

Blazor: sviluppo Desktop, Web e Mobile x-platform con un unico framework



Sabato 30 settembre 2023



<Michele Aponte>



SPONSORS

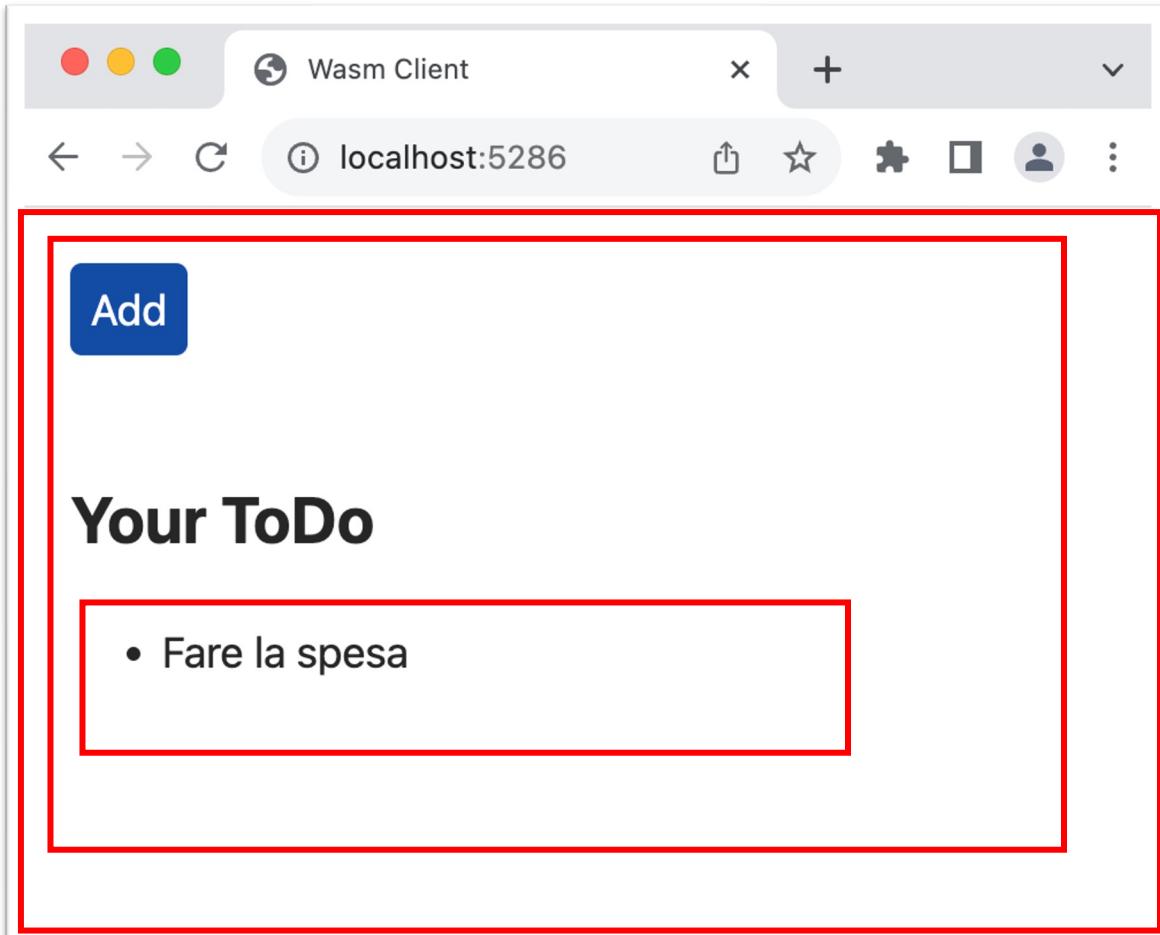


Chi siete?

- Quanti programmatori .NET?
- Quanti conoscono Blazor?
- Quanti lo stanno usando in produzione?
- Quanti di voi sanno come funziona, veramente, un componente in Blazor?

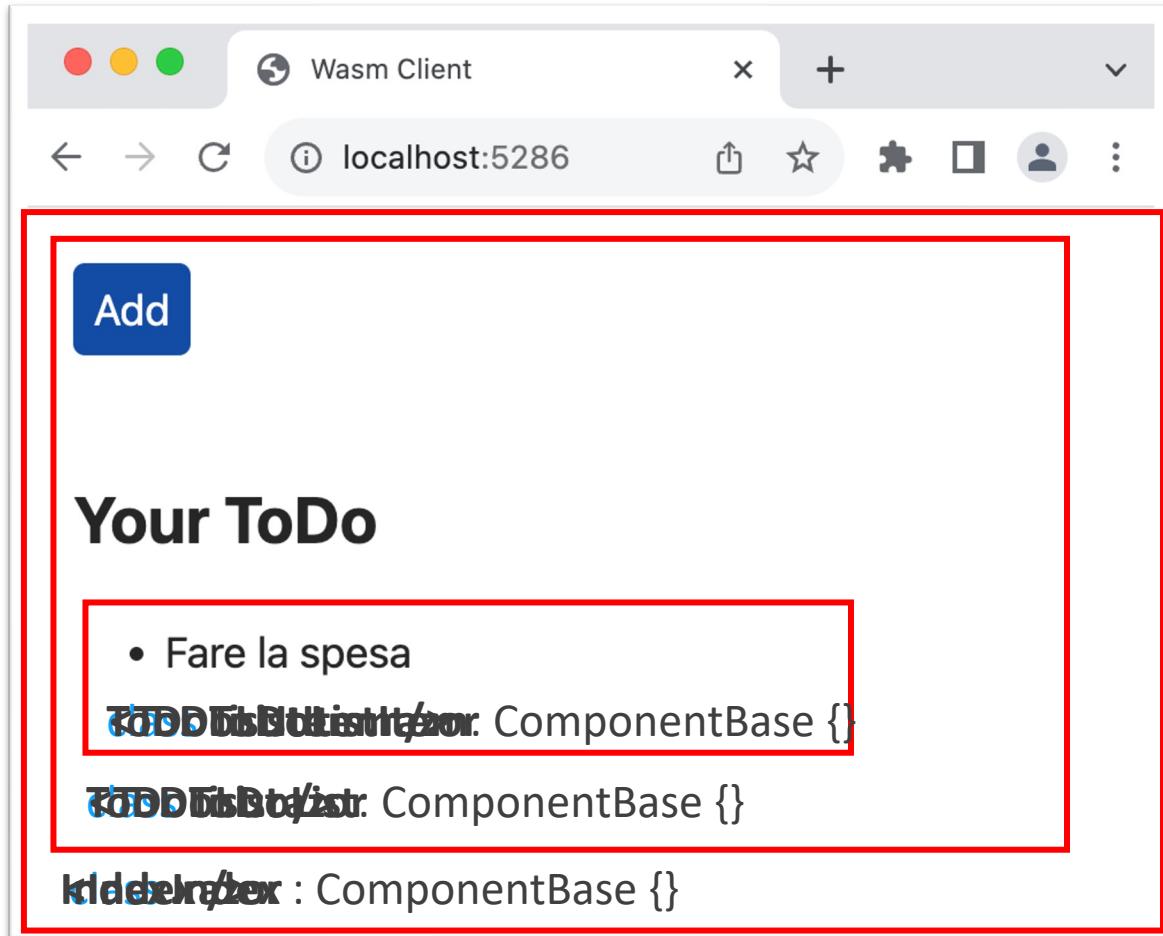


Che cos'è, davvero, un componente



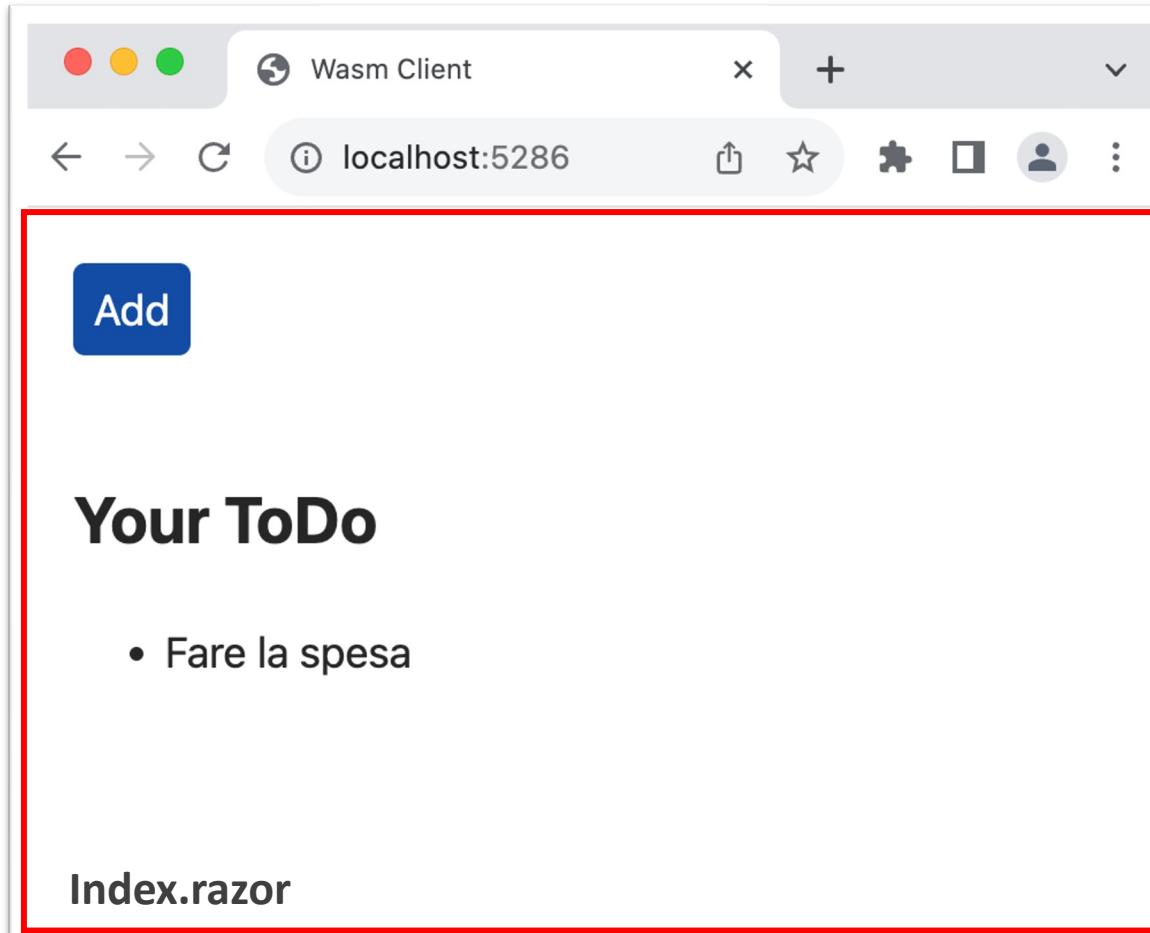
- Un componente è **una porzione della nostra interfaccia**
- Può **contenere** altri componenti
- La sua **responsabilità** è mostrare quella porzione di interfaccia e **catturare l'input dell'utente.**
- Deve essere **disaccoppiato** dagli altri componenti
- **Riutilizzabile**

Che cos'è un componente in Blazor



- Una **classe che estende la classe base ComponentBase**
- Può essere generato a partire da un file **.razor**
- Il **nome della classe** individua anche il suo **selettore**
- Permette di generare l'**HTML** necessario al browser per visualizzarlo

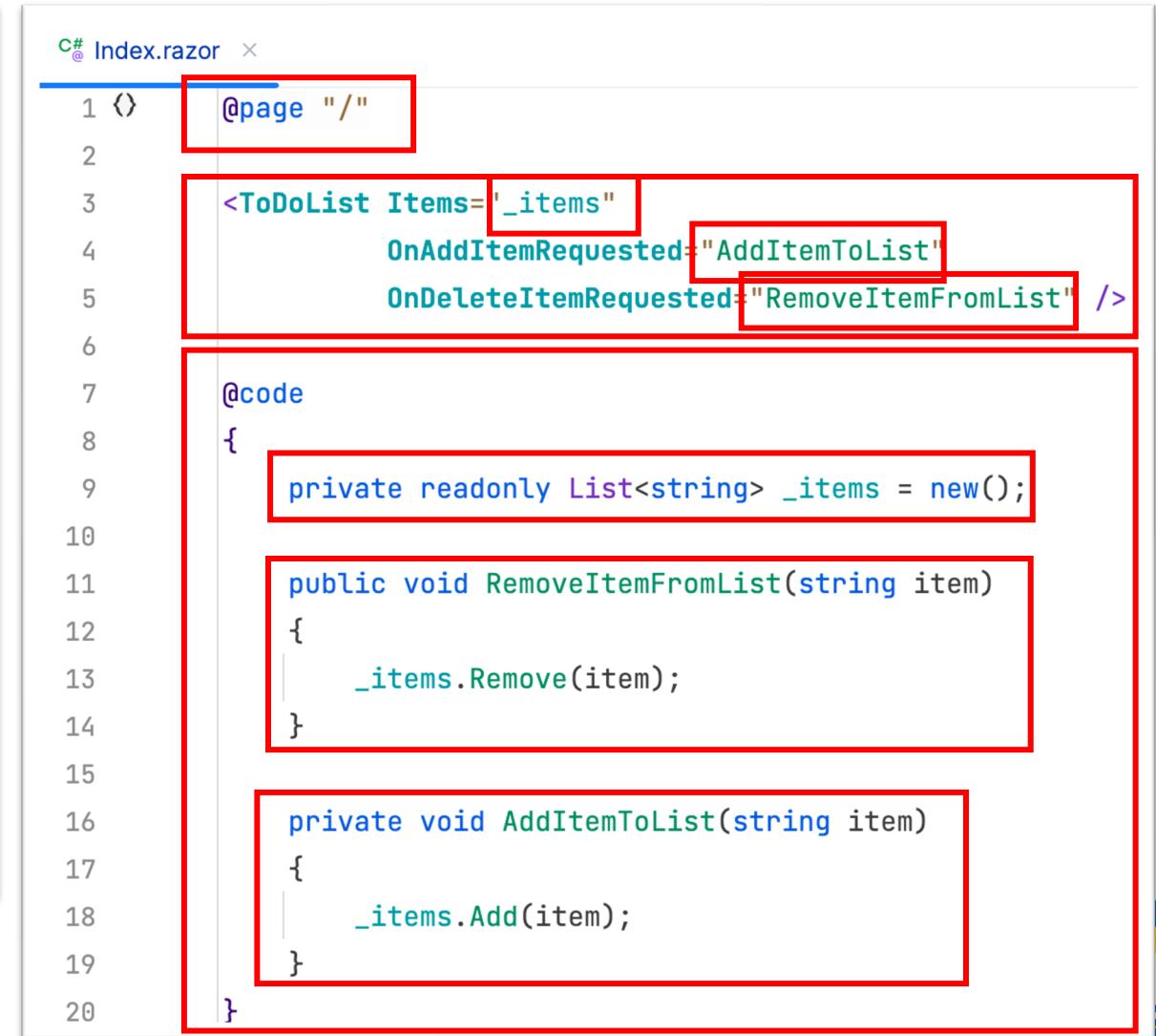
Un componente Blazor generato da un file .razor



The screenshot shows a browser window titled "Wasm Client" with the URL "localhost:5286". Inside the browser, there is a Blazor component with the following structure:

```
1 <div>
2   <h2>Your ToDo</h2>
3   <ul>
4     <li>• Fare la spesa</li>
5   </ul>
6   <button>Add</button>
7 </div>
```

The entire component area is highlighted with a red border.



```
C# Index.razor
1 @page "/"
2
3 <ToDoList Items="[_items]" OnAddItemRequested="AddItemToList" OnDeleteItemRequested="RemoveItemFromList" />
4
5
6 @code
7 {
8   private readonly List<string> _items = new();
9
10  public void RemoveItemFromList(string item)
11  {
12    _items.Remove(item);
13  }
14
15  private void AddItemToList(string item)
16  {
17    _items.Add(item);
18  }
19}
```

Specific sections of the code are highlighted with red boxes:

- The first three lines of the component declaration (`@page "/"`, `<ToDoList ...>`, and the event handlers) are highlighted.
- The entire `@code` block and its contents are highlighted.
- The `_items` field and its usage in the `RemoveItemFromList` and `AddItemToList` methods are highlighted.

Un componente Blazor generato da un file .razor

The screenshot shows a browser window titled "Wasm Client" displaying a "ToDoList.razor" component. The component has a blue "Add" button at the top left. Below it, the heading "Your ToDo" is displayed in large bold letters. A single item, "• Fare la spesa", is listed under the heading. At the bottom left, the file name "ToDoList.razor" is shown. The entire content area of the browser is highlighted with a red box.

The right side of the screenshot shows the "ToDoList.razor" file in a code editor. The code uses C# syntax and Razor components. It injects a `IDataService` named `DataService`. The UI consists of a button labeled "Add", a heading "Your ToDo", and a conditional block. If there are items, it displays them in an `ul` list; otherwise, it shows a "No Items" message. The code editor's syntax highlighting includes colors for different language elements like keywords, comments, and punctuation.

```
c# ToDoList.razor x
1 @<> @inject IDataService DataService
2
3 <button type="button" @onclick="RequestItemData">Add</button>
4 <h3>Your ToDo</h3>
5 <@if (Items?.Count > 0)>
6 {
7     <ul>
8         <@foreach (var item:string in Items)>
9             {
10                 <ToDoListItem Item="@item"
11                     OnItemClicked="OnDeleteItemRequested" />
12             }
13         </ul>
14     <@else>
15     {
16         <span>No Items</span>
17     }
18
19     <@code
20         DataService = new DataService();
21     </code>
22 }
```

Un componente Blazor generato da un file .razor

The image shows a split-screen view. On the left is a screenshot of a web browser window titled "Wasm Client" at "localhost:5286". The page displays a "Your ToDo" section with a single item: "• Fare la spesa". Below this is a "ToDo.razor" component header. A red box highlights the entire content area of the browser. On the right is a code editor window showing the "ToDoList.razor" file. The code defines a component with three parameters: "Items", "OnAddItemRequested", and "OnDeleteItemRequested". It also includes a private method "RequestItemData" that uses await DataService.RequestData() to get an item, trims it, and then calls the OnAddItemRequested event if it's not empty. A second red box highlights the parameter definitions.

```
#@ToDoList.razor
@code
{
    [Parameter]
    public List<string>? Items { get; set; }

    [Parameter]
    public EventCallback<string> OnAddItemRequested { get; set; }

    [Parameter]
    public EventCallback<string> OnDeleteItemRequested { get; set; }

    private async Task RequestItemData()
    {
        var item:string = await DataService.RequestData();
        item = item.Trim();
        if(!string.IsNullOrEmpty(item))
            await OnAddItemRequested.InvokeAsync(item);
    }
}
```

Un componente Blazor generato da un file .razor

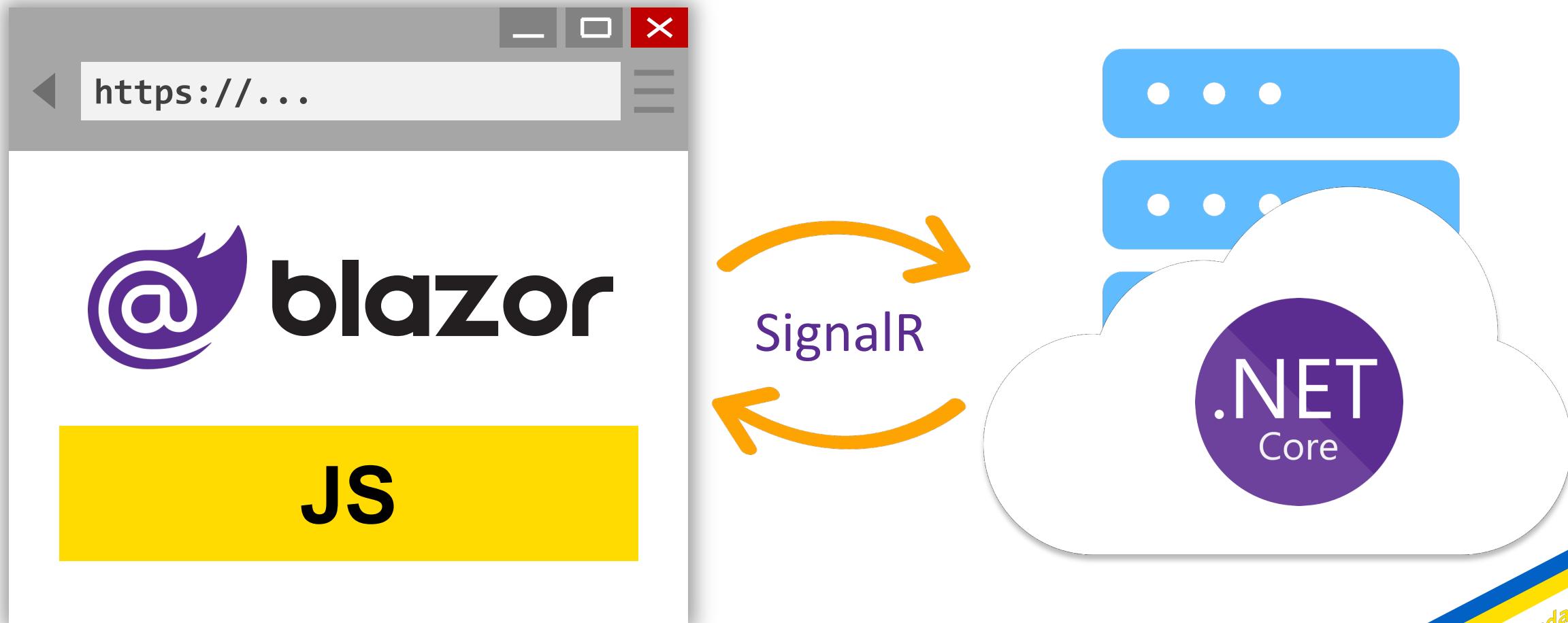
C# ToDoList.razor ×

```
20 @code
21 {
22     [Parameter]
23     public List<string>? Items { get; set; }
24
25     [Parameter]
26     public EventCallback<string> OnAddItemRequested { get; set; }
27
28     [Parameter]
29     public EventCallback<string> OnDeleteItemRequested { get; set; }
30
31     private async Task RequestItemData()
32     {
33         await ...
34     }
35 }
```

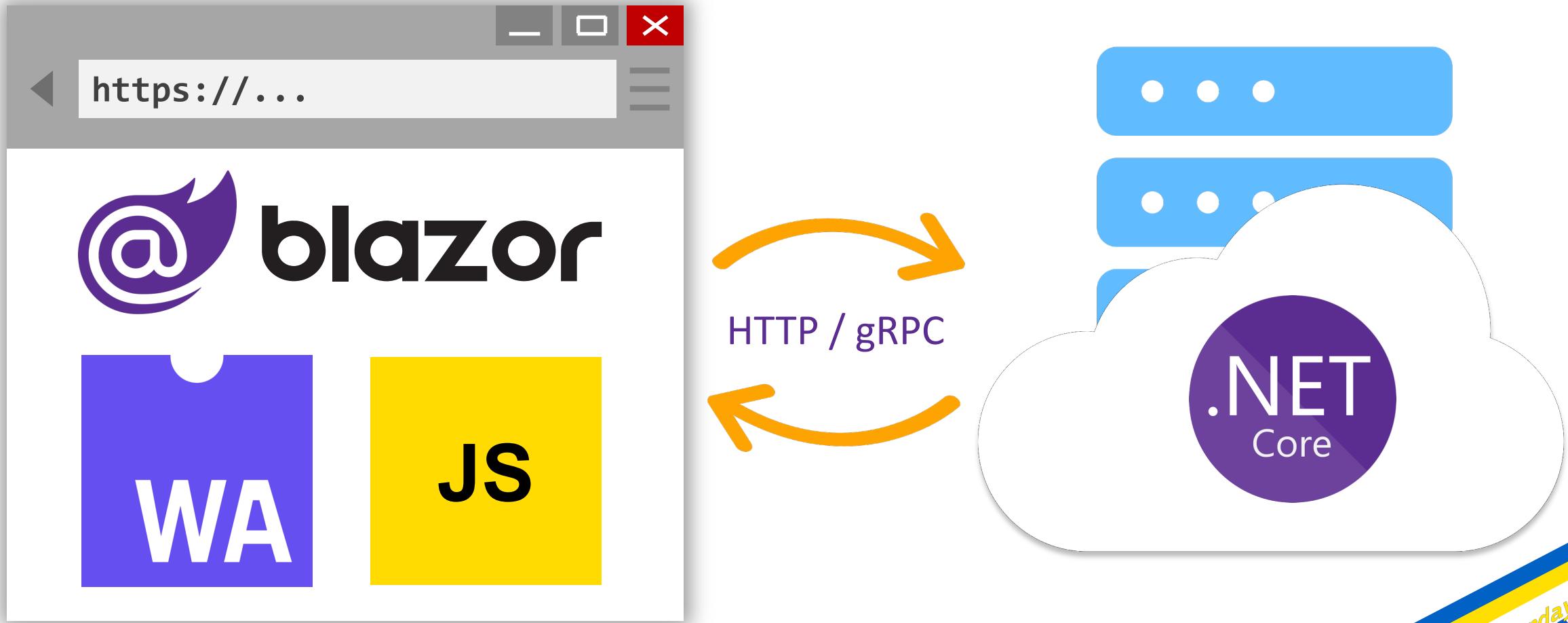


```
23 public List<string> Items { get; set; }  
24  
25 [Parameter]  
26 public EventCallback<string> OnAddItemRequested { get; set; }  
27  
28 [Parameter]  
29 public EventCallback<string> OnDeleteItemRequested { get; set; }  
30  
31 private async Task RequestItemData()  
32 {  
33     var item:string = await DataService.RequestData();  
34     item = item.Trim();  
35     if(!string.IsNullOrEmpty(item))  
36         await OnAddItemRequested.InvokeAsync(item);  
37 }  
38 }
```

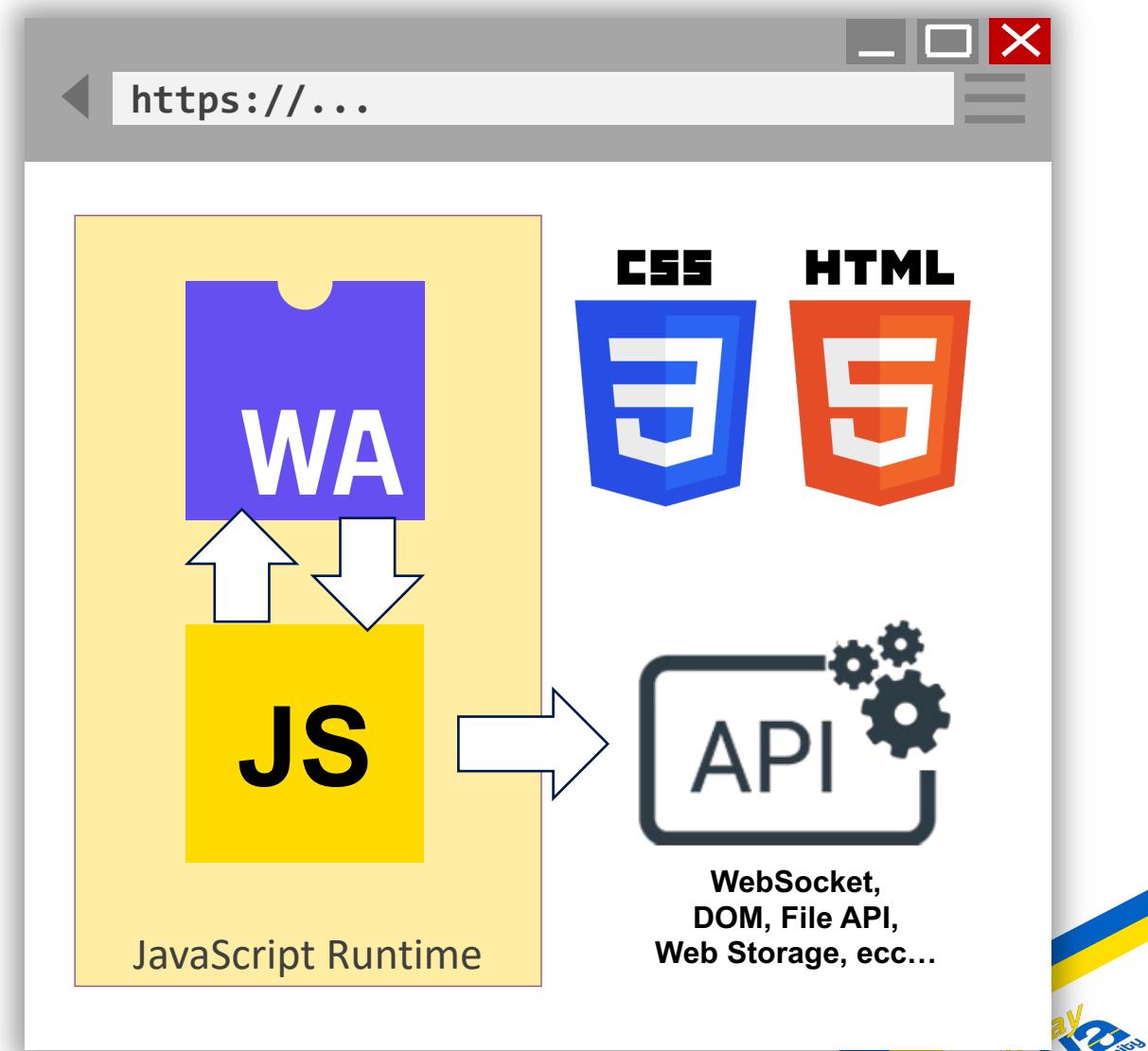
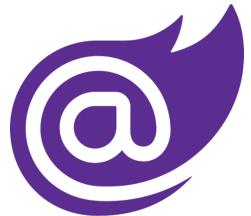
Blazor Server



Blazor WebAssembly



WebAssembly



WebSocket,
DOM, File API,
Web Storage, ecc...

API

JavaScript Runtime



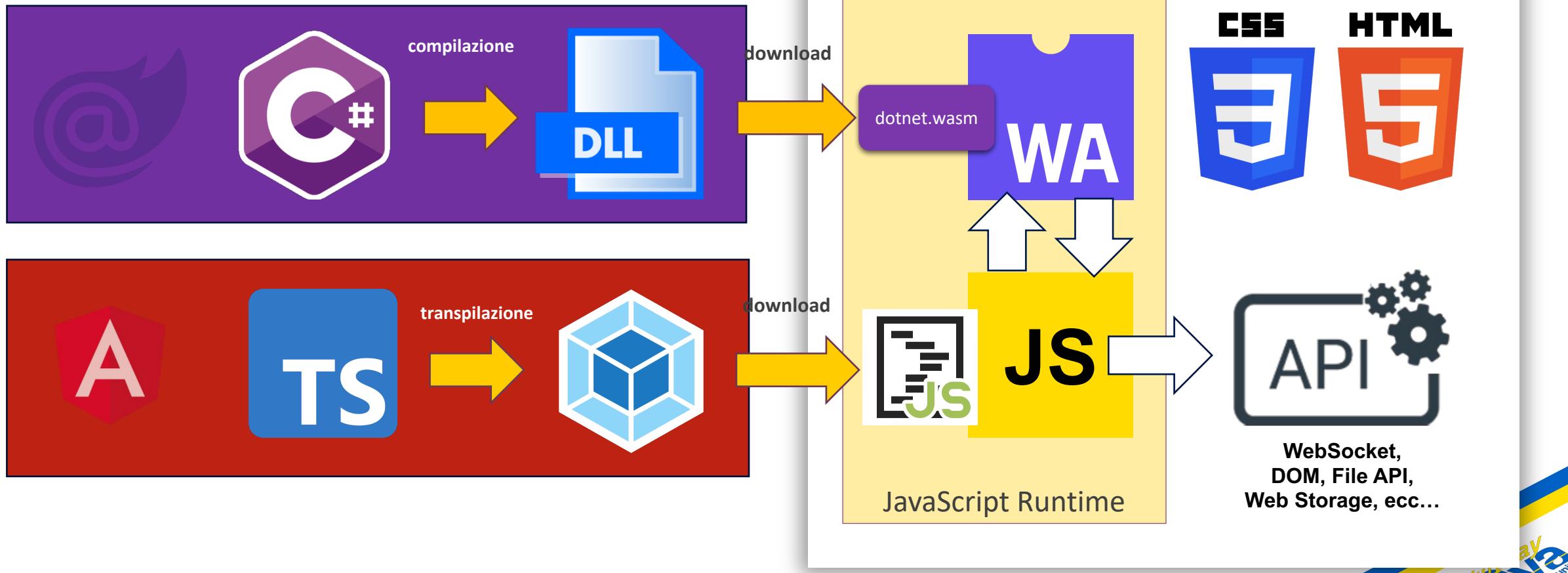
https://...



JS



WebAssembly



Anatomia di un componente



Sabato 30 settembre 2023

Michele Aponte 

Come avviene l'aggiornamento della UI

- Come fa Blazor a capire quando eseguire di nuovo il rendering di un componente?
 1. Quando viene sollevato un evento
 2. Quando vengono aggiornate le proprietà
 3. Quando viene esplicitamente chiesto

The screenshot shows the Blazor development environment with two panes. The left pane displays the `ToDoList.razor` component:

```
1 @inject IDataService Data
2
3 <button type="button" @on
4 <h3>Your ToDo</h3>
5 @if (Items?.Count > 0)
6 {
7     <ul>
8         @foreach (var item in Items)
9         {
10            <ToDoListItem OnItemClicked="OnDeleteItemRequested" />
11        }
12    </ul>
13}
14
```

The button element is highlighted with a red box. The right pane shows the generated `Components_ToDoList_razor.g.cs` file:

```
61 public partial class ToDoList : global::Microsoft.AspNetCore.Components.ComponentBase
62 {
63     #pragma warning disable 1998
64
65     protected override void BuildRenderTree(
66         global::Microsoft.AspNetCore.Components.Rendering.RenderTreeBuilder __builder)
67     {
68         __builder.OpenElement(0, "button");
69         __builder.AddAttribute(1, "type", "button");
70         __builder.AddAttribute(2, "onclick",
71             global::Microsoft.AspNetCore.Components.EventCallback.Factory
72                 .Create<global::Microsoft.AspNetCore.Components.Web.MouseEventArgs>(this,
73
74                     RequestItemData));
75
76         __builder.AddContent(3, "Add");
77         __builder.CloseElement();
78         __builder.AddMarkupContent(4, "\n");
79         __builder.OpenElement(5, "h3");
80         __builder.AddContent(6, "Your ToDo (");
81
82     }
83 }
```

The `__builder.AddAttribute(2, "onclick", ...)` line is highlighted with a red box, indicating the point where the Blazor runtime generates the event handling logic.

c# Components_ToDoList_razor.g.cs ×

```
61     public partial class ToDoList : global::Microsoft.AspNetCore.Components.ComponentBase
62     {
63         #pragma warning disable 1998
64         protected override void BuildRenderTree(
65             global::Microsoft.AspNetCore.Components.Rendering.RenderTreeBuilder __builder)
66         {
67             __builder.OpenElement(0, "button");
68             __builder.AddAttribute(1, "type", "button");
69             __builder.AddAttribute(2, "onclick",
70                 global::Microsoft.AspNetCore.Components.EventCallback.Factory
71                     .Create<global::Microsoft.AspNetCore.Components.Web.MouseEventArgs>(this,
72                         RequestItemData));
73             __builder.AddContent(3, "Add");
74             __builder.CloseElement();
75             __builder.AddMarkupContent(4, "\n");
76             __builder.OpenElement(5, "h3");
77             __builder.AddContent(6, "Your ToDo (");
```



Come avviene l'aggiornamento della UI

```
c# ComponentBase.cs ✘
◆ IL code

322 ^ I Task IHandleEvent.HandleEventAsync(EventCallbackWorkItem callback, object? arg)
323 {
324     var task = callback.InvokeAsync(arg);
325     var shouldAwaitTask:bool = task.Status != TaskStatus.RanToCompletion &&
326         task.Status != TaskStatus.Canceled;
327
328     // After each event, we synchronously re-render (unless !ShouldRender())
329     // This just saves the developer the trouble of putting "StateHasChanged();"
330     // at the end of every event callback.
331     StateHasChanged();
332
333     return shouldAwaitTask ?
334         CallStateHasChangedOnAsyncCompletion(task) :
335         Task.CompletedTask;
336 }
```

Come avviene l'aggiornamento della UI

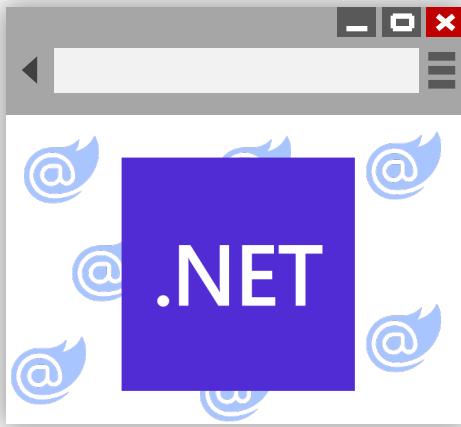
- Come fa Blazor a capire quando eseguire di nuovo il rendering di un componente?
 1. Quando viene sollevato un evento
 2. Quando vengono aggiornati i parametri di un componente
 3. Quando viene esplicitamente chiamato **StateHasChanged()**
- Quando posso considerare aggiornati i parametri di un componente?
 - <https://github.com/dotnet/aspnetcore/blob/main/src/Components/Components/src/ChangeDetection.cs>
- Per approfondire vi lascio un articolo molto ben fatto:
 - <https://jonhilton.net/blazor-rendering/>



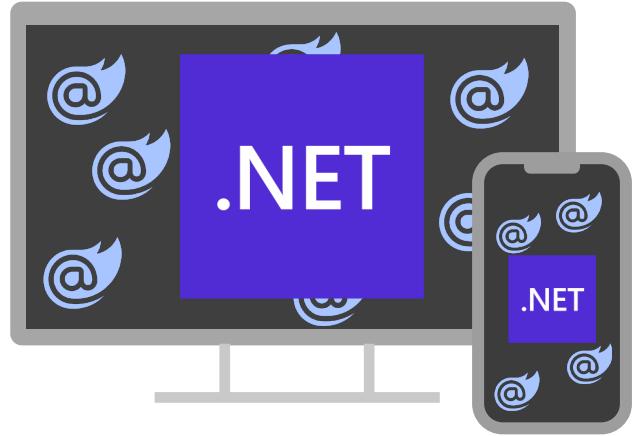
Blazor Hybrid (.NET 6)



**Blazor
Server**

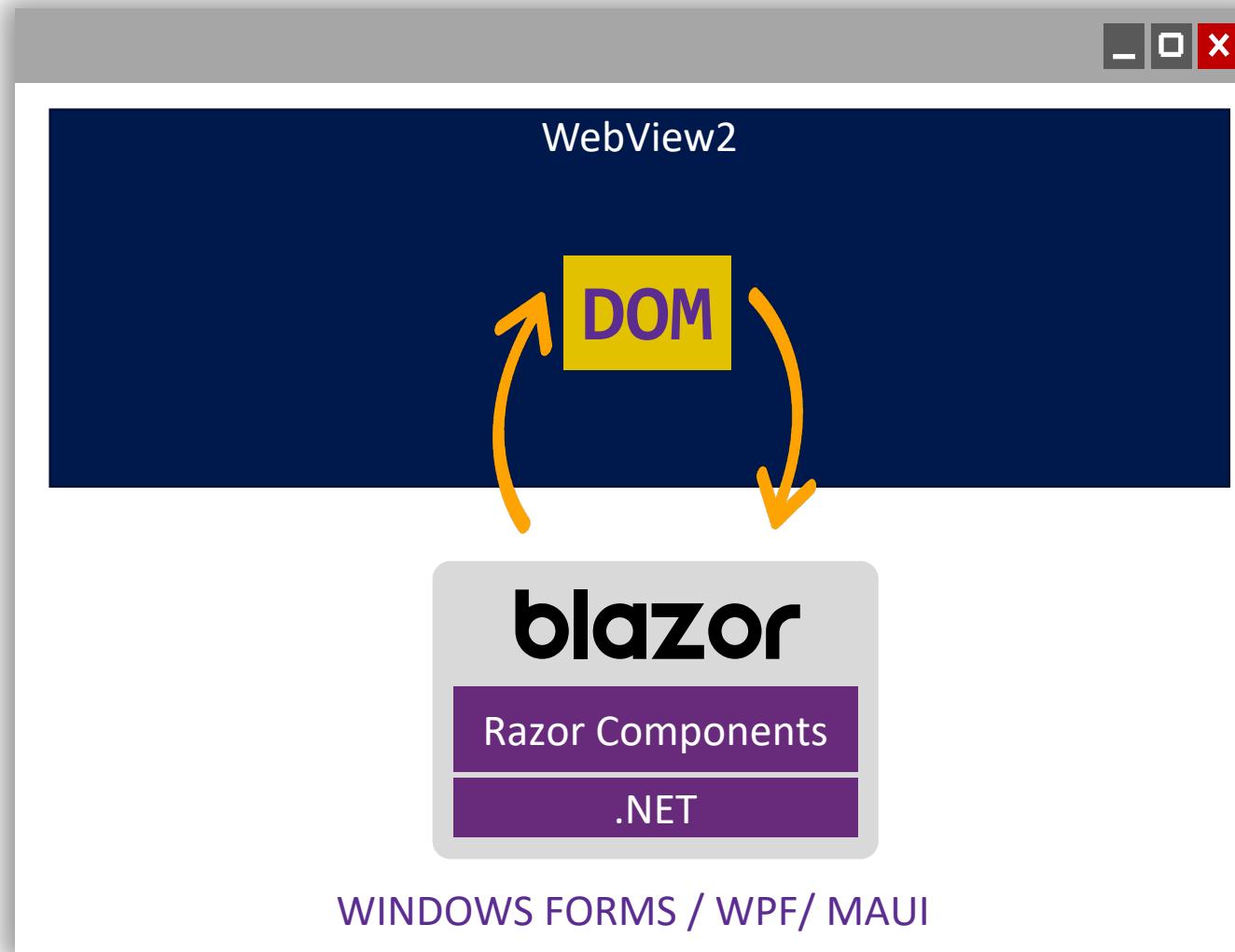


**Blazor
WebAssembly**

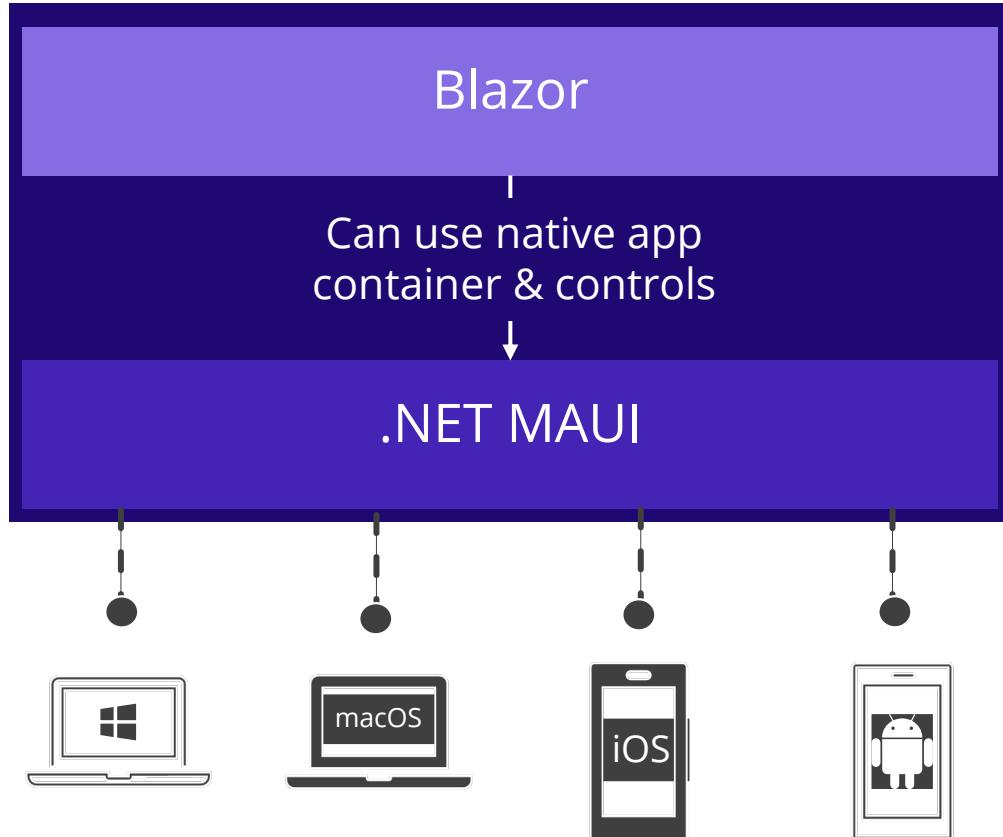


**Blazor
Hybrid**

Che cos'è Blazor Hybrid



Blazor in MAUI: come funziona



Reuse UI components across native and web

Built on top of .NET Multi-platform App UI

Native app container & embedded controls

Generally available

App Desktop, Mobile e Web con Blazor

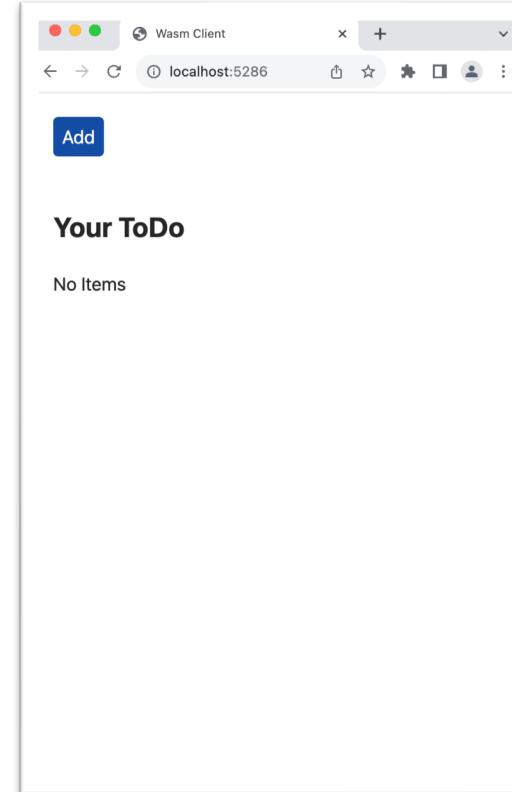


Sabato 30 settembre 2023

Michele Aponte 

Bello, ma che differenza c'è con l'app Web?

- La User eXperience è (quasi) la stessa!
- Nessuno dice che dobbiamo usare **un'unica istanza di BlazorWebView**
- Possiamo utilizzare un specifico componente, **ma dobbiamo passargli i parametri**



MAUI e Blazor nella realtà



Sabato 30 settembre 2023

Michele Aponte 

Perchè non funziona?

- Come mai secondo voi l'aggiornamento della UI non ha funzionato?

Nessuno ha chiamato **StateHasChanged()**



Come possiamo risolvere

- Proviamo a creare un **message broker** che **notifichi** ai componenti che **un parametro è cambiato**

```
public class PropertyChangeNotifier
{
    public event EventHandler<string> PropertyChanged;

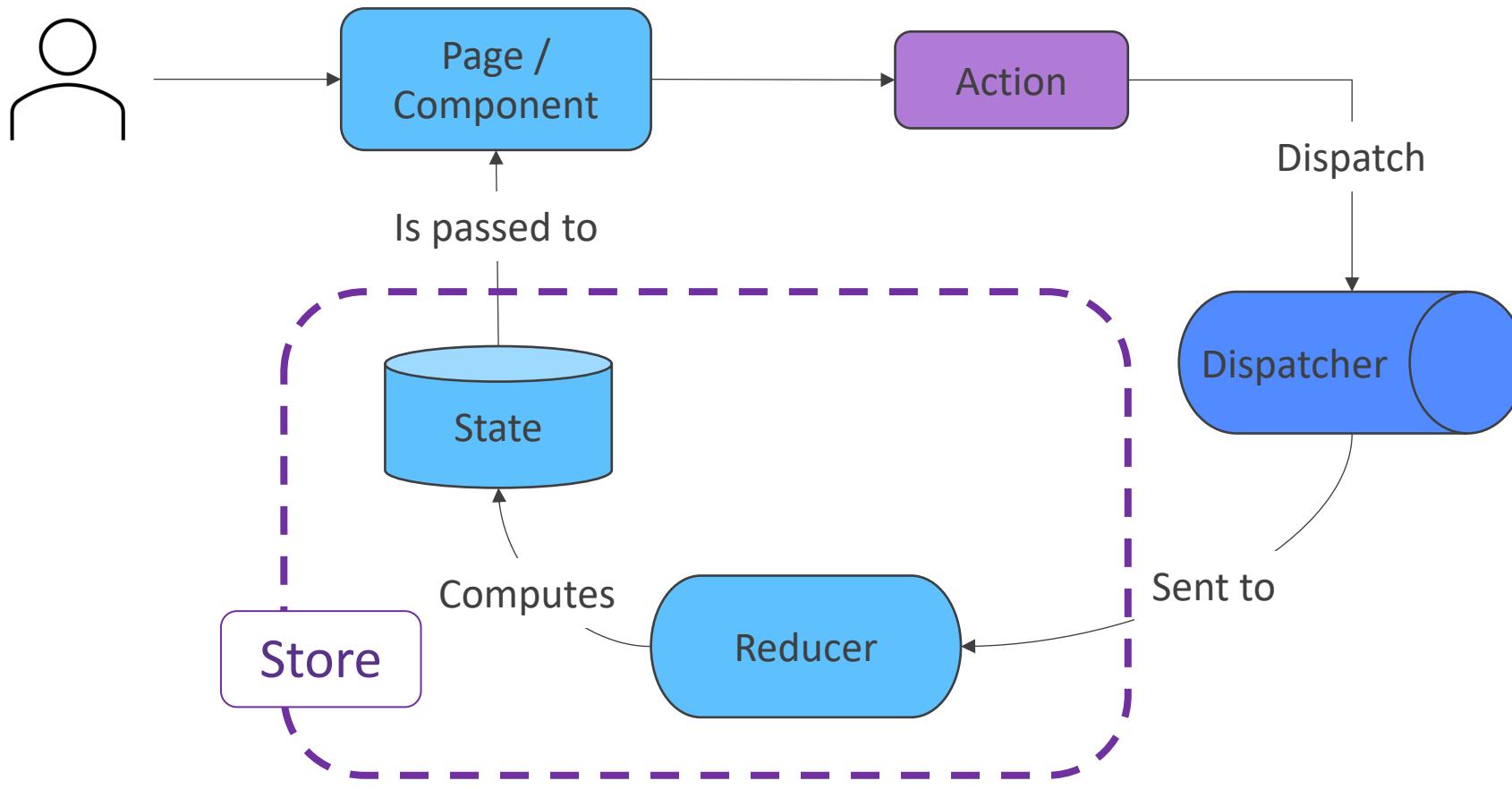
    public void NotifyPropertyChanged(string property)
    {
        PropertyChanged?.Invoke(this, property);
    }
}
```

Far comunicare MAUI e Blazor

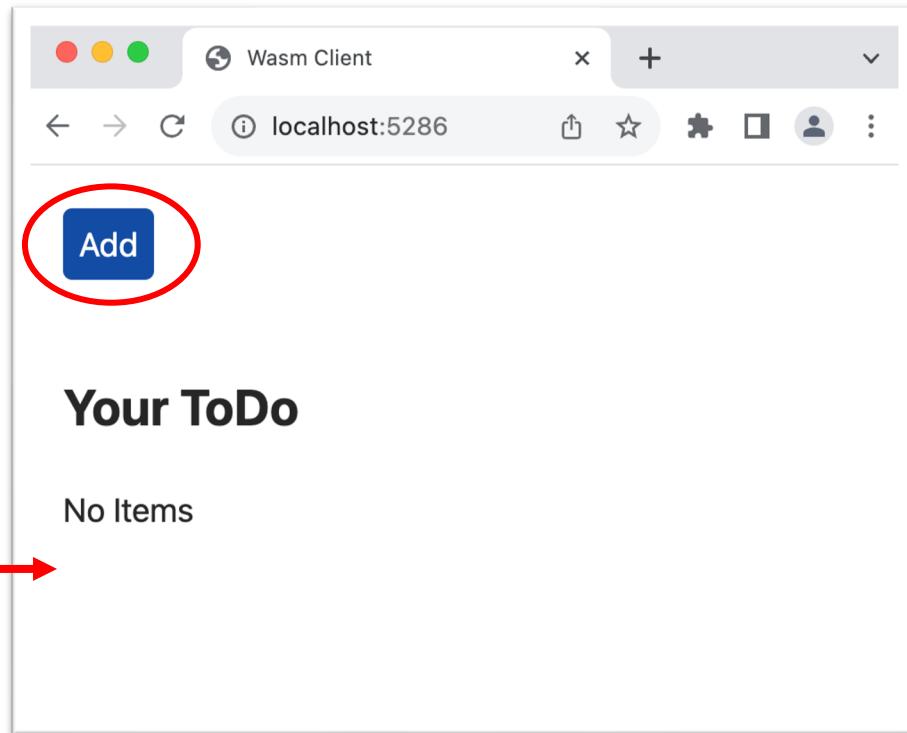
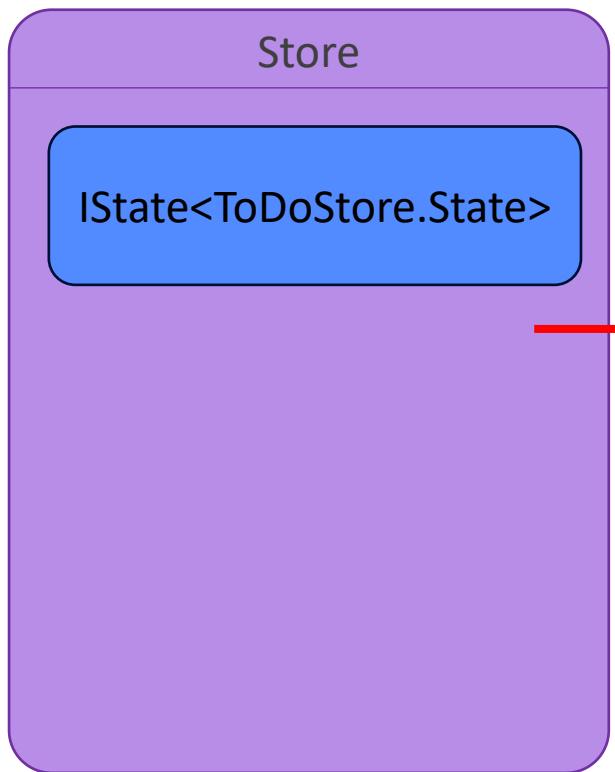


Ci sarebbe una soluzione alternativa ...

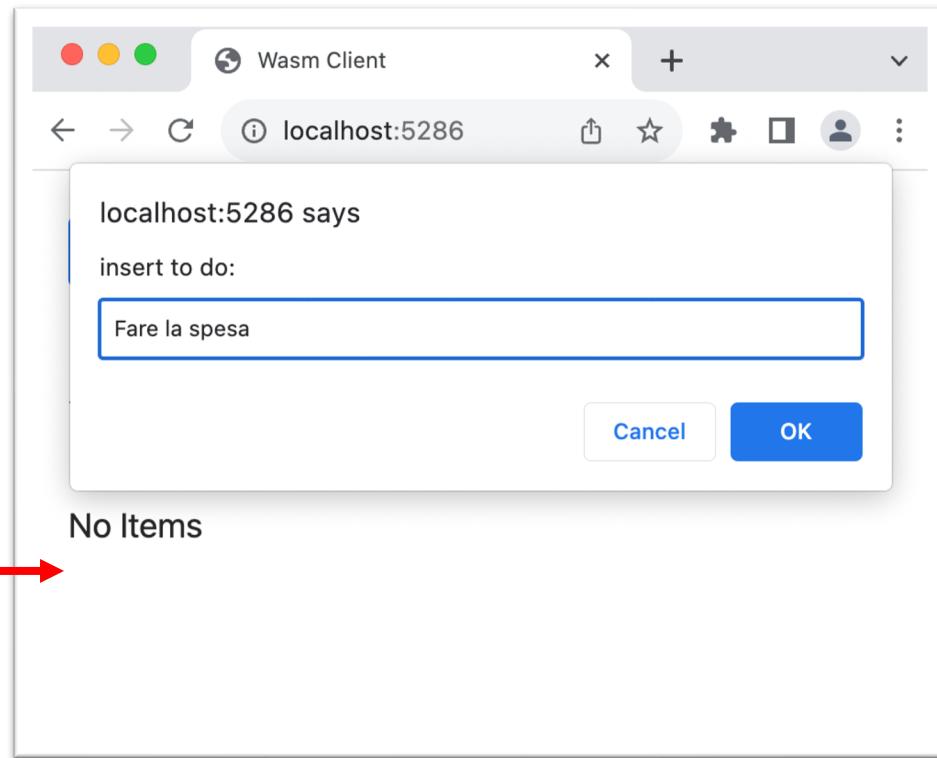
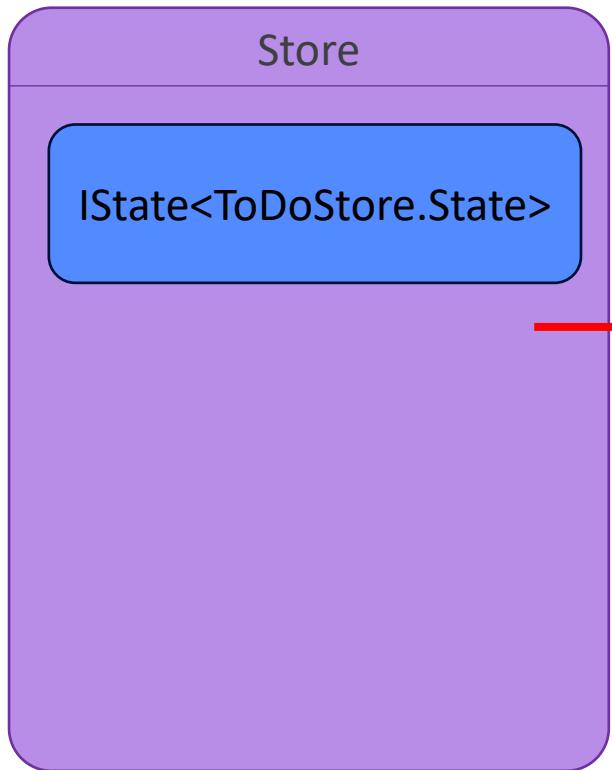
- Se il problema sono i parametri, **eliminiamo i parametri!**
- Potremmo **esternalizzare lo stato dei componenti** in uno **store esterno**, utilizzando il pattern **Flux** con una libreria già pronta chiamata **Fluxor**



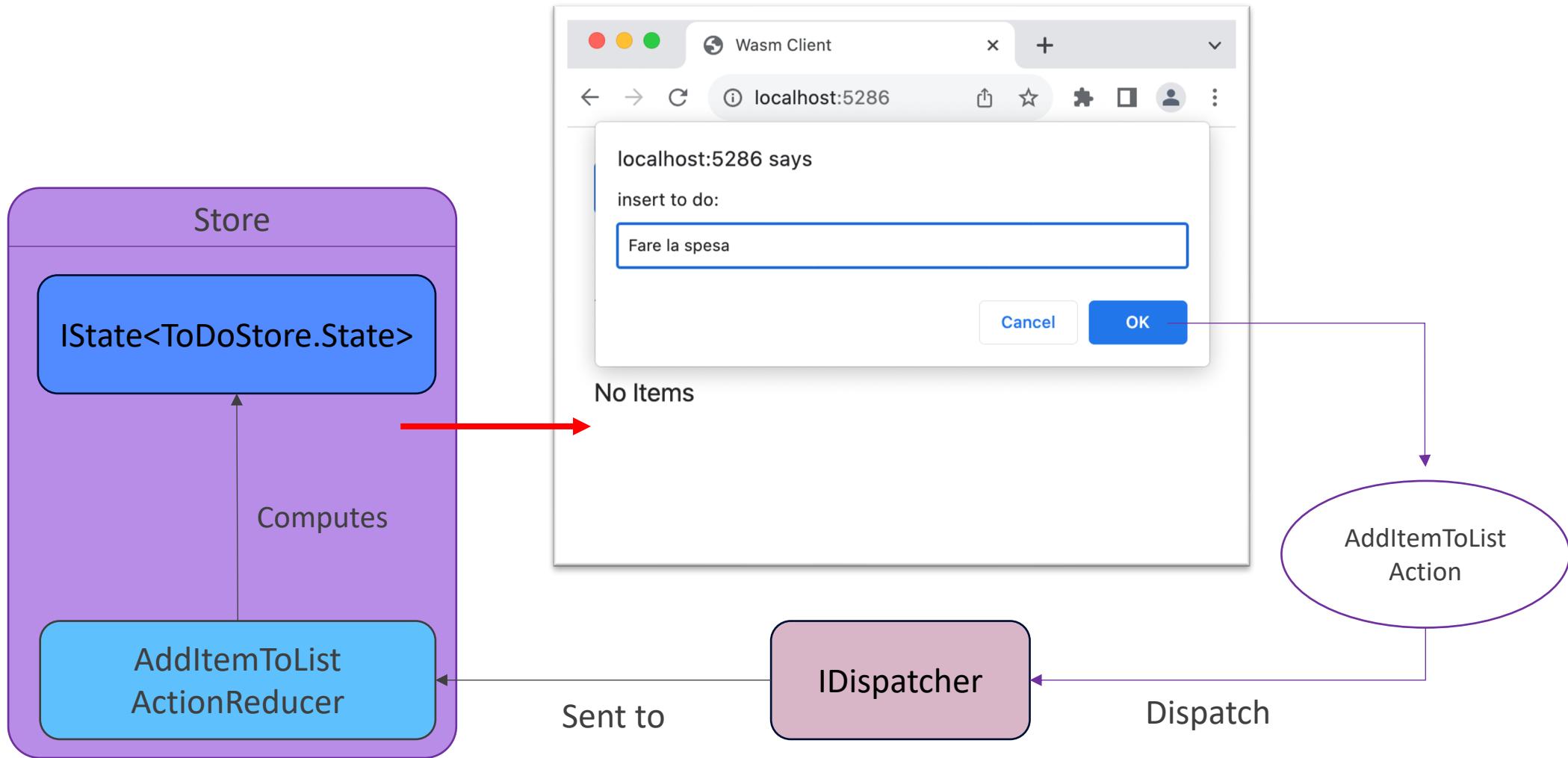
Fluxor nel nostro caso



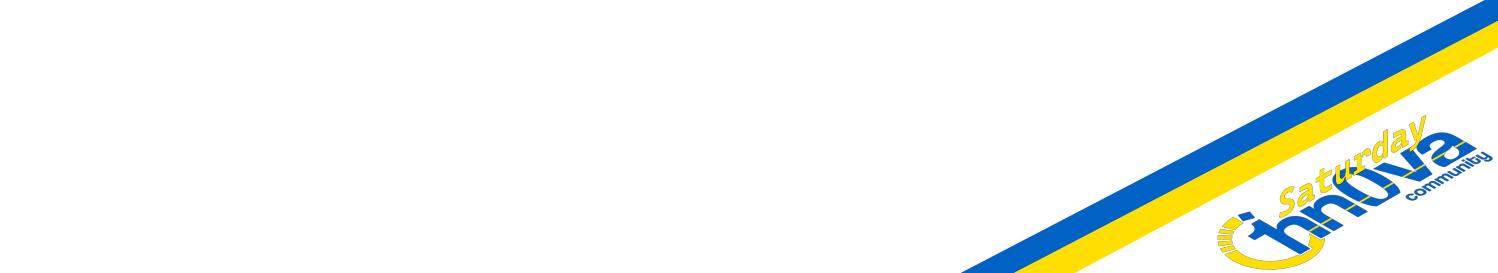
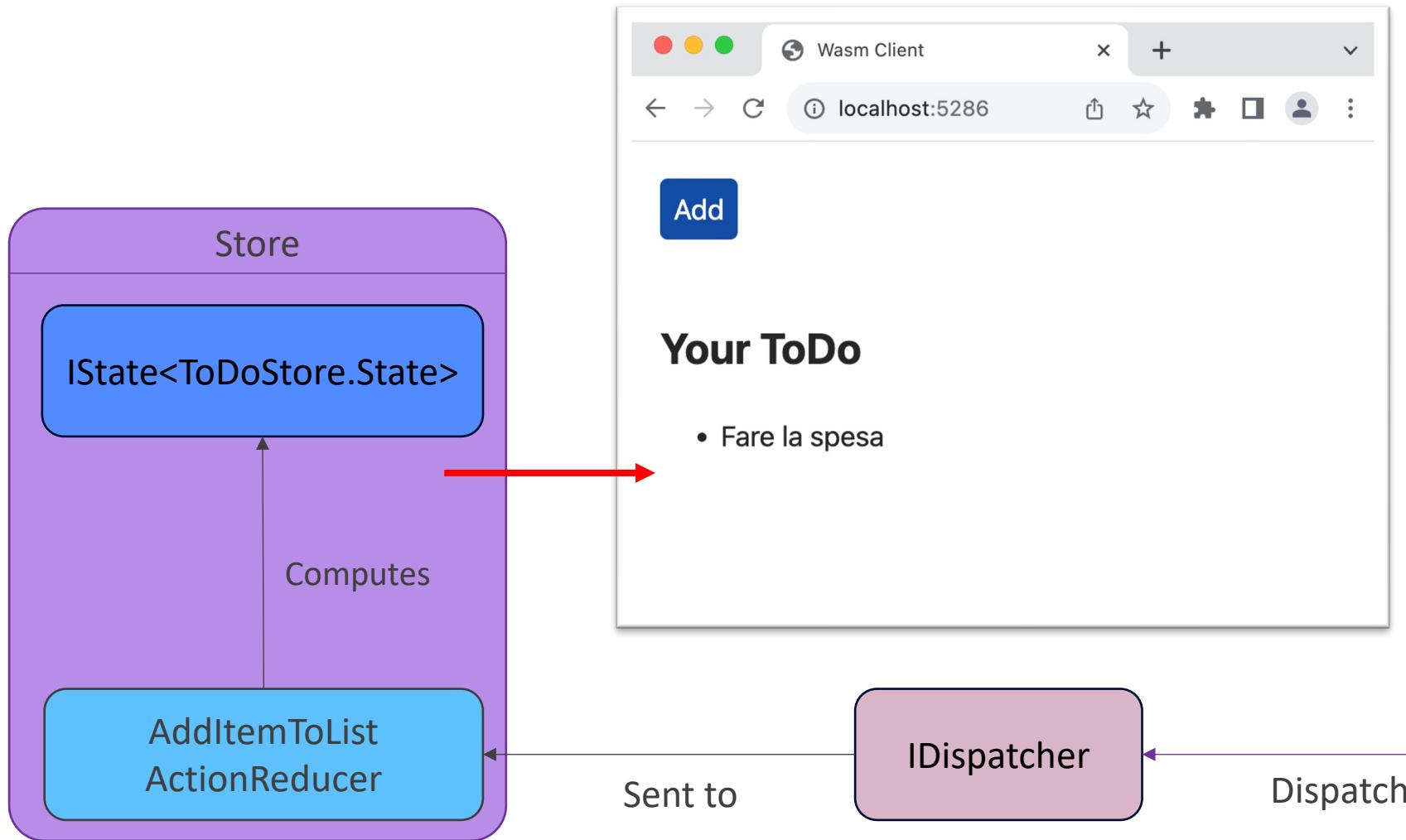
Fluxor nel nostro caso



Fluxor nel nostro caso



Fluxor nel nostro caso



Fluxor: spostiamo la logica della UI



Cosa ci portiamo a casa

- Ha senso utilizzare **Blazor** se già conosco **.NET framework**
- Possiamo **condividere componenti di UI tra diversi client**, usando **Blazor Hybrid** e la potenza della **Dependency Injection**
- Blazor non è **Silverlight**, Microsoft ci sta investendo tantissimo e vale la pena dargli una possibilità
- Valutiamo la possibilità di spostare la logica fuori dai **componenti**, rendendola **riutilizzabile, manutenibile e testabile**



Interessati a Blazor?



**BLAZOR
DEVELOPER
ITALIANI**

Blazor Developer Italiani è la prima
community italiana dedicata allo sviluppo
di applicazioni con **Blazor**

Dove ci incontriamo?

The screenshot shows a web browser window with the URL blazordev.it in the address bar. The page features a large purple header with the Blazor Developer Italiani logo (@ icon) and the text "Blazor Developer Italiani Meetup – Febbraio 2022". Below this, it says "22 Febbraio 2022 alle 18.30: il nostro primo meetup mensile con Dario Benevento e Niccolò Armellini". A "LEGGI DI PIÙ" button is visible at the bottom left of the purple area. The rest of the page has a white background with navigation links like Home, Chi Siamo, Staff, Articoli, Eventi, Video, Sponsor, Contatti, and a search icon.

<https://blazordev.it>

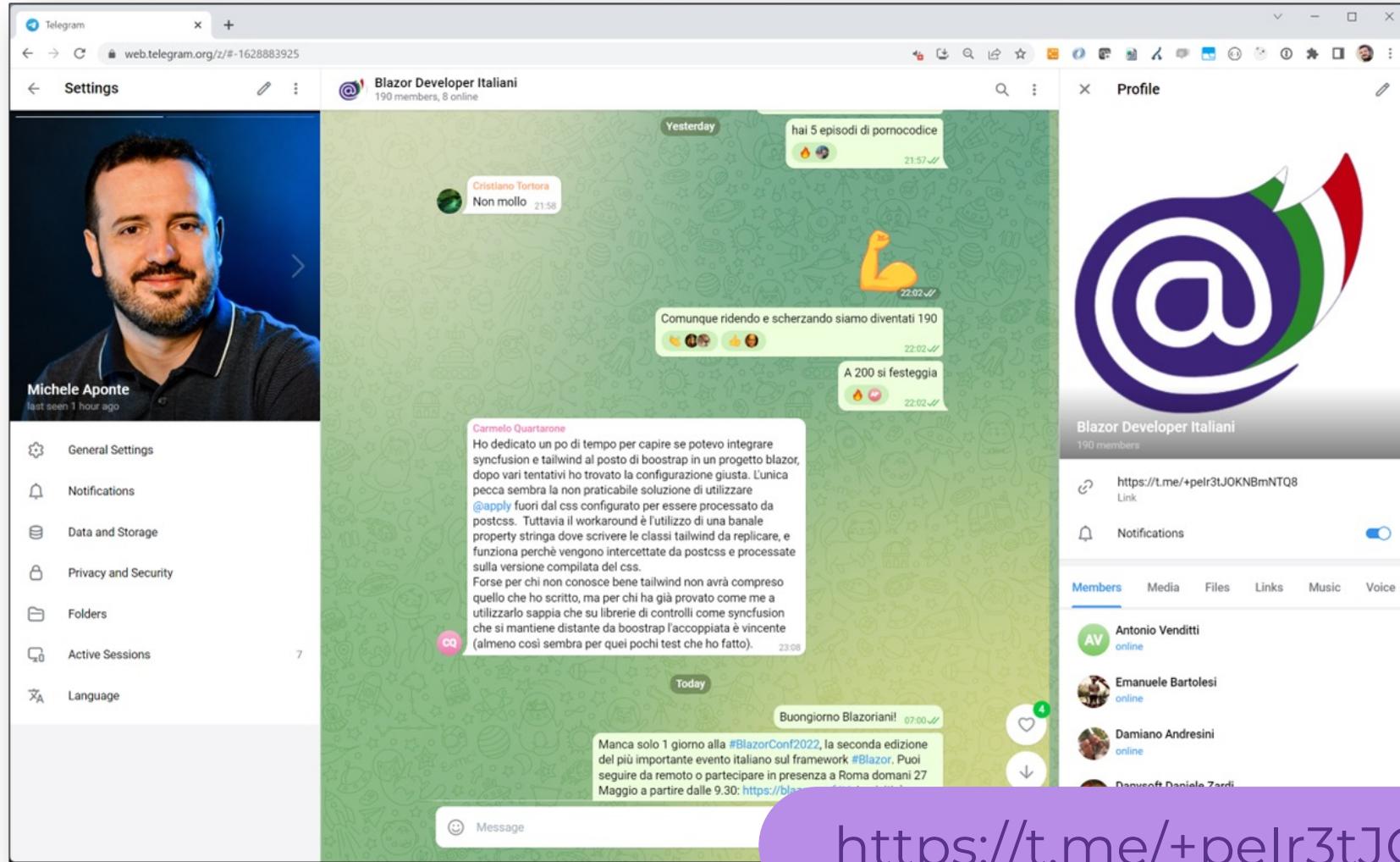


Dove ci incontriamo?

The screenshot shows a Facebook group page titled "BLAZOR DEVELOPER ITALIANI". The page header features the group name in large white letters and the subtitle "La Community italiana su Blazor". Below the header are icons for various web browsers (Chrome, Firefox, Safari) and development frameworks (.NET Core, C#). The left sidebar, titled "Gestisci il gruppo", contains a list of administrative tools and group statistics. The main content area displays the group's profile picture, member count (779), and a recent post by Michele Aponte. The URL of the group page is visible at the bottom of the screenshot.

<https://www.facebook.com/groups/998755440506950>

Dove ci incontriamo?



<https://t.me/+peIr3tJOKNBmNTQ8>

Dove seguirci?

<https://twitter.com/blazordevita>



<https://fb.me/blazordeveloperitaliani>



<https://www.linkedin.com/company/blazor-developer-italiani>



<https://youtu.be/c/BlazorDeveloperItaliani>



<https://www.twitch.tv/blazordevita>



<https://github.com/blazordevita>



Thanks!



Sabato 30 settembre 2023



<Michele Aponte>



apomic80



apomic80



apomic80



apomic80



apomic80

