Table of Contents

Namespace Microsoft.AspNetCore.Blazor

Classes

DataTransfer

The DataTransfer object is used to hold the data that is being dragged during a drag and drop operation. It may hold one or more UIDataTransferItem, each of one or more data types. For more information about drag and drop, see HTML Drag and Drop API.

ElementRef

Represents a reference to a rendered element.

HttpClientJsonExtensions

Extension methods for working with JSON APIs.

JsonUtil

Provides mechanisms for converting between .NET objects and JSON strings.

MarkupString

A string value that can be rendered as markup such as HTML.

RenderFragment

Represents a segment of UI content, implemented as a delegate that writes the content to a RenderTreeBuilder.

ServerSideBlazorOptions

Options for Server-Side Blazor.

UIChangeEventArgs

Supplies information about an input change event that is being raised.

UIClipboardEventArgs

Supplies information about an clipboard event that is being raised.

UIDataTransferItem

The UIDataTransferItem object represents one drag data item. During a drag operation, each drag event has a dataTransfer property which contains a list of drag data items. Each item in the list is a UIDataTransferItem object.

UIDragEventArgs

Supplies information about an drag event that is being raised.

UIErrorEventArgs

Supplies information about an error event that is being raised.

UIEventArgs

Supplies information about an event that is being raised.

UIEventArgsRenderTreeBuilderExtensions

Extensions methods on RenderTreeBuilder for event handlers.

UIFocusEventArgs

Supplies information about a focus event that is being raised.

UIKeyboardEventArgs

Supplies information about a keyboard event that is being raised.

UIMouseEventArgs

Supplies information about a mouse event that is being raised.

UIPointerEventArgs

Supplies information about a mouse event that is being raised.

UIProgressEventArgs

Supplies information about a progress event that is being raised.

UITouchEventArgs

Supplies information about a touch event that is being raised.

UITouchPoint

Represents a single contact point on a touch-sensitive device. The contact point is commonly a finger or stylus and the device may be a touchscreen or trackpad.

UIWheelEventArgs

Supplies information about a mouse wheel event that is being raised.

Class DataTransfer

The DataTransfer object is used to hold the data that is being dragged during a drag and drop operation. It may hold one or more UIDataTransferItem, each of one or more data types. For more information about drag and drop, see HTML Drag and Drop API.

Inheritance

System.Object
DataTransfer

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class DataTransfer : Object
```

Constructors

DataTransfer()

Declaration

```
public DataTransfer()
```

Properties

DropEffect

Gets the type of drag-and-drop operation currently selected or sets the operation to a new type. The value must be none, copy, link or move.

Declaration

```
public string DropEffect { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

EffectAllowed

Provides all of the types of operations that are possible. Must be one of none, copy, copyLink, copyMove, link, linkMove, move, all or uninitialized.

Declaration

```
public string EffectAllowed { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Files

Contains a list of all the local files available on the data transfer. If the drag operation doesn't involve dragging files, this property is an empty list.

Declaration

```
public string[] Files { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

Items

Gives a UIDataTransferItem array which is a list of all of the drag data.

Declaration

```
public UIDataTransferItem[] Items { get; set; }
```

Property Value

TYPE	DESCRIPTION
UIDataTransferItem[]	

Types

An array of System.String giving the formats that were set in the dragstart event.

Declaration

```
public string[] Types { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

Class ElementRef

Represents a reference to a rendered element.

Inheritance

System.Object

ElementRef

Implements

[ICustomArgSerializer](#)

Namespace: [Microsoft.AspNetCore.Blazor](#)

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class ElementRef : ValueType, ICustomArgSerializer
```

Explicit Interface Implementations

ICustomArgSerializer.ToJsonPrimitive()

Declaration

```
object ICustomArgSerializer.ToJsonPrimitive()
```

Returns

TYPE	DESCRIPTION
System.Object	

Implements

[ICustomArgSerializer](#)

Class HttpClientJsonExtensions

Extension methods for working with JSON APIs.

Inheritance

System.Object

HttpClientJsonExtensions

Namespace: **Microsoft.AspNetCore.Blazor**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public static class HttpClientJsonExtensions : Object
```

Methods

GetJsonAsync<T>(HttpClient, String)

Sends a GET request to the specified URI, and parses the JSON response body to create an object of the generic type.

Declaration

```
public static Task<T> GetJsonAsync<T>(this HttpClient httpClient, string requestUri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpClient	httpClient	The System.Net.Http.HttpClient.
System.String	requestUri	The URI that the request will be sent to.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<T>	The response parsed as an object of the generic type.

Type Parameters

NAME	DESCRIPTION
T	A type into which the response body can be JSON-deserialized.

PostJsonAsync(HttpClient, String, Object)

Sends a POST request to the specified URI, including the specified `content` in JSON-encoded format, and parses the JSON response body to create an object of the generic type.

Declaration

```
public static Task PostJsonAsync(this HttpClient httpClient, string requestUri, object content)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpClient	httpClient	The System.Net.Http.HttpClient.
System.String	requestUri	The URI that the request will be sent to.
System.Object	content	Content for the request body. This will be JSON-encoded and sent as a string.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	The response parsed as an object of the generic type.

PostJsonAsync<T>(HttpClient, String, Object)

Sends a POST request to the specified URI, including the specified `content` in JSON-encoded format, and parses the JSON response body to create an object of the generic type.

Declaration

```
public static Task<T> PostJsonAsync<T>(this HttpClient httpClient, string requestUri, object content)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpClient	httpClient	The System.Net.Http.HttpClient.
System.String	requestUri	The URI that the request will be sent to.
System.Object	content	Content for the request body. This will be JSON-encoded and sent as a string.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<T>	The response parsed as an object of the generic type.

Type Parameters

NAME	DESCRIPTION
T	A type into which the response body can be JSON-deserialized.

PutJsonAsync(HttpClient, String, Object)

Sends a PUT request to the specified URI, including the specified `content` in JSON-encoded format.

Declaration

```
public static Task PutJsonAsync(this HttpClient httpClient, string requestUri, object content)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpClient	httpClient	The System.Net.Http.HttpClient.
System.String	requestUri	The URI that the request will be sent to.
System.Object	content	Content for the request body. This will be JSON-encoded and sent as a string.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

PutJsonAsync<T>(HttpClient, String, Object)

Sends a PUT request to the specified URI, including the specified `content` in JSON-encoded format, and parses the JSON response body to create an object of the generic type.

Declaration

```
public static Task<T> PutJsonAsync<T>(this HttpClient httpClient, string requestUri, object content)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpClient	httpClient	The System.Net.Http.HttpClient.
System.String	requestUri	The URI that the request will be sent to.
System.Object	content	Content for the request body. This will be JSON-encoded and sent as a string.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<T>	The response parsed as an object of the generic type.

Type Parameters

NAME	DESCRIPTION
T	A type into which the response body can be JSON-deserialized.

SendJsonAsync(HttpClient, HttpMethod, String, Object)

Sends an HTTP request to the specified URI, including the specified `content` in JSON-encoded format.

Declaration

```
public static Task SendJsonAsync(this HttpClient httpClient, HttpMethod method, string requestUri, object
content)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpClient	httpClient	The System.Net.Http.HttpClient.
System.Net.Http.HttpMethod	method	The HTTP method.
System.String	requestUri	The URI that the request will be sent to.
System.Object	content	Content for the request body. This will be JSON-encoded and sent as a string.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

SendJsonAsync<T>(HttpClient, HttpMethod, String, Object)

Sends an HTTP request to the specified URI, including the specified `content` in JSON-encoded format, and parses the JSON response body to create an object of the generic type.

Declaration

```
public static Task<T> SendJsonAsync<T>(this HttpClient httpClient, HttpMethod method, string requestUri,
object content)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpClient	httpClient	The System.Net.Http.HttpClient.
System.Net.Http.HttpMethod	method	The HTTP method.

TYPE	NAME	DESCRIPTION
System.String	requestUri	The URI that the request will be sent to.
System.Object	content	Content for the request body. This will be JSON-encoded and sent as a string.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<T>	The response parsed as an object of the generic type.

Type Parameters

NAME	DESCRIPTION
T	A type into which the response body can be JSON-deserialized.

Class JsonUtil

Provides mechanisms for converting between .NET objects and JSON strings.

Inheritance

System.Object

JsonUtil

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public static class JsonUtil : Object
```

Methods

Deserialize<T>(String)

Deserializes the JSON string, creating an object of the specified generic type.

Declaration

```
public static T Deserialize<T>(string json)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	json	The JSON string.

Returns

TYPE	DESCRIPTION
T	An object of the specified type.

Type Parameters

NAME	DESCRIPTION
T	The type of object to create.

Serialize(Object)

Serializes the value as a JSON string.

Declaration

```
public static string Serialize(object value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	The value to serialize.

Returns

TYPE	DESCRIPTION
System.String	The JSON string.

TYPE	DESCRIPTION
System.String	The JSON string.

Class MarkupString

A string value that can be rendered as markup such as HTML.

Inheritance

System.Object

MarkupString

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class MarkupString : ValueType
```

Constructors

MarkupString(String)

Constructs an instance of MarkupString.

Declaration

```
public MarkupString(string value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	The value for the new instance.

Properties

Value

Gets the value of the MarkupString.

Declaration

```
public string Value { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ToString()

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
System.String	

Operators

Explicit(String to MarkupString)

Casts a System.String to a MarkupString.

Declaration

```
public static explicit operator MarkupString(string value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	The System.String value.

Returns

TYPE	DESCRIPTION
MarkupString	

Explicit(String to MarkupString)

Casts a System.String to a MarkupString.

Declaration

```
public static explicit operator MarkupString(string value)
```

Class RenderFragment

Represents a segment of UI content, implemented as a delegate that writes the content to a RenderTreeBuilder.

Inheritance

System.Object

RenderFragment

Namespace: **Microsoft.AspNetCore.Blazor**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class RenderFragment : MulticastDelegate
```

Constructors

RenderFragment(Object, IntPtr)

Declaration

```
public RenderFragment(object object, IntPtr method)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	object	
System.IntPtr	method	

Methods

BeginInvoke(RenderTreeBuilder, AsyncCallback, Object)

Declaration

```
public virtual IAsyncResult BeginInvoke(RenderTreeBuilder builder, AsyncCallback callback, object object)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	
System.AsyncCallback	callback	
System.Object	object	

Returns

TYPE	DESCRIPTION
System.IAsyncResult	

EndInvoke(IAsyncResult)

Declaration

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IAsyncResult	result	

Invoke(RenderTreeBuilder)

Declaration

```
public virtual void Invoke(RenderTreeBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	

Class ServerSideBlazorOptions

Options for Server-Side Blazor.

Inheritance

System.Object

ServerSideBlazorOptions

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public class ServerSideBlazorOptions : Object
```

Constructors

ServerSideBlazorOptions()

Declaration

```
public ServerSideBlazorOptions()
```

Class UIChangeEventArgs

Supplies information about an input change event that is being raised.

Inheritance

System.Object

UIEventArgs

UIChangeEventArgs

Inherited Members

UIEventArgs.Type

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIChangeEventArgs : UIEventArgs
```

Constructors

UIChangeEventArgs()

Declaration

```
public UIChangeEventArgs()
```

Properties

Value

Gets or sets the new value of the input. This may be a System.String or a System.Boolean.

Declaration

```
public object Value { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

Class UIClipboardEventArgs

Supplies information about an clipboard event that is being raised.

Inheritance

System.Object

UIEventArgs

UIClipboardEventArgs

Inherited Members

UIEventArgs.Type

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIClipboardEventArgs : UIEventArgs
```

Constructors

UIClipboardEventArgs()

Declaration

```
public UIClipboardEventArgs()
```

Class UIDataTransferItem

The UIDataTransferItem object represents one drag data item. During a drag operation, each drag event has a dataTransfer property which contains a list of drag data items. Each item in the list is a UIDataTransferItem object.

Inheritance

System.Object
UIDataTransferItem

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIDataTransferItem : Object
```

Constructors

UIDataTransferItem()

Declaration

```
public UIDataTransferItem()
```

Properties

Kind

The kind of drag data item, string or file

Declaration

```
public string Kind { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Type

The drag data item's type, typically a MIME type

Declaration

```
public string Type { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class UIDragEventArgs

Supplies information about an drag event that is being raised.

Inheritance

System.Object

[UIEventArgs](#)

UIDragEventArgs

Inherited Members

[UIEventArgs.Type](#)

Namespace: **Microsoft.AspNetCore.Blazor**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIDragEventArgs : UIEventArgs
```

Constructors

UIDragEventArgs()

Declaration

```
public UIDragEventArgs()
```

Properties

AltKey

true if the alt key was down when the event was fired. false otherwise.

Declaration

```
public bool AltKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Button

The button number that was pressed when the mouse event was fired: Left button=0, middle button=1 (if present), right button=2. For mice configured for left handed use in which the button actions are reversed the values are instead read from right to left.

Declaration

```
public long Button { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

Buttons

The buttons being pressed when the mouse event was fired: Left button=1, Right button=2, Middle (wheel) button=4, 4th button

(typically, "Browser Back" button)=8, 5th button (typically, "Browser Forward" button)=16. If two or more buttons are pressed, returns the logical sum of the values. E.g., if Left button and Right button are pressed, returns 3 (=1 | 2).

Declaration

```
public long Buttons { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ClientX

The X coordinate of the mouse pointer in local (DOM content) coordinates.

Declaration

```
public long ClientX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ClientY

The Y coordinate of the mouse pointer in local (DOM content) coordinates.

Declaration

```
public long ClientY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

CtrlKey

true if the control key was down when the event was fired. false otherwise.

Declaration

```
public bool CtrlKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

DataTransfer

The data that underlies a drag-and-drop operation, known as the drag data store. See DataTransfer.

Declaration

```
public DataTransfer DataTransfer { get; set; }
```

Property Value

TYPE	DESCRIPTION
DataTransfer	

Detail

A count of consecutive clicks that happened in a short amount of time, incremented by one.

Declaration

```
public float Detail { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MetaKey

true if the meta key was down when the event was fired. false otherwise.

Declaration

```
public bool MetaKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

ScreenX

The X coordinate of the mouse pointer in global (screen) coordinates.

Declaration

```
public long ScreenX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ScreenY

The Y coordinate of the mouse pointer in global (screen) coordinates.

Declaration

```
public long ScreenY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ShiftKey

true if the shift key was down when the event was fired. false otherwise.

Declaration

```
public bool ShiftKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Class UIErrorEventArgs

Supplies information about an error event that is being raised.

Syntax

```
public class UIErrorEventArgs : UIEventArgs
```

Constructors

UIErrorEventArgs()

Declaration

```
public UIErrorEventArgs()
```

Properties

Colno

Gets the column number of the script file on which the error occurred.

Declaration

```
public int Colno { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Filename

Gets the name of the script file in which the error occurred.

Declaration

```
public string Filename { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Lineno

Gets the line number of the script file on which the error occurred.

Declaration

```
public int Lineno { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Message

Gets a a human-readable error message describing the problem.

Declaration

```
public string Message { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class UIEventArgs

Supplies information about an event that is being raised.

Inheritance

System.Object

UIEventArgs

UIChangeEventArgs

UIClipboardEventArgs

UIDragEventArgs

UIErrorEventArgs

UIFocusEventArgs

UIKeyboardEventArgs

UIMouseEventArgs

UIProgressEventArgs

UITouchEventArgs

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIEventArgs : Object
```

Constructors

UIEventArgs()

Declaration

```
public UIEventArgs()
```

Properties

Type

Gets or sets the type of the event.

Declaration

```
public string Type { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class UIEventArgsRenderTreeBuilderExtensions

Extensions methods on RenderTreeBuilder for event handlers.

Inheritance

System.Object

UIEventArgsRenderTreeBuilderExtensions

Namespace: **Microsoft.AspNetCore.Blazor**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public static class UIEventArgsRenderTreeBuilderExtensions : Object
```

Remarks

These methods enable method-group to delegate conversion for delegates and methods that accept types derived from UIEventArgs.

This enhances the programming experience for using event handlers with the render tree builder in components written in pure C#. These extension methods make it possible to write code like:

```
builder.AddAttribute(0, "onkeypress", MyKeyPressHandler);
```

Where `void MyKeyPressHandler(UIKeyboardEventArgs e)` is a method defined in the same class. In this example, the author knows that the `onclick` event is associated with the UIKeyboardEventArgs event args type. The component author is responsible for providing a delegate that matches the expected event args type, an error will result in a failure at runtime.

When a component is authored in Razor (.cshtml), the Razor code generator will maintain a mapping between event names and event arg types that can be used to generate more strongly typed code. Generated code for the same case will look like:

```
builder.AddAttribute(0, "onkeypress", BindMethods.GetEventHandlerValue<UIKeyboardEventArgs>
(MyKeyPressHandler));
```

Methods

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIChangeEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIChangeEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.

TYPE	NAME	DESCRIPTION
System.String	name	The name of the attribute.
System.Action<UIChangeEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIClipboardEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIClipboardEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIClipboardEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIDragEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIDragEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIDragEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIErrorEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIErrorEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIErrorEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIFocusEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIFocusEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIFocusEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIKeyboardEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIKeyboardEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIKeyboardEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIMouseEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIMouseEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIMouseEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIPointerEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIPointerEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIPointerEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIProgressEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIProgressEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIProgressEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UITouchEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UITouchEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UITouchEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Action<UIWheelEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Action<UIWheelEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIWheelEventArgs>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIChangeEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIChangeEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIChangeEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIClipboardEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIClipboardEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIClipboardEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIDragEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIDragEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIDragEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIErrorEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIErrorEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIErrorEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIFocusEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIFocusEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIFocusEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIKeyboardEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIKeyboardEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIKeyboardEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIMouseEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIMouseEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIMouseEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIPointerEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIPointerEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIPointerEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIProgressEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIProgressEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIProgressEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UITouchEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UITouchEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UITouchEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(RenderTreeBuilder, Int32, String, Func<UIWheelEventArgs, Task>)

Appends a frame representing an System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public static void AddAttribute(this RenderTreeBuilder builder, int sequence, string name,
Func<UIWheelEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	The RenderTreeBuilder.
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIWheelEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

Class UIFocusEventArgs

Supplies information about a focus event that is being raised.

Inheritance

System.Object

UIEventArgs

UIFocusEventArgs

Inherited Members

UIEventArgs.Type

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIFocusEventArgs : UIEventArgs
```

Constructors

UIFocusEventArgs()

Declaration

```
public UIFocusEventArgs()
```

Class UIKeyboardEventArgs

Supplies information about a keyboard event that is being raised.

Inheritance

System.Object

UIEventArgs

UIKeyboardEventArgs

Inherited Members

UIEventArgs.Type

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIKeyboardEventArgs : UIEventArgs
```

Constructors

UIKeyboardEventArgs()

Declaration

```
public UIKeyboardEventArgs()
```

Properties

AltKey

true if the alt key was down when the event was fired. false otherwise.

Declaration

```
public bool AltKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Code

Holds a string that identifies the physical key being pressed. The value is not affected by the current keyboard layout or modifier state, so a particular key will always return the same value.

Declaration

```
public string Code { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

CtrlKey

true if the control key was down when the event was fired. false otherwise.

Declaration

```
public bool CtrlKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Key

The key value of the key represented by the event. If the value has a printed representation, this attribute's value is the same as the char attribute. Otherwise, it's one of the key value strings specified in 'Key values'. If the key can't be identified, this is the string "Unidentified"

Declaration

```
public string Key { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Location

The location of the key on the device.

Declaration

```
public float Location { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MetaKey

true if the meta key was down when the event was fired. false otherwise.

Declaration

```
public bool MetaKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Repeat

true if a key has been depressed long enough to trigger key repetition, otherwise false.

Declaration

```
public bool Repeat { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

ShiftKey

true if the shift key was down when the event was fired. false otherwise.

Declaration

```
public bool ShiftKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Class UIMouseEventArgs

Supplies information about a mouse event that is being raised.

Inheritance

System.Object

[UIEventArgs](#)

UIMouseEventArgs

[UIPointerEventArgs](#)

[UIWheelEventArgs](#)

Inherited Members

[UIEventArgs.Type](#)

Namespace: [Microsoft.AspNetCore.Blazor](#)

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIMouseEventArgs : UIEventArgs
```

Constructors

UIMouseEventArgs()

Declaration

```
public UIMouseEventArgs()
```

Properties

AltKey

true if the alt key was down when the event was fired. false otherwise.

Declaration

```
public bool AltKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Button

The button number that was pressed when the mouse event was fired: Left button=0, middle button=1 (if present), right button=2. For mice configured for left handed use in which the button actions are reversed the values are instead read from right to left.

Declaration

```
public long Button { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

Buttons

The buttons being pressed when the mouse event was fired: Left button=1, Right button=2, Middle (wheel) button=4, 4th button (typically, "Browser Back" button)=8, 5th button (typically, "Browser Forward" button)=16. If two or more buttons are pressed, returns the logical sum of the values. E.g., if Left button and Right button are pressed, returns 3 (=1 | 2).

Declaration

```
public long Buttons { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ClientX

The X coordinate of the mouse pointer in local (DOM content) coordinates.

Declaration

```
public long ClientX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ClientY

The Y coordinate of the mouse pointer in local (DOM content) coordinates.

Declaration

```
public long ClientY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

CtrlKey

true if the control key was down when the event was fired. false otherwise.

Declaration

```
public bool CtrlKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Detail

A count of consecutive clicks that happened in a short amount of time, incremented by one.

Declaration

```
public float Detail { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MetaKey

true if the meta key was down when the event was fired. false otherwise.

Declaration

```
public bool MetaKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

ScreenX

The X coordinate of the mouse pointer in global (screen) coordinates.

Declaration

```
public long ScreenX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ScreenY

The Y coordinate of the mouse pointer in global (screen) coordinates.

Declaration

```
public long ScreenY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ShiftKey

true if the shift key was down when the event was fired. false otherwise.

Declaration

```
public bool ShiftKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Class UIPointerEventArgs

Supplies information about a mouse event that is being raised.

Inheritance

System.Object

[UIEventArgs](#)

[UIMouseEventArgs](#)

UIPointerEventArgs

Inherited Members

[UIMouseEventArgs.Detail](#)

[UIMouseEventArgs.ScreenX](#)

[UIMouseEventArgs.ScreenY](#)

[UIMouseEventArgs.ClientX](#)

[UIMouseEventArgs.ClientY](#)

[UIMouseEventArgs.Button](#)

[UIMouseEventArgs.Buttons](#)

[UIMouseEventArgs.CtrlKey](#)

[UIMouseEventArgs.ShiftKey](#)

[UIMouseEventArgs.AltKey](#)

[UIMouseEventArgs.MetaKey](#)

[UIEventArgs.Type](#)

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIPointerEventArgs : UIMouseEventArgs
```

Constructors

UIPointerEventArgs()

Declaration

```
public UIPointerEventArgs()
```

Properties

Height

The height (magnitude on the Y axis), in CSS pixels, of the contact geometry of the pointer.

Declaration

```
public float Height { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

IsPrimary

Indicates if the pointer represents the primary pointer of this pointer type.

Declaration

```
public bool IsPrimary { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

PointerId

A unique identifier for the pointer causing the event.

Declaration

```
public string PointerId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

PointerType

Indicates the device type that caused the event. Must be one of the strings mouse, pen or touch, or an empty string.

Declaration

```
public string PointerType { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Pressure

The normalized pressure of the pointer input in the range of 0 to 1, where 0 and 1 represent the minimum and maximum pressure the hardware is capable of detecting, respectively.

Declaration

```
public float Pressure { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

TiltX

The plane angle (in degrees, in the range of -90 to 90) between the Y-Z plane and the plane containing both the transducer (e.g. pen stylus) axis and the Y axis.

Declaration

```
public float TiltX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

TiltY

The plane angle (in degrees, in the range of -90 to 90) between the X-Z plane and the plane containing both the transducer (e.g. pen stylus) axis and the X axis.

Declaration

```
public float TiltY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Width

The width (magnitude on the X axis), in CSS pixels, of the contact geometry of the pointer.

Declaration

```
public float Width { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Class UIProgressEventArgs

Supplies information about a progress event that is being raised.

Inheritance

System.Object

[UIEventArgs](#)

UIProgressEventArgs

Inherited Members

[UIEventArgs.Type](#)

Namespace: **Microsoft.AspNetCore.Blazor**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIProgressEventArgs : UIEventArgs
```

Constructors

UIProgressEventArgs()

Declaration

```
public UIProgressEventArgs()
```

Properties

LengthComputable

Whether or not the total size of the transfer is known.

Declaration

```
public bool LengthComputable { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Loaded

The number of bytes transferred since the beginning of the operation. This doesn't include headers and other overhead, but only the content itself.

Declaration

```
public long Loaded { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

Total

The total number of bytes of content that will be transferred during the operation. If the total size is unknown, this value is zero.

Declaration

```csharp
public long Total { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

```csharp
public long Total { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

Class UITouchEventArgs

Supplies information about a touch event that is being raised.

Inheritance

System.Object

[UIEventArgs](#)

UITouchEventArgs

Inherited Members

[UIEventArgs.Type](#)

Syntax

```
public class UITouchEventArgs : UIEventArgs
```

Constructors

UITouchEventArgs()

Declaration

```
public UITouchEventArgs()
```

Properties

AltKey

true if the alt key was down when the event was fired. false otherwise.

Declaration

```
public bool AltKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

ChangedTouches

A list of Touches for every point of contact which contributed to the event. For the touchstart event this must be a list of the touch points that just became active with the current event. For the touchmove event this must be a list of the touch points that have moved since the last event. For the touchend and touchcancel events this must be a list of the touch points that have just been removed from the surface.

Declaration

```
public UITouchPoint[] ChangedTouches { get; set; }
```

Property Value

TYPE	DESCRIPTION
[UITouchPoint](#)[]	

CtrlKey

true if the control key was down when the event was fired. false otherwise.

Declaration

```
public bool CtrlKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Detail

A count of consecutive clicks that happened in a short amount of time, incremented by one.

Declaration

```
public float Detail { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MetaKey

true if the meta key was down when the event was fired. false otherwise.

Declaration

```
public bool MetaKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

ShiftKey

true if the shift key was down when the event was fired. false otherwise.

Declaration

```
public bool ShiftKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

TargetTouches

A list of UITouchPoint for every point of contact that is touching the surface and started on the element that is the target of the current event.

Declaration

```
public UITouchPoint[] TargetTouches { get; set; }
```

Property Value

TYPE	DESCRIPTION
UITouchPoint[]	

Touches

A list of UITouchPoint for every point of contact currently touching the surface.

Declaration

```
public UITouchPoint[] Touches { get; set; }
```

Property Value

TYPE	DESCRIPTION
UITouchPoint[]	

Class UITouchPoint

Represents a single contact point on a touch-sensitive device. The contact point is commonly a finger or stylus and the device may be a touchscreen or trackpad.

Inheritance

System.Object
UITouchPoint

Namespace: Microsoft.AspNetCore.Blazor

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UITouchPoint : Object
```

Constructors

UITouchPoint()

Declaration

```
public UITouchPoint()
```

Properties

ClientX

The X coordinate of the touch point relative to the left edge of the browser viewport, not including any scroll offset.

Declaration

```
public long ClientX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ClientY

The Y coordinate of the touch point relative to the top edge of the browser viewport, not including any scroll offset.

Declaration

```
public long ClientY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

Identifier

A unique identifier for this Touch object. A given touch point (say, by a finger) will have the same identifier for the duration of its movement around the surface. This lets you ensure that you're tracking the same touch all the time.

Declaration

```
public long Identifier { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

PageX

The X coordinate of the touch point relative to the left edge of the document. Unlike ClientX, this value includes the horizontal scroll offset, if any.

Declaration

```
public long PageX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

PageY

The Y coordinate of the touch point relative to the top of the document. Unlike ClientY, this value includes the vertical scroll offset, if any.

Declaration

```
public long PageY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ScreenX

The X coordinate of the touch point relative to the left edge of the screen.

Declaration

```
public long ScreenX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

ScreenY

The Y coordinate of the touch point relative to the top edge of the screen.

Declaration

```
public long ScreenY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

Class UIWheelEventArgs

Supplies information about a mouse wheel event that is being raised.

Inheritance

System.Object

UIEventArgs

UIMouseEventArgs

UIWheelEventArgs

Inherited Members

UIMouseEventArgs.Detail

UIMouseEventArgs.ScreenX

UIMouseEventArgs.ScreenY

UIMouseEventArgs.ClientX

UIMouseEventArgs.ClientY

UIMouseEventArgs.Button

UIMouseEventArgs.Buttons

UIMouseEventArgs.CtrlKey

UIMouseEventArgs.ShiftKey

UIMouseEventArgs.AltKey

UIMouseEventArgs.MetaKey

UIEventArgs.Type

Namespace: **Microsoft.AspNetCore.Blazor**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class UIWheelEventArgs : UIMouseEventArgs
```

Constructors

UIWheelEventArgs()

Declaration

```
public UIWheelEventArgs()
```

Properties

DeltaMode

The unit of the delta values scroll amount.

Declaration

```
public long DeltaMode { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

DeltaX

The horizontal scroll amount.

Declaration

```
public double DeltaX { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

DeltaY

The vertical scroll amount.

Declaration

```
public double DeltaY { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

DeltaZ

The scroll amount for the z-axis.

Declaration

```
public double DeltaZ { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Namespace Microsoft.AspNetCore.Blazor.Browser.Http

Classes

BrowserHttpMessageHandler

A browser-compatible implementation of System.Net.Http.HttpMessageHandler

FetchCredentialsOption

Specifies a value for the 'credentials' option on outbound HTTP requests.

Class BrowserHttpMessageHandler

A browser-compatible implementation of System.Net.Http.HttpMessageHandler

Inheritance

System.Object

BrowserHttpMessageHandler

Namespace: **Microsoft.AspNetCore.Blazor.Browser.Http**

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public class BrowserHttpMessageHandler : HttpMessageHandler
```

Constructors

BrowserHttpMessageHandler()

Declaration

```
public BrowserHttpMessageHandler()
```

Fields

FetchArgs

The name of a well-known property that can be added to System.Net.Http.HttpRequestMessage.Properties to control the arguments passed to the underlying JavaScript

```
fetch
```

API.

Declaration

```
public const string FetchArgs = "BrowserHttpMessageHandler.FetchArgs"
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

DefaultCredentials

Gets or sets the default value of the 'credentials' option on outbound HTTP requests. Defaults to SameOrigin.

Declaration

```
public static FetchCredentialsOption DefaultCredentials { get; set; }
```

Property Value

TYPE	DESCRIPTION
FetchCredentialsOption	

Methods

SendAsync(HttpRequestMessage, CancellationToken)

Declaration

```
protected override Task<HttpResponseMessage> SendAsync(HttpRequestMessage request, CancellationToken
cancellationToken)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpRequestMessage	request	
System.Threading.CancellationToken	cancellationToken	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Net.Http.HttpResponseMessage>	

Declaration

```
protected override Task<HttpResponseMessage> SendAsync(HttpRequestMessage request, CancellationToken
cancellationToken)
```

Class FetchCredentialsOption

Specifies a value for the 'credentials' option on outbound HTTP requests.

Inheritance

System.Object

FetchCredentialsOption

Namespace: Microsoft.AspNetCore.Blazor.Browser.Http

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public sealed class FetchCredentialsOption : Enum
```

Fields

Include

Advises the browser to send credentials (such as cookies or HTTP auth headers) even for cross-origin requests.

Declaration

```
public const FetchCredentialsOption Include
```

Field Value

TYPE	DESCRIPTION
FetchCredentialsOption	

Omit

Advises the browser never to send credentials (such as cookies or HTTP auth headers).

Declaration

```
public const FetchCredentialsOption Omit
```

Field Value

TYPE	DESCRIPTION
FetchCredentialsOption	

SameOrigin

Advises the browser to send credentials (such as cookies or HTTP auth headers) only if the target URL is on the same origin as the calling application.

Declaration

```
public const FetchCredentialsOption SameOrigin
```

Field Value

TYPE	DESCRIPTION
FetchCredentialsOption	

value__

Declaration

```
public int value__
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Declaration

```
public int value__
```

Field Value

TYPE	DESCRIPTION

Namespace Microsoft.AspNetCore.Blazor.Browser.Rendering

Classes

BrowserRenderer

Provides mechanisms for rendering IComponent instances in a web browser, dispatching events to them, and refreshing the UI as required.

BrowserRendererEventDispatcher

Provides mechanisms for dispatching events to components in a BrowserRenderer. This is marked 'internal' because it only gets invoked from JS code.

BrowserRendererEventDispatcher.BrowserEventDescriptor

For framework use only.

Class BrowserRenderer

Provides mechanisms for rendering IComponent instances in a web browser, dispatching events to them, and refreshing the UI as required.

Inheritance

System.Object

Renderer

BrowserRenderer

Implements

System.IDisposable

Inherited Members

Renderer.InstantiateComponent(Type)

Renderer.AssignComponentId(IComponent)

Renderer.DispatchEvent(Int32, Int32, UIEventArgs)

Namespace: Microsoft.AspNetCore.Blazor.Browser.Rendering

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public class BrowserRenderer : Renderer, IDisposable
```

Constructors

BrowserRenderer()

Constructs an instance of BrowserRenderer.

Declaration

```
public BrowserRenderer()
```

BrowserRenderer(IServiceProvider)

Constructs an instance of BrowserRenderer.

Declaration

```
public BrowserRenderer(IServiceProvider serviceProvider)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IServiceProvider	serviceProvider	The System.IServiceProvider to use when initializing components.

Methods

AddComponent(Type, String)

Associates the IComponent with the BrowserRenderer, causing it to be displayed in the specified DOM element.

Declaration

```
public void AddComponent(Type componentType, string domElementSelector)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	componentType	The type of the component.
System.String	domElementSelector	A CSS selector that uniquely identifies a DOM element.

AddComponent<TComponent>(String)

Attaches a new root component to the renderer, causing it to be displayed in the specified DOM element.

Declaration

```
public void AddComponent<TComponent>(string domElementSelector)
    where TComponent : IComponent
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	domElementSelector	A CSS selector that uniquely identifies a DOM element.

Type Parameters

NAME	DESCRIPTION
TComponent	The type of the component.

Dispose()

Disposes the instance.

Declaration

```
public void Dispose()
```

UpdateDisplay(RenderBatch)

Updates the visible UI.

Declaration

```
protected override void UpdateDisplay(in RenderBatch batch)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderBatch	batch	

Overrides

Renderer.UpdateDisplay(RenderBatch)

Implements

System.IDisposable

Class BrowserRendererEventDispatcher

Provides mechanisms for dispatching events to components in a BrowserRenderer. This is marked 'internal' because it only gets invoked from JS code.

Inheritance

System.Object

BrowserRendererEventDispatcher

Namespace: Microsoft.AspNetCore.Blazor.Browser.Rendering

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public static class BrowserRendererEventDispatcher : Object
```

Methods

DispatchEvent(BrowserRendererEventDispatcher.BrowserEventDescriptor, String)

For framework use only.

Declaration

```
[JSInvokable("DispatchEvent")]
public static void DispatchEvent(BrowserRendererEventDispatcher.BrowserEventDescriptor eventDescriptor, string eventArgsJson)
```

Parameters

TYPE	NAME	DESCRIPTION
BrowserRendererEventDispatcher.BrowserEventDescriptor	eventDescriptor	
System.String	eventArgsJson	

Class
BrowserRendererEventDispatcher.BrowserEventDescriptor

For framework use only.

Inheritance

System.Object

BrowserRendererEventDispatcher.BrowserEventDescriptor

Namespace: **Microsoft.AspNetCore.Blazor.Browser.Rendering**

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public class BrowserEventDescriptor : Object
```

Constructors

BrowserEventDescriptor()

Declaration

```
public BrowserEventDescriptor()
```

Properties

BrowserRendererId

For framework use only.

Declaration

```
public int BrowserRendererId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

ComponentId

For framework use only.

Declaration

```
public int ComponentId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

EventArgsType

For framework use only.

Declaration

```
public string EventArgsType { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

EventHandlerId

For framework use only.

Declaration

```
public int EventHandlerId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Namespace Microsoft.AspNetCore.Blazor.Browser.Services

Classes

ActivateMonoJSRuntime

Temporary mechanism for registering the Mono JS runtime. Developers do not need to use this directly, and it will be removed shortly.

BrowserServiceProvider

An implementation of System.IServiceProvider configured with default services suitable for use in a browser environment.

BrowserUriHelper

Default browser implementation of IUriHelper.

Class ActivateMonoJSRuntime

Temporary mechanism for registering the Mono JS runtime. Developers do not need to use this directly, and it will be removed shortly.

Inheritance

System.Object

ActivateMonoJSRuntime

Namespace: Microsoft.AspNetCore.Blazor.Browser.Services

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public class ActivateMonoJSRuntime : Object
```

Constructors

ActivateMonoJSRuntime()

Declaration

```
public ActivateMonoJSRuntime()
```

Methods

EnsureActivated()

Temporary mechanism for registering the Mono JS runtime. Developers do not need to use this directly, and it will be removed shortly. The return value is intended to be used with GC.KeepAlive purely as a way of ensuring the invocation doesn't get stripped out by the linker (if it's smart enough to do so).

Declaration

```
public static object EnsureActivated()
```

Returns

TYPE	DESCRIPTION
System.Object	

Class BrowserServiceProvider

An implementation of System.IServiceProvider configured with default services suitable for use in a browser environment.

Inheritance

System.Object

BrowserServiceProvider

Implements

System.IServiceProvider

Namespace: Microsoft.AspNetCore.Blazor.Browser.Services

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public class BrowserServiceProvider : Object, IServiceProvider
```

Constructors

BrowserServiceProvider()

Constructs an instance of BrowserServiceProvider.

Declaration

```
public BrowserServiceProvider()
```

BrowserServiceProvider(Action<IServiceCollection>)

Constructs an instance of BrowserServiceProvider.

Declaration

```
public BrowserServiceProvider(Action<IServiceCollection> configure)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<Microsoft.Extensions.DependencyInjection.IServiceCollection>	configure	A callback that can be used to configure the Microsoft.Extensions.DependencyInjection.IServiceCollection.

Methods

GetService(Type)

Declaration

```
public object GetService(Type serviceType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	serviceType	

Returns

TYPE	DESCRIPTION
System.Object	

Implements

System.IServiceProvider

Class BrowserUriHelper

Default browser implementation of IUriHelper.

Inheritance

System.Object

UriHelperBase

BrowserUriHelper

Implements

IUriHelper

Inherited Members

UriHelperBase.NavigateTo(String)

UriHelperBase.GetAbsoluteUri()

UriHelperBase.GetBaseUri()

UriHelperBase.ToAbsoluteUri(String)

UriHelperBase.ToBaseRelativePath(String, String)

UriHelperBase.SetAbsoluteUri(String)

UriHelperBase.SetAbsoluteBaseUri(String)

UriHelperBase.TriggerOnLocationChanged()

UriHelperBase.OnLocationChanged

Namespace: Microsoft.AspNetCore.Blazor.Browser.Services

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public class BrowserUriHelper : UriHelperBase, IUriHelper
```

Fields

Instance

Gets the instance of BrowserUriHelper.

Declaration

```
public static readonly BrowserUriHelper Instance
```

Field Value

TYPE	DESCRIPTION
BrowserUriHelper	

Methods

InitializeState()

Called to initialize BaseURI and current URI before those values the first time. Override this method to dynamically calculate the those values.

Declaration

```
protected override void InitializeState()
```

Overrides

UriHelperBase.InitializeState()

NavigateToCore(String)

Navigates to the specified URI.

Declaration

```
protected override void NavigateToCore(string uri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uri	The destination URI. This can be absolute, or relative to the base URI (as returned by GetBaseUri()).

Overrides

UriHelperBase.NavigateToCore(String)

NotifyLocationChanged(String)

For framework use only.

Declaration

```
[JSInvokable("NotifyLocationChanged")]
public static void NotifyLocationChanged(string newAbsoluteUri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	newAbsoluteUri	

Implements

IUriHelper

Namespace Microsoft.AspNetCore.Blazor.Builder

Classes

BlazorApplicationBuilderExtensions

Provides extension methods for IBlazorApplicationBuilder.

Interfaces

IBlazorApplicationBuilder

A builder for constructing a Blazor application.

Class BlazorApplicationBuilderExtensions

Provides extension methods for IBlazorApplicationBuilder.

Inheritance

System.Object

BlazorApplicationBuilderExtensions

Namespace: Microsoft.AspNetCore.Blazor.Builder

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public static class BlazorApplicationBuilderExtensions : Object
```

Methods

AddComponent<TComponent>(IBlazorApplicationBuilder, String)

Associates the component type with the application, causing it to be displayed in the specified DOM element.

Declaration

```
public static void AddComponent<TComponent>(this IBlazorApplicationBuilder app, string domElementSelector)
    where TComponent : IComponent
```

Parameters

TYPE	NAME	DESCRIPTION
IBlazorApplicationBuilder	app	The IBlazorApplicationBuilder.
System.String	domElementSelector	A CSS selector that uniquely identifies a DOM element.

Type Parameters

NAME	DESCRIPTION
TComponent	The type of the component.

Interface IBlazorApplicationBuilder

A builder for constructing a Blazor application.

Syntax

```
public interface IBlazorApplicationBuilder
```

Properties

Services

Gets the application services.

Declaration

```
IServiceProvider Services { get; }
```

Property Value

TYPE	DESCRIPTION
System.IServiceProvider	

Methods

AddComponent(Type, String)

Associates the IComponent with the application, causing it to be displayed in the specified DOM element.

Declaration

```
void AddComponent(Type componentType, string domElementSelector)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	componentType	The type of the component.
System.String	domElementSelector	A CSS selector that uniquely identifies a DOM element.

Extension Methods

BlazorApplicationBuilderExtensions.AddComponent<TComponent>(IBlazorApplicationBuilder, String)

Namespace Microsoft.AspNetCore.Blazor.Components

Classes

BindAttributes

Infrastructure for the discovery of `bind` attributes for markup elements.

BindElementAttribute

Configures options for binding specific element types.

BindInputElementAttribute

Configures options for binding subtypes of an HTML

```
input
```

element.

BindMethods

Methods used internally by @bind syntax. Not intended to be used directly.

BlazorComponent

Optional base class for Blazor components. Alternatively, Blazor components may implement IComponent directly.

EventHandlerAttribute

Associates an event argument type with an event attribute name.

EventHandlerInvoker

A bound event handler delegate.

EventHandlers

Holds System.EventHandler attributes to configure the mappings between event names and event argument types.

InjectAttribute

Indicates that the associated property should have a value injected from the service provider during initialization.

Parameter

Represents a single parameter supplied to an IComponent by its parent in the render tree.

ParameterAttribute

Denotes the target member as a component parameter.

ParameterCollection

Represents a collection of parameters supplied to an IComponent by its parent in the render tree.

ParameterCollectionExtensions

Extension methods for the ParameterCollection type.

ParameterEnumerator

An enumerator that iterates through a ParameterCollection.

RenderHandle

Allows a component to notify the renderer that it should be rendered.

RouteAttribute

Indicates that the associated component should match the specified route template pattern.

RuntimeHelpers

Used by generated code produced by the Blazor code generator. Not intended or supported for use in application code.

Interfaces

IComponent

Represents a UI component.

IHandleAfterRender

Interface implemented by components that receive notification that they have been rendered.

IHandleEvent

Interface implemented by components that receive notification of their events.

Class BindAttributes

Infrastructure for the discovery of `bind` attributes for markup elements.

Inheritance

System.Object

BindAttributes

Syntax

```
[BindInputElement(null, null, "value", "onchange")]
[BindInputElement("checkbox", null, "checked", "onchange")]
[BindInputElement("text", null, "value", "onchange")]
[BindElement("select", null, "value", "onchange")]
[BindElement("textarea", null, "value", "onchange")]
public static class BindAttributes : Object
```

Remarks

To extend the set of `bind` attributes, define a public class named `BindAttributes` and annotate it with the appropriate attributes.

Class BindElementAttribute

Configures options for binding specific element types.

Syntax

```
public sealed class BindElementAttribute : Attribute
```

Constructors

BindElementAttribute(String, String, String, String)

Constructs an instance of BindElementAttribute.

Declaration

```
public BindElementAttribute(string element, string suffix, string valueAttribute, string changeAttribute)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	element	The tag name of the element.
System.String	suffix	The suffix value. For example, set this to `value` for `bind-value`, or set this to `null` for `bind`.
System.String	valueAttribute	The name of the value attribute to be bound.
System.String	changeAttribute	The name of an attribute that will register an associated change event.

Properties

ChangeAttribute

Gets the name of an attribute that will register an associated change event.

Declaration

```
public string ChangeAttribute { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Element

Gets the tag name of the element.

Declaration

```
public string Element { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Suffix

Gets the suffix value. For example, this will be

```
value
```

to mean

```
bind-value
```

, or

```
null
```

to mean

```
bind
```

.

Declaration

```
public string Suffix { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ValueAttribute

Gets the name of the value attribute to be bound.

Declaration

```
public string ValueAttribute { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Declaration

```
public string ValueAttribute { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class BindInputElementAttribute

Configures options for binding subtypes of an HTML

```
input
```

element.

Inheritance

System.Object

BindInputElementAttribute

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class BindInputElementAttribute : Attribute
```

Constructors

BindInputElementAttribute(String, String, String, String)

Constructs an instance of BindInputElementAttribute.

Declaration

```
public BindInputElementAttribute(string type, string suffix, string valueAttribute, string changeAttribute)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	type	The value of the element's ``` type ``` attribute.
System.String	suffix	The suffix value.
System.String	valueAttribute	The name of the value attribute to be bound.
System.String	changeAttribute	The name of an attribute that will register an associated change event.

Properties

ChangeAttribute

Gets the name of an attribute that will register an associated change event.

Declaration

```
public string ChangeAttribute { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Suffix

Gets the suffix value.

Declaration

```
public string Suffix { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Type

Gets the value of the element's

```
type
```

attribute.

Declaration

```
public string Type { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ValueAttribute

Gets the name of the value attribute to be bound.

Declaration

```
public string ValueAttribute { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class BindMethods

Methods used internally by @bind syntax. Not intended to be used directly.

Inheritance

System.Object

BindMethods

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public static class BindMethods : Object
```

Methods

GetEventHandlerValue<T>(Action)

Not intended to be used directly.

Declaration

```
public static MulticastDelegate GetEventHandlerValue<T>(Action value)
    where T : UIEventArgs
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	value	

Returns

TYPE	DESCRIPTION
System.MulticastDelegate	

Type Parameters

NAME	DESCRIPTION
T	

GetEventHandlerValue<T>(Action<T>)

Not intended to be used directly.

Declaration

```
public static MulticastDelegate GetEventHandlerValue<T>(Action<T> value)
    where T : UIEventArgs
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<T>	value	

Returns

TYPE	DESCRIPTION
System.MulticastDelegate	

Type Parameters

NAME	DESCRIPTION
T	

GetEventHandlerValue<T>(Func<T, Task>)

Not intended to be used directly.

Declaration

```
public static MulticastDelegate GetEventHandlerValue<T>(Func<T, Task> value)
    where T : UIEventArgs
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<T, System.Threading.Tasks.Task>	value	

Returns

TYPE	DESCRIPTION
System.MulticastDelegate	

Type Parameters

NAME	DESCRIPTION
T	

GetEventHandlerValue<T>(Func<Task>)

Not intended to be used directly.

Declaration

```
public static MulticastDelegate GetEventHandlerValue<T>(Func<Task> value)
    where T : UIEventArgs
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.Threading.Tasks.Task>	value	

Returns

TYPE	DESCRIPTION
System.MulticastDelegate	

Type Parameters

NAME	DESCRIPTION
T	

GetEventHandlerValue<T>(String)

Not intended to be used directly.

Declaration

```
public static string GetEventHandlerValue<T>(string value)
    where T : UIEventArgs
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	

Returns

TYPE	DESCRIPTION
System.String	

Type Parameters

NAME	DESCRIPTION
T	

GetValue(DateTime, String)

Not intended to be used directly.

Declaration

```
public static string GetValue(DateTime value, string format)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	value	
System.String	format	

Returns

TYPE	DESCRIPTION
System.String	

GetValue<T>(T)

Not intended to be used directly.

Declaration

```
public static T GetValue<T>(T value)
```

Parameters

TYPE	NAME	DESCRIPTION
T	value	

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	

SetValueHandler(Action<Boolean>, Boolean)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<bool> setter, bool existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Boolean>	setter	
System.Boolean	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler(Action<DateTime>, DateTime)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<DateTime> setter, DateTime existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.DateTime>	setter	
System.DateTime	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler(Action<DateTime>, DateTime, String)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<DateTime> setter, DateTime existingValue, string format)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.DateTime>	setter	
System.DateTime	existingValue	
System.String	format	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler(Action<Decimal>, Decimal)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<Decimal> setter, Decimal existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Decimal>	setter	
System.Decimal	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler(Action<Double>, Double)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<double> setter, double existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Double>	setter	
System.Double	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler(Action<Int32>, Int32)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<int> setter, int existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Int32>	setter	
System.Int32	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler(Action<Int64>, Int64)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<long> setter, long existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Int64>	setter	
System.Int64	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler(Action<Single>, Single)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<float> setter, float existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Single>	setter	
System.Single	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler(Action<String>, String)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler(Action<string> setter, string existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.String>	setter	
System.String	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

SetValueHandler<T>(Action<T>, T)

Not intended to be used directly.

Declaration

```
public static Action<UIEventArgs> SetValueHandler<T>(Action<T> setter, T existingValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<T>	setter	
T	existingValue	

Returns

TYPE	DESCRIPTION
System.Action<UIEventArgs>	

Type Parameters

NAME	DESCRIPTION
T	

Class BlazorComponent

Optional base class for Blazor components. Alternatively, Blazor components may implement IComponent directly.

Inheritance

System.Object

BlazorComponent

BlazorLayoutComponent

Implements

IComponent

IHandleEvent

IHandleAfterRender

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public abstract class BlazorComponent : Object, IComponent, IHandleEvent, IHandleAfterRender
```

Constructors

BlazorComponent()

Constructs an instance of BlazorComponent.

Declaration

```
public BlazorComponent()
```

Fields

BuildRenderTreeMethodName

Specifies the name of the Microsoft.AspNetCore.Blazor.RenderTree-building method.

Declaration

```
public const string BuildRenderTreeMethodName = "BuildRenderTree"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

BuildRenderTree(RenderTreeBuilder)

Renders the component to the supplied RenderTreeBuilder.

Declaration

```
protected virtual void BuildRenderTree(RenderTreeBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	A RenderTreeBuilder that will receive the render output.

OnAfterRender()

Method invoked after each time the component has been rendered.

Declaration

```
protected virtual void OnAfterRender()
```

OnAfterRenderAsync()

Method invoked after each time the component has been rendered. Note that the component does not automatically re-render after the completion of any returned System.Threading.Tasks.Task, because that would cause an infinite render loop.

Declaration

```
protected virtual Task OnAfterRenderAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	A System.Threading.Tasks.Task representing any asynchronous operation, or null.

OnInit()

Method invoked when the component is ready to start, having received its initial parameters from its parent in the render tree.

Declaration

```
protected virtual void OnInit()
```

OnInitAsync()

Method invoked when the component is ready to start, having received its initial parameters from its parent in the render tree.

Override this method if you will perform an asynchronous operation and want the component to refresh when that operation is completed.

Declaration

```
protected virtual Task OnInitAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	A System.Threading.Tasks.Task representing any asynchronous operation, or null.

OnParametersSet()

Method invoked when the component has received parameters from its parent in the render tree, and the incoming values have been assigned to properties.

Declaration

```
protected virtual void OnParametersSet()
```

OnParametersSetAsync()

Method invoked when the component has received parameters from its parent in the render tree, and the incoming values have been assigned to properties.

Declaration

```
protected virtual Task OnParametersSetAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	A System.Threading.Tasks.Task representing any asynchronous operation, or null.

SetParameters(ParameterCollection)

Method invoked to apply initial or updated parameters to the component.

Declaration

```
public virtual void SetParameters(ParameterCollection parameters)
```

Parameters

TYPE	NAME	DESCRIPTION
ParameterCollection	parameters	The parameters to apply.

ShouldRender()

Returns a flag to indicate whether the component should render.

Declaration

```
protected virtual bool ShouldRender()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

StateHasChanged()

Notifies the component that its state has changed. When applicable, this will cause the component to be re-rendered.

Declaration

```
protected void StateHasChanged()
```

Explicit Interface Implementations

IComponent.Init(RenderHandle)

Declaration

```
void IComponent.Init(RenderHandle renderHandle)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderHandle	renderHandle	

IHandleAfterRender.OnAfterRender()

Declaration

```
void IHandleAfterRender.OnAfterRender()
```

IHandleEvent.HandleEvent(EventHandlerInvoker, UIEventArgs)

Declaration

```
void IHandleEvent.HandleEvent(EventHandlerInvoker binding, UIEventArgs args)
```

Parameters

TYPE	NAME	DESCRIPTION
EventHandlerInvoker	binding	
UIEventArgs	args	

Implements

IComponent
IHandleEvent
IHandleAfterRender

Class EventHandlerAttribute

Associates an event argument type with an event attribute name.

Inheritance

System.Object

EventHandlerAttribute

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class EventHandlerAttribute : Attribute
```

Constructors

EventHandlerAttribute(String, Type)

Constructs an instance of EventHandlerAttribute.

Declaration

```
public EventHandlerAttribute(string attributeName, Type eventArgsType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	attributeName	
System.Type	eventArgsType	

Properties

AttributeName

Gets the attribute name.

Declaration

```
public string AttributeName { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

EventArgsType

Gets the event argument type.

Declaration

```
public Type EventArgsType { get; }
```

Property Value

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.Type	

Class EventHandlerInvoker

A bound event handler delegate.

Inheritance

System.Object

EventHandlerInvoker

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class EventHandlerInvoker : ValueType
```

Constructors

EventHandlerInvoker(MulticastDelegate)

Creates the new EventHandlerInvoker.

Declaration

```
public EventHandlerInvoker(MulticastDelegate delegate)
```

Parameters

TYPE	NAME	DESCRIPTION
System.MulticastDelegate	delegate	The delegate to bind.

Methods

Invoke(UIEventArgs)

Invokes the delegate associated with this binding.

Declaration

```
public Task Invoke(UIEventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
UIEventArgs	e	The UIEventArgs.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Class EventHandlers

Holds System.EventHandler attributes to configure the mappings between event names and event argument types.

Inheritance

System.Object

EventHandlers

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
```

```
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
[EventHandler]
public static class EventHandlers : Object
```

Interface IComponent

Represents a UI component.

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public interface IComponent
```

Methods

Init(RenderHandle)

Initializes the component.

Declaration

```
void Init(RenderHandle renderHandle)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderHandle	renderHandle	A RenderHandle that allows the component to be rendered.

SetParameters(ParameterCollection)

Sets parameters supplied by the component's parent in the render tree.

Declaration

```
void SetParameters(ParameterCollection parameters)
```

Parameters

TYPE	NAME	DESCRIPTION
ParameterCollection	parameters	The parameters.

Interface IHandleAfterRender

Interface implemented by components that receive notification that they have been rendered.

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public interface IHandleAfterRender
```

Methods

OnAfterRender()

Notifies the component that it has been rendered.

Declaration

```
void OnAfterRender()
```

Interface IHandleEvent

Interface implemented by components that receive notification of their events.

Syntax

```
public interface IHandleEvent
```

Methods

HandleEvent(EventHandlerInvoker, UIEventArgs)

Notifies the component that one of its event handlers has been triggered.

Declaration

```
void HandleEvent(EventHandlerInvoker binding, UIEventArgs args)
```

Parameters

TYPE	NAME	DESCRIPTION
EventHandlerInvoker	binding	The event binding.
UIEventArgs	args	Arguments for the event handler.

Class InjectAttribute

Indicates that the associated property should have a value injected from the service provider during initialization.

Inheritance

System.Object

InjectAttribute

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class InjectAttribute : Attribute
```

Constructors

InjectAttribute()

Declaration

```
public InjectAttribute()
```

Class Parameter

Represents a single parameter supplied to an IComponent by its parent in the render tree.

Inheritance

System.Object

Parameter

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class Parameter : ValueType
```

Properties

Name

Gets the name of the parameter.

Declaration

```
public string Name { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Value

Gets the value of the parameter.

Declaration

```
public object Value { get; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

Class ParameterAttribute

Denotes the target member as a component parameter.

Inheritance

System.Object

ParameterAttribute

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class ParameterAttribute : Attribute
```

Constructors

ParameterAttribute()

Declaration

```
public ParameterAttribute()
```

Class ParameterCollection

Represents a collection of parameters supplied to an IComponent by its parent in the render tree.

Inheritance

System.Object

ParameterCollection

Namespace: **Microsoft.AspNetCore.Blazor.Components**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class ParameterCollection : ValueType
```

Properties

Empty

Gets an empty ParameterCollection.

Declaration

```
public static ParameterCollection Empty { get; }
```

Property Value

TYPE	DESCRIPTION
ParameterCollection	

Methods

GetEnumerator()

Returns an enumerator that iterates through the ParameterCollection.

Declaration

```
public ParameterEnumerator GetEnumerator()
```

Returns

TYPE	DESCRIPTION
ParameterEnumerator	The enumerator.

GetValueOrDefault<T>(String)

Gets the value of the parameter with the specified name, or a default value if no such parameter exists in the collection.

Declaration

```
public T GetValueOrDefault<T>(string parameterName)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	parameterName	The name of the parameter.

Returns

TYPE	DESCRIPTION
T	The parameter value if found; otherwise the default value for the specified type.

Type Parameters

NAME	DESCRIPTION
T	The type of the value.

GetValueOrDefault<T>(String, T)

Gets the value of the parameter with the specified name, or a specified default value if no such parameter exists in the collection.

Declaration

```
public T GetValueOrDefault<T>(string parameterName, T defaultValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	parameterName	The name of the parameter.
T	defaultValue	The default value to return if no such parameter exists in the collection.

Returns

TYPE	DESCRIPTION
T	The parameter value if found; otherwise `defaultValue`.

Type Parameters

NAME	DESCRIPTION
T	The type of the value.

ToDictionary()

Returns a dictionary populated with the contents of the ParameterCollection.

Declaration

```
public IReadOnlyDictionary<string, object> ToDictionary()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IReadOnlyDictionary<System.String, System.Object>	A dictionary populated with the contents of the ParameterCollection.

TryGetValue<T>(String, out T)

Gets the value of the parameter with the specified name.

Declaration

```
public bool TryGetValue<T>(string parameterName, out T result)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	parameterName	The name of the parameter.
T	result	Receives the result, if any.

Returns

TYPE	DESCRIPTION
System.Boolean	True if a matching parameter was found; false otherwise.

Type Parameters

NAME	DESCRIPTION
T	The type of the value.

Extension Methods

ParameterCollectionExtensions.AssignToProperties(ParameterCollection, Object)

Class ParameterCollectionExtensions

Extension methods for the ParameterCollection type.

Inheritance

System.Object

ParameterCollectionExtensions

Namespace: **Microsoft.AspNetCore.Blazor.Components**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public static class ParameterCollectionExtensions : Object
```

Methods

AssignToProperties(ParameterCollection, Object)

Iterates through the ParameterCollection, assigning each parameter to a property of the same name on `target`.

Declaration

```
public static void AssignToProperties(this in ParameterCollection parameterCollection, object target)
```

Parameters

TYPE	NAME	DESCRIPTION
ParameterCollection	parameterCollection	The ParameterCollection.
System.Object	target	An object that has a public writable property matching each parameter's name and type.

Class ParameterEnumerator

An enumerator that iterates through a ParameterCollection.

Inheritance

System.Object

ParameterEnumerator

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class ParameterEnumerator : ValueType
```

Properties

Current

Gets the current value of the enumerator.

Declaration

```
public Parameter Current { get; }
```

Property Value

TYPE	DESCRIPTION
Parameter	

Methods

MoveNext()

Instructs the enumerator to move to the next value in the sequence.

Declaration

```
public bool MoveNext()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

Class RenderHandle

Allows a component to notify the renderer that it should be rendered.

Inheritance

System.Object

RenderHandle

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class RenderHandle : ValueType
```

Properties

IsInitialized

Gets a value that indicates whether the RenderHandle has been initialised and is ready to use.

Declaration

```
public bool IsInitialized { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Render(RenderFragment)

Notifies the renderer that the component should be rendered.

Declaration

```
public void Render(RenderFragment renderFragment)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderFragment	renderFragment	The content that should be rendered.

Class RouteAttribute

Indicates that the associated component should match the specified route template pattern.

Inheritance

System.Object

RouteAttribute

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class RouteAttribute : Attribute
```

Constructors

RouteAttribute(String)

Constructs an instance of RouteAttribute.

Declaration

```
public RouteAttribute(string template)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	template	The route template.

Properties

Template

Gets the route template.

Declaration

```
public string Template { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class RuntimeHelpers

Used by generated code produced by the Blazor code generator. Not intended or supported for use in application code.

Inheritance

System.Object

RuntimeHelpers

Namespace: Microsoft.AspNetCore.Blazor.Components

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public static class RuntimeHelpers : Object
```

Methods

TypeCheck<T>(T)

Not intended for use by application code.

Declaration

```
public static T TypeCheck<T>(T value)
```

Parameters

TYPE	NAME	DESCRIPTION
T	value	

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	

Namespace Microsoft.AspNetCore.Blazor.Hosting

Classes

BlazorWebAssemblyHost

Used to to create instances a Blazor host builder for a Browser application.

WebAssemblyHostBuilderContext

Context containing the common services on the IWebAssemblyHost. Some properties may be null until set by the IWebAssemblyHost.

WebAssemblyHostBuilderExtensions

Provides Blazor-specific support for IWebAssemblyHost.

WebAssemblyHostExtensions

Extension methods for IWebAssemblyHost.

Interfaces

IWebAssemblyHost

A program abstraction.

IWebAssemblyHostBuilder

Abstraction for configuring a Blazor browser-based application.

Class BlazorWebAssemblyHost

Used to to create instances a Blazor host builder for a Browser application.

Inheritance

System.Object

BlazorWebAssemblyHost

Namespace: Microsoft.AspNetCore.Blazor.Hosting

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public static class BlazorWebAssemblyHost : Object
```

Methods

CreateDefaultBuilder()

Creates a an instance of IWebAssemblyHostBuilder.

Declaration

```
public static IWebAssemblyHostBuilder CreateDefaultBuilder()
```

Returns

TYPE	DESCRIPTION
IWebAssemblyHostBuilder	The IWebAssemblyHostBuilder.

Interface IWebAssemblyHost

A program abstraction.

Namespace: Microsoft.AspNetCore.Blazor.Hosting

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public interface IWebAssemblyHost : IDisposable
```

Properties

Services

The programs configured services.

Declaration

```
IServiceProvider Services { get; }
```

Property Value

TYPE	DESCRIPTION
System.IServiceProvider	

Methods

StartAsync(CancellationToken)

Start the program.

Declaration

```
Task StartAsync(CancellationToken cancellationToken = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Threading.CancellationToken	cancellationToken	Used to abort program start.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

StopAsync(CancellationToken)

Attempts to gracefully stop the program.

Declaration

```
Task StopAsync(CancellationToken cancellationToken = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Threading.CancellationToken	cancellationToken	Used to indicate when stop should no longer be graceful.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Extension Methods

WebAssemblyHostExtensions.Run(IWebAssemblyHost)

Interface IWebAssemblyHostBuilder

Abstraction for configuring a Blazor browser-based application.

Syntax

```
public interface IWebAssemblyHostBuilder
```

Properties

Properties

A central location for sharing state between components during the host building process.

Declaration

```
IDictionary<object, object> Properties { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IDictionary<System.Object, System.Object>	

Methods

Build()

Run the given actions to initialize the host. This can only be called once.

Declaration

```
IWebAssemblyHost Build()
```

Returns

TYPE	DESCRIPTION
IWebAssemblyHost	An initialized IWebAssemblyHost

ConfigureServices(Action<WebAssemblyHostBuilderContext, IServiceCollection>)

Adds services to the container. This can be called multiple times and the results will be additive.

Declaration

```
IWebAssemblyHostBuilder ConfigureServices(Action<WebAssemblyHostBuilderContext, IServiceCollection>
configureDelegate)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<WebAssemblyHostBuilderContext, Microsoft.Extensions.DependencyInjection.IServiceCollection>	configureDelegate	The delegate for configuring the Microsoft.Extensions.DependencyInjection.IServiceCollection that will be used to construct the System.IServiceProvider.

Returns

TYPE	DESCRIPTION
IWebAssemblyHostBuilder	The same instance of the IWebAssemblyHostBuilder for chaining.

Extension Methods

WebAssemblyHostBuilderExtensions.ConfigureServices(IWebAssemblyHostBuilder, Action<IServiceCollection>)

WebAssemblyHostBuilderExtensions.UseBlazorStartup(IWebAssemblyHostBuilder, Type)

WebAssemblyHostBuilderExtensions.UseBlazorStartup<TStartup>(IWebAssemblyHostBuilder)

Class WebAssemblyHostBuilderContext

Context containing the common services on the IWebAssemblyHost. Some properties may be null until set by the IWebAssemblyHost.

Inheritance

System.Object

WebAssemblyHostBuilderContext

Namespace: Microsoft.AspNetCore.Blazor.Hosting

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public sealed class WebAssemblyHostBuilderContext : Object
```

Constructors

WebAssemblyHostBuilderContext(IDictionary<Object, Object>)

Creates a new WebAssemblyHostBuilderContext.

Declaration

```
public WebAssemblyHostBuilderContext(IDictionary<object, object> properties)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IDictionary<System.Object, System.Object>	properties	The property collection.

Properties

Properties

A central location for sharing state between components during the host building process.

Declaration

```
public IDictionary<object, object> Properties { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IDictionary<System.Object, System.Object>	

Class WebAssemblyHostBuilderExtensions

Provides Blazor-specific support for IWebAssemblyHost.

Inheritance

System.Object

WebAssemblyHostBuilderExtensions

Namespace: Microsoft.AspNetCore.Blazor.Hosting

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public static class WebAssemblyHostBuilderExtensions : Object
```

Methods

ConfigureServices(IWebAssemblyHostBuilder, Action<IServiceCollection>)

Adds services to the container. This can be called multiple times and the results will be additive.

Declaration

```
public static IWebAssemblyHostBuilder ConfigureServices(this IWebAssemblyHostBuilder hostBuilder,
Action<IServiceCollection> configureDelegate)
```

Parameters

TYPE	NAME	DESCRIPTION
IWebAssemblyHostBuilder	hostBuilder	The IWebAssemblyHostBuilder to configure.
System.Action<Microsoft.Extensions.DependencyInjection.IServiceCollection>	configureDelegate	

Returns

TYPE	DESCRIPTION
IWebAssemblyHostBuilder	The same instance of the IWebAssemblyHostBuilder for chaining.

UseBlazorStartup(IWebAssemblyHostBuilder, Type)

Configures the IWebAssemblyHostBuilder to use the provided startup class.

Declaration

```
public static IWebAssemblyHostBuilder UseBlazorStartup(this IWebAssemblyHostBuilder builder, Type startupType)
```

Parameters

TYPE	NAME	DESCRIPTION
IWebAssemblyHostBuilder	builder	The IWebAssemblyHostBuilder.

TYPE	NAME	DESCRIPTION
System.Type	startupType	A type that configures a Blazor application.

Returns

TYPE	DESCRIPTION
IWebAssemblyHostBuilder	The IWebAssemblyHostBuilder.

UseBlazorStartup<TStartup>(IWebAssemblyHostBuilder)

Configures the IWebAssemblyHostBuilder to use the provided startup class.

Declaration

```
public static IWebAssemblyHostBuilder UseBlazorStartup<TStartup>(this IWebAssemblyHostBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
IWebAssemblyHostBuilder	builder	The IWebAssemblyHostBuilder.

Returns

TYPE	DESCRIPTION
IWebAssemblyHostBuilder	The IWebAssemblyHostBuilder.

Type Parameters

NAME	DESCRIPTION
TStartup	A type that configures a Blazor application.

Class WebAssemblyHostExtensions

Extension methods for IWebAssemblyHost.

Inheritance

System.Object

WebAssemblyHostExtensions

Namespace: Microsoft.AspNetCore.Blazor.Hosting

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public static class WebAssemblyHostExtensions : Object
```

Methods

Run(IWebAssemblyHost)

Runs the application.

Declaration

```
public static void Run(this IWebAssemblyHost host)
```

Parameters

TYPE	NAME	DESCRIPTION
IWebAssemblyHost	host	The IWebAssemblyHost to run.

Remarks

Currently, Blazor applications running in the browser don't have a lifecycle - the application does not get a chance to gracefully shut down. For now, Run(IWebAssemblyHost) simply starts the host and allows execution to continue.

Namespace Microsoft.AspNetCore.Blazor.Layouts

Classes

BlazorLayoutComponent

Optional base class for components that represent a layout. Alternatively, Blazor components may implement IComponent directly and declare their own parameter named Body.

LayoutAttribute

Indicates that the associated component type uses a specified layout.

LayoutDisplay

Displays the specified page component, rendering it inside its layout and any further nested layouts.

Class BlazorLayoutComponent

Optional base class for components that represent a layout. Alternatively, Blazor components may implement IComponent directly and declare their own parameter named Body.

Inheritance

System.Object

BlazorComponent

BlazorLayoutComponent

Implements

IComponent

IHandleEvent

IHandleAfterRender

Inherited Members

BlazorComponent.BuildRenderTreeMethodName

BlazorComponent.BuildRenderTree(RenderTreeBuilder)

BlazorComponent.OnInit()

BlazorComponent.OnInitAsync()

BlazorComponent.OnParametersSet()

BlazorComponent.OnParametersSetAsync()

BlazorComponent.StateHasChanged()

BlazorComponent.ShouldRender()

BlazorComponent.OnAfterRender()

BlazorComponent.OnAfterRenderAsync()

BlazorComponent.IComponent.Init(RenderHandle)

BlazorComponent.SetParameters(ParameterCollection)

BlazorComponent.IHandleEvent.HandleEvent(EventHandlerInvoker, UIEventArgs)

BlazorComponent.IHandleAfterRender.OnAfterRender()

Namespace: Microsoft.AspNetCore.Blazor.Layouts

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public abstract class BlazorLayoutComponent : BlazorComponent, IComponent, IHandleEvent, IHandleAfterRender
```

Constructors

BlazorLayoutComponent()

Declaration

```
protected BlazorLayoutComponent()
```

Properties

Body

Gets the content to be rendered inside the layout.

Declaration

```
[Parameter]
protected RenderFragment Body { get; }
```

Property Value

TYPE	DESCRIPTION
RenderFragment	

Implements

IComponent
IHandleEvent
IHandleAfterRender

Class LayoutAttribute

Indicates that the associated component type uses a specified layout.

Inheritance

System.Object

LayoutAttribute

Namespace: Microsoft.AspNetCore.Blazor.Layouts

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class LayoutAttribute : Attribute
```

Constructors

LayoutAttribute(Type)

Constructs an instance of LayoutAttribute.

Declaration

```
public LayoutAttribute(Type layoutType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	layoutType	The type of the layout.

Properties

LayoutType

The type of the layout. The type myst implement IComponent and must accept a parameter with the name 'Body'.

Declaration

```
public Type LayoutType { get; }
```

Property Value

TYPE	DESCRIPTION
System.Type	

Class LayoutDisplay

Displays the specified page component, rendering it inside its layout and any further nested layouts.

Inheritance

System.Object

LayoutDisplay

Implements

[IComponent](#)

Namespace: **Microsoft.AspNetCore.Blazor.Layouts**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class LayoutDisplay : Object, IComponent
```

Constructors

LayoutDisplay()

Declaration

```
public LayoutDisplay()
```

Methods

Init(RenderHandle)

Initializes the component.

Declaration

```
public void Init(RenderHandle renderHandle)
```

Parameters

TYPE	NAME	DESCRIPTION
[RenderHandle](#)	renderHandle	A [RenderHandle](#) that allows the component to be rendered.

SetParameters(ParameterCollection)

Sets parameters supplied by the component's parent in the render tree.

Declaration

```
public void SetParameters(ParameterCollection parameters)
```

Parameters

TYPE	NAME	DESCRIPTION
[ParameterCollection](#)	parameters	The parameters.

Implements

[IComponent](#)

Namespace Microsoft.AspNetCore.Blazor.Rendering

Classes

RenderBatch

Describes a set of UI changes.

Renderer

Provides mechanisms for rendering hierarchies of IComponent instances, dispatching events to them, and notifying when the user interface is being updated.

RendererRegistry

Framework infrastructure, not intended to be used by application code.

Class RenderBatch

Describes a set of UI changes.

Inheritance

System.Object

RenderBatch

Namespace: **Microsoft.AspNetCore.Blazor.Rendering**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class RenderBatch : ValueType
```

Properties

DisposedComponentIDs

Gets the IDs of the components that were disposed.

Declaration

```
public ArrayRange<int> DisposedComponentIDs { get; }
```

Property Value

TYPE	DESCRIPTION
ArrayRange<System.Int32>	

DisposedEventHandlerIDs

Gets the IDs of the event handlers that were disposed.

Declaration

```
public ArrayRange<int> DisposedEventHandlerIDs { get; }
```

Property Value

TYPE	DESCRIPTION
ArrayRange<System.Int32>	

ReferenceFrames

Gets render frames that may be referenced by entries in UpdatedComponents. For example, edit entries of type PrependFrame will point to an entry in this array to specify the subtree to be prepended.

Declaration

```
public ArrayRange<RenderTreeFrame> ReferenceFrames { get; }
```

Property Value

TYPE	DESCRIPTION
ArrayRange<RenderTreeFrame>	

UpdatedComponents

Gets the changes to components that were added or updated.

Declaration

```
public ArrayRange<RenderTreeDiff> UpdatedComponents { get; }
```

Property Value

TYPE	DESCRIPTION
ArrayRange<RenderTreeDiff>	

Class Renderer

Provides mechanisms for rendering hierarchies of IComponent instances, dispatching events to them, and notifying when the user interface is being updated.

Inheritance

System.Object

Renderer

BrowserRenderer

Namespace: Microsoft.AspNetCore.Blazor.Rendering

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public abstract class Renderer : Object
```

Constructors

Renderer(IServiceProvider)

Constructs an instance of Renderer.

Declaration

```
public Renderer(IServiceProvider serviceProvider)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IServiceProvider	serviceProvider	The System.IServiceProvider to be used when initialising components.

Methods

AssignComponentId(IComponent)

Associates the IComponent with the Renderer, assigning an identifier that is unique within the scope of the Renderer.

Declaration

```
protected int AssignComponentId(IComponent component)
```

Parameters

TYPE	NAME	DESCRIPTION
IComponent	component	The component.

Returns

TYPE	DESCRIPTION
System.Int32	The component's assigned identifier.

DispatchEvent(Int32, Int32, UIEventArgs)

Notifies the specified component that an event has occurred.

Declaration

```
public void DispatchEvent(int componentId, int eventHandlerId, UIEventArgs eventArgs)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	componentId	The unique identifier for the component within the scope of this Renderer.
System.Int32	eventHandlerId	The AttributeEventHandlerId value from the original event attribute.
UIEventArgs	eventArgs	Arguments to be passed to the event handler.

InstantiateComponent(Type)

Constructs a new component of the specified type.

Declaration

```
protected IComponent InstantiateComponent(Type componentType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	componentType	The type of the component to instantiate.

Returns

TYPE	DESCRIPTION
IComponent	The component instance.

UpdateDisplay(RenderBatch)

Updates the visible UI.

Declaration

```
protected abstract void UpdateDisplay(in RenderBatch renderBatch)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderBatch	renderBatch	The changes to the UI since the previous call.

Class RendererRegistry

Framework infrastructure, not intended to be used by application code.

Inheritance

System.Object

RendererRegistry

Namespace: Microsoft.AspNetCore.Blazor.Rendering

Assembly: Microsoft.AspNetCore.Blazor.Browser.dll

Syntax

```
public class RendererRegistry : Object
```

Constructors

RendererRegistry()

Declaration

```
public RendererRegistry()
```

Properties

Current

Framework infrastructure, not intended to be used by application code.

Declaration

```
public static RendererRegistry Current { get; }
```

Property Value

TYPE	DESCRIPTION
RendererRegistry	

Methods

Add(Renderer)

Framework infrastructure, not intended by used by application code.

Declaration

```
public int Add(Renderer renderer)
```

Parameters

TYPE	NAME	DESCRIPTION
Renderer	renderer	

Returns

TYPE	DESCRIPTION
System.Int32	

Find(Int32)

Framework infrastructure, not intended by used by application code.

Declaration

```
public Renderer Find(int browserRendererId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	browserRendererId	

Returns

TYPE	DESCRIPTION
Renderer	

SetCurrentRendererRegistry(RendererRegistry)

Framework infrastructure, not intended by used by application code.

Declaration

```
public static void SetCurrentRendererRegistry(RendererRegistry registry)
```

Parameters

TYPE	NAME	DESCRIPTION
RendererRegistry	registry	

TryRemove(Int32)

Framework infrastructure, not intended by used by application code.

Declaration

```
public bool TryRemove(int browserRendererId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	browserRendererId	

Returns

TYPE	DESCRIPTION
System.Boolean	

Namespace Microsoft.AspNetCore.Blazor.RenderTree

Classes

ArrayRange<T>

Represents a range of elements in an array that are in use.

RenderTreeBuilder

Provides methods for building a collection of RenderTreeFrame entries.

RenderTreeDiff

Describes changes to a component's render tree between successive renders.

RenderTreeEdit

Represents a single edit operation on a component's render tree.

RenderTreeEditType

Describes the type of a render tree edit operation.

RenderTreeFrame

Represents an entry in a tree of user interface (UI) items.

RenderTreeFrameType

Describes the type of a RenderTreeFrame.

Class ArrayRange<T>

Represents a range of elements in an array that are in use.

Inheritance

System.Object

ArrayRange<T>

Implements

System.Collections.IEnumerable

System.Collections.Generic.IEnumerable<T>

Namespace: Microsoft.AspNetCore.Blazor.RenderTree

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class ArrayRange<T> : ValueType, IEnumerable, IEnumerable<T>
```

Type Parameters

NAME	DESCRIPTION
T	The array item type.

Constructors

ArrayRange(T[], Int32)

Constructs an instance of ArrayRange<T>.

Declaration

```
public ArrayRange(T[] array, int count)
```

Parameters

TYPE	NAME	DESCRIPTION
T[]	array	The array.
System.Int32	count	The number of items in the array that are in use.

Fields

Array

Gets the underlying array instance.

Declaration

```
public readonly T[] Array
```

Field Value

TYPE	DESCRIPTION
T[]	

Count

Gets the number of items in the array that are considered to be in use.

Declaration

```
public readonly int Count
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Implements

System.Collections.IEnumerable

System.Collections.Generic.IEnumerable<>

Class RenderTreeBuilder

Provides methods for building a collection of RenderTreeFrame entries.

Inheritance

System.Object

RenderTreeBuilder

Namespace: **Microsoft.AspNetCore.Blazor.RenderTree**

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class RenderTreeBuilder : Object
```

Constructors

RenderTreeBuilder(Renderer)

Constructs an instance of RenderTreeBuilder.

Declaration

```
public RenderTreeBuilder(Renderer renderer)
```

Parameters

TYPE	NAME	DESCRIPTION
Renderer	renderer	The associated Renderer.

Fields

ChildContent

The reserved parameter name used for supplying child content.

Declaration

```
public const string ChildContent = "ChildContent"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

AddAttribute(Int32, RenderTreeFrame)

Appends a frame representing an attribute.

The attribute is associated with the most recently added element.

Declaration

```
public void AddAttribute(int sequence, in RenderTreeFrame frame)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
RenderTreeFrame	frame	A RenderTreeFrame holding the name and value of the attribute.

AddAttribute(Int32, String, Action)

Appends a frame representing an System.Action-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public void AddAttribute(int sequence, string name, Action value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action	value	The value of the attribute.

AddAttribute(Int32, String, Action<UIEventArgs>)

Appends a frame representing an System.Action<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public void AddAttribute(int sequence, string name, Action<UIEventArgs> value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Action<UIEventArgs>	value	The value of the attribute.

AddAttribute(Int32, String, Boolean)

Appends a frame representing a bool-valued attribute.

The attribute is associated with the most recently added element. If the value is `false` and the current element is not a component, the frame will be omitted.

Declaration

```
public void AddAttribute(int sequence, string name, bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Boolean	value	The value of the attribute.

AddAttribute(Int32, String, Func<UIEventArgs, Task>)

Appends a frame representing a System.Func<, >-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public void AddAttribute(int sequence, string name, Func<UIEventArgs, Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<UIEventArgs, System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(Int32, String, Func<Task>)

Appends a frame representing a System.Func<>-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public void AddAttribute(int sequence, string name, Func<Task> value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Func<System.Threading.Tasks.Task>	value	The value of the attribute.

AddAttribute(Int32, String, MulticastDelegate)

Appends a frame representing a delegate-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public void AddAttribute(int sequence, string name, MulticastDelegate value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.MulticastDelegate	value	The value of the attribute.

Remarks

This method is provided for infrastructure purposes, and is used to be UIEventArgsRenderTreeBuilderExtensions to provide support for delegates of specific types. For a good programming experience when using a custom delegate type, define an extension method similar to AddAttribute(RenderTreeBuilder, Int32, String, Action<UIChangeEventArgs>) that calls this method.

AddAttribute(Int32, String, Object)

Appends a frame representing a string-valued attribute. The attribute is associated with the most recently added element. If the value is `null`, or the System.Boolean value `false` and the current element is not a component, the frame will be omitted.

Declaration

```
public void AddAttribute(int sequence, string name, object value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.Object	value	The value of the attribute.

AddAttribute(Int32, String, String)

Appends a frame representing a string-valued attribute.

The attribute is associated with the most recently added element. If the value is `null` and the current element is not a component, the frame will be omitted.

Declaration

```
public void AddAttribute(int sequence, string name, string value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	name	The name of the attribute.
System.String	value	The value of the attribute.

AddComponentReferenceCapture(Int32, Action<Object>)

Appends a frame representing an instruction to capture a reference to the parent component.

Declaration

```
public void AddComponentReferenceCapture(int sequence, Action<object> componentReferenceCaptureAction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.Action<System.Object>	componentReferenceCaptureAction	An action to be invoked whenever the reference value changes.

AddContent(Int32, MarkupString)

Appends a frame representing markup content.

Declaration

```
public void AddContent(int sequence, MarkupString markupContent)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
MarkupString	markupContent	Content for the new markup frame.

AddContent(Int32, RenderFragment)

Appends frames representing an arbitrary fragment of content.

Declaration

```
public void AddContent(int sequence, RenderFragment fragment)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
RenderFragment	fragment	Content to append.

AddContent(Int32, Object)

Appends a frame representing text content.

Declaration

```
public void AddContent(int sequence, object textContent)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.Object	textContent	Content for the new text frame.

AddContent(Int32, String)

Appends a frame representing text content.

Declaration

```
public void AddContent(int sequence, string textContent)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	textContent	Content for the new text frame.

AddElementReferenceCapture(Int32, Action<ElementRef>)

Appends a frame representing an instruction to capture a reference to the parent element.

Declaration

```
public void AddElementReferenceCapture(int sequence, Action<ElementRef> elementReferenceCaptureAction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.Action<ElementRef>	elementReferenceCaptureAction	An action to be invoked whenever the reference value changes.

AddMarkupContent(Int32, String)

Appends a frame representing markup content.

Declaration

```
public void AddMarkupContent(int sequence, string markupContent)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	markupContent	Content for the new markup frame.

Clear()

Clears the builder.

Declaration

```
public void Clear()
```

CloseComponent()

Marks a previously appended component frame as closed. Calls to this method must be balanced with calls to

OpenComponent<TComponent>(Int32).

Declaration

```
public void CloseComponent()
```

CloseElement()

Marks a previously appended element frame as closed. Calls to this method must be balanced with calls to OpenElement(Int32, String).

Declaration

```
public void CloseElement()
```

GetFrames()

Returns the RenderTreeFrame values that have been appended.

Declaration

```
public ArrayRange<RenderTreeFrame> GetFrames()
```

Returns

TYPE	DESCRIPTION
ArrayRange<RenderTreeFrame>	An array range of RenderTreeFrame values.

OpenComponent(Int32, Type)

Appends a frame representing a child component.

Declaration

```
public void OpenComponent(int sequence, Type componentType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.Type	componentType	The type of the child component.

OpenComponent<TComponent>(Int32)

Appends a frame representing a child component.

Declaration

```
public void OpenComponent<TComponent>(int sequence)
    where TComponent : IComponent
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.

Type Parameters

NAME	DESCRIPTION
TComponent	The type of the child component.

OpenElement(Int32, String)

Appends a frame representing an element, i.e., a container for other frames. In order for the RenderTreeBuilder state to be valid, you must also call CloseElement() immediately after appending the new element's child frames.

Declaration

```
public void OpenElement(int sequence, string elementName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sequence	An integer that represents the position of the instruction in the source code.
System.String	elementName	A value representing the type of the element.

Extension Methods

UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIChangeEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIChangeEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIDragEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIDragEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIClipboardEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIClipboardEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIErrorEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIErrorEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIFocusEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIFocusEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIKeyboardEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIKeyboardEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIMouseEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIMouseEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIPointerEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIPointerEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIProgressEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIProgressEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UITouchEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UITouchEventArgs, Task>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Action<UIWheelEventArgs>)
UIEventArgsRenderTreeBuilderExtensions.AddAttribute(RenderTreeBuilder, Int32, String, Func<UIWheelEventArgs, Task>)

Class RenderTreeDiff

Describes changes to a component's render tree between successive renders.

Inheritance

System.Object

RenderTreeDiff

Namespace: Microsoft.AspNetCore.Blazor.RenderTree

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class RenderTreeDiff : ValueType
```

Fields

ComponentId

Gets the ID of the component.

Declaration

```
public readonly int ComponentId
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Edits

Gets the changes to the render tree since a previous state.

Declaration

```
public readonly ArraySegment<RenderTreeEdit> Edits
```

Field Value

TYPE	DESCRIPTION
System.ArraySegment<RenderTreeEdit>	

Class RenderTreeEdit

Represents a single edit operation on a component's render tree.

Inheritance

System.Object

RenderTreeEdit

Namespace: Microsoft.AspNetCore.Blazor.RenderTree

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class RenderTreeEdit : ValueType
```

Fields

ReferenceFrameIndex

Gets the index of related data in an associated render frames array. For example, if the Type value is PrependFrame, gets the index of the new frame data in an associated render tree.

Declaration

```
public readonly int ReferenceFrameIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

RemovedAttributeName

If the Type value is RemoveAttribute, gets the name of the attribute that is being removed.

Declaration

```
public readonly string RemovedAttributeName
```

Field Value

TYPE	DESCRIPTION
System.String	

SiblingIndex

Gets the index of the sibling frame that the edit relates to.

Declaration

```
public readonly int SiblingIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Type

Gets the type of the edit operation.

Declaration

```
public readonly RenderTreeEditType Type
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

Class RenderTreeEditType

Describes the type of a render tree edit operation.

Inheritance

System.Object

RenderTreeEditType

Namespace: `Microsoft.AspNetCore.Blazor.RenderTree`

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class RenderTreeEditType : Enum
```

Fields

PrependFrame

Indicates that a new frame should be inserted before the specified tree frame.

Declaration

```
public const RenderTreeEditType PrependFrame
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

RemoveAttribute

Indicates that a named attribute should be removed from the specified frame.

Declaration

```
public const RenderTreeEditType RemoveAttribute
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

RemoveFrame

Indicates that the specified tree frame should be removed.

Declaration

```
public const RenderTreeEditType RemoveFrame
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

SetAttribute

Indicates that an attribute value should be applied to the specified frame. This may be a change to an existing attribute, or the

addition of a new attribute.

Declaration

```
public const RenderTreeEditType SetAttribute
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

StepIn

Indicates that the edit position should move inside the specified frame.

Declaration

```
public const RenderTreeEditType StepIn
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

StepOut

Indicates that there are no further edit operations on the current frame, and the edit position should move back to the parent frame.

Declaration

```
public const RenderTreeEditType StepOut
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

UpdateMarkup

Indicates that the markup content of the specified frame (which must be a markup frame) should be updated.

Declaration

```
public const RenderTreeEditType UpdateMarkup
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

UpdateText

Indicates that the text content of the specified frame (which must be a text frame) should be updated.

Declaration

```
public const RenderTreeEditType UpdateText
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

value__

Declaration

```
public int value__
```

Field Value

TYPE	DESCRIPTION
System.Int32	

```
public const RenderTreeEditType UpdateText
```

Field Value

TYPE	DESCRIPTION
RenderTreeEditType	

Class RenderTreeFrame

Represents an entry in a tree of user interface (UI) items.

Inheritance

System.Object

RenderTreeFrame

Namespace: Microsoft.AspNetCore.Blazor.RenderTree

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class RenderTreeFrame : ValueType
```

Fields

AttributeEventHandlerId

If the FrameType property equals Attribute gets the ID of the corresponding event handler, if any.

Declaration

```
public readonly int AttributeEventHandlerId
```

Field Value

TYPE	DESCRIPTION
System.Int32	

AttributeName

If the FrameType property equals Attribute, gets the attribute name. Otherwise, the value is undefined.

Declaration

```
public readonly string AttributeName
```

Field Value

TYPE	DESCRIPTION
System.String	

AttributeValue

If the FrameType property equals Attribute, gets the attribute value. Otherwise, the value is undefined.

Declaration

```
public readonly object AttributeValue
```

Field Value

TYPE	DESCRIPTION
System.Object	

Component

If the FrameType property equals Component, gets the child component instance. Otherwise, the value is undefined.

Declaration

```
public readonly IComponent Component
```

Field Value

TYPE	DESCRIPTION
IComponent	

ComponentId

If the FrameType property equals Component, gets the child component instance identifier.

Declaration

```
public readonly int ComponentId
```

Field Value

TYPE	DESCRIPTION
System.Int32	

ComponentReferenceCaptureAction

If the FrameType property equals ComponentReferenceCapture, gets the action that writes the reference to its target. Otherwise, the value is undefined.

Declaration

```
public readonly Action<object> ComponentReferenceCaptureAction
```

Field Value

TYPE	DESCRIPTION
System.Action<System.Object>	

ComponentReferenceCaptureParentFrameIndex

If the FrameType property equals ComponentReferenceCapture, gets the index of the parent frame representing the component being captured. Otherwise, the value is undefined. WARNING: This index can only be used in the context of the frame's original render tree. If the frame is copied elsewhere, such as to the ReferenceFrames buffer of a RenderTreeDiff, then the index will not relate to entries in that other buffer. Currently there's no scenario where this matters, but if there was, we could change all of the subtree initialization logic in RenderTreeDiffBuilder to walk the frames hierarchically, then it would know the parent index at the point where it wants to initialize the ComponentReferenceCapture frame.

Declaration

```
public readonly int ComponentReferenceCaptureParentFrameIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

ComponentSubtreeLength

If the FrameType property equals Component gets the number of frames in the subtree for which this frame is the root. The value is zero if the frame has not yet been closed.

Declaration

```
public readonly int ComponentSubtreeLength
```

Field Value

TYPE	DESCRIPTION
System.Int32	

ComponentType

If the FrameType property equals Component, gets the type of the child component.

Declaration

```
public readonly Type ComponentType
```

Field Value

TYPE	DESCRIPTION
System.Type	

ElementName

If the FrameType property equals Element, gets a name representing the type of the element. Otherwise, the value is undefined.

Declaration

```
public readonly string ElementName
```

Field Value

TYPE	DESCRIPTION
System.String	

ElementReferenceCaptureAction

If the FrameType property equals ElementReferenceCapture, gets the action that writes the reference to its target. Otherwise, the value is undefined.

Declaration

```
public readonly Action<ElementRef> ElementReferenceCaptureAction
```

Field Value

TYPE	DESCRIPTION
System.Action<ElementRef>	

ElementReferenceCaptureId

If the FrameType property equals ElementReferenceCapture, gets the ID of the reference capture. Otherwise, the value is undefined.

Declaration

```
public readonly string ElementReferenceCaptureId
```

Field Value

TYPE	DESCRIPTION
System.String	

ElementSubtreeLength

If the FrameType property equals Element gets the number of frames in the subtree for which this frame is the root. The value is zero if the frame has not yet been closed.

Declaration

```
public readonly int ElementSubtreeLength
```

Field Value

TYPE	DESCRIPTION
System.Int32	

FrameType

Describes the type of this frame.

Declaration

```
public readonly RenderTreeFrameType FrameType
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

MarkupContent

If the FrameType property equals Markup, gets the content of the markup frame. Otherwise, the value is undefined.

Declaration

```
public readonly string MarkupContent
```

Field Value

TYPE	DESCRIPTION
System.String	

RegionSubtreeLength

If the FrameType property equals Region gets the number of frames in the subtree for which this frame is the root. The value is zero if the frame has not yet been closed.

Declaration

```
public readonly int RegionSubtreeLength
```

TYPE	DESCRIPTION
System.Int32	

Sequence

Gets the sequence number of the frame. Sequence numbers indicate the relative source positions of the instructions that inserted the frames. Sequence numbers are only comparable within the same sequence (typically, the same source method).

Declaration

```
public readonly int Sequence
```

Field Value

TYPE	DESCRIPTION
System.Int32	

TextContent

If the FrameType property equals Text, gets the content of the text frame. Otherwise, the value is undefined.

Declaration

```
public readonly string TextContent
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

ToString()

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
System.String	

Class RenderTreeFrameType

Describes the type of a RenderTreeFrame.

Inheritance

System.Object

RenderTreeFrameType

Namespace: Microsoft.AspNetCore.Blazor.RenderTree

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class RenderTreeFrameType : Enum
```

Fields

Attribute

Represents a key-value pair associated with another RenderTreeFrame.

Declaration

```
public const RenderTreeFrameType Attribute
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

Component

Represents a child component.

Declaration

```
public const RenderTreeFrameType Component
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

ComponentReferenceCapture

Represents an instruction to capture or update a reference to the parent component.

Declaration

```
public const RenderTreeFrameType ComponentReferenceCapture
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

Element

Represents a container for other frames.

Declaration

```
public const RenderTreeFrameType Element
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

ElementReferenceCapture

Represents an instruction to capture or update a reference to the parent element.

Declaration

```
public const RenderTreeFrameType ElementReferenceCapture
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

Markup

Represents a block of markup content.

Declaration

```
public const RenderTreeFrameType Markup
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

Region

Defines the boundary around range of sibling frames that should be treated as an unsplittable group for the purposes of diffing. This is typically used when appending a tree fragment generated by external code, because the sequence numbers in that tree fragment are not comparable to sequence numbers outside it.

Declaration

```
public const RenderTreeFrameType Region
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

Text

Represents text content.

Declaration

```
public const RenderTreeFrameType Text
```

Field Value

TYPE	DESCRIPTION
RenderTreeFrameType	

value__

Declaration

```
public int value__
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Namespace Microsoft.AspNetCore.Blazor.Routing

Classes

NavLink

A component that renders an anchor tag, automatically toggling its 'active' class based on whether its 'href' matches the current URI.

NavLinkMatch

Modifies the URL matching behavior for a NavLink.

Router

A component that displays whichever other component corresponds to the current navigation location.

Class NavLink

A component that renders an anchor tag, automatically toggling its 'active' class based on whether its 'href' matches the current URI.

Inheritance

System.Object

NavLink

Implements

IComponent

System.IDisposable

Namespace: Microsoft.AspNetCore.Blazor.Routing

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class NavLink : Object, IComponent, IDisposable
```

Constructors

NavLink()

Declaration

```
public NavLink()
```

Methods

Dispose()

Declaration

```
public void Dispose()
```

Init(RenderHandle)

Initializes the component.

Declaration

```
public void Init(RenderHandle renderHandle)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderHandle	renderHandle	A RenderHandle that allows the component to be rendered.

SetParameters(ParameterCollection)

Sets parameters supplied by the component's parent in the render tree.

Declaration

```
public void SetParameters(ParameterCollection parameters)
```

Parameters

TYPE	NAME	DESCRIPTION
ParameterCollection	parameters	The parameters.

Implements

IComponent

System.IDisposable

Class NavLinkMatch

Modifies the URL matching behavior for a NavLink.

Inheritance

System.Object

NavLinkMatch

Namespace: Microsoft.AspNetCore.Blazor.Routing

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public sealed class NavLinkMatch : Enum
```

Fields

All

Specifies that the NavLink should be active when it matches the entire current URL.

Declaration

```
public const NavLinkMatch All
```

Field Value

TYPE	DESCRIPTION
NavLinkMatch	

Prefix

Specifies that the NavLink should be active when it matches any prefix of the current URL.

Declaration

```
public const NavLinkMatch Prefix
```

Field Value

TYPE	DESCRIPTION
NavLinkMatch	

value__

Declaration

```
public int value__
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class Router

A component that displays whichever other component corresponds to the current navigation location.

Inheritance

System.Object

Router

Implements

IComponent

System.IDisposable

Namespace: Microsoft.AspNetCore.Blazor.Routing

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public class Router : Object, IComponent, IDisposable
```

Constructors

Router()

Declaration

```
public Router()
```

Methods

Dispose()

Declaration

```
public void Dispose()
```

Init(RenderHandle)

Initializes the component.

Declaration

```
public void Init(RenderHandle renderHandle)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderHandle	renderHandle	A RenderHandle that allows the component to be rendered.

Render(RenderTreeBuilder, Type, IDictionary<String, Object>)

Declaration

```
protected virtual void Render(RenderTreeBuilder builder, Type handler, IDictionary<string, object> parameters)
```

Parameters

TYPE	NAME	DESCRIPTION
RenderTreeBuilder	builder	

TYPE	NAME	DESCRIPTION
System.Type	handler	
System.Collections.Generic.IDictionary<System.String, System.Object>	parameters	

SetParameters(ParameterCollection)

Sets parameters supplied by the component's parent in the render tree.

Declaration

```
public void SetParameters(ParameterCollection parameters)
```

Parameters

TYPE	NAME	DESCRIPTION
ParameterCollection	parameters	The parameters.

Implements

[IComponent](#)
System.IDisposable

Namespace Microsoft.AspNetCore.Blazor.Server

Classes

WasmMediaTypeNames

Contains values for WASM-related media types.

WasmMediaTypeNames.Application

Contains values for WASM-related media types within the "application." namespace

Class WasmMediaTypeNames

Contains values for WASM-related media types.

Inheritance

System.Object

WasmMediaTypeNames

Namespace: Microsoft.AspNetCore.Blazor.Server

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public static class WasmMediaTypeNames : Object
```

Class WasmMediaTypeNames.Application

Contains values for WASM-related media types within the "application." namespace

Inheritance

System.Object

WasmMediaTypeNames.Application

Namespace: Microsoft.AspNetCore.Blazor.Server

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public static class Application : Object
```

Fields

Wasm

The standard media type name for WebAssembly binary files.

Declaration

```
public const string Wasm = "application/wasm"
```

Field Value

TYPE	DESCRIPTION
System.String	

Namespace Microsoft.AspNetCore.Blazor.Server.Circuits

Classes

Circuit

Represents an active connection between a Blazor server and a client.

RemoteUriHelper

A Server-Side Blazor implemenation of IUriHelper.

Class Circuit

Represents an active connection between a Blazor server and a client.

Inheritance

System.Object

Circuit

Namespace: Microsoft.AspNetCore.Blazor.Server.Circuits

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public class Circuit : Object
```

Properties

Current

Gets the current Circuit.

Declaration

```
public static Circuit Current { get; }
```

Property Value

TYPE	DESCRIPTION
Circuit	

JSRuntime

Gets the IJSRuntime associated with this circuit.

Declaration

```
public IJSRuntime JSRuntime { get; }
```

Property Value

TYPE	DESCRIPTION
IJSRuntime	

Services

Gets the System.IServiceProvider associated with this circuit.

Declaration

```
public IServiceProvider Services { get; }
```

Property Value

TYPE	DESCRIPTION
System.IServiceProvider	

Class RemoteUriHelper

A Server-Side Blazor implemenation of IUriHelper.

Inheritance

System.Object

UriHelperBase

RemoteUriHelper

Implements

IUriHelper

Inherited Members

UriHelperBase.NavigateTo(String)

UriHelperBase.InitializeState()

UriHelperBase.GetAbsoluteUri()

UriHelperBase.GetBaseUri()

UriHelperBase.ToAbsoluteUri(String)

UriHelperBase.ToBaseRelativePath(String, String)

UriHelperBase.SetAbsoluteUri(String)

UriHelperBase.SetAbsoluteBaseUri(String)

UriHelperBase.TriggerOnLocationChanged()

UriHelperBase.OnLocationChanged

Namespace: Microsoft.AspNetCore.Blazor.Server.Circuits

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public class RemoteUriHelper : UriHelperBase, IUriHelper
```

Constructors

RemoteUriHelper(IJSRuntime)

Creates a new RemoteUriHelper.

Declaration

```
public RemoteUriHelper(IJSRuntime jsRuntime)
```

Parameters

TYPE	NAME	DESCRIPTION
IJSRuntime	jsRuntime	

Methods

Initialize(String, String)

Initializes the RemoteUriHelper.

Declaration

```
public void Initialize(string uriAbsolute, string baseUriAbsolute)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uriAbsolute	The absolute URI of the current page.
System.String	baseUriAbsolute	The absolute base URI of the current page.

NavigateToCore(String)

Navigates to the specified URI.

Declaration

```
protected override void NavigateToCore(string uri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uri	The destination URI. This can be absolute, or relative to the base URI (as returned by GetBaseUri()).

Overrides

UriHelperBase.NavigateToCore(String)

NotifyLocationChanged(String)

For framework use only.

Declaration

```
[JSInvokable("NotifyLocationChanged")]
public static void NotifyLocationChanged(string uriAbsolute)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uriAbsolute	

Implements

IUriHelper

Namespace Microsoft.AspNetCore.Blazor.Services

Classes

UriHelperBase

A base class for IUriHelper implementations.

Interfaces

IUriHelper

Helpers for working with URIs and navigation state.

Interface IUriHelper

Helpers for working with URIs and navigation state.

Namespace: Microsoft.AspNetCore.Blazor.Services

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public interface IUriHelper
```

Methods

GetAbsoluteUri()

Gets the current absolute URI.

Declaration

```
string GetAbsoluteUri()
```

Returns

TYPE	DESCRIPTION
System.String	The current absolute URI.

GetBaseUri()

Gets the base URI (with trailing slash) that can be prepended before relative URI paths to produce an absolute URI. Typically this corresponds to the 'href' attribute on the document's <base> element.

Declaration

```
string GetBaseUri()
```

Returns

TYPE	DESCRIPTION
System.String	The URI prefix, which has a trailing slash.

NavigateTo(String)

Navigates to the specified URI.

Declaration

```
void NavigateTo(string uri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uri	The destination URI. This can be absolute, or relative to the base URI (as returned by GetBaseUri()).

ToAbsoluteUri(String)

Converts a relative URI into an absolute one (by resolving it relative to the current absolute URI).

Declaration

```
Uri ToAbsoluteUri(string href)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	href	The relative URI.

Returns

TYPE	DESCRIPTION
System.Uri	The absolute URI.

ToBaseRelativePath(String, String)

Given a base URI (e.g., one previously returned by GetBaseUri()), converts an absolute URI into one relative to the base URI prefix.

Declaration

```
string ToBaseRelativePath(string baseUri, string locationAbsolute)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	baseUri	The base URI prefix (e.g., previously returned by GetBaseUri()).
System.String	locationAbsolute	An absolute URI that is within the space of the base URI.

Returns

TYPE	DESCRIPTION
System.String	A relative URI path.

Events

OnLocationChanged

An event that fires when the navigation location has changed.

Declaration

```
event EventHandler<string> OnLocationChanged
```

Event Type

TYPE	DESCRIPTION
System.EventHandler<System.String>	

System.EventHandler<System.String>

Class UriHelperBase

A base class for IUriHelper implementations.

Inheritance

System.Object

UriHelperBase

BrowserUriHelper

RemoteUriHelper

Implements

IUriHelper

Namespace: Microsoft.AspNetCore.Blazor.Services

Assembly: Microsoft.AspNetCore.Blazor.dll

Syntax

```
public abstract class UriHelperBase : Object, IUriHelper
```

Constructors

UriHelperBase()

Declaration

```
protected UriHelperBase()
```

Methods

GetAbsoluteUri()

Gets the current absolute URI.

Declaration

```
public string GetAbsoluteUri()
```

Returns

TYPE	DESCRIPTION
System.String	The current absolute URI.

GetBaseUri()

Gets the base URI (with trailing slash) that can be prepended before relative URI paths to produce an absolute URI. Typically this corresponds to the 'href' attribute on the document's <base> element.

Declaration

```
public virtual string GetBaseUri()
```

Returns

TYPE	DESCRIPTION
System.String	The URI prefix, which has a trailing slash.

InitializeState()

Called to initialize BaseURI and current URI before those values the first time. Override this method to dynamically calculate the those values.

Declaration

```
protected virtual void InitializeState()
```

NavigateTo(String)

Navigates to the specified URI.

Declaration

```
public void NavigateTo(string uri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uri	The destination URI. This can be absolute, or relative to the base URI (as returned by GetBaseUri()).

NavigateToCore(String)

Navigates to the specified URI.

Declaration

```
protected abstract void NavigateToCore(string uri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uri	The destination URI. This can be absolute, or relative to the base URI (as returned by GetBaseUri()).

SetAbsoluteBaseUri(String)

Sets the base URI to the provided value (after normalization).

Declaration

```
protected void SetAbsoluteBaseUri(string baseUri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	baseUri	The base URI. Must be an absolute URI.

Remarks

Calling SetAbsoluteBaseUri(String) does not trigger OnLocationChanged.

SetAbsoluteUri(String)

Set the URI to the provided value.

Declaration

```
protected void SetAbsoluteUri(string uri)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uri	The URI. Must be an absolute URI.

Remarks

Calling SetAbsoluteUri(String) does not trigger OnLocationChanged.

ToAbsoluteUri(String)

Converts a relative URI into an absolute one (by resolving it relative to the current absolute URI).

Declaration

```
public Uri ToAbsoluteUri(string href)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	href	The relative URI.

Returns

TYPE	DESCRIPTION
System.Uri	The absolute URI.

ToBaseRelativePath(String, String)

Given a base URI (e.g., one previously returned by GetBaseUri()), converts an absolute URI into one relative to the base URI prefix.

Declaration

```
public string ToBaseRelativePath(string baseUri, string locationAbsolute)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	baseUri	The base URI prefix (e.g., previously returned by GetBaseUri()).
System.String	locationAbsolute	An absolute URI that is within the space of the base URI.

Returns

TYPE	DESCRIPTION
System.String	A relative URI path.

TriggerOnLocationChanged()

Triggers the OnLocationChanged event with the current URI value.

Declaration

```
protected void TriggerOnLocationChanged()
```

Events

OnLocationChanged

An event that fires when the navigation location has changed.

Declaration

```
public event EventHandler<string> OnLocationChanged
```

Event Type

TYPE	DESCRIPTION
System.EventHandler<System.String>	

Implements

IUriHelper

Namespace Microsoft.AspNetCore.Builder

Classes

BlazorApplicationBuilderExtensions

Provides extension methods that add Blazor-related middleware to the ASP.NET pipeline.

BlazorOptions

Provides configuration options to the UseBlazor(IApplicationBuilder, BlazorOptions) middleware.

ServerSideBlazorApplicationBuilderExtensions

Extension methods to configure an Microsoft.AspNetCore.Builder.IApplicationBuilder for Server-Side Blazor.

Class BlazorApplicationBuilderExtensions

Provides extension methods that add Blazor-related middleware to the ASP.NET pipeline.

Inheritance

System.Object

BlazorApplicationBuilderExtensions

Namespace: Microsoft.AspNetCore.Builder

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public static class BlazorApplicationBuilderExtensions : Object
```

Methods

UseBlazor(IApplicationBuilder, BlazorOptions)

Configures the middleware pipeline to work with Blazor.

Declaration

```
public static void UseBlazor(this IApplicationBuilder app, BlazorOptions options)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.AspNetCore.Builder.IApplicationBuilder	app	
BlazorOptions	options	

UseBlazor<TProgram>(IApplicationBuilder)

Configures the middleware pipeline to work with Blazor.

Declaration

```
public static void UseBlazor<TProgram>(this IApplicationBuilder app)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.AspNetCore.Builder.IApplicationBuilder	app	

Type Parameters

NAME	DESCRIPTION
TProgram	Any type from the client app project. This is used to identify the client app assembly.

Class BlazorOptions

Provides configuration options to the UseBlazor(IApplicationBuilder, BlazorOptions) middleware.

Inheritance

System.Object
BlazorOptions

Namespace: Microsoft.AspNetCore.Builder

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public class BlazorOptions : Object
```

Constructors

BlazorOptions()

Declaration

```
public BlazorOptions()
```

Properties

ClientAssemblyPath

Full path to the client assembly.

Declaration

```
public string ClientAssemblyPath { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class ServerSideBlazorApplicationBuilderExtensions

Extension methods to configure an Microsoft.AspNetCore.Builder.IApplicationBuilder for Server-Side Blazor.

Inheritance

System.Object

ServerSideBlazorApplicationBuilderExtensions

Namespace: **Microsoft.AspNetCore.Builder**

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public static class ServerSideBlazorApplicationBuilderExtensions : Object
```

Methods

UseServerSideBlazor(IApplicationBuilder, BlazorOptions, Action<IBlazorApplicationBuilder>)

Registers middleware for Server-Side Blazor.

Declaration

```
public static IApplicationBuilder UseServerSideBlazor(this IApplicationBuilder builder, BlazorOptions options,
Action<IBlazorApplicationBuilder> startupAction)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.AspNetCore.Builder.IApplicationBuilder	builder	The Microsoft.AspNetCore.Builder.IApplicationBuilder.
BlazorOptions	options	A BlazorOptions instance used to configure the Blazor file provider.
System.Action<IBlazorApplicationBuilder>	startupAction	A delegate used to configure the renderer.

Returns

TYPE	DESCRIPTION
Microsoft.AspNetCore.Builder.IApplicationBuilder	The Microsoft.AspNetCore.Builder.IApplicationBuilder.

UseServerSideBlazor(IApplicationBuilder, Type)

Registers Server-Side Blazor in the pipeline.

Declaration

```
public static IApplicationBuilder UseServerSideBlazor(this IApplicationBuilder builder, Type startupType)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
Microsoft.AspNetCore.Builder.IApplicationBuilder	builder	The Microsoft.AspNetCore.Builder.IApplicationBuilder.
System.Type	startupType	A Blazor startup type.

Returns

TYPE	DESCRIPTION
Microsoft.AspNetCore.Builder.IApplicationBuilder	The Microsoft.AspNetCore.Builder.IApplicationBuilder.

UseServerSideBlazor<TStartup>(IApplicationBuilder)

Registers Server-Side Blazor in the pipeline.

Declaration

```
public static IApplicationBuilder UseServerSideBlazor<TStartup>(this IApplicationBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.AspNetCore.Builder.IApplicationBuilder	builder	The Microsoft.AspNetCore.Builder.IApplicationBuilder.

Returns

TYPE	DESCRIPTION
Microsoft.AspNetCore.Builder.IApplicationBuilder	The Microsoft.AspNetCore.Builder.IApplicationBuilder.

Type Parameters

NAME	DESCRIPTION
TStartup	A Blazor startup type.

Namespace Microsoft.Extensions.DependencyInjection

Classes

[ServerSideBlazorServiceCollectionExtensions](#)

Extension methods to configure an Microsoft.Extensions.DependencyInjection.IServiceCollection for Server-Side Blazor.

Class ServerSideBlazorServiceCollectionExtensions

Extension methods to configure an Microsoft.Extensions.DependencyInjection.IServiceCollection for Server-Side Blazor.

Inheritance

System.Object

ServerSideBlazorServiceCollectionExtensions

Namespace: `Microsoft.Extensions.DependencyInjection`

Assembly: Microsoft.AspNetCore.Blazor.Server.dll

Syntax

```
public static class ServerSideBlazorServiceCollectionExtensions : Object
```

Methods

AddServerSideBlazor(IServiceCollection)

Adds Server-Side Blazor services to the service collection.

Declaration

```
public static IServiceCollection AddServerSideBlazor(this IServiceCollection services)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The Microsoft.Extensions.DependencyInjection.IServiceCollection.

Returns

TYPE	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	The Microsoft.Extensions.DependencyInjection.IServiceCollection.

AddServerSideBlazor(IServiceCollection, Action<ServerSideBlazorOptions>)

Adds Server-Side Blazor services to the service collection.

Declaration

```
public static IServiceCollection AddServerSideBlazor(this IServiceCollection services,
Action<ServerSideBlazorOptions> configure)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The Microsoft.Extensions.DependencyInjection.IServiceCollection.
System.Action<ServerSideBlazorOptions>	configure	A delegate to configure the ServerSideBlazorOptions.

Returns

TYPE	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	The Microsoft.Extensions.DependencyInjection.IServiceCollection.

AddServerSideBlazor(IServiceCollection, Type)

Adds Server-Side Blazor services to the service collection.

Declaration

```
public static IServiceCollection AddServerSideBlazor(this IServiceCollection services, Type startupType)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The Microsoft.Extensions.DependencyInjection.IServiceCollection.
System.Type	startupType	A Blazor startup type.

Returns

TYPE	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	The Microsoft.Extensions.DependencyInjection.IServiceCollection.

AddServerSideBlazor(IServiceCollection, Type, Action<ServerSideBlazorOptions>)

Adds Server-Side Blazor services to the service collection.

Declaration

```
public static IServiceCollection AddServerSideBlazor(this IServiceCollection services, Type startupType,
Action<ServerSideBlazorOptions> configure)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The Microsoft.Extensions.DependencyInjection.IServiceCollection.
System.Type	startupType	A Blazor startup type.
System.Action<ServerSideBlazorOptions>	configure	A delegate to configure the ServerSideBlazorOptions.

Returns

TYPE	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	The Microsoft.Extensions.DependencyInjection.IServiceCollection.

AddServerSideBlazor<TStartup>(IServiceCollection)

Adds Server-Side Blazor services to the service collection.

Declaration

```
public static IServiceCollection AddServerSideBlazor<TStartup>(this IServiceCollection services)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The Microsoft.Extensions.DependencyInjection.IServiceCollection.

Returns

TYPE	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	The Microsoft.Extensions.DependencyInjection.IServiceCollection.

Type Parameters

NAME	DESCRIPTION
TStartup	A Blazor startup type.

AddServerSideBlazor<TStartup>(IServiceCollection, Action<ServerSideBlazorOptions>)

Adds Server-Side Blazor services to the service collection.

Declaration

```
public static IServiceCollection AddServerSideBlazor<TStartup>(this IServiceCollection services,
Action<ServerSideBlazorOptions> configure)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The Microsoft.Extensions.DependencyInjection.IServiceCollection.
System.Action<ServerSideBlazorOptions>	configure	A delegate to configure the ServerSideBlazorOptions.

Returns

TYPE	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	The Microsoft.Extensions.DependencyInjection.IServiceCollection.

Type Parameters

NAME	DESCRIPTION
TStartup	A Blazor startup type.

Namespace Microsoft.JSInterop

Classes

DotNetDispatcher

Provides methods that receive incoming calls from JS to .NET.

DotNetObjectRef

Wraps a JS interop argument, indicating that the value should not be serialized as JSON but instead should be passed as a reference.

To avoid leaking memory, the reference must later be disposed by JS code or by .NET code.

JSException

Represents errors that occur during an interop call from .NET to JavaScript.

JSInProcessRuntimeBase

Abstract base class for an in-process JavaScript runtime.

JSInvokableAttribute

Identifies a .NET method as allowing invocation from JavaScript code. Any method marked with this attribute may receive arbitrary parameter values from untrusted callers. All inputs should be validated carefully.

Json

Provides mechanisms for converting between .NET objects and JSON strings for use when making calls to JavaScript functions via IJSRuntime.

Warning: This is not intended as a general-purpose JSON library. It is only intended for use when making calls via IJSRuntime. Eventually its implementation will be replaced by something more general-purpose.

JSRuntime

Provides mechanisms for accessing the current IJSRuntime.

JSRuntimeBase

Abstract base class for a JavaScript runtime.

Interfaces

IJSInProcessRuntime

Represents an instance of a JavaScript runtime to which calls may be dispatched.

IJSRuntime

Represents an instance of a JavaScript runtime to which calls may be dispatched.

Class DotNetDispatcher

Provides methods that receive incoming calls from JS to .NET.

Inheritance

System.Object

DotNetDispatcher

Syntax

```
public static class DotNetDispatcher : Object
```

Methods

BeginInvoke(String, String, String, Int64, String)

Receives a call from JS to .NET, locating and invoking the specified method asynchronously.

Declaration

```
public static void BeginInvoke(string callId, string assemblyName, string methodIdentifier, long
dotNetObjectId, string argsJson)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	callId	A value identifying the asynchronous call that should be passed back with the result, or null if no result notification is required.
System.String	assemblyName	The assembly containing the method to be invoked.
System.String	methodIdentifier	The identifier of the method to be invoked. The method must be annotated with a JSInvokableAttribute matching this identifier string.
System.Int64	dotNetObjectId	For instance method calls, identifies the target object.
System.String	argsJson	A JSON representation of the parameters.

EndInvoke(Int64, Boolean, JSAsyncCallResult)

Receives notification that a call from .NET to JS has finished, marking the associated System.Threading.Tasks.Task as completed.

Declaration

```
[JSInvokable("DotNetDispatcher.EndInvoke")]
public static void EndInvoke(long asyncHandle, bool succeeded, JSAsyncCallResult result)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	asyncHandle	The identifier for the function invocation.
System.Boolean	succeeded	A flag to indicate whether the invocation succeeded.
JSAsyncCallResult	result	If `succeeded` is `true`, specifies the invocation result. If `succeeded` is `false`, gives the System.Exception corresponding to the invocation failure.

Invoke(String, String, Int64, String)

Receives a call from JS to .NET, locating and invoking the specified method.

Declaration

```
public static string Invoke(string assemblyName, string methodIdentifier, long dotNetObjectId, string
argsJson)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	assemblyName	The assembly containing the method to be invoked.
System.String	methodIdentifier	The identifier of the method to be invoked. The method must be annotated with a JSInvokableAttribute matching this identifier string.
System.Int64	dotNetObjectId	For instance method calls, identifies the target object.
System.String	argsJson	A JSON representation of the parameters.

Returns

TYPE	DESCRIPTION
System.String	A JSON representation of the return value, or null.

ReleaseDotNetObject(Int64)

Releases the reference to the specified .NET object. This allows the .NET runtime to garbage collect that object if there are no other references to it.

To avoid leaking memory, the JavaScript side code must call this for every .NET object it obtains a reference to. The exception is if that object is used for the entire lifetime of a given user's session, in which case it is released automatically when the JavaScript runtime is disposed.

Declaration

```
[JSInvokable("DotNetDispatcher.ReleaseDotNetObject")]
public static void ReleaseDotNetObject(long dotNetObjectId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	dotNetObjectId	The identifier previously passed to JavaScript code.

Class DotNetObjectRef

Wraps a JS interop argument, indicating that the value should not be serialized as JSON but instead should be passed as a reference.

To avoid leaking memory, the reference must later be disposed by JS code or by .NET code.

Inheritance

System.Object

DotNetObjectRef

Implements

System.IDisposable

Syntax

```
public class DotNetObjectRef : Object, IDisposable
```

Constructors

DotNetObjectRef(Object)

Constructs an instance of DotNetObjectRef.

Declaration

```
public DotNetObjectRef(object value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	The value being wrapped.

Properties

Value

Gets the object instance represented by this wrapper.

Declaration

```
public object Value { get; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

Methods

Dispose()

Stops tracking this object reference, allowing it to be garbage collected (if there are no other references to it). Once the instance is disposed, it can no longer be used in interop calls from JavaScript code.

Declaration

```
public void Dispose()
```

EnsureAttachedToJsRuntime(IJSRuntime)

Ensures the DotNetObjectRef is associated with the specified IJSRuntime. Developers do not normally need to invoke this manually, since it is called automatically by framework code.

Declaration

```
public void EnsureAttachedToJsRuntime(IJSRuntime runtime)
```

Parameters

TYPE	NAME	DESCRIPTION
IJSRuntime	runtime	The IJSRuntime.

Implements

System.IDisposable

Interface IJSInProcessRuntime

Represents an instance of a JavaScript runtime to which calls may be dispatched.

Inherited Members

[IJSRuntime.InvokeAsync<T>(String, Object[])](#)

[IJSRuntime.UntrackObjectRef(DotNetObjectRef)](#)

Namespace: [Microsoft.JSInterop](#)

Assembly: Microsoft.JSInterop.dll

Syntax

```
public interface IJSInProcessRuntime : IJSRuntime
```

Methods

Invoke<T>(String, Object[])

Invokes the specified JavaScript function synchronously.

Declaration

```
T Invoke<T>(string identifier, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	identifier	An identifier for the function to invoke. For example, the value `"someScope.someFunction"` will invoke the function `window.someScope.someFunction` .
System.Object[]	args	JSON-serializable arguments.

Returns

TYPE	DESCRIPTION
T	An instance of T obtained by JSON-deserializing the return value.

Type Parameters

NAME	DESCRIPTION
T	The JSON-serializable return type.

Interface IJSRuntime

Represents an instance of a JavaScript runtime to which calls may be dispatched.

Namespace: Microsoft.JSInterop

Assembly: Microsoft.JSInterop.dll

Syntax

```
public interface IJSRuntime
```

Methods

InvokeAsync<T>(String, Object[])

Invokes the specified JavaScript function asynchronously.

Declaration

```
Task<T> InvokeAsync<T>(string identifier, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	identifier	An identifier for the function to invoke. For example, the value `"someScope.someFunction"` will invoke the function `window.someScope.someFunction` .
System.Object[]	args	JSON-serializable arguments.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<T>	An instance of T obtained by JSON-deserializing the return value.

Type Parameters

NAME	DESCRIPTION
T	The JSON-serializable return type.

UntrackObjectRef(DotNetObjectRef)

Stops tracking the .NET object represented by the DotNetObjectRef. This allows it to be garbage collected (if nothing else holds a reference to it) and means the JS-side code can no longer invoke methods on the instance or pass it as an argument to subsequent calls.

Declaration

void UntrackObjectRef(DotNetObjectRef dotNetObjectRef)

Parameters

TYPE	NAME	DESCRIPTION
DotNetObjectRef	dotNetObjectRef	The reference to stop tracking.

Remarks

This method is called automaticallly by Dispose().

void UntrackObjectRef(DotNetObjectRef dotNetObjectRef)

Parameters

TYPE	NAME	DESCRIPTION
DotNetObjectRef	dotNetObjectRef	The reference to stop tracking.

Class JSException

Represents errors that occur during an interop call from .NET to JavaScript.

Inheritance

System.Object

JSException

Namespace: Microsoft.JSInterop

Assembly: Microsoft.JSInterop.dll

Syntax

```
public class JSException : Exception
```

Constructors

JSException(String)

Constructs an instance of JSException.

Declaration

```
public JSException(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	The exception message.

Class JSInProcessRuntimeBase

Abstract base class for an in-process JavaScript runtime.

Inheritance

System.Object

[JSRuntimeBase](#)

JSInProcessRuntimeBase

Implements

[IJSInProcessRuntime](#)

[IJSRuntime](#)

Inherited Members

[JSRuntimeBase.UntrackObjectRef(DotNetObjectRef)](#)

[JSRuntimeBase.InvokeAsync<T>(String, Object[])](#)

[JSRuntimeBase.BeginInvokeJS(Int64, String, String)](#)

Syntax

```
public abstract class JSInProcessRuntimeBase : JSRuntimeBase, IJSInProcessRuntime, IJSRuntime
```

Constructors

JSInProcessRuntimeBase()

Declaration

```
protected JSInProcessRuntimeBase()
```

Methods

Invoke<T>(String, Object[])

Invokes the specified JavaScript function synchronously.

Declaration

```
public T Invoke<T>(string identifier, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION

System.String	identifier	An identifier for the function to invoke. For example, the value
		
		`"someScope.someFunction"`
		
		 will invoke the function
		
		`window.someScope.someFunction`
		
		 .

System.Object[]	args	JSON-serializable arguments.

Returns

TYPE	DESCRIPTION
T	An instance of T obtained by JSON-deserializing the return value.

Type Parameters

NAME	DESCRIPTION
T	The JSON-serializable return type.

InvokeJS(String, String)

Performs a synchronous function invocation.

Declaration

```
protected abstract string InvokeJS(string identifier, string argsJson)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	identifier	The identifier for the function to invoke.
System.String	argsJson	A JSON representation of the arguments.

Returns

TYPE	DESCRIPTION
System.String	A JSON representation of the result.

Implements

[IJSInProcessRuntime](#)
[IJSRuntime](#)

Class JSInvokableAttribute

Identifies a .NET method as allowing invocation from JavaScript code. Any method marked with this attribute may receive arbitrary parameter values from untrusted callers. All inputs should be validated carefully.

Inheritance

System.Object

JSInvokableAttribute

Namespace: Microsoft.JSInterop

Assembly: Microsoft.JSInterop.dll

Syntax

```
public class JSInvokableAttribute : Attribute
```

Constructors

JSInvokableAttribute()

Constructs an instance of JSInvokableAttribute without setting an identifier for the method.

Declaration

```
public JSInvokableAttribute()
```

JSInvokableAttribute(String)

Constructs an instance of JSInvokableAttribute using the specified identifier.

Declaration

```
public JSInvokableAttribute(string identifier)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	identifier	An identifier for the method, which must be unique within the scope of the assembly.

Properties

Identifier

Gets the identifier for the method. The identifier must be unique within the scope of an assembly.

If not set, the identifier is taken from the name of the method. In this case the method name must be unique within the assembly.

Declaration

```
public string Identifier { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class Json

Provides mechanisms for converting between .NET objects and JSON strings for use when making calls to JavaScript functions via IJSRuntime.

Warning: This is not intended as a general-purpose JSON library. It is only intended for use when making calls via IJSRuntime. Eventually its implementation will be replaced by something more general-purpose.

Inheritance

System.Object

Json

Namespace: Microsoft.JSInterop

Assembly: Microsoft.JSInterop.dll

Syntax

```
public static class Json : Object
```

Methods

Deserialize<T>(String)

Deserializes the JSON string, creating an object of the specified generic type.

Declaration

```
public static T Deserialize<T>(string json)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	json	The JSON string.

Returns

TYPE	DESCRIPTION
T	An object of the specified type.

Type Parameters

NAME	DESCRIPTION
T	The type of object to create.

Serialize(Object)

Serializes the value as a JSON string.

Declaration

```
public static string Serialize(object value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	The value to serialize.

Returns

TYPE	DESCRIPTION
System.String	The JSON string.

Class JSRuntime

Provides mechanisms for accessing the current IJSRuntime.

Inheritance

System.Object

JSRuntime

Namespace: Microsoft.JSInterop

Assembly: Microsoft.JSInterop.dll

Syntax

```
public static class JSRuntime : Object
```

Properties

Current

Gets the current IJSRuntime, if any.

Declaration

```
public static IJSRuntime Current { get; }
```

Property Value

TYPE	DESCRIPTION
IJSRuntime	

Methods

SetCurrentJSRuntime(IJSRuntime)

Sets the current JS runtime to the supplied instance.

This is intended for framework use. Developers should not normally need to call this method.

Declaration

```
public static void SetCurrentJSRuntime(IJSRuntime instance)
```

Parameters

TYPE	NAME	DESCRIPTION
IJSRuntime	instance	The new current IJSRuntime.

Class JSRuntimeBase

Abstract base class for a JavaScript runtime.

Inheritance

System.Object

JSRuntimeBase

JSInProcessRuntimeBase

Implements

IJSRuntime

Namespace: Microsoft.JSInterop

Assembly: Microsoft.JSInterop.dll

Syntax

```
public abstract class JSRuntimeBase : Object, IJSRuntime
```

Constructors

JSRuntimeBase()

Constructs an instance of JSRuntimeBase.

Declaration

```
public JSRuntimeBase()
```

Methods

BeginInvokeJS(Int64, String, String)

Begins an asynchronous function invocation.

Declaration

```
protected abstract void BeginInvokeJS(long asyncHandle, string identifier, string argsJson)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	asyncHandle	The identifier for the function invocation, or zero if no async callback is required.
System.String	identifier	The identifier for the function to invoke.
System.String	argsJson	A JSON representation of the arguments.

InvokeAsync<T>(String, Object[])

Invokes the specified JavaScript function asynchronously.

Declaration

```
public Task<T> InvokeAsync<T>(string identifier, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	identifier	An identifier for the function to invoke. For example, the value ``` "someScope.someFunction" ``` will invoke the function ``` window.someScope.someFunction ``` .
System.Object[]	args	JSON-serializable arguments.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<T>	An instance of `T` obtained by JSON-deserializing the return value.

Type Parameters

NAME	DESCRIPTION
T	The JSON-serializable return type.

UntrackObjectRef(DotNetObjectRef)

Stops tracking the .NET object represented by the DotNetObjectRef. This allows it to be garbage collected (if nothing else holds a reference to it) and means the JS-side code can no longer invoke methods on the instance or pass it as an argument to subsequent calls.

Declaration

```
public void UntrackObjectRef(DotNetObjectRef dotNetObjectRef)
```

Parameters

TYPE	NAME	DESCRIPTION
DotNetObjectRef	dotNetObjectRef	The reference to stop tracking.

Remarks

This method is called automaticallly by Dispose().

Implements

IJSRuntime

Namespace Microsoft.JSInterop.Internal

Classes

JSAsyncCallResult

Intended for framework use only.

Interfaces

ICustomArgSerializer

Internal. Intended for framework use only.

Interface ICustomArgSerializer

Internal. Intended for framework use only.

Namespace: Microsoft.JSInterop.Internal

Assembly: Microsoft.JSInterop.dll

Syntax

```
public interface ICustomArgSerializer
```

Methods

ToJsonPrimitive()

Internal. Intended for framework use only.

Declaration

```
object ToJsonPrimitive()
```

Returns

TYPE	DESCRIPTION
System.Object	

Class JSAsyncCallResult

Intended for framework use only.

Inheritance

System.Object

JSAsyncCallResult

Namespace: Microsoft.JSInterop.Internal

Assembly: Microsoft.JSInterop.dll

Syntax

```
public class JSAsyncCallResult : Object
```