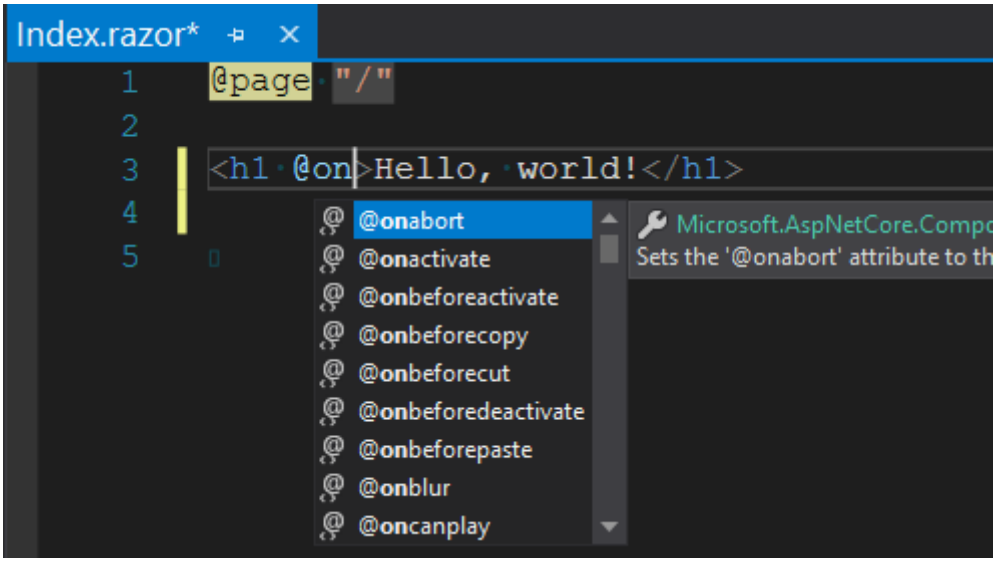
Browser DOM events

Edit

When rendering any mark-up, it is possible to assign standard JavaScript events on the rendered HTML elements so that our own Blazor C# methods are executed. For example, we have used the **`@onclick`** **Directive** in many samples elsewhere:

```
<button @onclick=ButtonClicked>Click me</button>
```

These event directives have full IntelliSense support within the Visual Studio editor, so starting to type the **@** symbol should present us with a comprehensive list of available directives, along with a description identifying which argument class type the event passes us in our event handler. DOM events start with **@on**.



> ***"Sets the '@onabort' attribute to the provided string or delegate value. A delegate value should be of type 'Microsoft.AspNetCore.Components.Web.ProgressEventArgs'"***

Warning: When writing a Blazor app that runs entirely on the server, Blazor will hook events in the browser and send them to server so our C#

methods can be invoked. This can lead to a noticeable slow-down for frequently fired events such as **`onmousemove`**.

Note: Because JavaScript invocation of C# methods is asynchronous, this means that in C# methods we cannot cancel events as we can in JavaScript. This is because cancelling browser DOM events is a synchronous operation, by the time our C# has been asynchronously invoked it is already too late to cancel the event.

Available DOM events and their parameter types include:

General events

DOM event	Argument type
on*a*ctivate	**`EventArgs`**
*on*b*efore*a*ctivate	**`EventArgs`**
on*b*efore*d*eactivate	**`EventArgs`**
on*d*eactivate	**`EventArgs`**
on*e*nded	**`EventArgs`**
on*f*ullscreen*c*hange	**`EventArgs`**
on*f*ullscreen*e*rror	**`EventArgs`**
on*l*oaded*d*ata	**`EventArgs`**
on*l*oaded*m*etadata	**`EventArgs`**
on*p*ointer*l*ock*c*hange	**`EventArgs`**
on*p*ointer*l*ock*e*rror	**`EventArgs`**
on*r*eady*s*tate*c*hange	**`EventArgs`**
on*s*croll	**`EventArgs`**

Focus events

DOM event	Argument type

on*f*ocus	`FocusEventArgs`
on*b*lur	`FocusEventArgs`
on*f*ocus*i*n	`FocusEventArgs`
on*f*ocus*o*ut	`FocusEventArgs`

Mouse events

DOM event	Argument type
on*m*ouse*o*ver	`MouseEventArgs`
on*m*ouse*o*ut	`MouseEventArgs`
on*m*ouse*m*ove	`MouseEventArgs`
on*m*ouse*d*own	`MouseEventArgs`
on*m*ouse*u*p	`MouseEventArgs`
on*c*lick	`MouseEventArgs`
on*d*blclick	`MouseEventArgs`
on*c*ontextmenu	`MouseEventArgs`
on*w*heel	`WheelEventArgs`
on*m*ouse*w*heel	`WheelEventArgs`

Drag events

DOM event	Argument type
on*d*rag	`DragEventArgs`
on*d*rag*e*nd	`DragEventArgs`
on*d*rag*e*nter	`DragEventArgs`
on*d*rag*l*eave	`DragEventArgs`
on*d*rag*o*ver	`DragEventArgs`
on*d*rag*s*tart	`DragEventArgs`

on*d*rop	`DragEventArgs`

Keyboard events

DOM event	Argument type
on*key**d***own	`KeyboardEventArgs`
on*key**u***p	`KeyboardEventArgs`
on*key**p***ress	`KeyboardEventArgs`

Input events

DOM event	Argument type
onchange	`ChangeEventArgs`
on*i*nput	`ChangeEventArgs`
on*i*nvalid	`EventArgs`
on*r*eset	`EventArgs`
on*s*elect	`EventArgs`
on*s*elect*s*tart	`EventArgs`
on*s*electionchange	`EventArgs`
on*s*ubmit	`EventArgs`

Clipboard events

DOM event	Argument type
on*b*eforecopy	`EventArgs`
on*b*eforecut	`EventArgs`
on*b*efore*p*aste	`EventArgs`
oncopy	`ClipboardEventArgs`

oncut	`ClipboardEventArgs`
onpaste	`ClipboardEventArgs`

Touch events

DOM event	Argument type
ontouchcancel	`TouchEventArgs`
ontouchend	`TouchEventArgs`
ontouchmove	`TouchEventArgs`
ontouchstart	`TouchEventArgs`
ontouchenter	`TouchEventArgs`
ontouchleave	`TouchEventArgs`

Pointer events

DOM event	Argument type
ongotpointercapture	`PointerEventArgs`
onlostpointercapture	`PointerEventArgs`
onpointercancel	`PointerEventArgs`
onpointerdown	`PointerEventArgs`
onpointerenter	`PointerEventArgs`
onpointerleave	`PointerEventArgs`
onpointermove	`PointerEventArgs`
onpointerout	`PointerEventArgs`
onpointerover	`PointerEventArgs`
onpointerup	`PointerEventArgs`

Media events

DOM event	Argument type
oncanplay	`EventArgs`
oncanplaythrough	`EventArgs`
oncuechange	`EventArgs`
ondurationchange	`EventArgs`
onemptied	`EventArgs`
onpause	`EventArgs`
onplay	`EventArgs`
onplaying	`EventArgs`
onratechange	`EventArgs`
onseeked	`EventArgs`
onseeking	`EventArgs`
onstalled	`EventArgs`
onstop	`EventArgs`
onsuspend	`EventArgs`
ontimeupdate	`EventArgs`
onvolumechange	`EventArgs`
onwaiting	`EventArgs`

Progress events

DOM event	Argument type
onloadstart	`ProgressEventArgs`
ontimeout	`ProgressEventArgs`
onabort	`ProgressEventArgs`
onload	`ProgressEventArgs`

on*loadend*	**ProgressEventArgs**
on*progress*	**ProgressEventArgs**
on*error*	**ErrorEventArgs**