Blazor event handling

In this video we will learn event handling in Blazor with examples.

Display X and Y mouse coordinates

As we hover the mouse over an image element, we want to display X and Y mouse coordinates. In the following example,

* mousemove event is handled.
* Mouse\_Move is the event handler method.
* The event handler stores the mouse X and Y coordinates in Coordinates property.
* The component view binds to this Coordinates property to display the X and Y coordinates.

Component View

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

@onmousemove="@Mouse\_Move" />

<h1>@Coordinates</h1>

Copy

Component Class

protected string Coordinates { get; set; }

protected void Mouse\_Move(MouseEventArgs e)

{

Coordinates = $"X = {e.ClientX } Y = {e.ClientY}";

}

Copy

Event handling using a lambda expression

Instead of creating a separate named method in the component class and then specifying that method as the event handler we can use a lambda expression as the event handler directly in the HTML. An example is shown below.

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

@onmousemove="@(e => Coordinates = $"X = {e.ClientX} Y = {e.ClientY}")" />

Copy

Show and Hide Employee Card Footer

We want to show and hide the card footer when a button is clicked. The text on the button should also change from Show Footer to Hide Footer and vice-versa depending on the visibility of the footer.

Component View

In the example below

* onclick event is being handled
* Button\_Click is the event handler method
* The text on the button is bound to ButtonText property in the component class
* The class attribute of the card-footer <div> eletemt is bound to CssClass property in the component class

<button class="btn btn-primary" @onclick="@Button\_Click">@ButtonText</button>

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

<a href="/" class="btn btn-primary">Back</a>

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

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

</div>

Copy

Component Class

ButtonText and CssClass properties are declared in the component class. In the Button\_Click event handler the values of these 2 properties (i.e ButtonText & CssClass) are being changed depending on thier current values.

protected string ButtonText { get; set; } = "Hide Footer";

protected string CssClass { get; set; } = null;

protected void Button\_Click()

{

if (ButtonText == "Hide Footer")

{

ButtonText = "Show Footer";

CssClass = "HideFooter";

}

else

{

CssClass = null;

ButtonText = "Hide Footer";

}

}

Copy

CSS Class

.HideFooter{

display:none;

}