```
<button disabled="@IsDisabled"
        @onclick="() => ButtonClicked.InvokeAsync()">
    Click Me
</button>

@code {
    [Parameter]
    public bool IsDisabled { get; set; }

    [Parameter]
    public EventCallback ButtonClicked { get; set; }
}
```

```
[Fact]
public void DontInvokeEventWhenButtonDisabled()
{
    var buttonClicked = false;
    var cut = Render(@<MyButton
        IsDisabled="true"
        ButtonClicked="() => buttonClicked = true"/>);

    cut.Find("button").Click();

    buttonClicked.Should().BeFalse();
}
```

Conclusion

- bUnit gives you the ability to decouple your code from the 3rd party code
- Don’t test 3rd party code - That includes Blazor itself

```
[Fact]
public void DontInvokeEventWhenButtonDisabled()
{
    var buttonClicked = false;
    var cut = Render(@<MyButton
        IsDisabled="true"
        ButtonClicked="() => buttonClicked = true"/>);

    cut.Find("button").Click();

    buttonClicked.Should().BeFalse();
}
```

```
<button disabled="@IsDisabled"
        @onclick="() => ButtonClicked.InvokeAsync()”>
    Click Me
</button>

@code {
    [Parameter]
    public bool IsDisabled { get; set; }

    [Parameter]
    public EventCallback ButtonClicked { get; set; }
}
```

Resources

- bUnit repository: https://github.com/bUnit-dev/bUnit
- Documentation: https://bunit.dev/
- Starter Tutorial: https://steven-giesel.com/