

Theodo

HTMX, le nouvel atout perf pour vos projets SSR

Ewen Quimerc'h

We Love Speed 17.09.2024



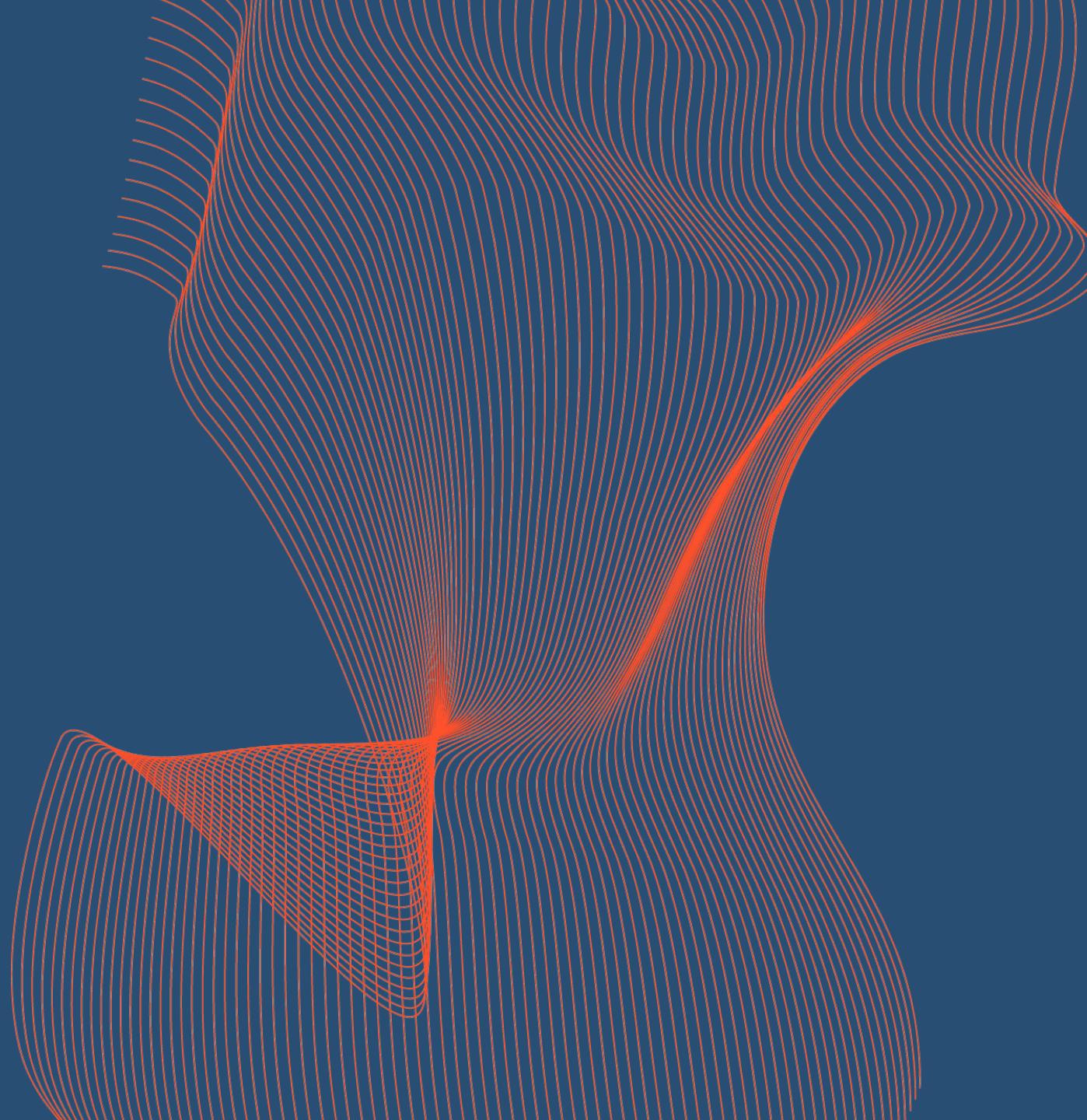
Ewen Quimerc'h

Developer @Theodo depuis 2 ans 1/2

- Plein de missions avec diverses technologies
- Spécialité performance
- Open-source : Fuego, un framework Go pour OpenAPI
- Amateur de gastronomie



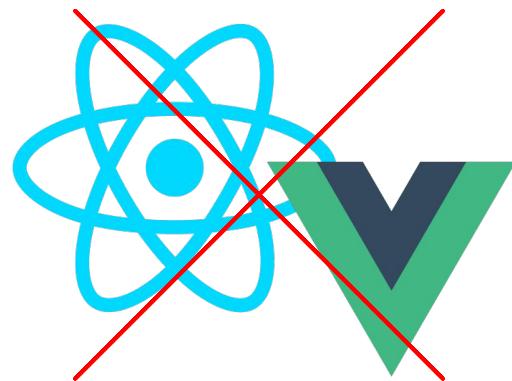
Server-Side Rendering





C'est quoi le SSR ?

Le Server Side Rendering, la manière classique de faire un site.



```
templ hello(name string) {  
  <div>Hello, { name }</div>  
}
```

django **=GO**
Symfony



WORDPRESS
Theodo 4

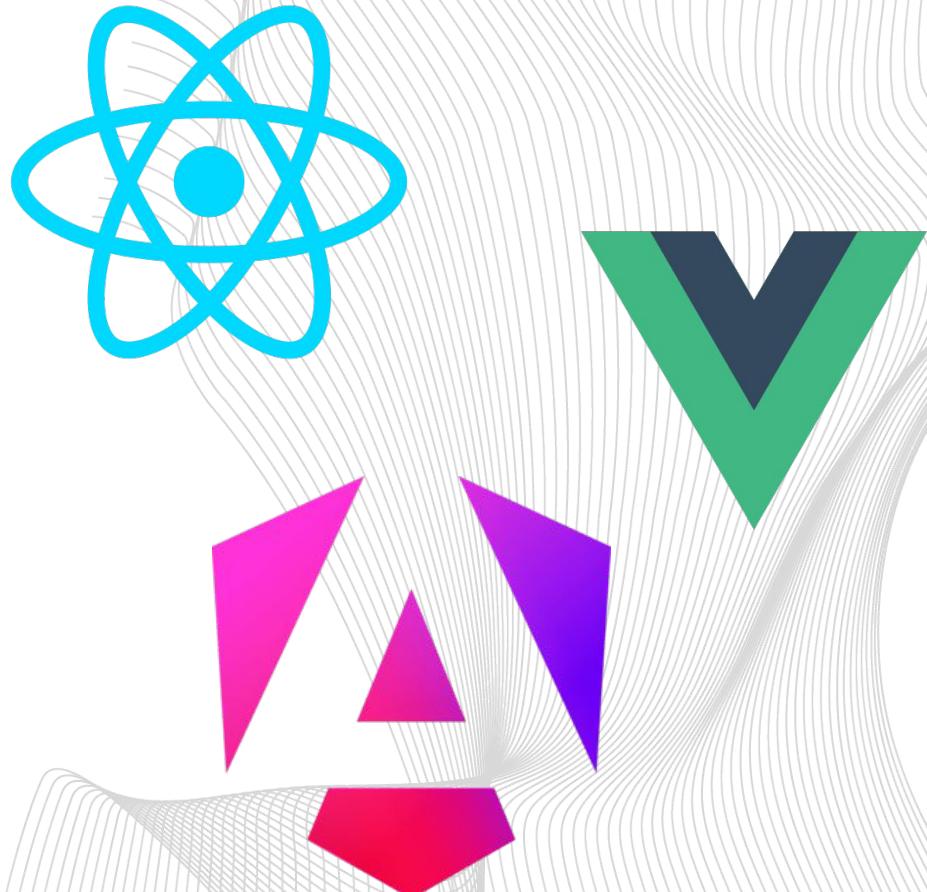
En SSR, le navigateur est juste **un écran** qui affiche **des pages**, pas un logiciel !

La guerre des modes de rendu

Pour ceux qui n'ont pas été en conférence depuis 2015



VS



Les frameworks front s'approprient le terme SSR

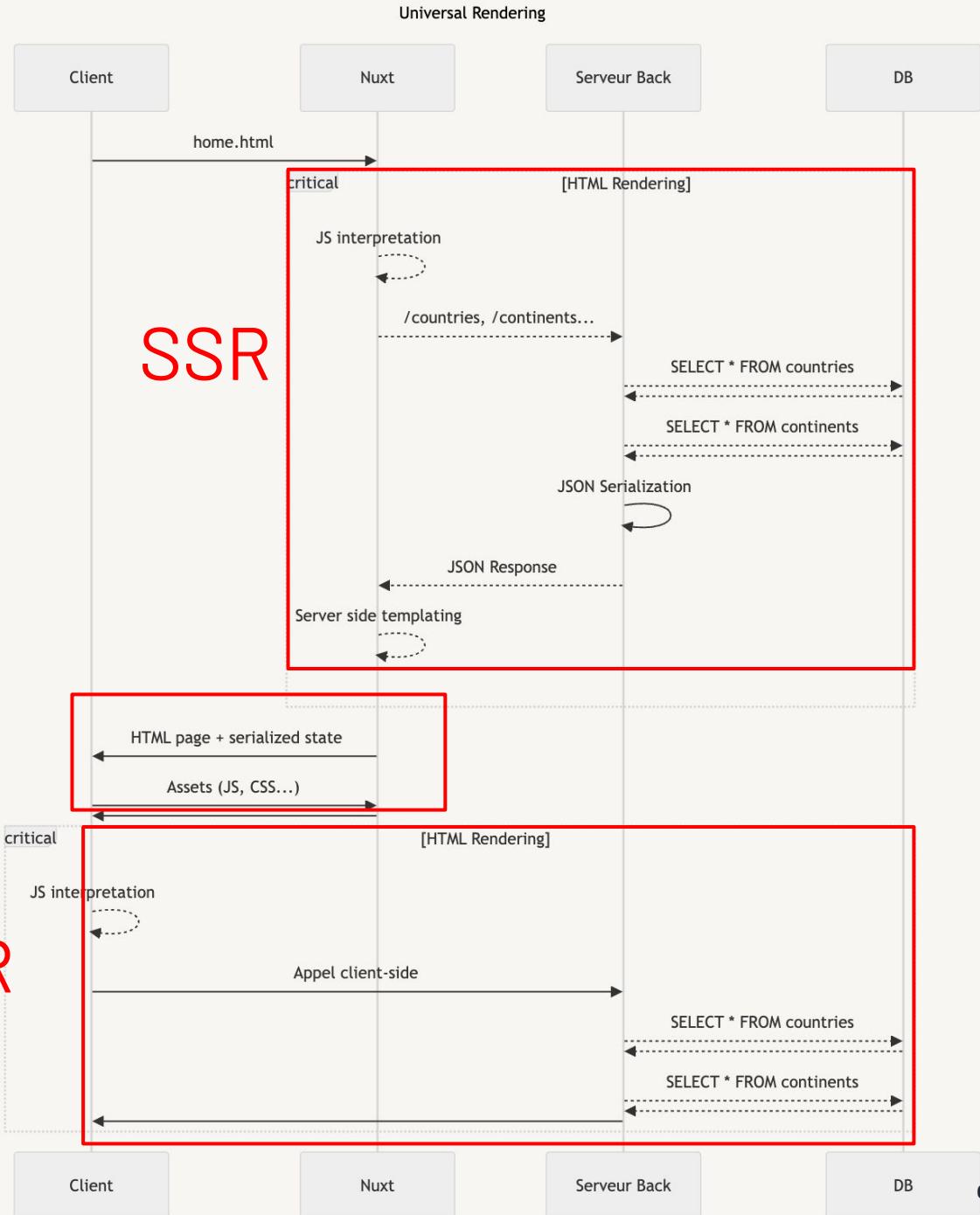
Alors qu'ils font du Universal Rendering :
SSR puis CSR (avec réhydratation)

NEXT.JS



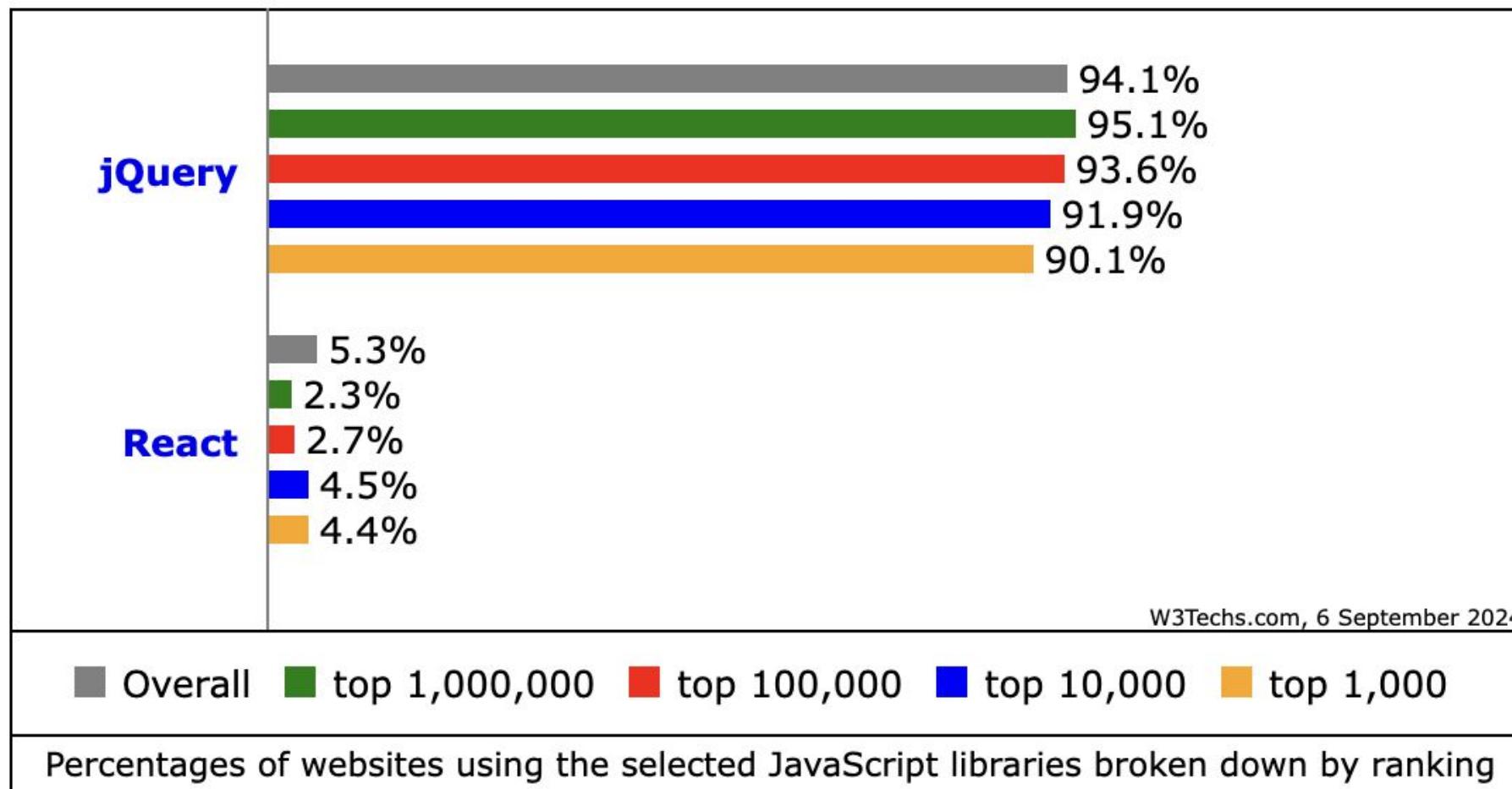
???

CSR



La majorité des sites actuels sont en SSR

Comment peut-on améliorer leur performance sans changer de paradigme ?



HTMX, amélioration progressive de l'UX



On va travailler sur un exemple live

Vous pouvez aller sur <https://gourmet.quimerch.com>

The screenshot shows a web browser window with the URL <https://gourmet.quimerch.com/recipes/one?id=crepes-de-ble-noir>. The page title is "Gourmet". Below it is a search bar with the text "orange, crème brûlée...". The main navigation menu includes "Planner", "Recipes", "Healthy", and "Quick".

Crêpes de blé noir (with a heart icon)

★★★★☆ by admin

DISH

⌚ Preparation: 5 min ⌚ Cooking: 2 min

⌚ Calories: ? kcal 🏃 Difficulty: Medium

Aussi appelées Galettes par les non-bretons

Ingredients

- Eau - 100 ml
- Farine de blé noir - 125 g
- Oeuf - 1 piece

Instructions

Dans un saladier, mélanger la farine et le gros sel. A l'aide d'un fouet, verser l'eau en deux ou trois fois, tout en mélangeant la préparation. On obtient une pâte lisse et épaisse à laquelle on ajoute un oeuf pour donner une belle coloration à la cuisson.

Filmer et laisser reposer 1 à 2 heures au réfrigérateur.

Related Recipes

Pâtes Carbonara DISH New La classique italienne

Le problème du SSR

C'est la navigation !

→ Pas de notion de **composant** !
On ne parle que de **pages**

Heureusement, HTMX est là !

</>htmx

high power tools for HTML

HTMX fluidifie l'expérience utilisateur

Une UX digne d'une app!

The screenshot shows a recipe card for "Crêpes de blé noir". At the top, there's a thumbnail of the dish, the title "Crêpes de blé noir" with a heart icon, and a rating of 4 stars. Below the title, it says "by admin" and "DISH". It lists preparation time as 5 min and cooking time as 2 min. The ingredients section includes "Eau - 100 ml", "Farine de blé noir - 125 g", and "Oeuf - 1 pièce". The instructions section provides steps for mixing flour and water, adding an egg, and letting it rest in the refrigerator. A "Related Recipes" sidebar shows a card for "Pâtes Carbonara" with a thumbnail, the title, a brief description, and a "New" badge.

The screenshot shows a recipe card for "Pâtes carbonara". It has a similar layout to the first card. The title is "Pâtes carbonara" with a heart icon and a 4-star rating. It's also by "admin" and categorized as "DISH". Preparation time is 5 min and cooking time is 10 min. The ingredients section is identical to the first card. The instructions section provides detailed steps for cooking spaghetti in salted water, adding bacon, and finishing with cheese. A red border highlights the main content area (title, author, category, time, ingredients, and instructions). A red arrow points from the "Related Recipes" section of the first card to the "Pâtes carbonara" card here, indicating that only the recipe content has been updated.

→ Uniquement la Recette est changée !

Comment est-ce que ça marche ?

```
<script src="https://unpkg.com/htmxx.org/dist/htmxx.min.js"></script>
```

```
templ Card(props CardProps) {
  <div class="rounded-lg border dark:border-zinc-600">
    <a href={ templ.URL(props.Link) }>
      hx-boost="true"
      hx-target="#page"
      hx-select="#page"
    </a>
    <div>
      <img alt="Card image" />
      if props.ImageURL != "" {
```

HTML

```
<!doctype html><html lang="en"><head><title>Crêpes de blé noir</title><meta
{
  "@context": "https://schema.org",
  "@type": "WebSite",
  "name": "Gourmet",
  "url": "https://gourmet.quimerch.com",
  "potentialAction": {
    "@type": "SearchAction",
    "target": "https://gourmet.quimerch.com/search?q={search_term_string}"
    "query-input": "required name=search_term_string"
  }
}></script><!-- tailwindcss --><link href="/static/tailwind.min.css" rel="stylesheet" type="text/css"/>
<div class="flex flex-col items-center justify-center gap-4 p-4 md:p-8 f
px-8 py-2 md:py-3"><a class="h-full whitespace nowrap text-stone-500 hover:dark:text-stone-300" href="#">Search</a>
<script>document.addEventListener("keydown", function (e) {
  if ((e.ctrlKey || e.metaKey) && (e.key === "k" || e.key === "s")) {
    e.preventDefault();
    document.getElementById("searchInput").select();
  }
});</script><main id="page" class="container mx-auto"><div class="p-4 md:p-8 f
```

302	GET	🔒 unp...	htmx.min.js	script	js	16.61 kB	47.76
200	GET	🔒 unp...	htmx.min.js	script	js	16.68 kB	47.76
200	GET	🔒 gou...	related?id=lasagnes	htmx....	html	1.72 kB	10.12
200	GET	🔒 gou...	one?id=crepes-de-ble-	htmx....	html	4.81 kB	18.52
200	GET	🔒 gou...	related?id=crepes-de-t	htmx....	html	1.73 kB	10.12 kB
200	GET	4.69 kB	17.56 kB

Des propriétés intéressantes

```
<script defer src="https://unpkg.com/htmrx.org/dist/htmrx.min.js">
```

```
<nav
  hx-boost="true"
  hx-target="#page"
  hx-select="#page"
>
  <div>
    <a href="/planner">
      Planner
    </a>
    <a href="/recipes">
      Recipes
    </a>
    <a href="/healthy">
      Healthy
    </a>
    <a href="/fast">
      Quick
    </a>
```

- Héritage
- DOM morphing (automatic select)
- **Progressive enhancement** : le site continue à marcher même si HTMX ne fonctionne pas

HTMX pour améliorer la performance à la navigation



L'UX est la même qu'avec React

<https://gourmet.quimerch.com/recipes/one?id=crepes-de-ble-noir>

Gourmet orange, crème brûlée... 3K

Planner Recipes Healthy Quick

Crêpes de blé noir 5 stars
by admin
DISH
Preparation: 5 min Cooking: 2 min
Calories: ? kcal Difficulty: Medium
Aussi appelées Galettes par les non-bretons

Ingredients

- Eau - 100 ml
- Farine de blé noir - 125 g
- Oeuf - 1 piece

Instructions

Dans un saladier, mélanger la farine et le gros sel. A l'aide d'un fouet, verser l'eau en deux ou trois fois, tout en mélangeant la préparation. On obtient une pâte lisse et épaisse à laquelle on ajoute un œuf pour donner une belle coloration à la cuisson. Filmer et laisser reposer 1 à 2 heures au réfrigérateur.

Related Recipes

Pâtes Carbonara La classique italienne

Galettes par les non-bretons

<https://gourmet.quimerch.com/recipes/one?id=pates-carbonara>

Gourmet orange, crème brûlée... 3K

Planner Recipes Healthy Quick

Pâtes carbonara 5 stars
by admin
DISH
Preparation: 5 min Cooking: 10 min
Calories: ? kcal Difficulty: Medium
La classique italienne

Ingredients

Instructions

Dans une grande casserole, portez de l'eau salée à ébullition et faites cuire les spaghetti selon les instructions sur l'emballage jusqu'à ce qu'ils soient al dente. Égouttez les pâtes, en réservant environ 1/2 tasse d'eau de cuisson. Dans une poêle séparée, faites cuire les dés de bacon à feu moyen jusqu'à ce qu'ils soient croustillants. Retirez le bacon de la poêle et mettez-le de côté, en laissant la graisse de cuisson dans la poêle. Dans la même poêle, ajoutez l'ail haché à la graisse de bacon et faites cuire pendant une minute jusqu'à ce qu'il dégage son parfum. Retirez la poêle du feu. Dans un petit bol, fouettez ensemble les œufs, le Parmesan râpé et le Pecorino Romano râpé jusqu'à obtention d'un mélange.

Related Recipes

Crêpes de blé noir

Croque Monsieur Le classique français

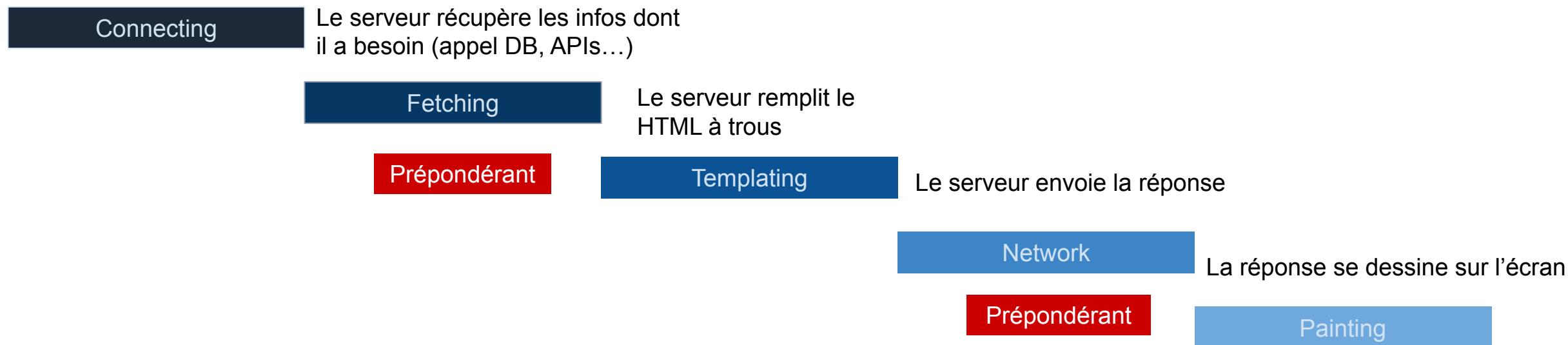


Uniquement le #page est changé !

En terme de performance, ça donne quoi ?

On veut une waterfall !

Envoi d'une requête au serveur



La différence avant/après HTMX ?

Bon, peut mieux faire

Navigation normale

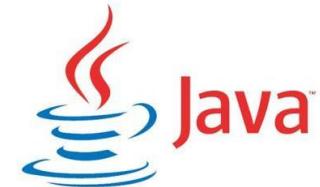
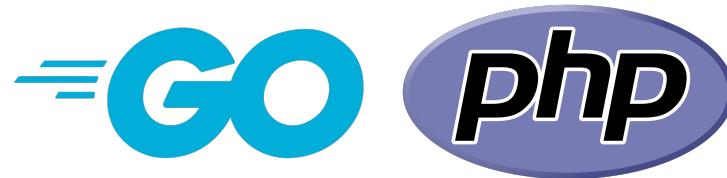


Navigation avec HTMX



- L'UX est quand même bien améliorée
- Sur des devices de basse qualité, l'amélioration peut être plus significative
- Gain/effort quand même intéressant

Code d'un controller /*ingredients*



```
func (rs Resource) showIngredients(c fuego.ContextNoBody) (fuego.Templ, error) {
    ingredients, _ := rs.IngredientsQueries.GetIngredients(c.Context())
    headerInfo, _ := rs.MetaQueries.GetHeaderInfo(c.Context())      Fetching
}

return templa.IngredientPage(templa.IngredientPageProps{
    Ingredients: ingredients,
    Header:       headerInfo,
}), nil                                         Templating
}
```

Mais si je suis sur /recette/carbonara, j'ai pas **besoin de charger encore toute la page !**

On ne veut changer qu'un “bout” de HTML au lieu d'une page complète

hx-target va juste charger le premier élément avec l'id `#page`

```
▼ <main id="page" class="container mx-auto">
  ▼ <div class="p-4 md:p-8 flex flex-col items-center md:items-start md:ween gap-6"> flex
    ▼ <div class="grid grid-cols-1 md:grid-cols-3 gap-4 md:gap-8 mb-4"> grid
      ► <div id="img"> ... </div>
      ▼ <div id="meta" class="md:col-span-2">
        ▼ <div class="flex items-center justify-between" style="flex">
          ▼ <hgroup>
            ▼ <span class="inline-flex flex-col gap-2 items-baseline pb-2">
              ▼ <span class="inline-flex gap-4 items-baseline" style="flex">
                <h1 class="text-4xl font-bold text-gray-800 dark:text-gray-200 carbonara">Carbonara</h1>
                ► <svg class="w-6 h-6 text-gray-300 hover:text-red-500" style="flex">
                  <img alt="fork and knife icon" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" style="width: 1em; height: 1em; fill: none; stroke: currentColor; stroke-width: 2px; stroke-linecap: round; stroke-linejoin: round; vertical-align: middle; margin-right: 0.5em;"/>
                </span>
              </span>
            </hgroup>
          </div>
        </div>
      </div>
    </div>
  </div>
</main>
```

Quitte à ne charger qu'un composant, autant ne traiter qu'un composant

```
func (rs Resource) showIngredients(c fuego.ContextNoBody) (fuego.Templ, error) {
    ingredients, _ := rs.IngredientsQueries.GetIngredients(c.Context())
    if c.Header("HX-Request") == "true" && c.Header("HX-Target") == "#page" {
        return templa.IngredientList(templa.IngredientListProps{
            Ingredients: ingredients,
        }), nil
    }
}
```

```
headerInfo, _ := rs.MetaQueries.GetHeaderInfo(c.Context())
return templa.IngredientPage(templa.IngredientPageProps{
    Ingredients: ingredients,
    Header:      headerInfo,
}), nil
}
```

Ca commence à être intéressant !

On économise dans toutes les étapes, y compris les plus chronophages !

Navigation normale



Navigation avec HTMX



- Une balise script
- Quelques attributs HTML
- Un early return dans le controller
- On a décomposé les templates

→ **On découpe la perf sans changement majeur !**



Parenthèse : navigation SSR vs SPA



Parenthèse SPA & Data fetching par le front

Renvoyer du HTML ou un object JSON ?

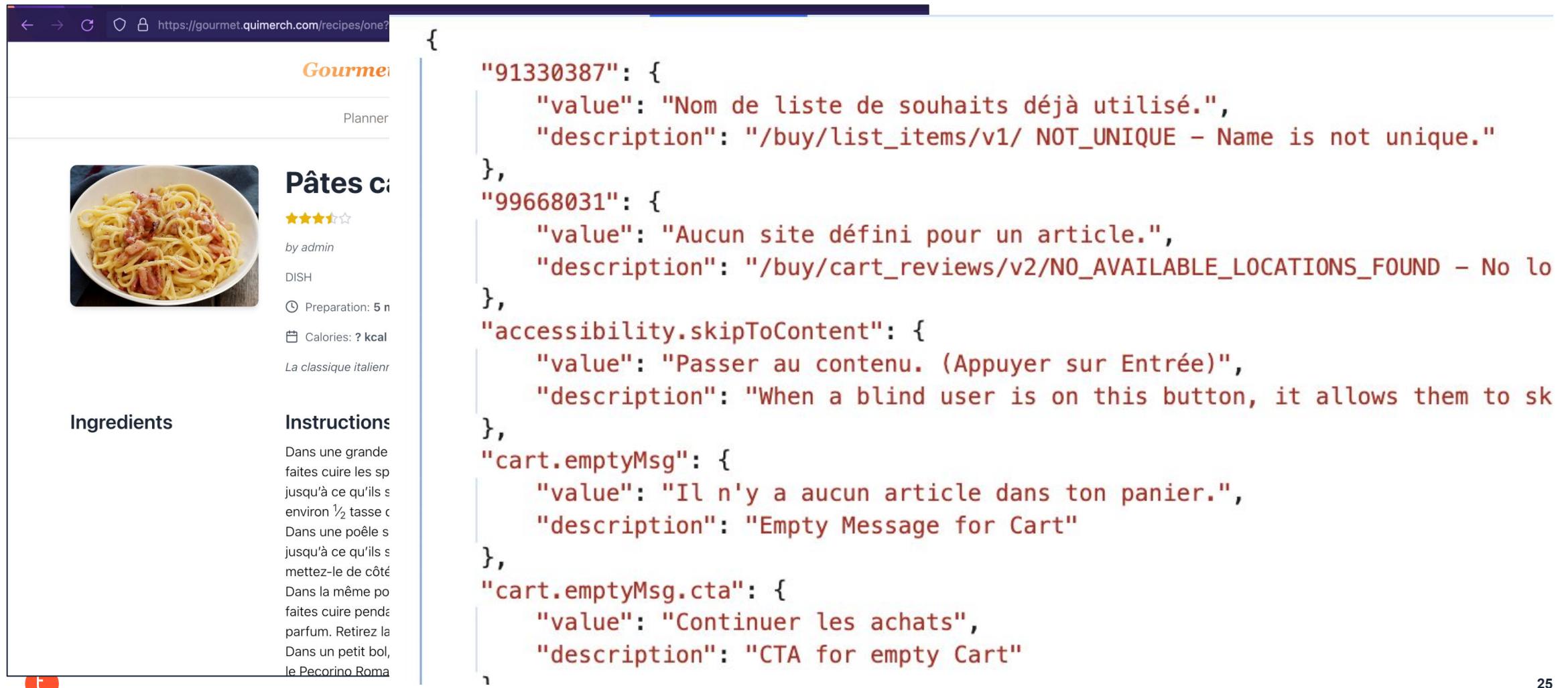
```
▼ <main id="page" class="container mx-auto">
  ▼ <div class="p-4 md:p-8 flex flex-col items-center md:items-start md:flex-row md:justify-between gap-6"> [flex]
    ▼ <div class="grid grid-cols-1 md:grid-cols-3 gap-4 md:gap-8 mb-4"> [grid]
      ▶ <div id="img"> ... </div>
      ▼ <div id="meta" class="md:col-span-2">
        ▼ <div class="flex items-center justify-between"> [flex]
          ▼ <hgroup>
            ▼ <span class="inline-flex flex-col gap-2 items-baseline pb-2 md:pb-4"> [flex]
              ▼ <span class="inline-flex gap-4 items-baseline"> [flex]
                <h1 class="text-4xl font-bold text-gray-800 dark:text-gray-200 md:mt-0">Pâtes carbonara</h1>
                ▶ <svg class="w-6 h-6 text-gray-300 hover:text-red-500" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"> ...
                  </svg>
                </span>
              ▶ <span id="stars" class="inline-flex items-baseline text-xs font-semibold"> ...
                </span> [flex]
              </span>
              <p id="created-by" class="text-sm text-gray-600 dark:text-gray-400 italic">by admin </p>
            </hgroup>
          </div>
        ▶ <div class="flex items-center justify-between my-4"> ... </div> [flex]
        ▶ <div class="flex justify-between items-center mb-4"> ... </div> [flex]
        ▶ <div class="flex justify-between items-center mb-4"> ... </div> [flex]
          <p class="text-sm text-gray-700 dark:text-gray-300 my-2 italic"> italienne</p>
        </div>
        ▶ <div> ... </div>
        ▶ <div class="md:col-span-2 space-y-2"> ... </div>
        </div>
      ▶ <aside class="w-full md:w-56"> ... </aside>
    </div>
</main>
```



Vous préférez votre liste entourée
de `<li class="...">` ou de `{ key: ?`

Parenthèse SPA & Data fetching par le front

En SSR, on ne charge que ce que l'on voit !



The screenshot shows a web browser window with the URL <https://gourmet.quimerch.com/recipes/one?>. On the left, there is a recipe card for "Pâtes carbonara". The card includes a thumbnail image of the dish, the title "Pâtes carbonara", a rating of 4 stars, author information ("by admin"), preparation time ("5 min"), and calorie count ("? kcal"). Below the card, sections for "Ingredients" and "Instructions" are visible. The "Ingredients" section lists various ingredients like bacon, eggs, and cheese. The "Instructions" section provides cooking steps. On the right side of the browser, a large JSON object is displayed, showing error messages and accessibility information. The JSON structure includes objects for error codes such as "91330387" and "99668031", their descriptions, and accessibility details.

```
{
  "91330387": {
    "value": "Nom de liste de souhaits déjà utilisé.",
    "description": "/buy/list_items/v1/ NOT_UNIQUE - Name is not unique."
  },
  "99668031": {
    "value": "Aucun site défini pour un article.",
    "description": "/buy/cart_reviews/v2/NO_AVAILABLE_LOCATIONS_FOUND - No lo
  },
  "accessibility.skipToContent": {
    "value": "Passer au contenu. (Appuyer sur Entrée)",
    "description": "When a blind user is on this button, it allows them to sk
  },
  "cart.emptyMsg": {
    "value": "Il n'y a aucun article dans ton panier.",
    "description": "Empty Message for Cart"
  },
  "cart.emptyMsg.cta": {
    "value": "Continuer les achats",
    "description": "CTA for empty Cart"
  }
}
```

Parenthèse SPA & Data fetching par le front

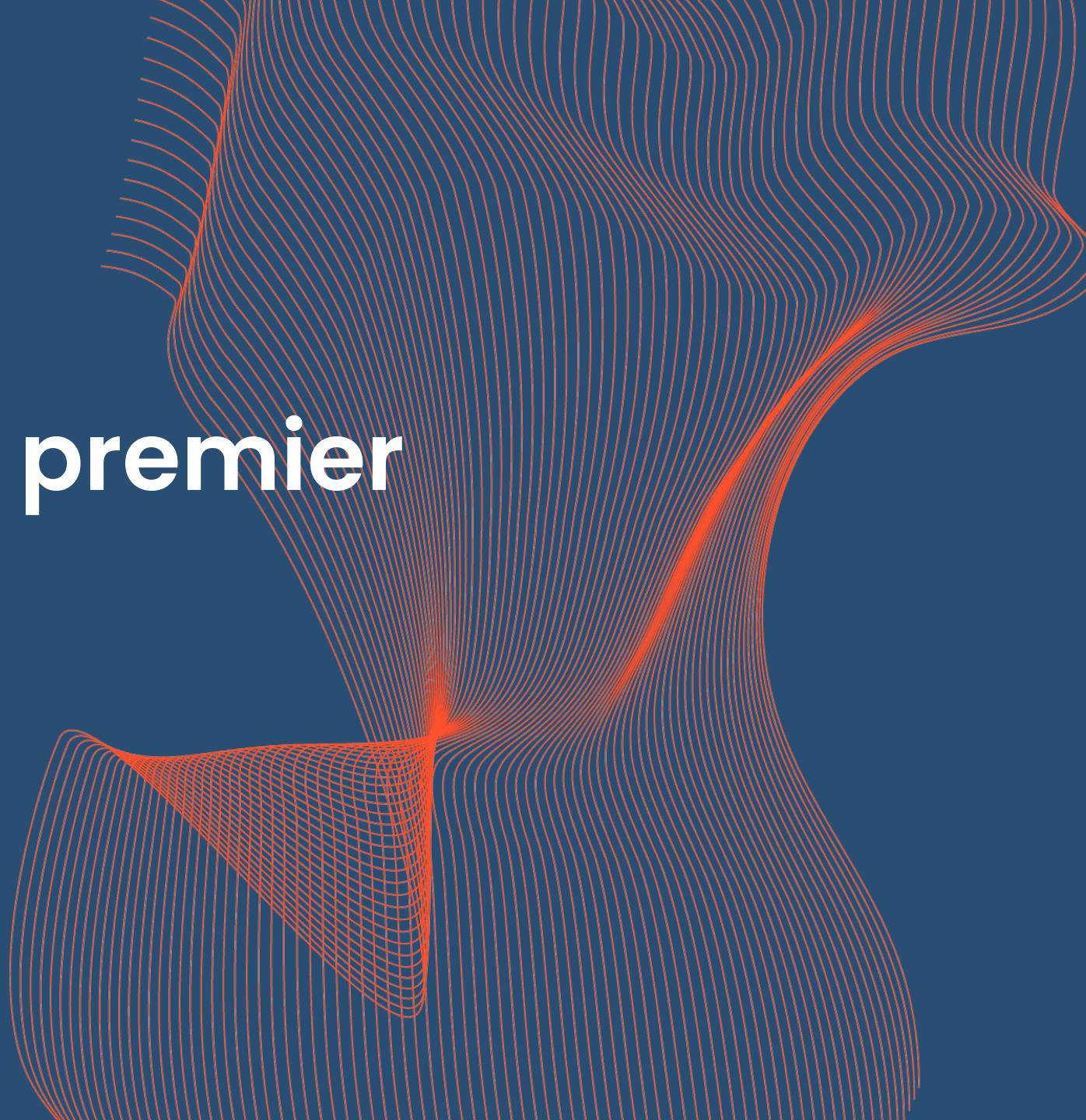
SSR avec HTMX

- On charge tout le layout
- On ne charge que ce que l'on voit

CSR

- On charge les templates + la donnée en JSON
- On peut potentiellement charger des données qui ne seront jamais affichées

Performance du premier chargement



Le lazy loading

On va envoyer des pages HTML incomplètes, qui chargeront plus tard avec un intersection observer

```
<aside class="w-full md:w-56">
  <h2
    class="text-2xl font-bold mb-4"
    hx-get="/recette/{id}"
    hx-trigger="ready"
    hx-target="#recette">
    Related Recip...
  </h2>
  <div id="related-...
    @RelatedRecip...
  </div>
</aside>
```

3rd-party requests

Name	Status	Type	Initiator	Size	Time
lasagnes	200	xhr	htmx.min.js:	4.4 kB	7.2...
related	200	xhr	htmx.min.js:	1.9 kB	2.1...

laquelle on ajoute un oeuf pour donner une belle coloration à la cuisson.

Filmer et laisser reposer 1 à 2 heures au réfrigérateur.



Le lazy loading : éviter le CLS

HTMX va remplacer la section qui est lazy-loadée.

Pour éviter le CLS, on peut renvoyer un squelette au premier chargement.

```
return templa.RecipePage(templa.RecipePageProps{  
    Recipe:          recipe,  
    Ingredients:    ingredients,  
    RelatedRecipes: []store.Recipe{}, {}, {}, {}, {}, {}},  
    {  
        id: "searchInput"  
        name: "q"  
        placeholder: "orange, crème brûlée ... "  
        hx-get: "/search"  
        hx-push-url: "true"  
        hx-trigger: "keyup changed delay:100ms, search"  
        hx-target: "#page"  
        hx-select: "#page"  
        hx-swap: "outerHTML"  
        hx-indicator: "#search-indicator"  
    }  
)
```

HTMX gère l'état de chargement

Le streaming

HTMX supporte les Server-Sent Events et les WebSockets

```
<h1>Chat</h1>
<p>This simple chat doesn't remember anything. Just stay connected!</p>
<div hx-ext="sse" sse-connect="/chat/events" sse-swap="ping" hx-swap="beforeend"></div>
<form method="POST" action="/chat" hx-boost="true" class="flex flex-row gap-2 mt-2">
  <input type="text" name="author" class="hidden" placeholder="name" value={ props.Author }/>
  <input type="text" name="content" class="input min-w-12" placeholder="message" />
  <button class="btn">Send</button>
</form>
```

Peut être utile pour charger des données de manière incrémentale (recherche en plusieurs phases, réponses de LLMs...)

Et plus !

On a vu :

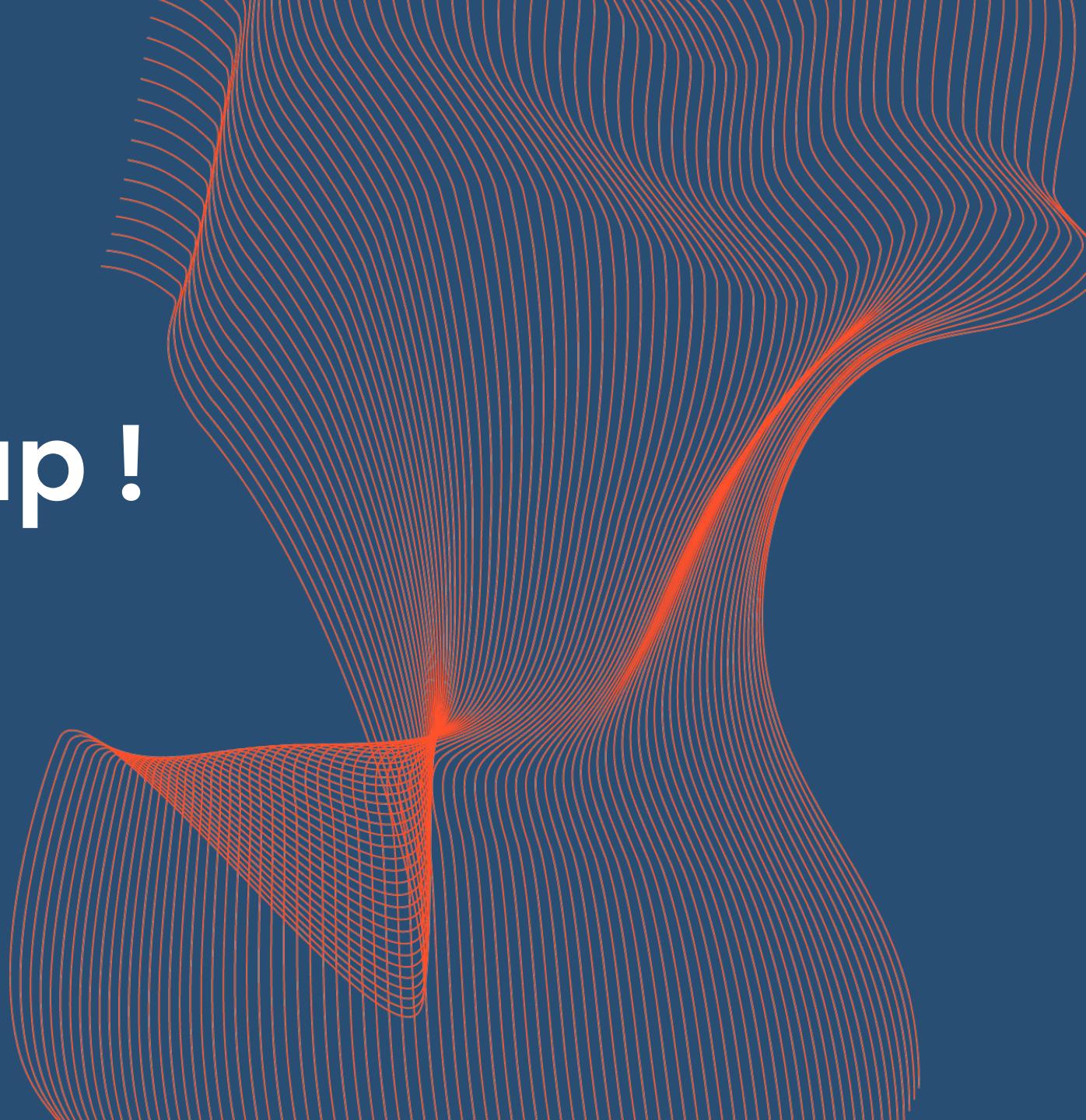
- Amélioration de l'UX
- Accélération de la navigation
- Le lazy-loading
- Le streaming

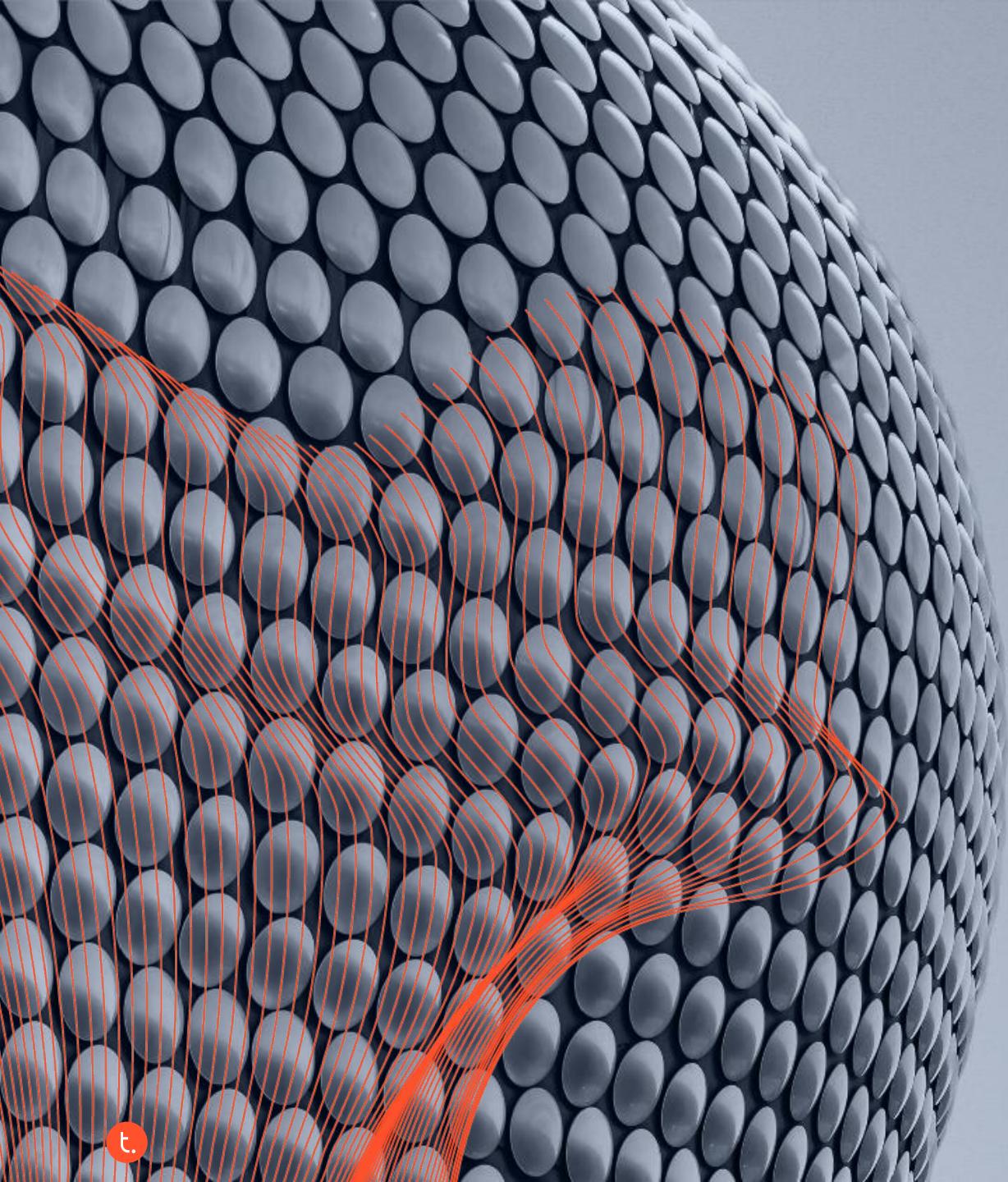
On n'a malheureusement pas pu tout voir :

- Cache
- Réactivité
- Debug
- Transitions & Animations
- ...

Merci beaucoup !

Des questions ?





theodo.

Cabinet de conseil et réalisation en digital

Les listes & la pagination

On va envoyer des pages HTML incomplètes puis les charger avec le même intersection observer

Des îlots interactifs

Alpine, Petite-Vue

_INGRÉDIENTS

-

6 personnes

+

00

00



50 g
d' emmental râpé



200 g
de lardons



1
butternut



1 rouleau
de chèvre



1 gousse
d' ail



40 cl
de crème fraîche
liquide

Des îlots interactifs

Alpine, Petite-Vue



Your new, lightweight, JavaScript framework.

```
<div x-data="{ open: false }">
  ...
</div>

<div x-bind:class="! open ? 'hidden' : ''">
  ...
</div>

<button x-on:click="open = ! open">
  Toggle
</button>
```