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

Saturday chNova community

<Michele Aponte>

apomic80 apomic80 apomic80 apomic80 apomic80

Sabato 30 settembre 2023

SPONSORS

altitudo

beanTech
IT moves your business

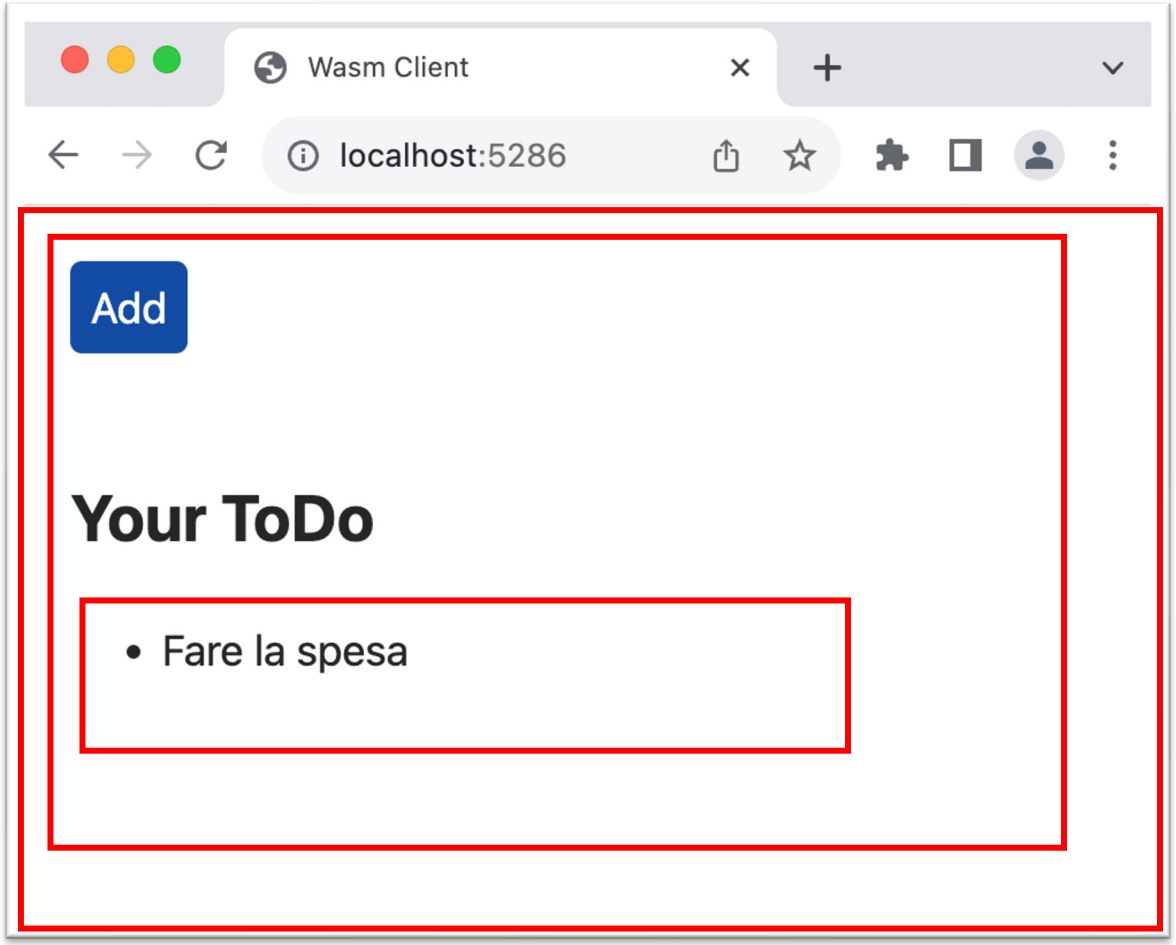
fluentis
the ERP ready to go live!

[stesi]
Powered by Innovation

Chi siete?

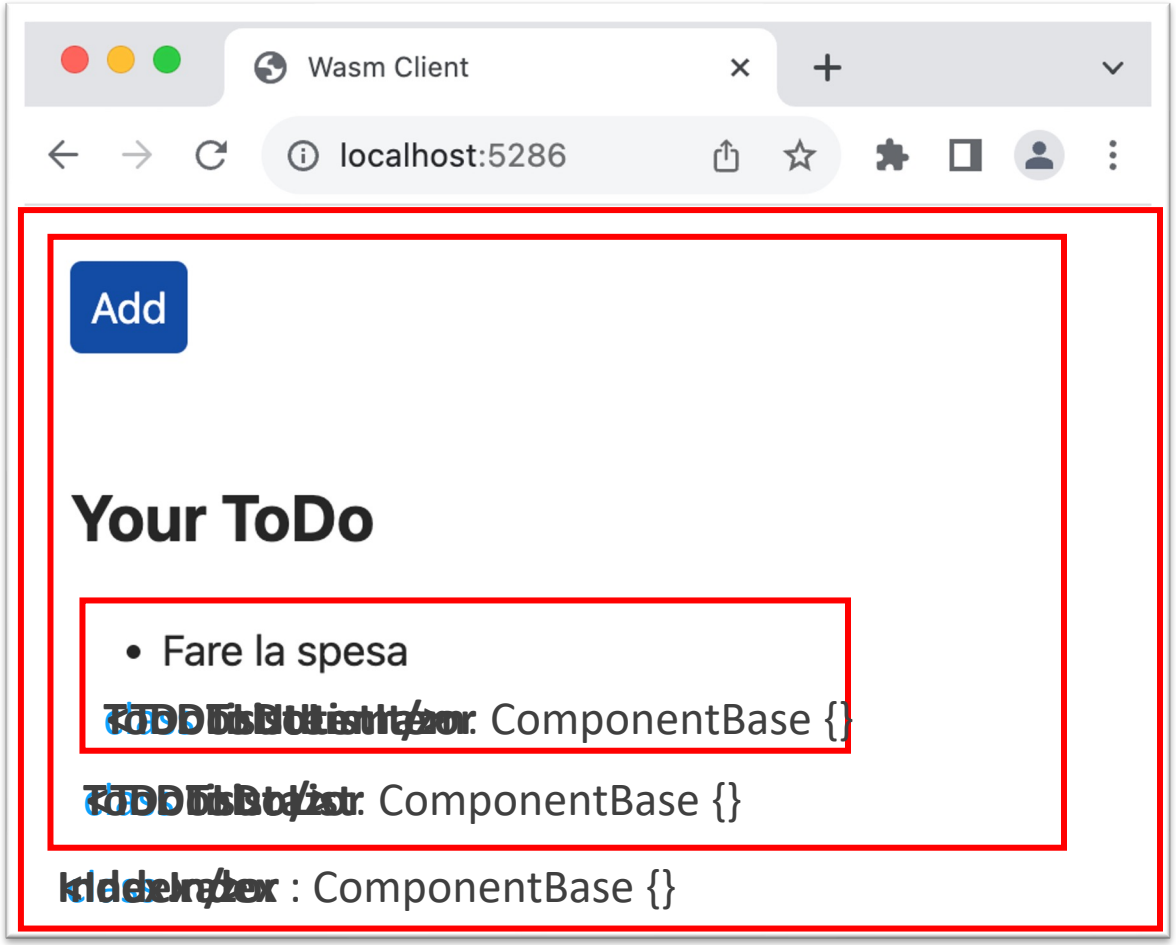
- Quanti programmatori .NET?
- Quanti conoscono Blazor?
- Quanti lo stanno usando in produzione?
- Quanti di voi sanno come funziona, veramente, un componente in Blazor?

Che cos'è, davvero, un componente



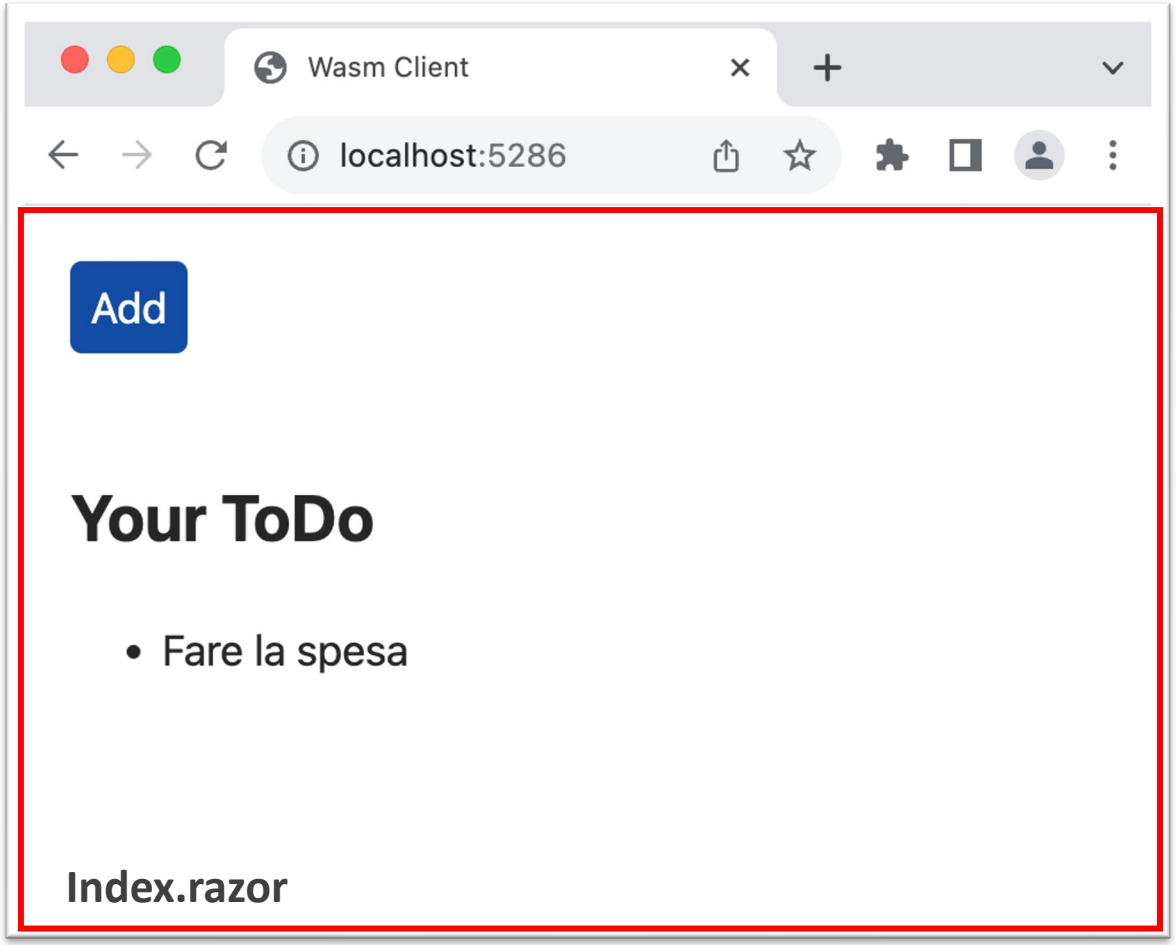
- Un componente è **una porzione della nostra interfaccia**
- Può **contenere** altri componenti
- La sua **responsabilità** è **mostrare** quella porzione di interfaccia e **catturare l'input dell'utente**.
- Deve essere **disaccoppiato** dagli altri componenti
- **Riutilizzabile**

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

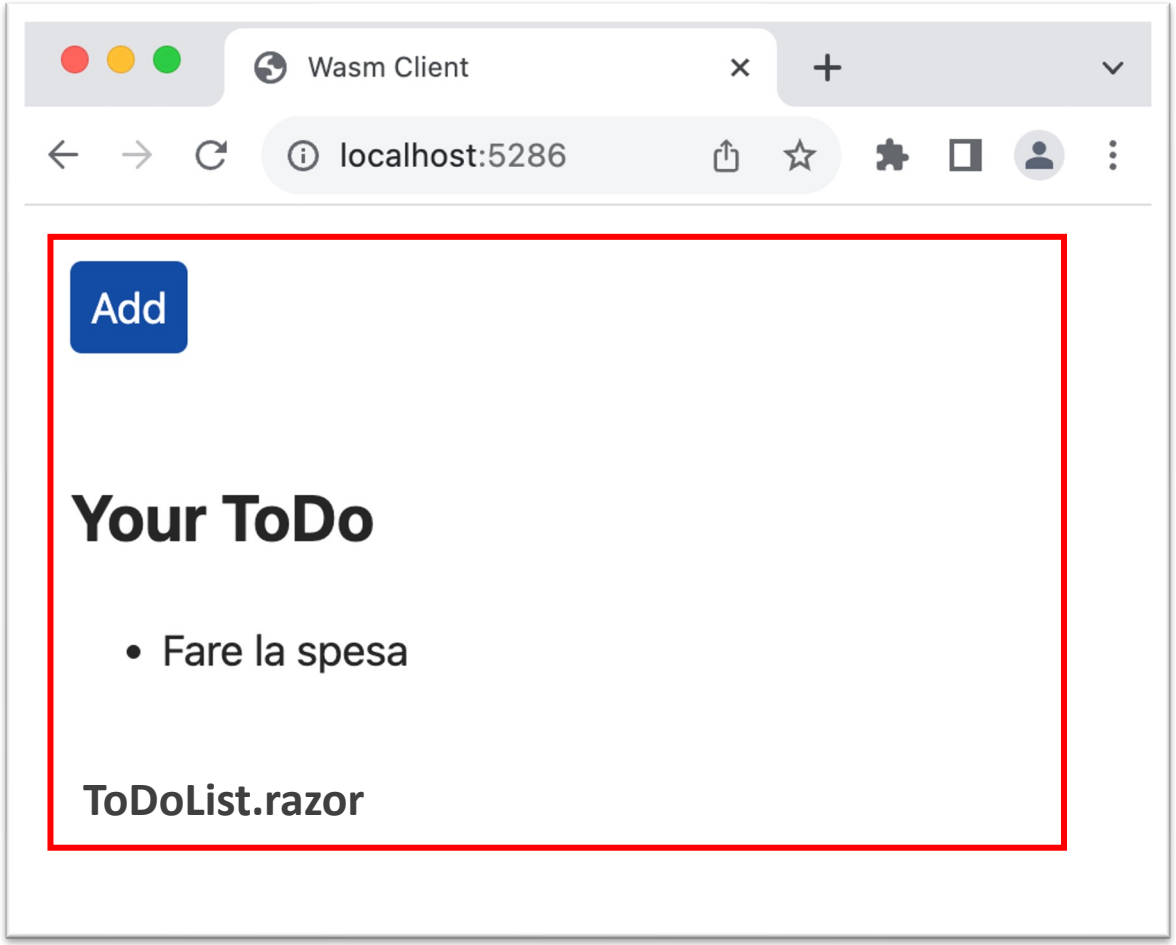
Un componente Blazor generato da un file .razor



Index.razor

```razor
@page "/"

<ToDoList Items="_items"
          OnAddItemRequested="AddItemToList"
          OnDeleteItemRequested="RemoveItemFromList" />

@code
{
    private readonly List<string> _items = new();

    public void RemoveItemFromList(string item)
    {
        _items.Remove(item);
    }

    private void AddItemToList(string item)
    {
        _items.Add(item);
    }
}
```

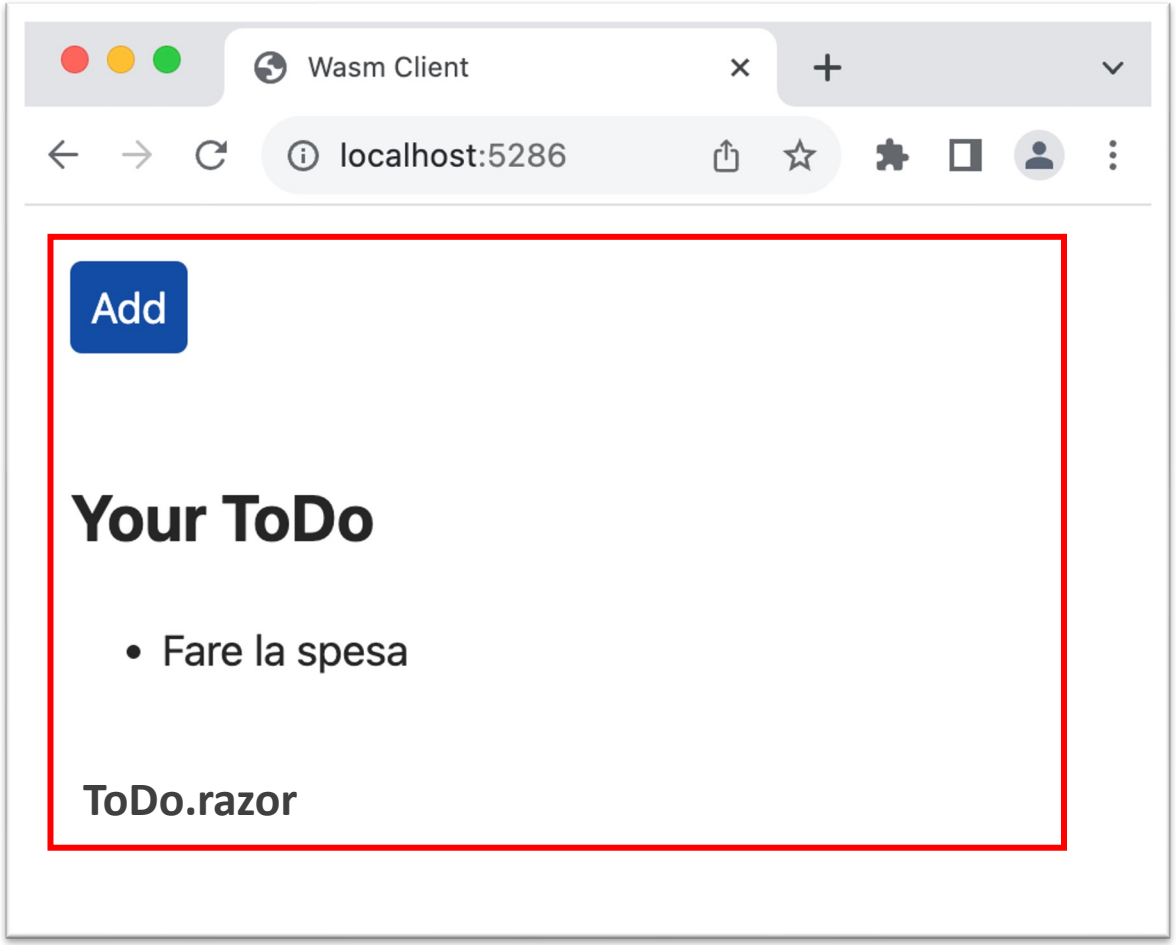
Un componente Blazor generato da un file .razor



ToDoList.razor

```razor
@inject IDataService DataService

<button type="button" @onclick="RequestItemData">Add</button>
<h3>Your ToDo</h3>
@if (Items?.Count > 0)
{
    <ul>
        @foreach (var item:string in Items)
        {
            <ToDoListItem Item="@item"
                          OnItemClicked="OnDeleteItemRequested" />
        }
    </ul>
}
else
{
    <span>No Items</span>
}

@code
{
```

Un componente Blazor generato da un file .razor



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          var item:string = await DataService.RequestData();
34          item = item.Trim();
35          if(!string.IsNullOrEmpty(item))
36              await OnAddItemRequested.InvokeAsync(item);
37      }
38  }
```

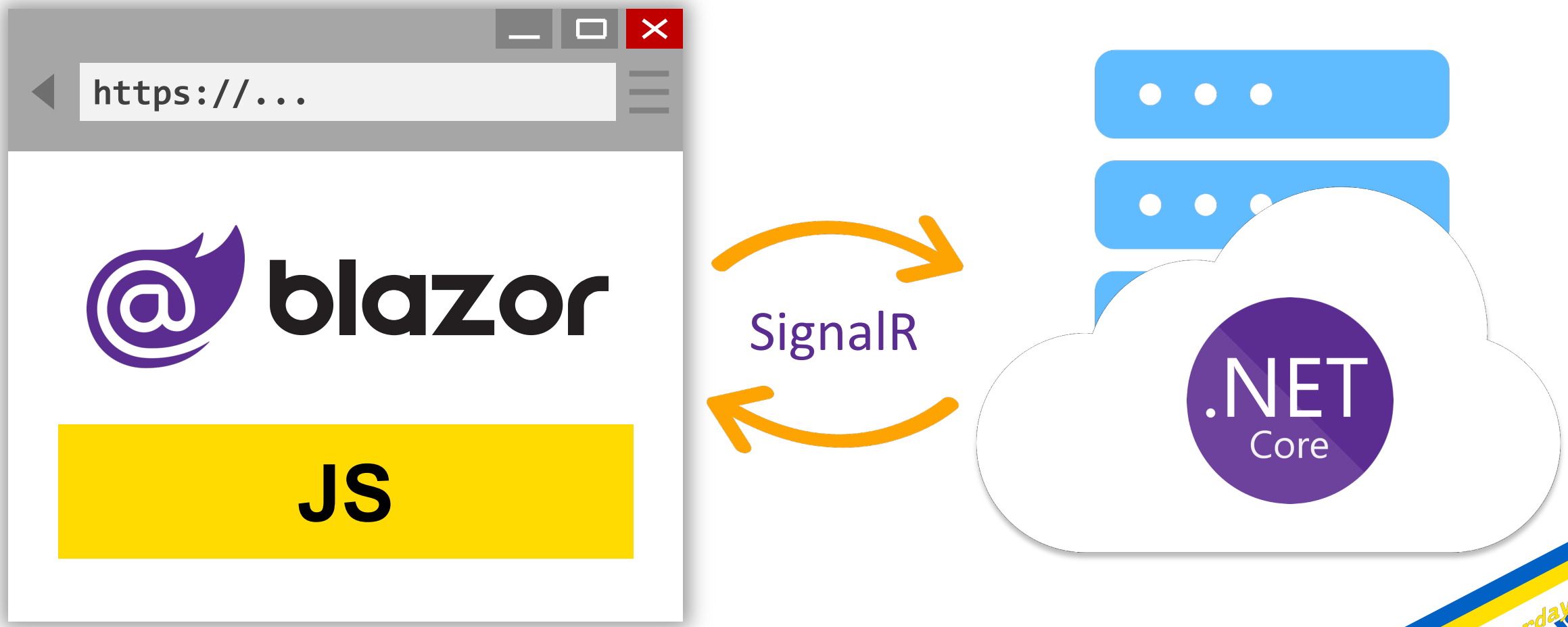
Un componente Blazor generato da un file .razor

ToDoList.razor

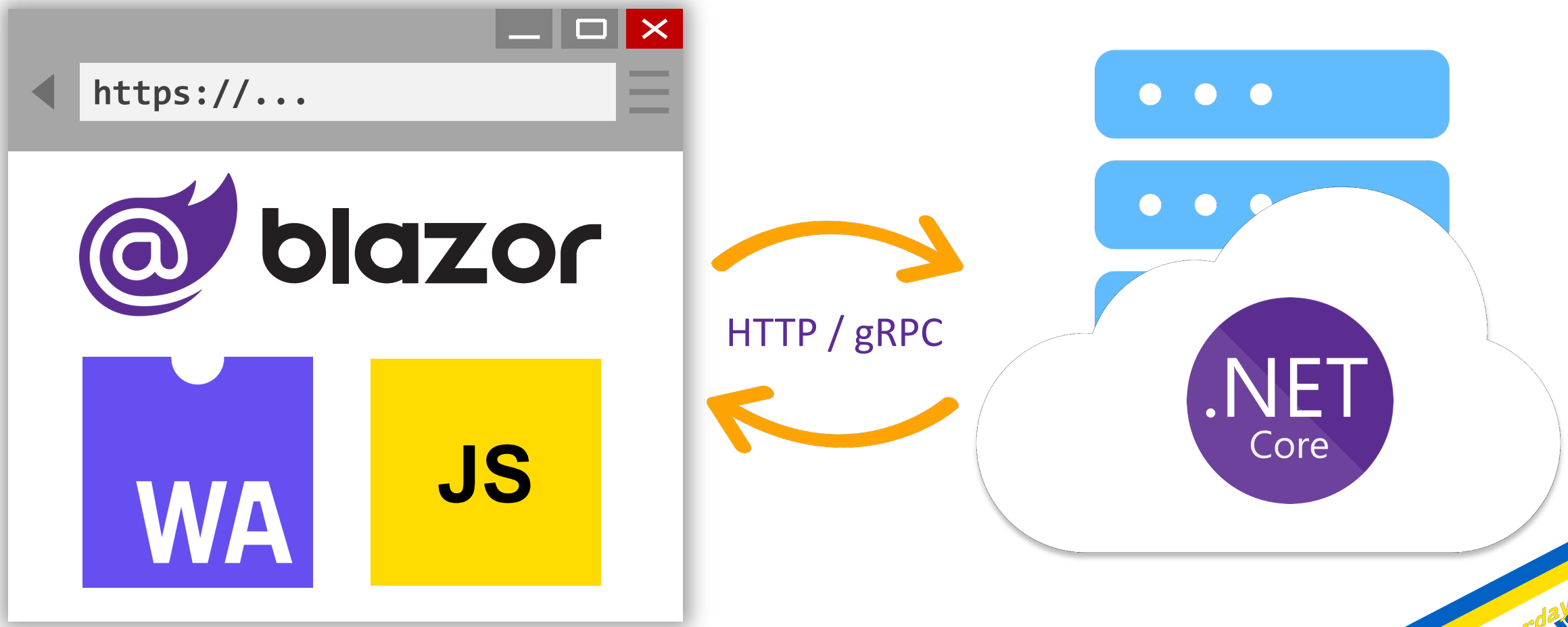
```
20  @code
21  {
22      [Parameter]
23      public List<string>? Items { get; set; }
24
25      [Parameter]
26      public EventCallback<string> OnAddItemRequested { get; set; }
27
28      [Parameter]
29      public EventCallback<string> OnDeleteItemRequested { get; set; }
30
31      private async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

```
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  }
```

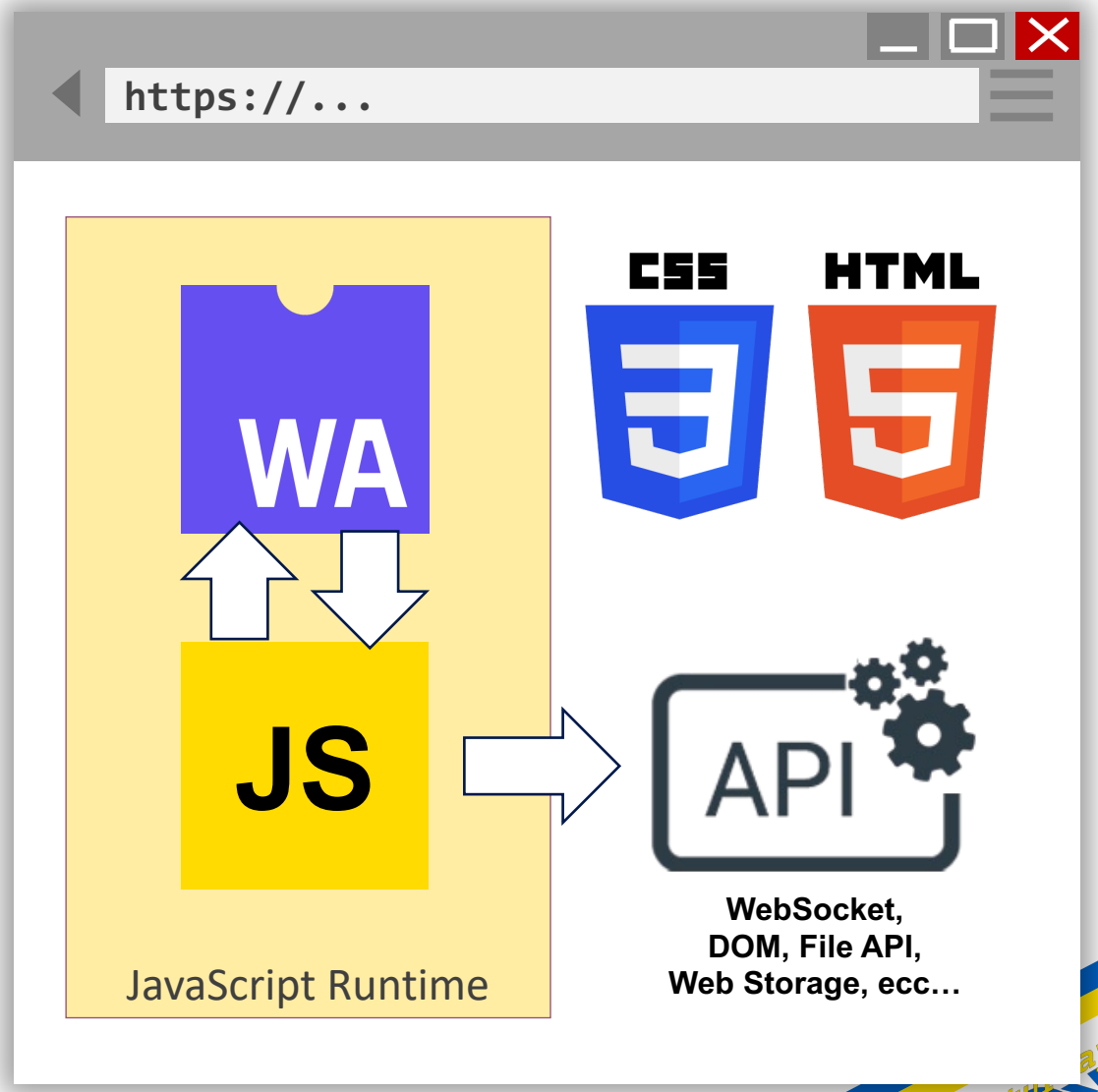
Blazor Server



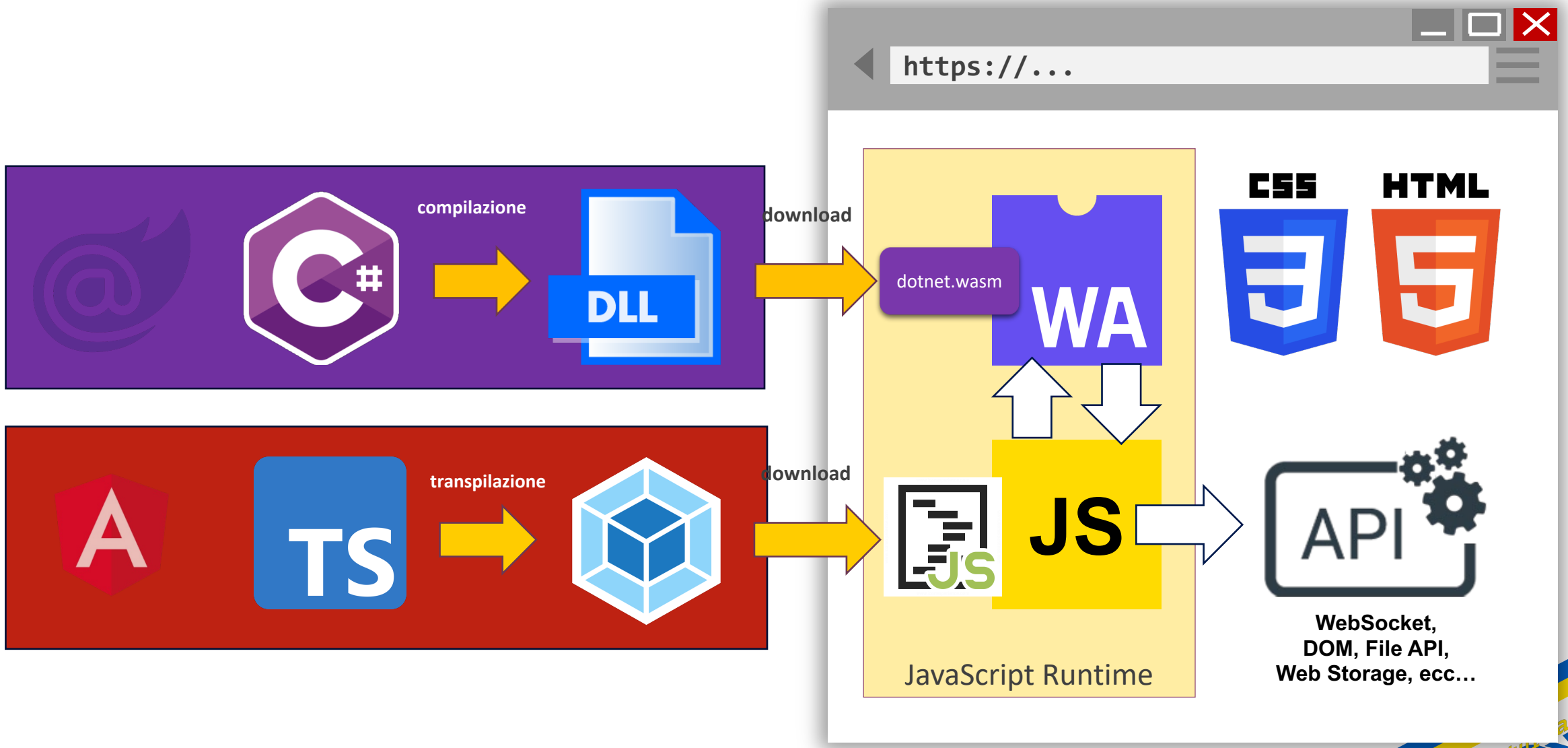
Blazor WebAssembly



WebAssembly



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**
  2. Quando vengono **aggiorn**
  3. Quando viene esplicitam

ToDoList.razor

```
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 ite
9          {
10             <ToDoListItem
11                 OnItemClicked="OnDeleteItemRequested" />
```

Components_ToDoList_razor.g.cs

```
61  public partial class ToDoList : global::Microsoft.AspNetCore.Components.ComponentBase
62  {
63      #pragma warning disable 1998
        IL code
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 (");
```

C# Components_ToDoList_razor.g.cs

```
61  public partial class ToDoList : global::Microsoft.AspNetCore.Components.ComponentBase
62  {
63      #pragma warning disable 1998
        IL code
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

```
322    IL code
       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/

StateHasChanged()

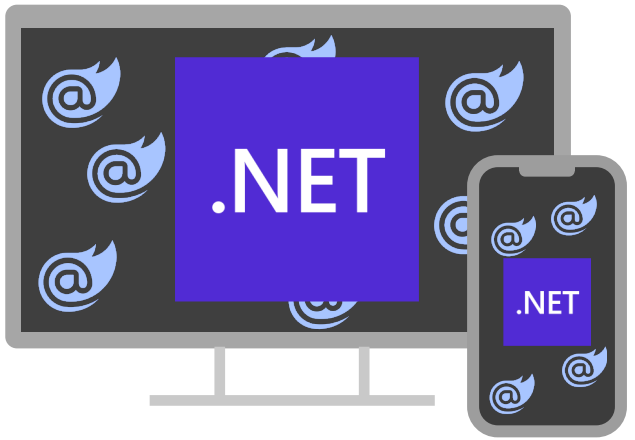
Sabato 30 settembre 2023

Michele Aponte

Blazor Hybrid (.NET 6)

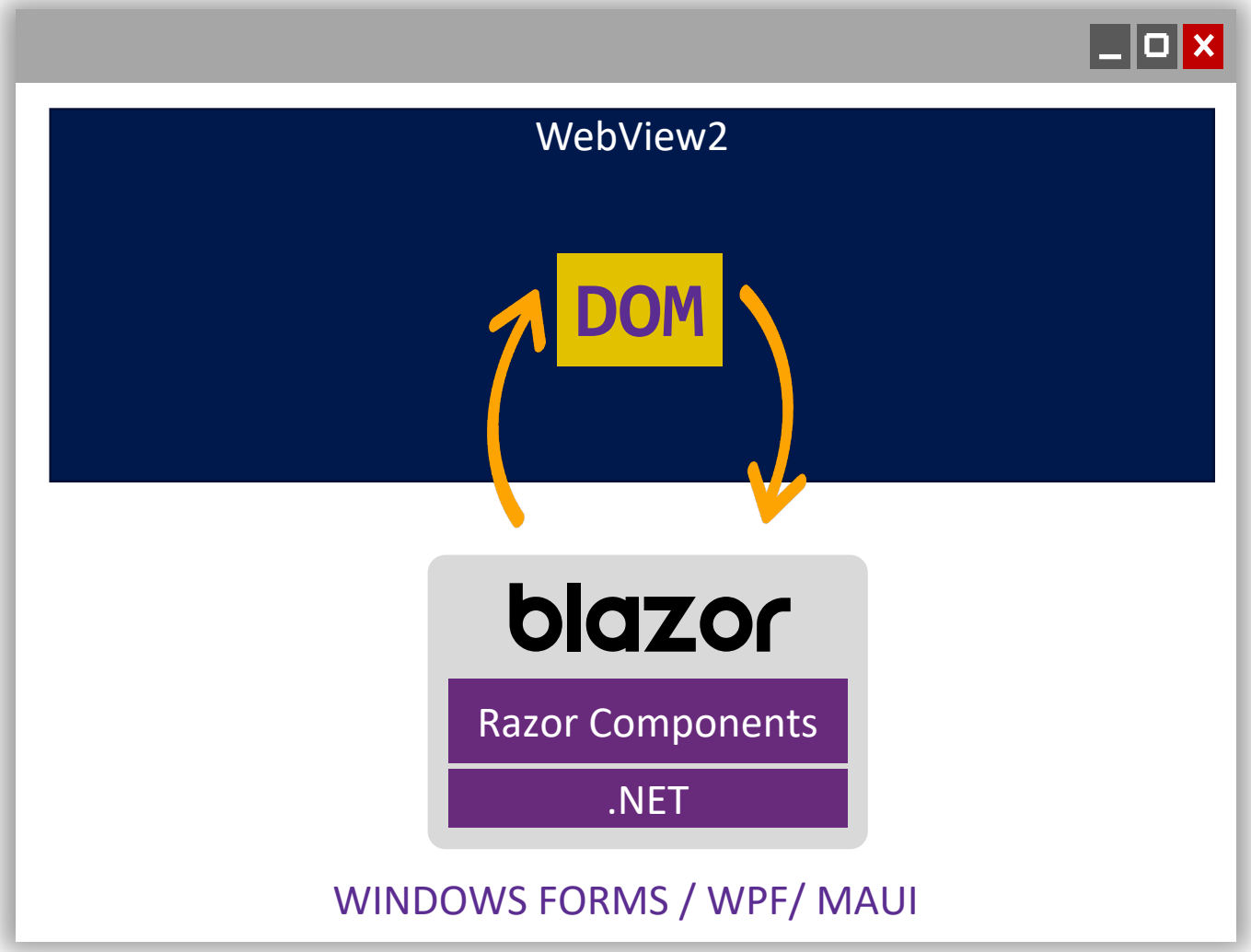
Blazor Server

Blazor WebAssembly

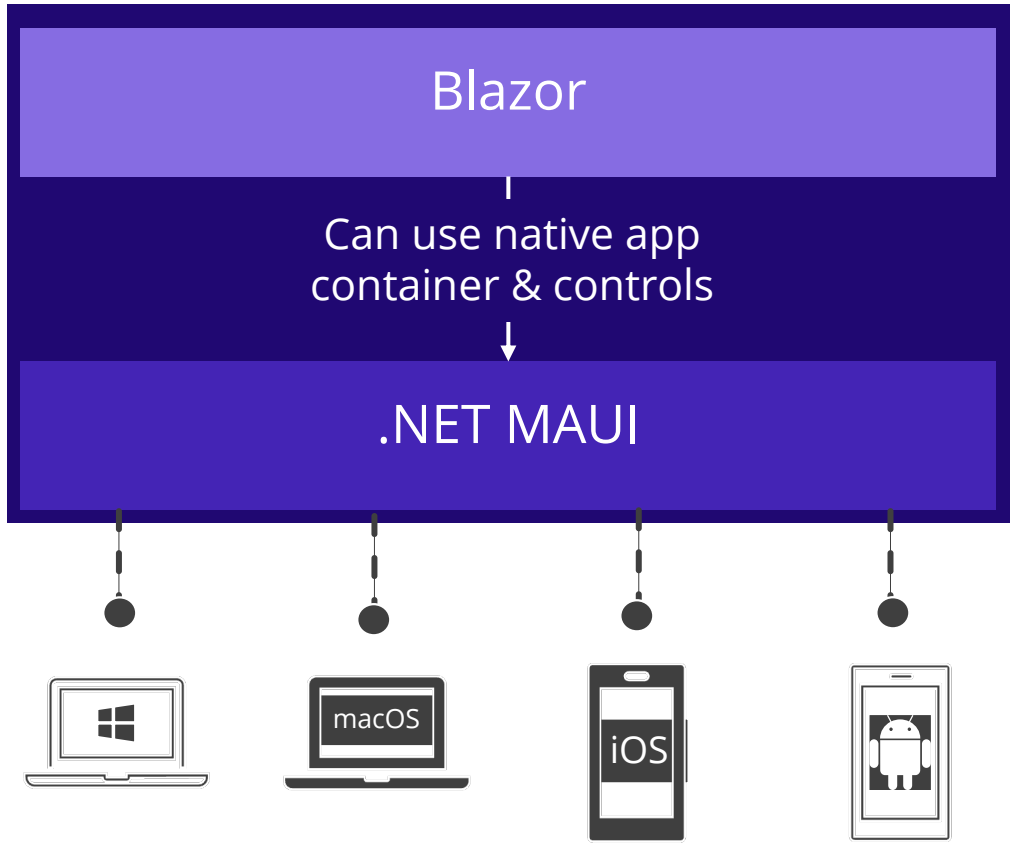


Blazor Hybrid

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

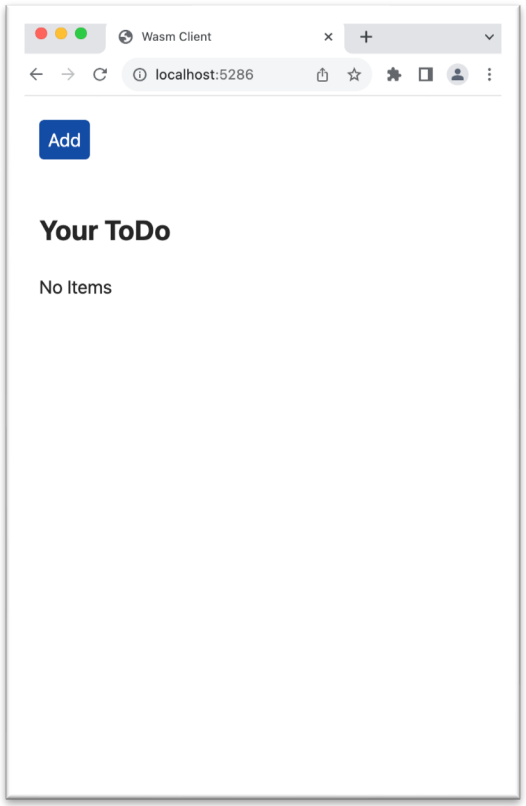
App Desktop, Mobile e Web con Blazor

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

Saturday chNOva community

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 NotifyPropertyChange(string property)
	{
		PropertyChanged?.Invoke(this, property);
	}
}
```

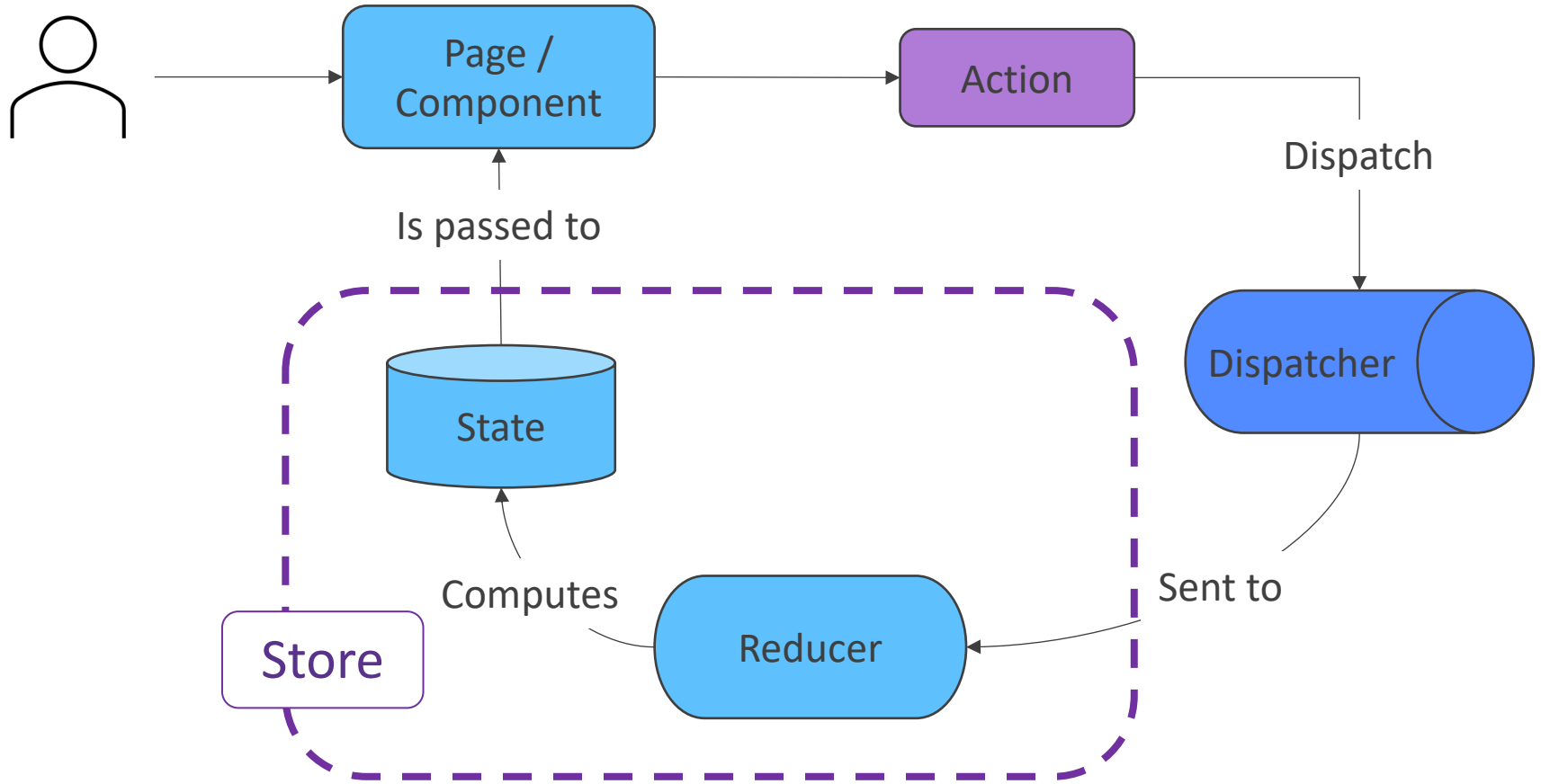
Far comunicare MAUI e Blazor

Sabato 30 settembre 2023

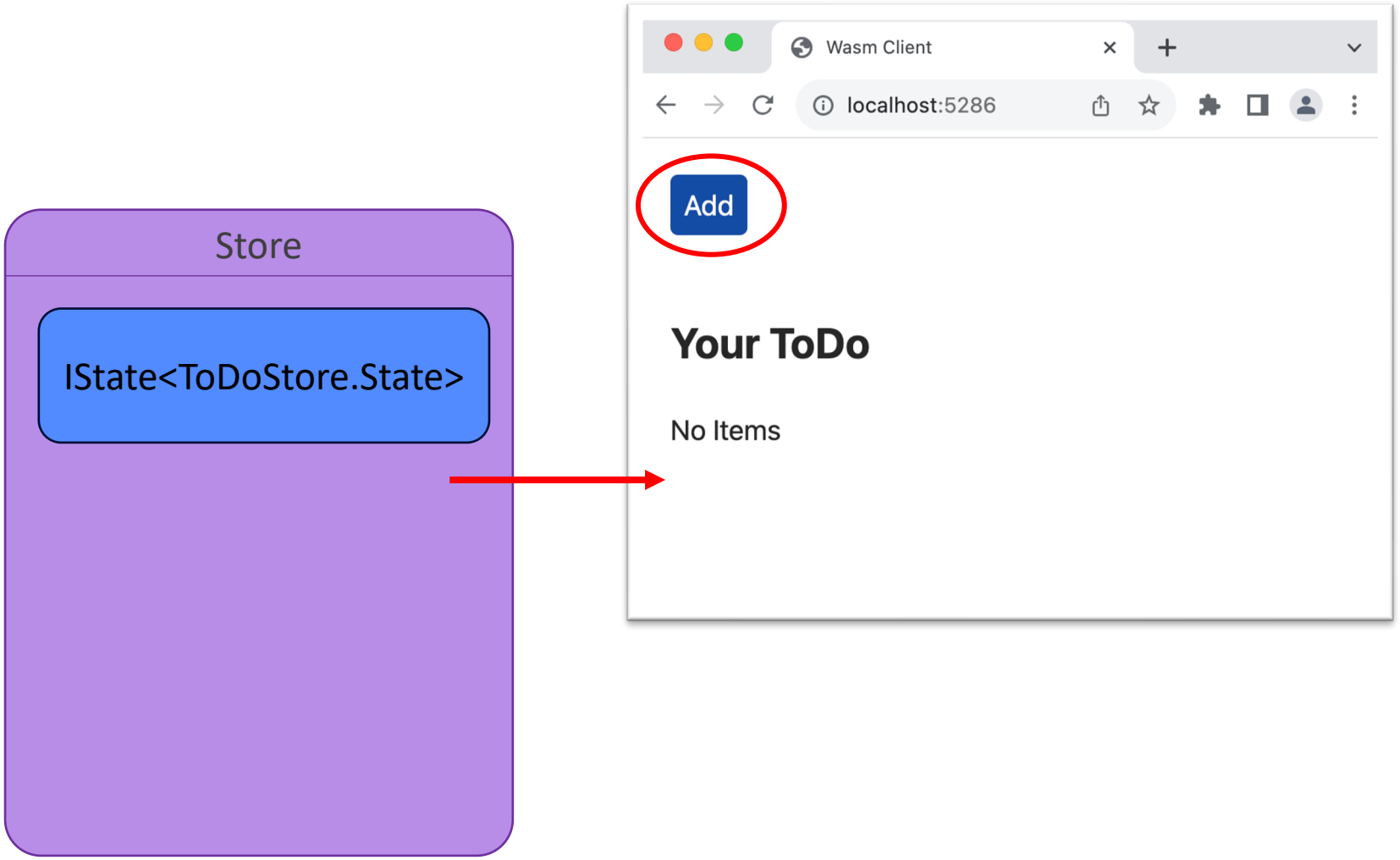
Michele Aponte

Ci sarebbe una soluzione alternativa …

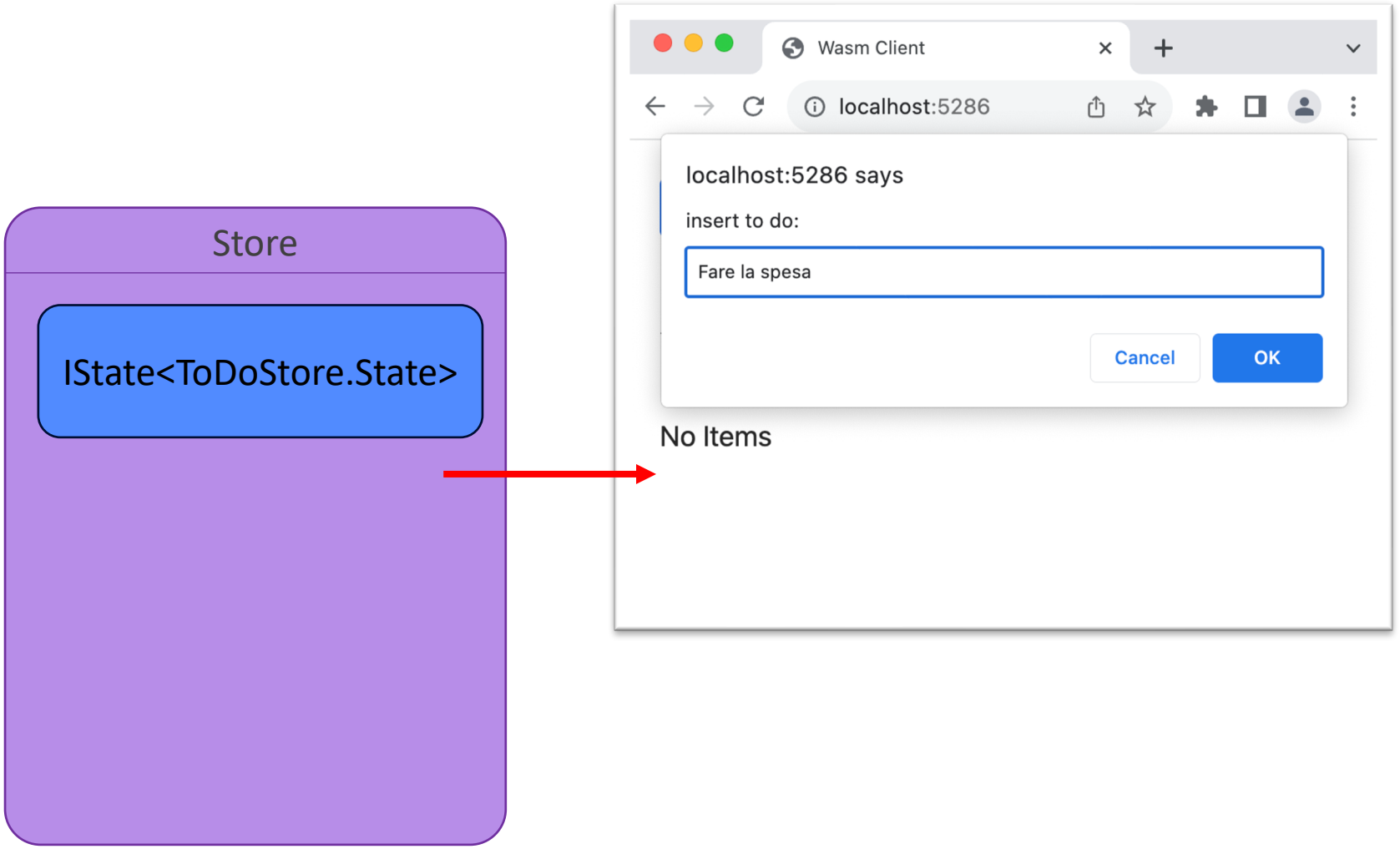
- Se il problema sono i parametri, **eliminiamo i parametri**!
- Potremmo **esternalizzare lo stato dei componenti** in uno **store esterno**, utilizzando il pattern **Flux** con una libreria già pronta chiamata **Fluxor**



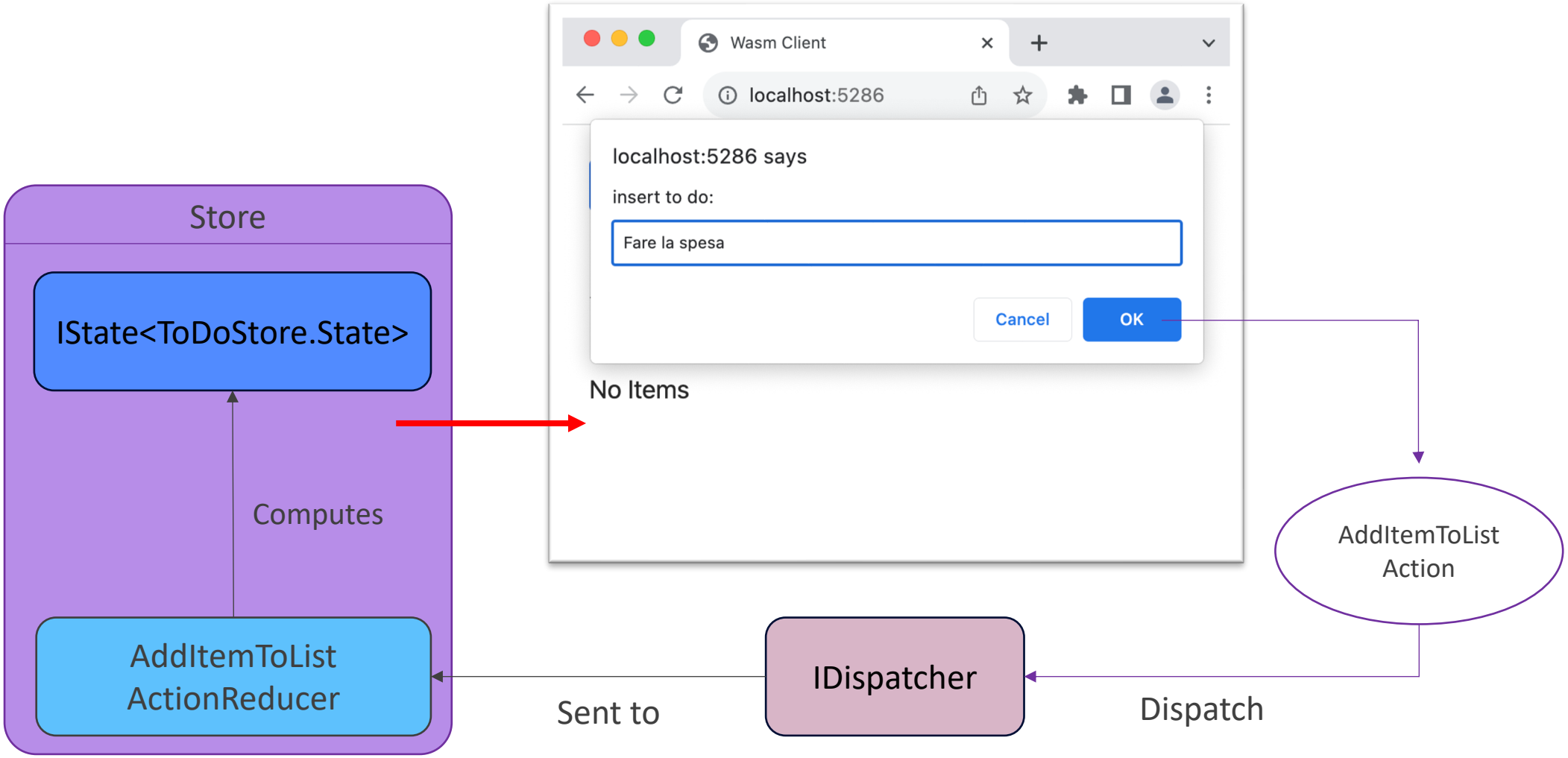
Fluxor nel nostro caso



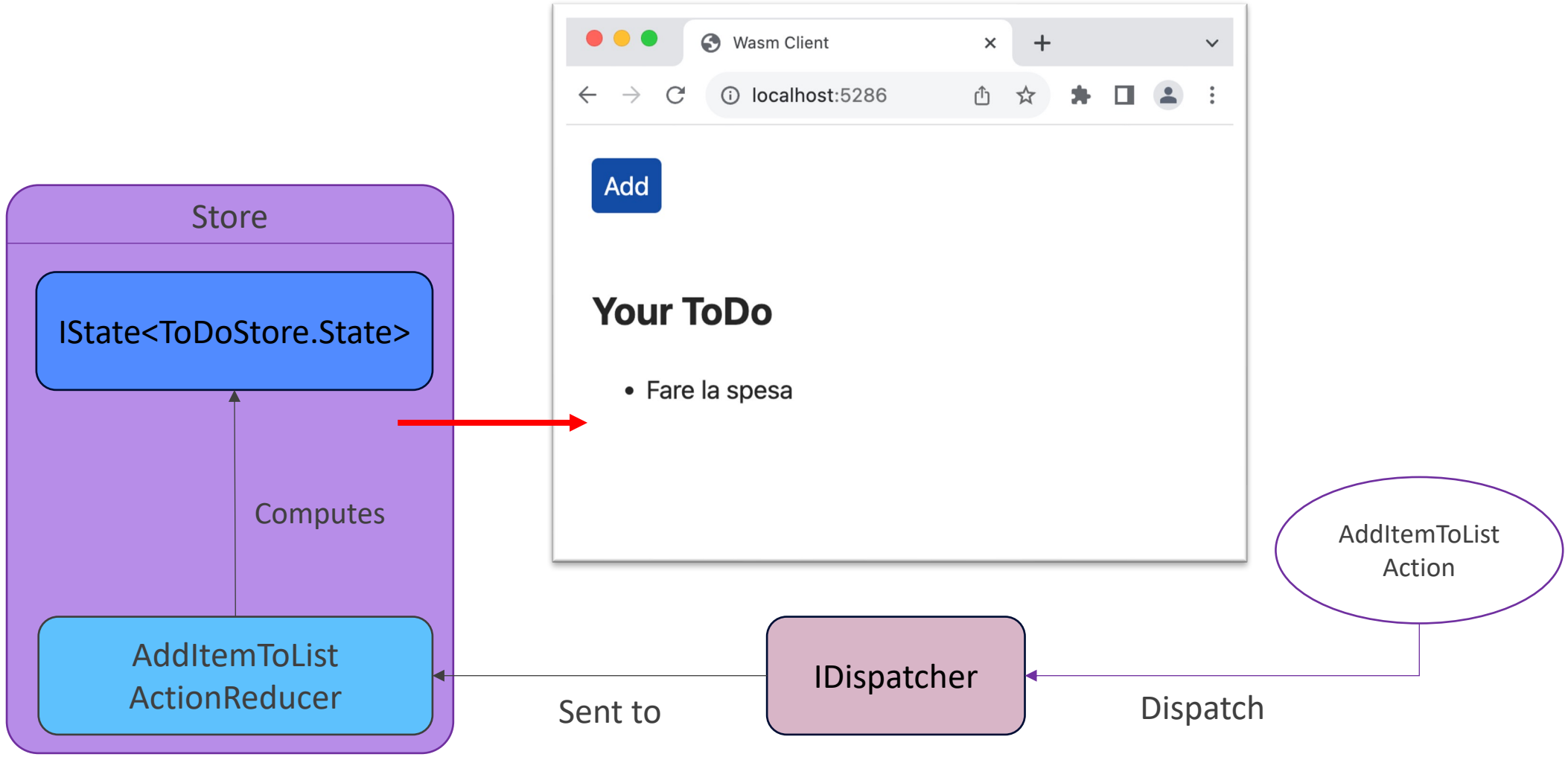
Fluxor nel nostro caso



Fluxor nel nostro caso



Fluxor nel nostro caso



Fluxor: spostiamo la logica della UI

Sabato 30 settembre 2023

Michele Aponte

Cosa ci portiamo a casa

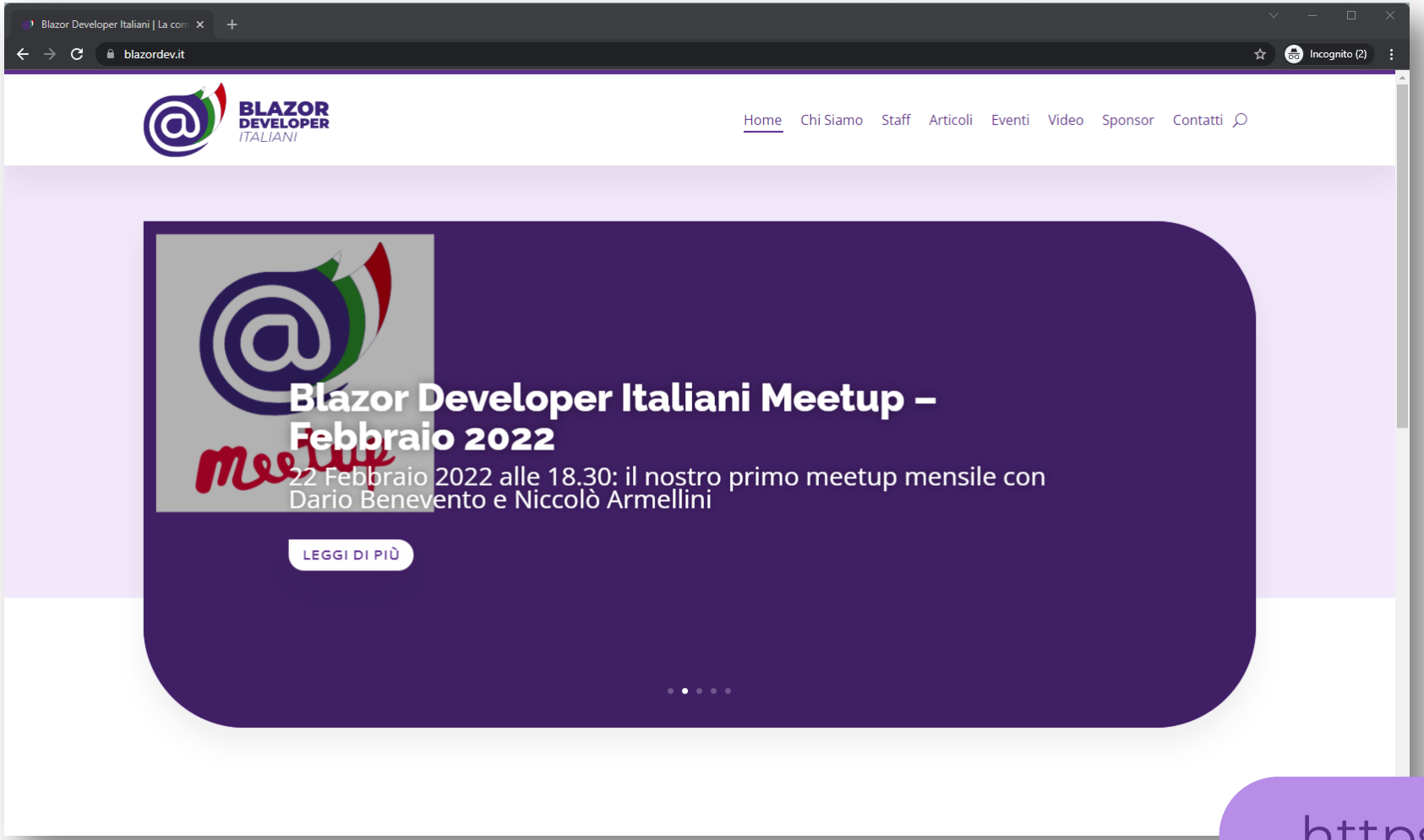
- Ha senso utilizzare **Blazor** se già conosco **.NET framework**
- Possiamo **condividere componenti di UI tra diversi client**, usando **Blazor Hybrid** e la potenza della **Dependency Injection**
- Blazor non è **Silverlight**, Microsoft ci sta investendo tantissimo e vale la pena dargli una possibilità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

Interessati a Blazor?

BLAZOR
DEVELOPER
ITALIANI

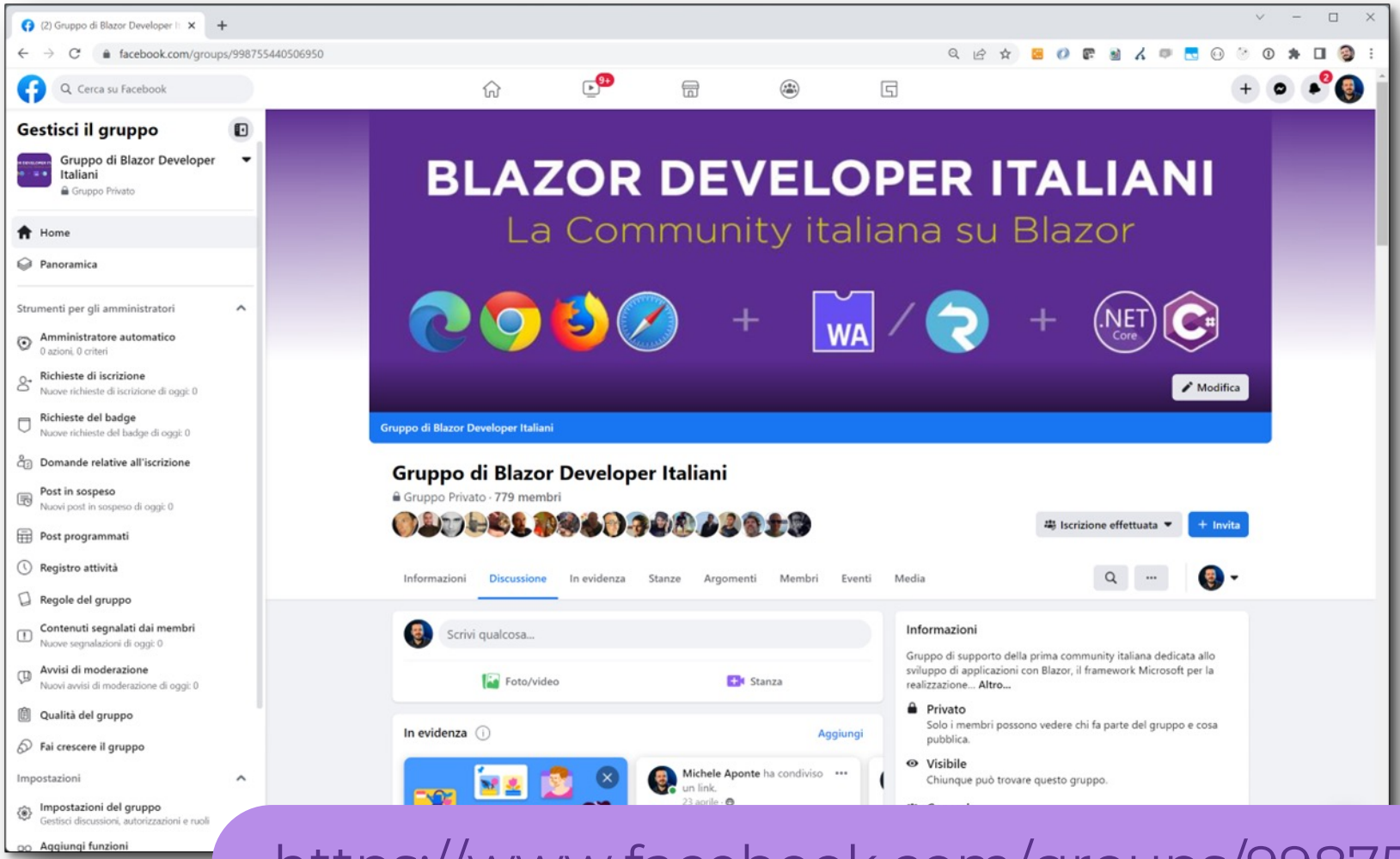
Blazor Developer Italiani è la prima
community italiana dedicata allo sviluppo
di applicazioni con **Blazor**

Dove ci incontriamo?



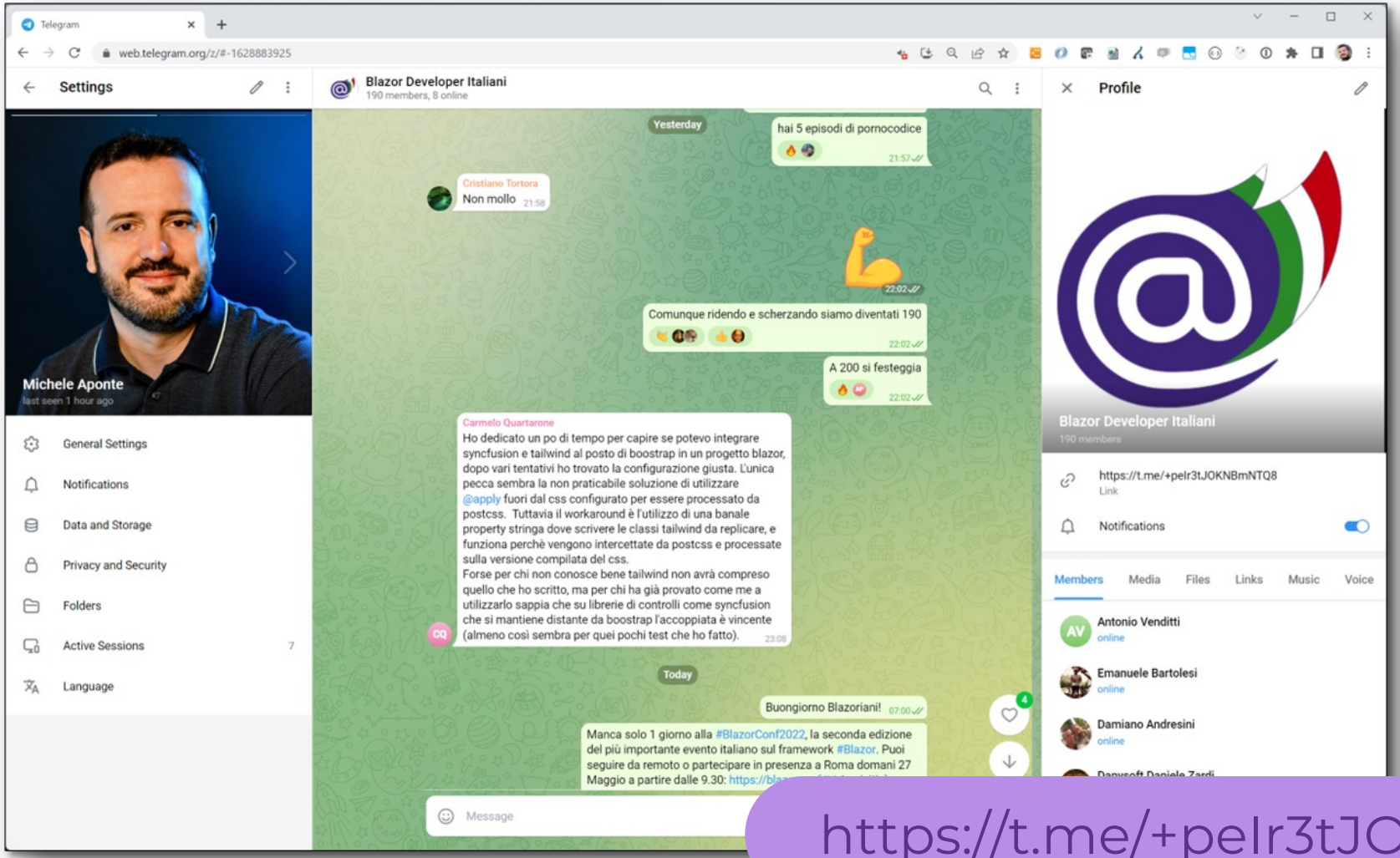
https://blazordev.it

Dove ci incontriamo?



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!

<Michele Aponte>

apomic80 apomic80 apomic80 apomic80 apomic80

Sabato 30 settembre 2023