged is fired everytime Delete or Cancel button is clicked.
* The boolean property ShowConfirmation controls the visibility of the dialog on the UI.

using Microsoft.AspNetCore.Components;

using System.Threading.Tasks;

namespace PragimTech.Components

{

public class ConfirmBase : ComponentBase

{

protected bool ShowConfirmation { get; set; }

[Parameter]

public string ConfirmationTitle { get; set; } = "Confirm Delete";

[Parameter]

public string ConfirmationMessage { get; set; } = "Are you sure you want to delete";

public void Show()

{

ShowConfirmation = true;

StateHasChanged();

}

[Parameter]

public EventCallback<bool> ConfirmationChanged { get; set; }

protected async Task OnConfirmationChange(bool value)

{

ShowConfirmation = false;

await ConfirmationChanged.InvokeAsync(value);

}

}

}

Copy

Confirmation Component View (Confirm.razor)

* The view binds to ConfirmationTitle and ConfirmationMessage properties. The default values specified in the component class are used, if the calling component does not pass values explicitly.
* OnConfirmationChange method in the component class is specified as the event handler for the following 3 buttons
  + Delete
  + Cancel
  + Close
* When Cancel and Close buttons are clicked, false is passed as the event payload (i.e event data). The calling component can subscribe to the custom event ConfirmationChanged and will have access to this event data.
* When Delete button is clicked, true is passed as the event data.
* In the calling component, we can use the event data to determine if a confiramtion is given or not. True indicates confirmation given, false indicates confirmation not given.

@inherits ConfirmBase

@if (ShowConfirmation)

{

<div class="modal fade show d-block" id="exampleModal" tabindex="-1" role="dialog">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">@ConfirmationTitle</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close"

@onclick="() => OnConfirmationChange(false)">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

@ConfirmationMessage

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal"

@onclick="() => OnConfirmationChange(false)">Cancel</button>

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

@onclick="() => OnConfirmationChange(true)">Delete</button>

</div>

</div>

</div>

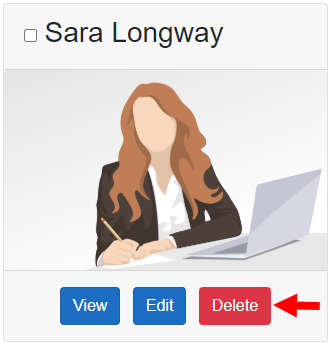
</div>

}

Copy

Using Confirmation Dialog Component

In our example, Delete confirmation must be displayed when Delete button on Employee card is clicked.



* @ref attribute creates a reference to the component instance
* The component reference (DeleteConfirmation) is used to call Show() method of the delete confirmation component which displays the dialog.
* ConfirmDelete\_Click is the event handler that handles the custom event ConfirmationChanged.

<Confirm @ref="DeleteConfirmation" ConfirmationChanged="ConfirmDelete\_Click"

ConfirmationMessage=@($"Are you sure you want to delete \"{Employee.FirstName}\"")>

</Confirm>

Copy

protected PragimTech.Components.ConfirmBase DeleteConfirmation { get; set; }

protected void Delete\_Click()

{

DeleteConfirmation.Show();

}

protected async Task ConfirmDelete\_Click(bool deleteConfirmed)

{

if(deleteConfirmed)

{

await EmployeeService.DeleteEmployee(Employee.EmployeeId);

await OnEmployeeDeleted.InvokeAsync(Employee.EmployeeId);

}

}