



Usare Blazor in un caso reale



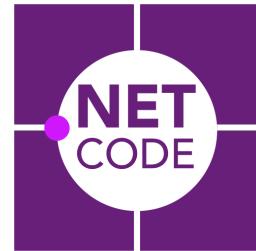
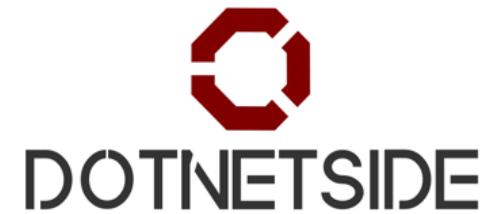
Diego Zunino

Responsabile Area Sviluppo @ Ellycode
diego.zunino@ellycode.com

Un grazie agli sponsor



E alle community che ci hanno supportato

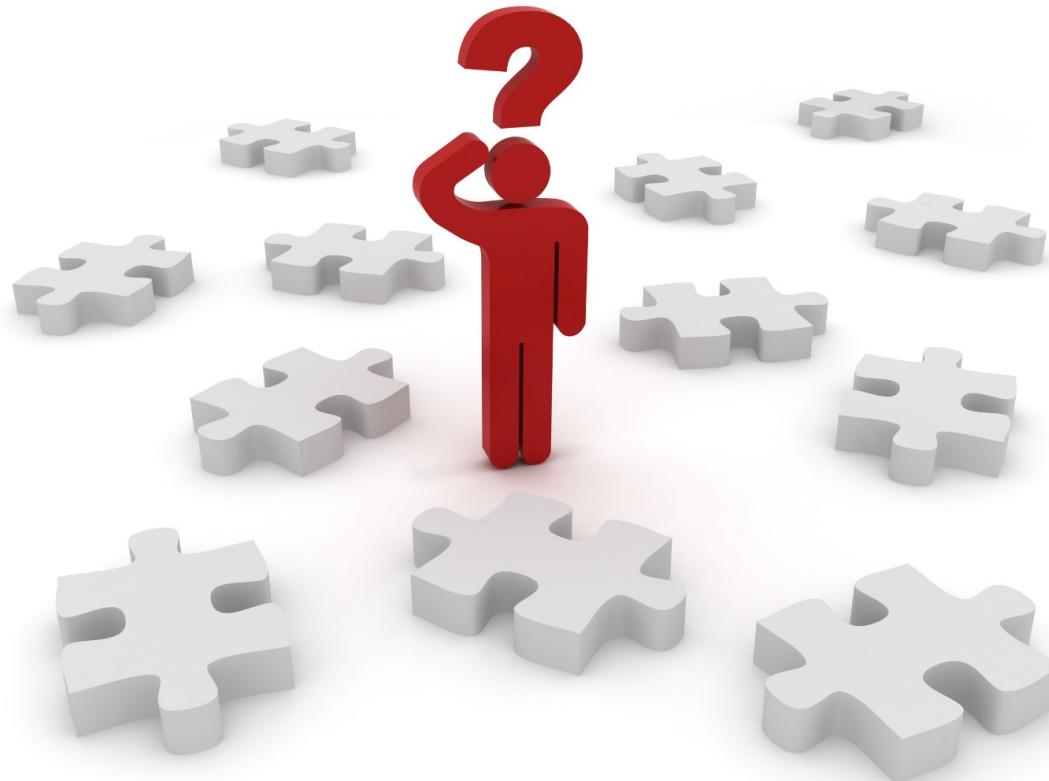




Un problema reale....

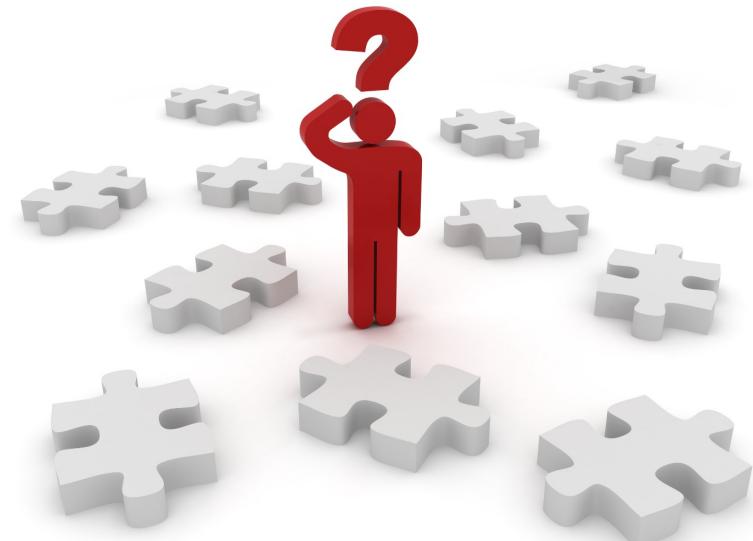


Estrarre i dati necessari a prendere una decisione richiede tempo



Analisi

- Dati **sparsi** in tutta l'azienda
- Estrarre **informazioni** dai dati
- Accesso **istantaneo** ai dati
- Nessun **intermediario**
- **Condivisione** delle informazioni
- **Sicurezza e Privacy**
- Data **Governance**

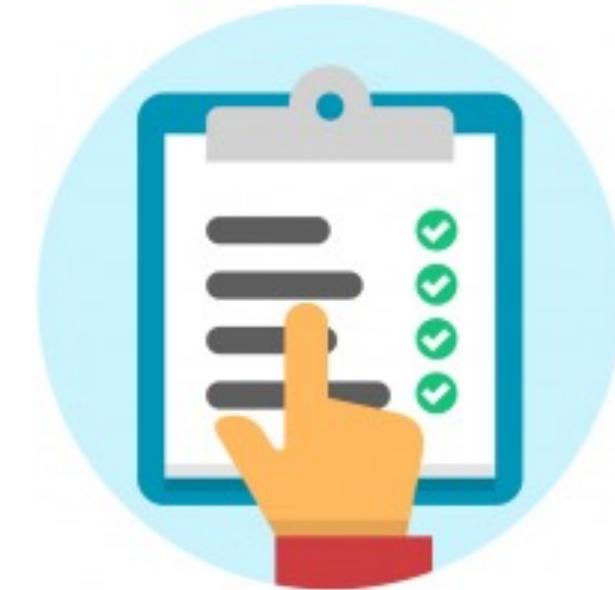


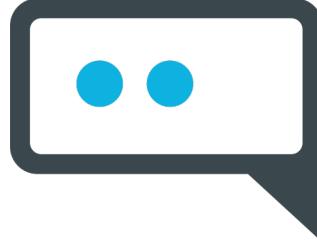
Negli scenari medio grandi ci sono prodotti consolidati



I requisiti

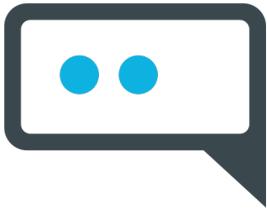
1. Nessuna preparazione di **report**
2. Nessun **linguaggio** di interrogazione dati
3. Utilizzo del **linguaggio naturale**
4. Poter **salvare il risultato**
5. Poter **rappresentare graficamente** il dato
6. Poter **condividere** i risultati ottenuti
7. **Integrazione** con i tool **esistenti**





elly
WHAT DO YOU NEED?

Una piattaforma per l'accesso, la manipolazione, la visualizzazione e la condivisione di dati aziendali utilizzando una User eXperience basata sul riconoscimento del linguaggio naturale



elly

WHAT DO YOU NEED?



Analisi Vendite 2022

Obiettivo fatturato 2022

423905534
84.78%

Articoli scontati

Andamento vendite

Mario Rossi
Ellycode Srl

Andamento vendite

Risultati: 2000

enti fidelizzati

Risultati: 2000

Dipendenti

ID	Nome	Email	Data assunzione	Ferie	Salario annuo	Data fine rapporto
1	Noreen Click	nclick0@storify.com	2019-03-21	true	1360.29	2021-06-26
2	Reeta Stut	rstut1@oracle.com	2021-02-13	true	2705.88	2022-02-24
3	Shelby Brodest	sbrodest2@unblog.fr	2022-02-26	true	2325.08	2021-06-17
4	Frannie Whyler	fwhyler3@fema.gov	2020-05-13	false	1255.82	2021-10-11
5	Nikkie Kinge	nkinge4@ihg.com	2019-03-19	false	2709.81	2021-11-07
6	Aluin McGunley	amcgunley5@cbslocal.com	2017-11-21	false	1841.36	
7	Oliy Anney	oanney6@symantec.com	2018-01-30	true	1643.66	2021-11-09
8	Giraldo Theodoris	gtheodoris7@istockphoto.com	2020-10-25	false	1278.67	2021-09-01

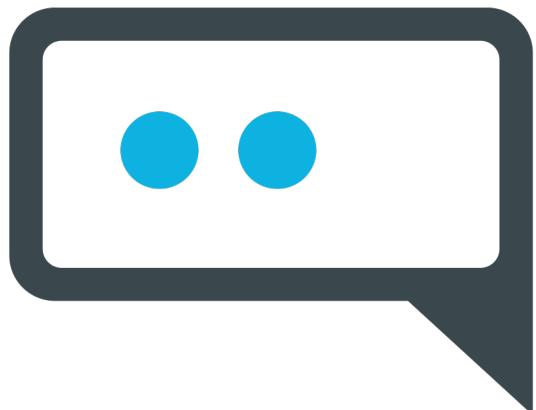
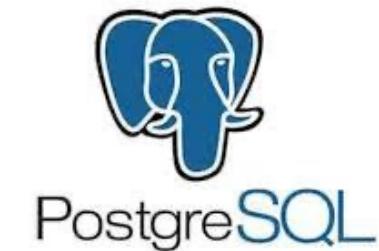
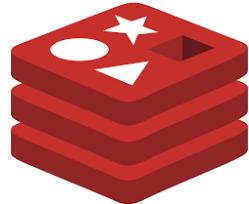
Risultati: 2000

Vendite

ID	Id cliente	Nome oggetto	Prezzo	Quantita	Tipologia	Scontato	Data
2	cust2	Fish - Soup Base, Bouillon	88	1	vendita	False	2021-07-27T05:42:57Z
4	cust1	Marjoram - Fresh	59	2	vendita	False	2021-03-01T01:40:54Z
9	cust9	Creme De Menth - White	52	3	vendita	False	2021-08-18T20:12:62Z
6	cust1	Beets - Mini Golden	34	3	vendita	True	2021-03-26T03:51:06Z
8	cust8	Nescafe - Frothy French Vanilla	64	4	vendita	False	2021-01-26T16:08:52Z
5	cust2	Fond - Chocolate	58	4	vendita	False	2021-10-16T10:11:01Z
10	cust10	Salad Dressing	28	5	vendita	True	2021-10-24T08:15:02Z
1	cust1	Doilies - 10, Paper	64	6	vendita	True	2021-09-11T05:04:06Z

Risultati: 10

Cosa posso fare per te?



elly

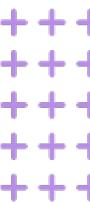
WHAT DO YOU NEED?



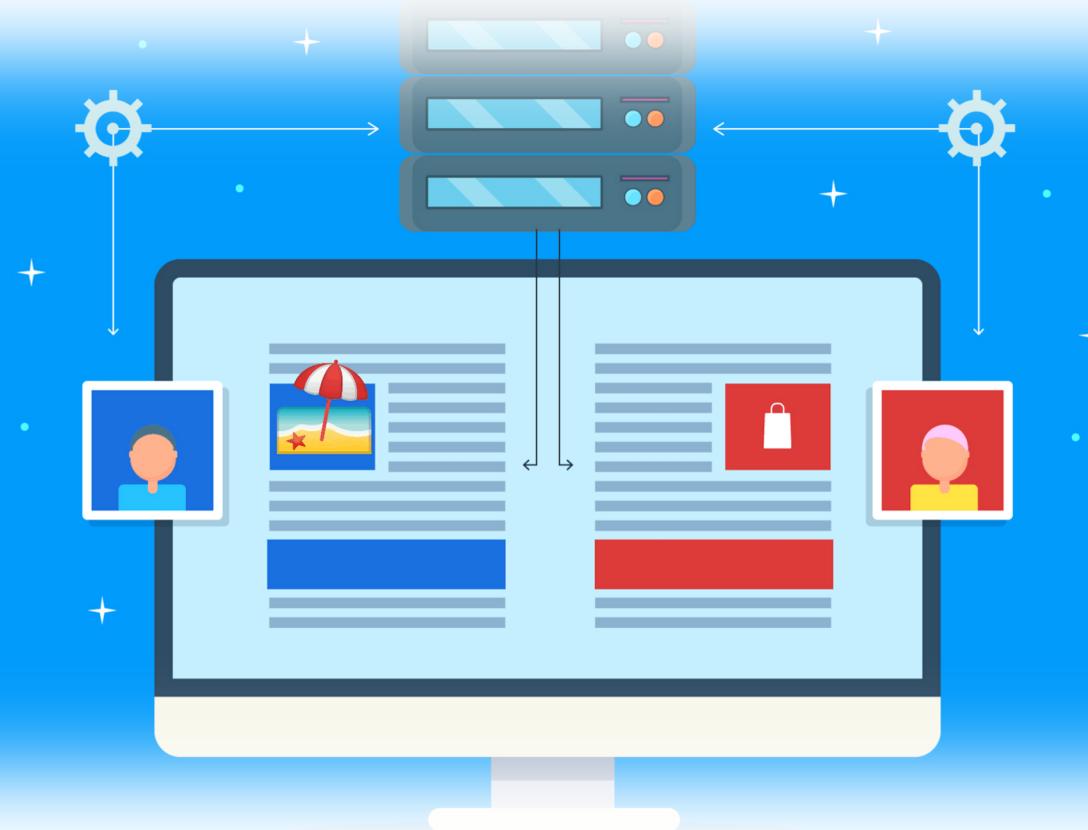
Perché Blazor?

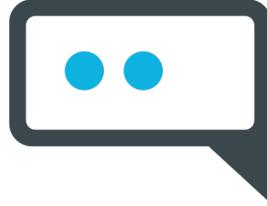
- C# e .NET
- Onboarding sviluppatori backend
- Riutilizzo e condivisione del codice
- Framework frontend moderno e completo





Generazione dinamica della UI





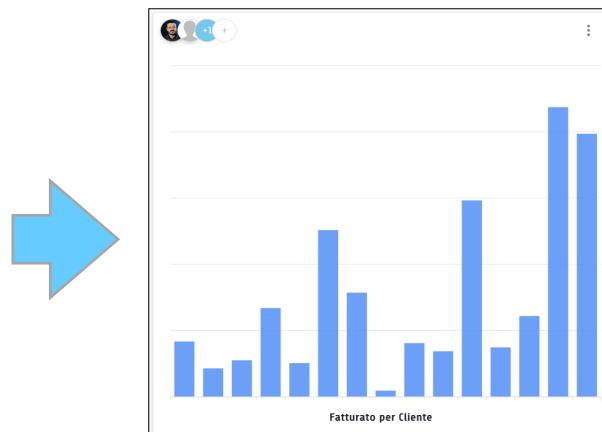
elly
WHAT DO YOU NEED?



Importatori

Fatture consolidate							
Id	Numero	Id cliente	Data	Numero Fattura	Data scadenza	Importo lordo	Importo netto
20059	10068	101	02-01-2009	20059	16-01-2009	1276,50	1110,00
20060	10069	101	24-01-2009	20060	07-02-2009	247,25	215,00
20061	10070	101	30-05-2009	20061	13-06-2009	1851,50	1610,00
20062	10071	101	31-05-2009	20062	14-06-2009	925,75	805,00
20063	10072	101	01-12-2009	20063	15-12-2009	5807,50	5050,00
20064	10073	101	19-12-2009	20064	02-01-2010	207,00	180,00
20065	10074	110	15-02-2009	20065	01-03-2009	3015,00	3015,00
20066	10075	110	15-05-2009	20066	29-05-2009	2510,00	2510,00
20067	10076	110	05-12-2009	20067	19-12-2009	2357,50	2050,00
20068	10090	109	05-01-2009	20068	19-01-2009	1621,50	1410,00
20069	10081	106	15-01-2009	20069	29-01-2009	247,25	215,00
20041	10050	106	19-12-2008	20041	02-01-2009	1150,00	1000,00
20042	10051	109	19-12-2008	20042	02-01-2009	920,00	800,00

Collezioni Originarie



Collezioni Derivate



Configura l'importatore WebApi

Test

Annulla

Aggiorna

Nome Intervallo sincronizzazione

WebApi Ogni ora

Gli intervalli di tempo consentiti partono da 30 minuti

Parametri

(*) obbligatori

Collezioni da importare



Products



Endpoint (*)

<https://dummyjson.com/products?limit=100>

Metodo HTTP (*)

GET

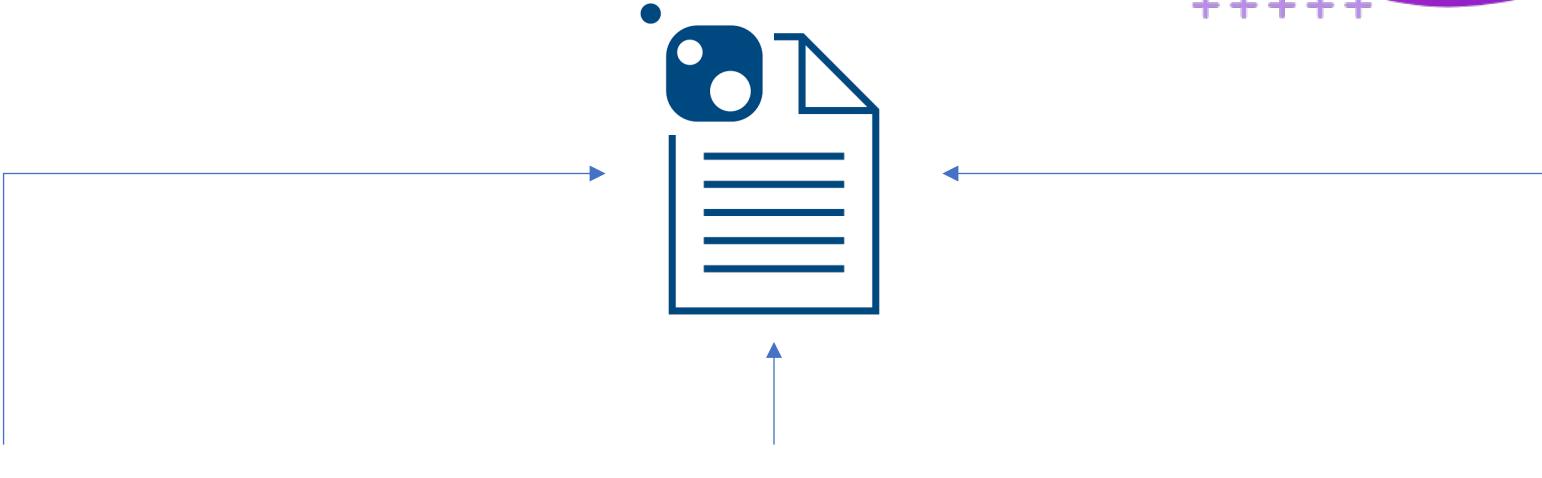
Body richiesta

Query string segreta



.....







```
1 public record EllyDataImporterMetaConfiguration
2 {
3     ...
4     [Required]
5     [MaxLength(20)]
6     public string Name { get; init; }
7     public int Version { get; init; }
8     public bool IsSchedulable { get; init; }
9     public bool RequiresSchedule { get; init; }
10    public TimeSpan MinScheduleInterval { get; init; }
11    public TimeSpan? MaxScheduleInterval { get; init; }
12
13    public List<EllyDataImporterMetaCollection> AvailableCollections {
14        public List<EllyDataImporterParameter> Fields { get; init; } = new()
15        public List<EllyDataImporterParameter> CollectionFields { get; init; }
16        public List<EllyDataImporterParameter> AttributeFields { get; init; }
17        ...
18    }
```



```
1 public record EllyDataImporterMetaCollection
2 {
3     public string Name { get; set; }
4     public string DisplayName { get; set; }
5 }
```



```
1 public record EllyDataImporterParameter
2 {
3     [Required]
4     [MaxLength(20)]
5     public string Name { get; set; }
6
7     [Required]
8     [DefaultValue(EllyDataImporterParameterType.String)]
9     public EllyDataImporterParameterType Type { get; set; }
10
11    public bool IsRequired { get; set; }
12    public bool IsSecret { get; set; }
13    public string[] AllowedValues { get; set; } = Array.Empty<string>();
14 }
```



```
1 @switch (Parameter.Type)
2 {
3     case EllyDataImporterParameterType.Enum:
4     {
5         <div>
6             <label for="@($"field-{Parameter.Name}-value")">
7                 @GetLocalizedParameterName()
8             </label>
9             <ComboBox
10                name="@($"field-{Parameter.Name}-value")"
11                TItem="string"
12                TValue="string"
13                @bind-Value="Parameter.Value"
14                DisableClear="Parameter.IsRequired"
15                SearchMethod="ParameterSearch">
16                 <ComboBoxSelectedTemplate Context="item">
17                     <span>@GetLocalizedParameterValue(item)</span>
18                 </ComboBoxSelectedTemplate>
19                 <ComboBoxResultTemplate Context="item">
20                     <span>@GetLocalizedParameterValue(item)</span>
21                 </ComboBoxResultTemplate>
22             </ComboBox>
23         </div>
24         break;
25     }
26
27 ...
```



```
1 ...
2
3     case EllyDataImporterParameterType.String:
4     {
5         <div class="form-group">
6             <label for="@($"field-{Parameter.Name}-value")">
7                 @GetLocalizedParameterName()
8             </label>
9             <InputText
10                id="@($"field-{Parameter.Name}-value")"
11                class="form-control"
12                type="text"
13                @bind-Value="Parameter.Value"/>
14         </div>
15         break;
16     }
17 }
```



Upload di grandi file





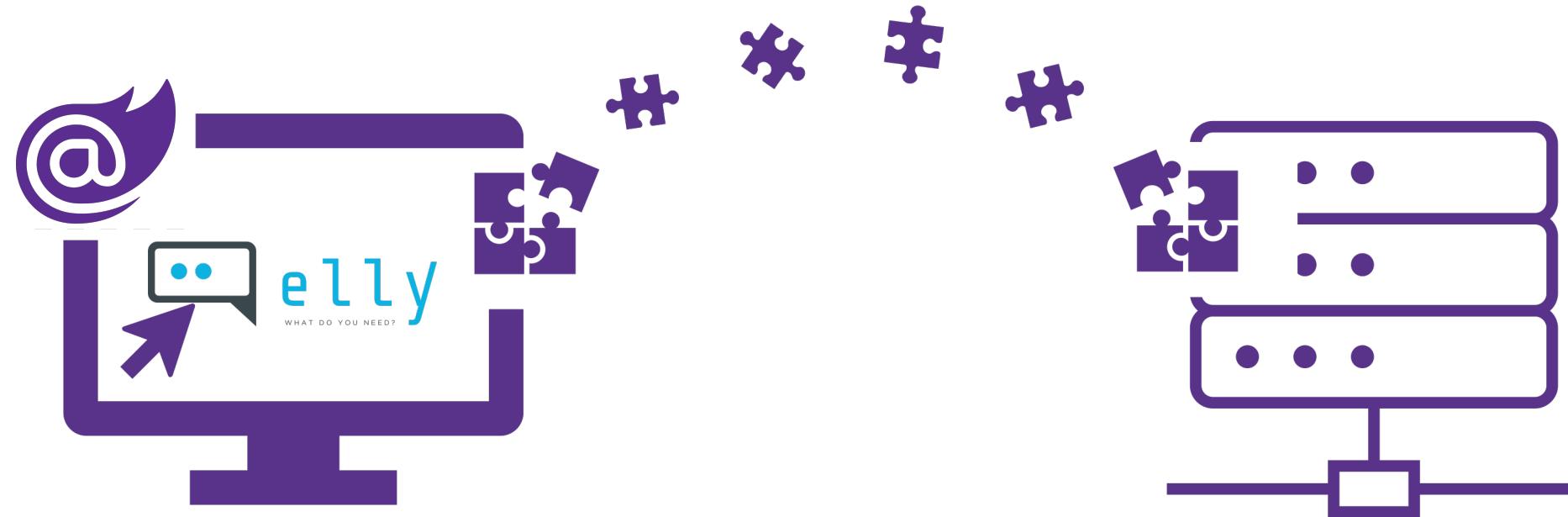
OutOfMemoryException

```
✖ Uncaught (in promise) Error: System.OutOfMemoryException: Out of memory
    at System.IO.MemoryStream.set_Capacity(Int32 value)
    at System.IO.MemoryStream.EnsureCapacity(Int32 value)
    at System.IO.MemoryStream.Write(Byte[] buffer, Int32 offset, Int32 count)
    at System.Net.Http.HttpContent.LimitMemoryStream.Write(Byte[] buffer, Int32 offset, Int32 count)
    at System.IO.MemoryStream.WriteAsync(ReadOnlyMemory`1 buffer, CancellationToken cancellationToken)
--- End of stack trace from previous location ---
    at System.IO.Stream.<CopyToAsync>g__Core|27_0(Stream source, Stream destination, Int32 bufferSize, CancellationToken cancellationToken)
    at System.Net.Http.StreamToStreamCopy.<CopyAsync>g__DisposeSourceAsync|1_0(Task copyTask, Stream source)
    at System.Net.Http.HttpContent.<CopyToAsync>g__WaitAsync|56_0(ValueTask copyTask)
    at System.Net.Http.MultipartContent.SerializeToStreamAsyncCore(Stream stream, TransportContext context, CancellationToken cancellationToken)
    at System.Net.Http.HttpContent.LoadIntoBufferAsyncCore(Task serializeToStreamTask, MemoryStream tempBuffer)
    at System.Net.Http.HttpContent.<WaitAndReturnAsync>d__82`2[[System.Net.Http.HttpContent, System.Net.Http, Version=7.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a],[System.Byte[], System.Private.CoreLib, Version=7.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e]].MoveNext()
    at System.Net.Http.BrowserHttpHandler.CallFetch(HttpRequestMessage request, CancellationToken cancellationToken, Nullable`1 allowAutoRedirect)
    at System.Net.Http.BrowserHttpHandler.<SendAsync>g__Impl|55_0(HttpRequestMessage request, CancellationToken cancellationToken, Nullable`1 allowAutoRedirect)
    at System.Net.Http.HttpClient.<SendAsync>g__Core|83_0(HttpRequestMessage request, HttpCompletionOption completionOption, CancellationTokenSource cts, Boolean
disposeCts, CancellationTokenSource pendingRequestsCts, CancellationToken originalCancellationToken)
    at FileUploadDotNet7.Client.Pages.Index.OnInputChange(InputFileChangeEventArgs e) in /Users/diego/Library/CloudStorage/OneDrive-
EllycodeSrl/Projects/Elly.BalzorConf/FileUploadDotNet7/FileUploadDotNet7/Client/Pages/Index.razor:line 106
    at Microsoft.AspNetCore.Components.ComponentBase.CallStateHasChangedOnAsyncCompletion(Task task)
    at Object.endInvokeDotNetFromJS (http://localhost:5093/\_framework/blazor.webassembly.js:1:3549)
    at Object.Xt [as endInvokeDotNetFromJS] (http://localhost:5093/\_framework/blazor.webassembly.js:1:63263)
    at Object.Gt [as invokeJSFromDotNet] (http://localhost:5093/\_framework/blazor.webassembly.js:1:62760)
    at Object.Ii (http://localhost:5093/\_framework/dotnet.7.0.5.eesj9oj0me.js:5:71974)
    at _mono_wasm_invoke_js_blazor (http://localhost:5093/\_framework/dotnet.7.0.5.eesj9oj0me.js:14:103886)
    at wasm://wasm/00992dae:wasm-function\[313\]:0x1d507
    at wasm://wasm/00992dae:wasm-function\[283\]:0x1c935
    at wasm://wasm/00992dae:wasm-function\[221\]:0xe025
    at wasm://wasm/00992dae:wasm-function\[220\]:0xce95
    at wasm://wasm/00992dae:wasm-function\[8115\]:0x1a21ff
```



```
● ● ●  
1 // We need to check for body content  
2 if (request.Content != null)  
3 {  
4     if (request.Content is StringContent)  
5     {  
6         requestObject.SetObjectProperty(  
7             "body",  
8             await request.Content.ReadAsStringAsync(cancellationToken)  
9                 .ConfigureAwait(continueOnCapturedContext: true)  
10            );  
11    }  
12 else  
13 {  
14     using (Uint8Array uint8Buffer = Uint8Array.From(  
15         await request.Content.ReadAsByteArrayAsync(cancellationToken)  
16             .ConfigureAwait(continueOnCapturedContext: true))  
17     )  
18     {  
19         requestObject.SetObjectProperty("body", uint8Buffer);  
20     }  
21 }  
22 }
```

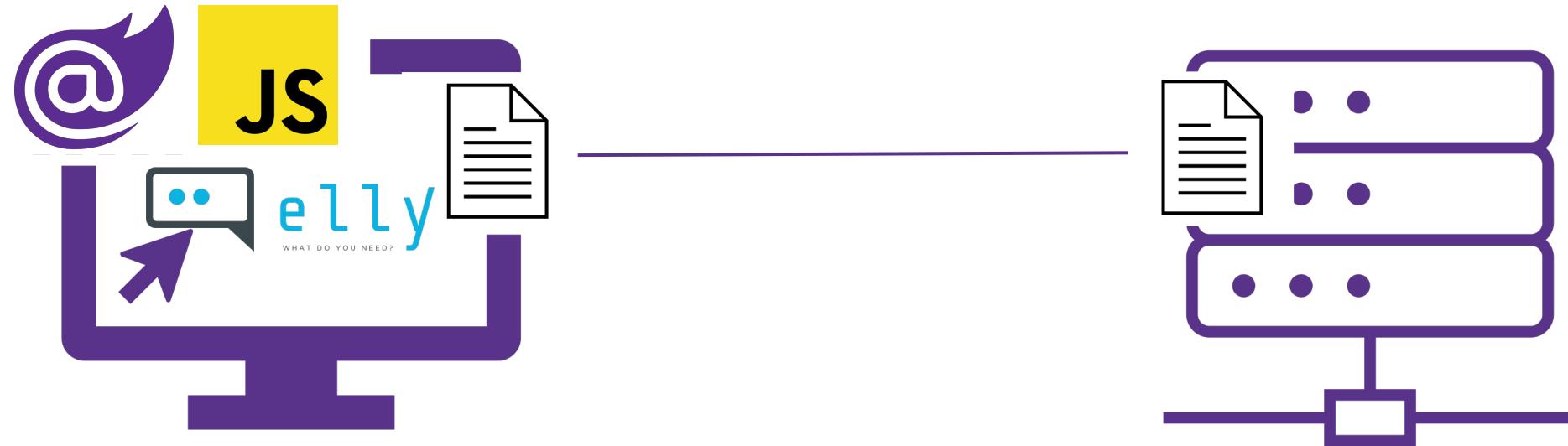
Invio di file in chunks



```
1 ...  
2  
3 await using var inStream = file.OpenReadStream(long.MaxValue);  
4 for (var i = 0; i < numChunks; i++)  
5 {  
6     var buffer = new byte[_chunkSize];  
7     await inStream.ReadAsync(buffer, 0, buffer.Length);  
8  
9     var chunk = new FileChunk  
10    {  
11        Data = buffer,  
12        FileName = fileName,  
13        Offset = offset += _chunkSize,  
14        First = first  
15    };  
16    await _apiClient.ChunkUpload(chunk);  
17    first = false;  
18 }  
19  
20 ...
```



Stream via JavaScript



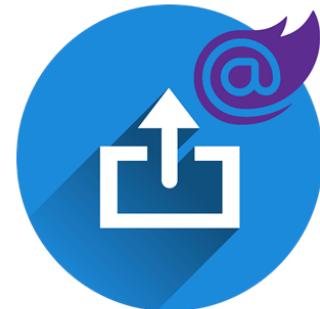
```
 1 'use strict';
 2
 3 export async function fileStream (fileInputElementId) {
 4   let fileInputElement = document.getElementById(fileInputElementId);
 5   let data = new FormData();
 6   data.append('File', fileInputElement.files[0])
 7   return { statusCode : await AJAXSubmit('/fileupload/stream', data).then(response => response.status) };
 8 }
 9
10 export async function AJAXSubmit (url, data) {
11   return await fetch(url, {
12     method: 'POST',
13     body: data
14   });
15 }
```

```
 1 public class FileUploadJsInterop : IAsyncDisposable
 2 {
 3     private readonly Lazy<Task<IJSObjectReference>> _moduleTask;
 4
 5     public FileUploadJsInterop(IJSRuntime jsRuntime)
 6     {
 7         var importPath = "./js/fileUploadJsInterop.js";
 8         _moduleTask = new(() => jsRuntime.InvokeAsync<IJSObjectReference>("import", importPath).AsTask());
 9     }
10
11     public async ValueTask<FileStreamResponse> FileStream(string fileInputElementId)
12     {
13         var module = await _moduleTask.Value;
14         return await module.InvokeAsync<FileStreamResponse>("fileStream", fileInputElementId);
15     }
16
17     public async ValueTask DisposeAsync()
18     {
19         if (_moduleTask.HasValue)
20         {
21             var module = await _moduleTask.Value;
22             await module.DisposeAsync();
23         }
24     }
25 }
```



```
1 [HttpPost("stream")]
2 [RequestSizeLimit(long.MaxValue)]
3 [DisableFormValueModelBinding]
4 public async Task<IActionResult> FileStream()
5 {
6     if (MultipartRequestHelper.IsMultipartContentType(Request.ContentType))
7     {
8         try
9         {
10             var boundary = MultipartRequestHelper.GetBoundary(MediaTypeHeaderValue.Parse(Request.ContentType));
11             var reader = new MultipartReader(boundary, HttpContext.Request.Body);
12             var section = await reader.ReadNextSectionAsync();
13             var contentDisposition = MultipartRequestHelper.GetContentDisposition(section);
14
15             if (MultipartRequestHelper.HasAllowedFileContentDisposition(contentDisposition))
16             {
17                 var fileName = $"{Guid.NewGuid()}{Path.GetExtension(contentDisposition.FileName.Value).ToLowerInvariant()}";
18                 await using var targetStream = System.IO.File.Create(Path.Combine(_dirName, fileName));
19                 await section.Body.CopyToAsync(targetStream);
20                 return Ok();
21             }
22         }
23         catch (InvalidDataException ex)
24         {
25             return BadRequest();
26         }
27     }
28     return BadRequest();
29 }
```





Upload di grandi file in Blazor WebAssembly

Lo scenario che analizzeremo in questo articolo è quello dell'invio di file di grandi dimensioni tramite richiesta HTTP nel contesto, più specifico, di un'applicazione realizzata con Blazor WebAssembly che effettua il caricamento di file su server tramite API REST....

[LEGGI TUTTO](#)





Attivazione vocale



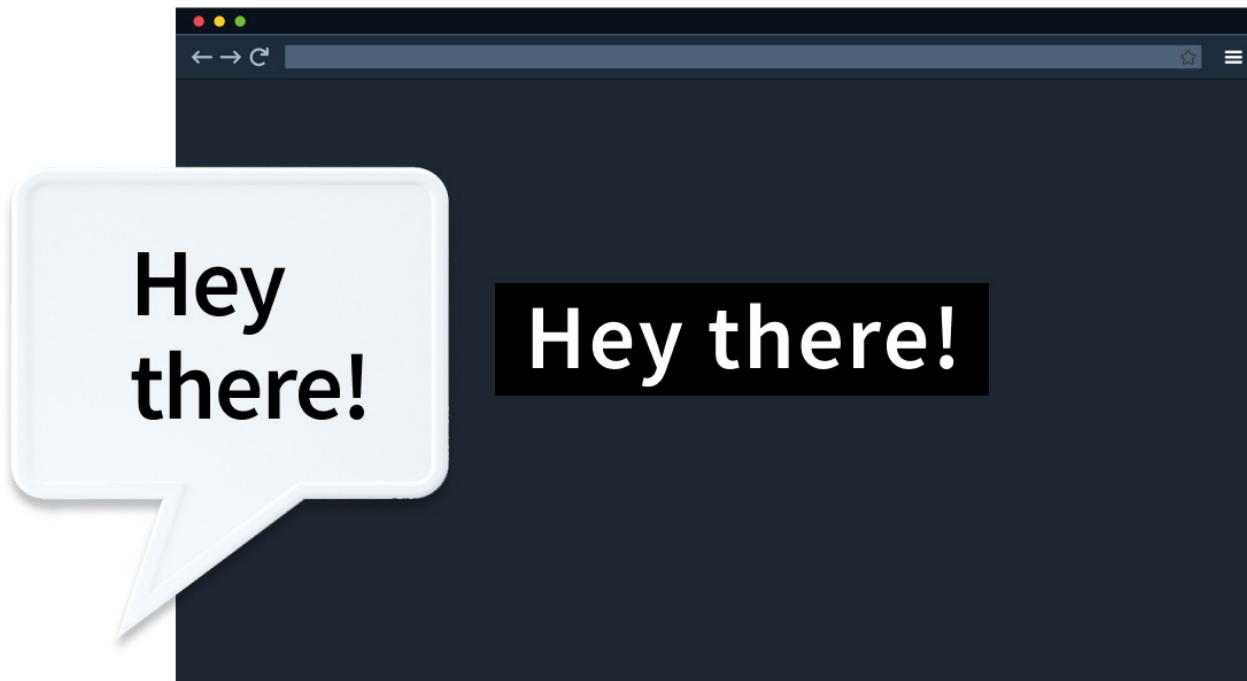
Costi

Accuracy

Latenza

Risorse

Privacy





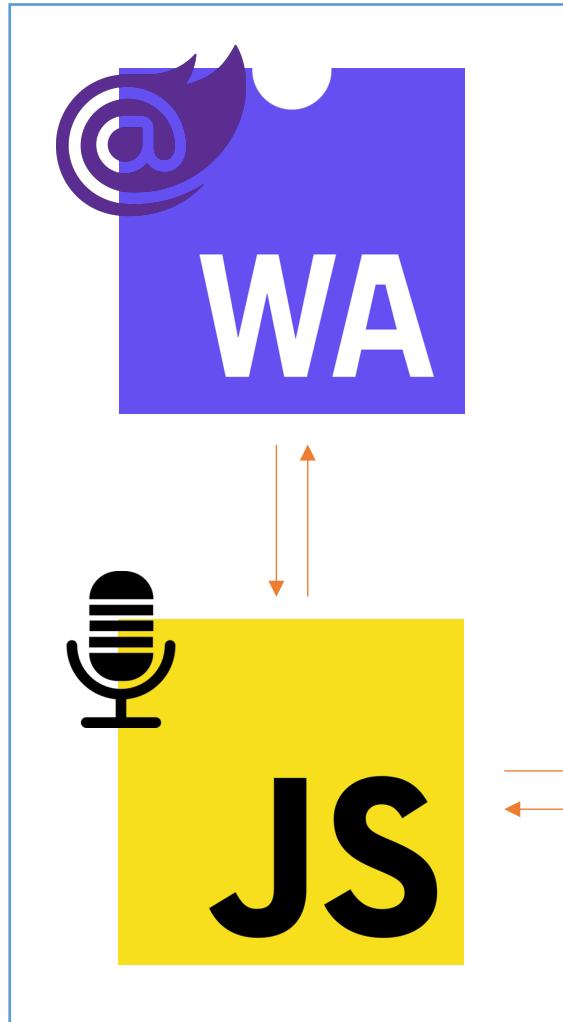
- Supports 20+ languages and dialects.
- Works offline, even on lightweight devices.
- Portable per-language models are only 50Mb each.
- Allows quick reconfiguration of vocabulary for best accuracy.



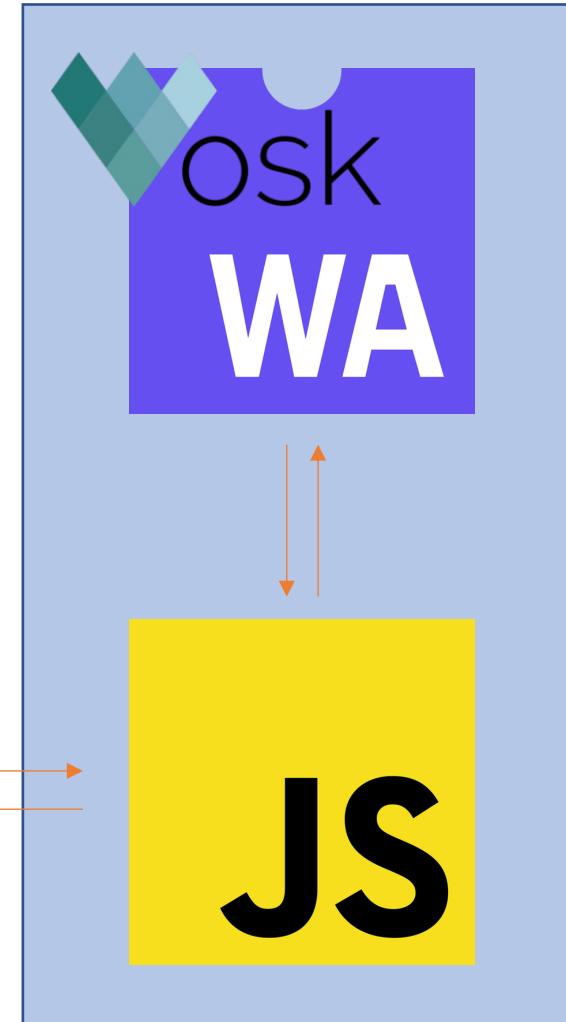
WEBASSEMBLY

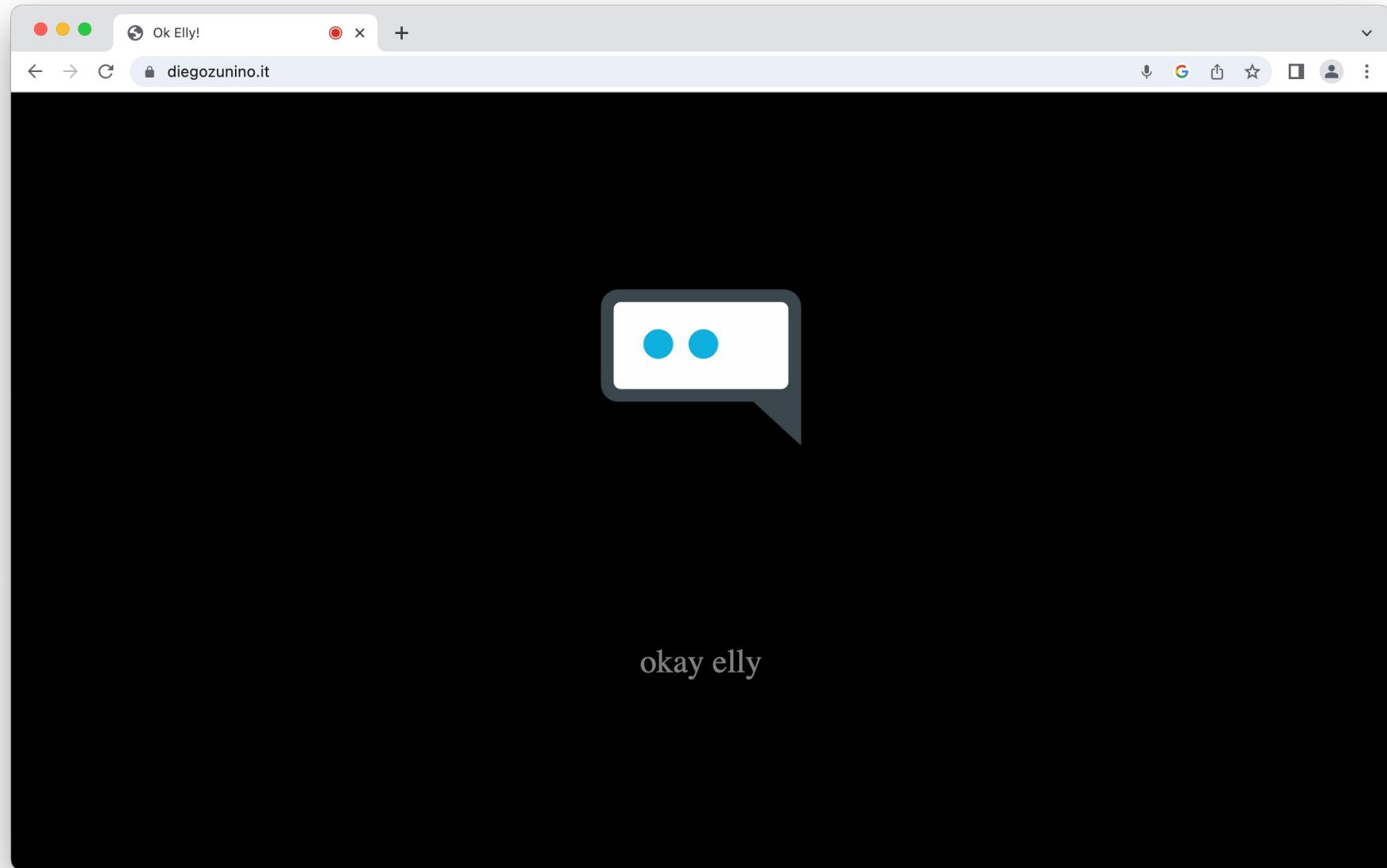
- Linguaggio di basso livello assembly-like con un formato binario compatto.
- Pensato per l'esecuzione all'interno del browser con prestazioni simil-native.
- Supportato da tutti i principali browser.
- Offre un target di compilazione per i molti linguaggi (C++, Rust, C#).
- Progettato per interoperare con JavaScript.

MAIN THREAD



WEB WORKER



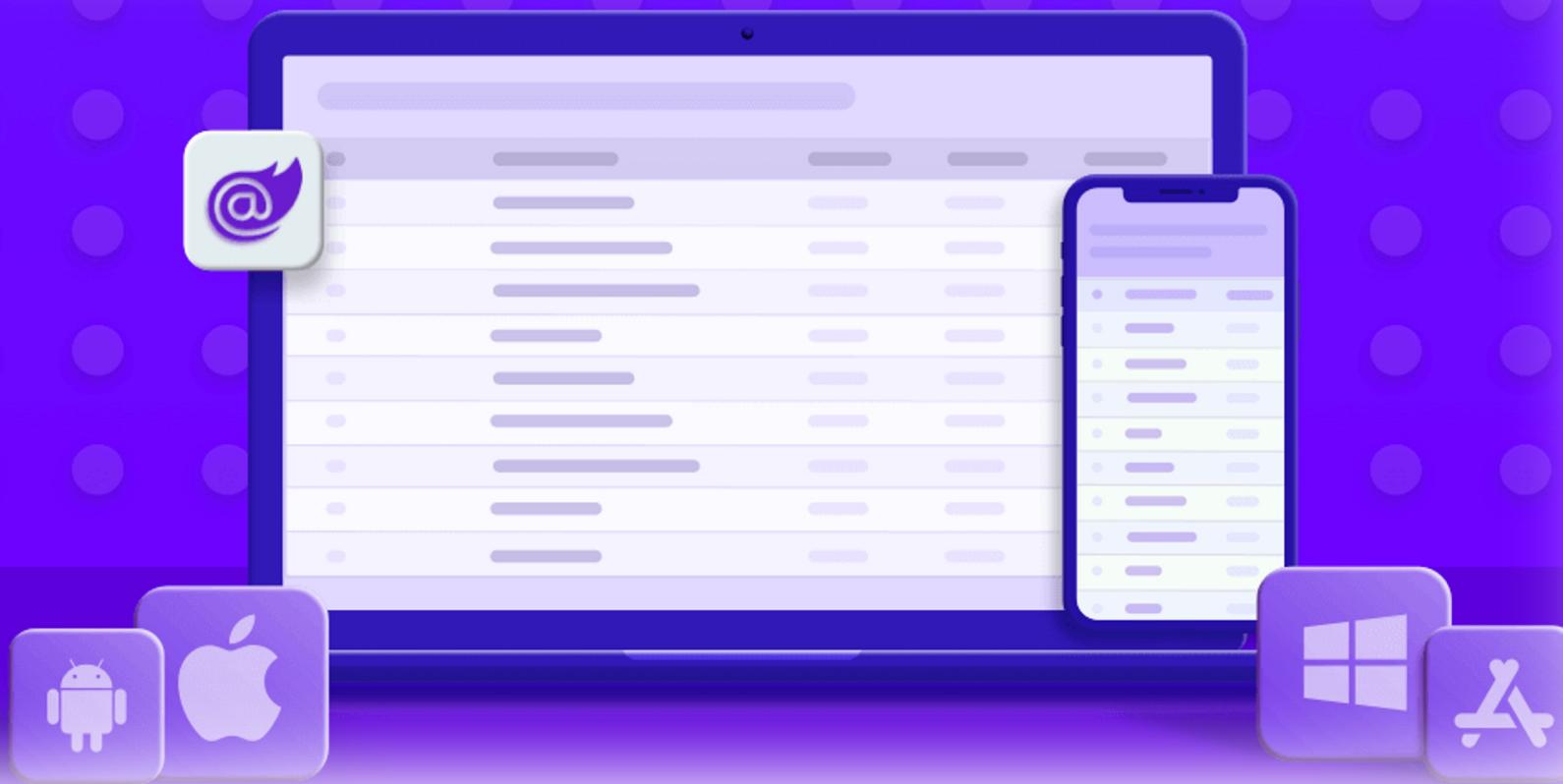




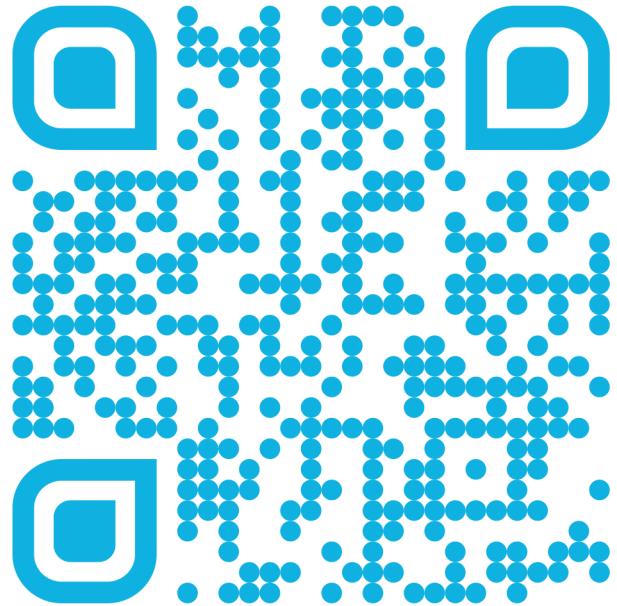
Doc, e ora che fai? Ritorni al futuro?



Blazor Hybrid



Attualmente siamo in beta privata



Stiamo cercando **scenari** in cui poter dare **valore** con la nostra soluzione:

- Accesso **gratuito** alla piattaforma
- Integrazioni **gratuite** con i dati aziendali
- **Prezzo scontato** per gli utenti aggiuntivi

Chiediamo in cambio di poter **analizzare l'uso della piattaforma** (non i dati) e che forniate **il vostro feedback**

**WE ARE
HIRING!**





Domande? Dubbi? Curiosità?



Diego Zunino

Responsabile Area Sviluppo @ Ellycode
diego.zunino@ellycode.com

