

NILS HARTMANN

<https://nilshartmann.net>

Moderne Webanwendungen: JavaScript oder HTMX?

NILS HARTMANN

nils@nilshartmann.net

Freiberuflicher Entwickler, Architekt, Trainer

Java, Spring, GraphQL, React, TypeScript



<https://graphql.schule/video-kurs>

<https://reactbuch.de>

[HTTPS://NILSHARTMANN.NET](https://nilshartmann.net)

NILS HARTMANN

<https://nilshartmann.net>



Moderne Webanwendungen: Java Planänderung! oder HTMX?

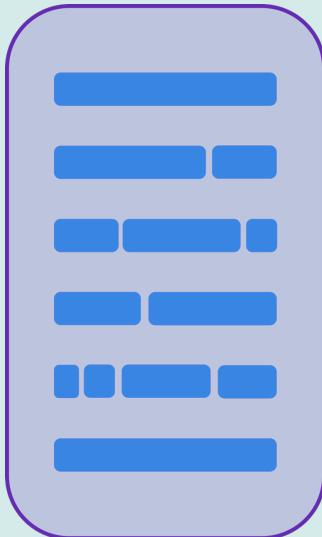
NILS HARTMANN

<https://nilshartmann.net>

Alternativen zur Single-Page-Anwendung: JavaScript oder HTMX?

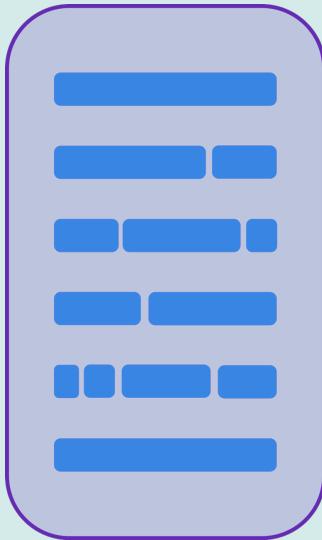
Architekturen für Webanwendungen

Single-Page-Anwendungen



- Darstellung erfolgt vollständig mit JavaScript
- Statisches HTML spielt (fast) keine Rolle
- Die Anwendung kommuniziert mit dem Backend über API
- Ausgetauscht werden Daten, aber keine UI
- Vertreter: Angular, **React**, Svelte, Vue

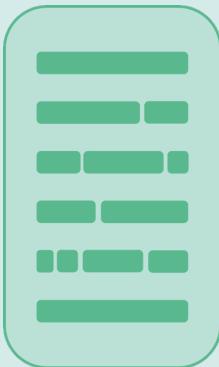
Single-Page-Anwendungen: Eure Meinung!



- Wo machen SPA Sinn? 🤔
- Welche Probleme gibt es damit? 🤯
- Gängige Vorurteile? 😡

Alternativen zur SPA

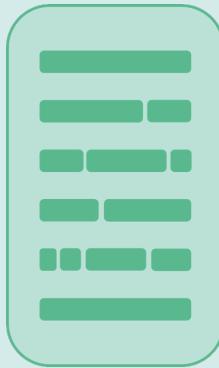
"Klassisch": Serverseitiges Rendern



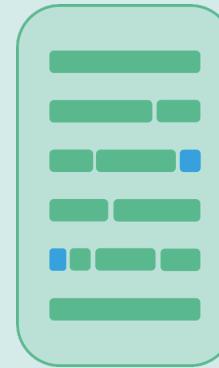
HTML-Seite

- Server liefert fertige HTML-Seite zum Client
- Für jede Interaktion dann Server-Roundtrip (Link, Formular)
- Typische Vertreter: PHP, Java, dotNET, ...

"Klassisch": Serverseitiges Rendern plus JavaScript "Schnipsel"



HTML-Seite



...mit JavaScript-Schnipseln

- Für feingranulare Updates wird JavaScript eingestreut

"Klassisch": Serverseitiges Rendern plus JavaScript "Schnipsel"



- Eigentlich optimal:
 - wir haben **JavaScript** nur da, wo wir es **wirklich** brauchen, für Interaktivität
 - alles andere kann statisches HTML und CSS sein ❤️

"Klassisch": Serverseitiges Rendern plus HTMX



HTML-Seite

- Statt Java Schnipsel HTMX

HTMX ist eine JavaScript-Bibliothek

introduction

htmx gives you access to [AJAX](#), [CSS Transitions](#), [WebSockets](#) and [Server Sent Events](#) directly in HTML, using [attributes](#), so you can build [modern user interfaces](#) with the [simplicity](#) and [power](#) of [hypertext](#)

htmx is small (~14k min.gz'd), [dependency-free](#), [extendable](#), IE11 compatible & has [reduced](#) code base sizes by 67% when compared with react

<https://htmx.org/>

Versprechen: Aktualisierung der Darstellung ohne selbst JavaScript **schreiben**
zu müssen
(stattdessen "hypertext" 🤯)

HTMX - Grundlagen

Demo: Hello World, HTMX!

```
<html lang="en">
  <body>
    <div hx-get="/hello-world"
          hx-trigger="click"
          hx-target="#result">
      <div id="result"></div>
    </div>
  </body>
</html>
```

The code snippet illustrates the basic structure of an HTMX request. It consists of an HTML document with a body containing a single div element. This div uses three HTMX attributes: `hx-get` to specify the URL for the initial request, `hx-trigger` to indicate when the request should be triggered (in this case, on a click event), and `hx-target` to specify the ID of the target element where the response should be inserted. The response itself contains another div with the ID `result`.

HTMX - Grundlagen

- HTML-Elemente werden mit HTMX-Attributen ergänzt

```
<html lang="en">
  <body>
    <div hx-get="/hello-world"
          hx-trigger="click"
          hx-target="#result">
      Get Greeting
    </div>

    <div id="result"></div>

    <script
      type="text/javascript"
      th:src="@{/htmx/htmx-1.9.10.min.js}"
    ></script>
  </body>
</html>
```

HTMX - Grundlagen

- HTML-Elemente werden mit HTMX-Attributen ergänzt
- Damit wird beschrieben, welche Server Requests bei einem "Trigger" ausgeführt werden sollen

```
<html lang="en">
  <body>
    <div hx-get="/hello-world"
          hx-trigger="click"
          hx-target="#result">
      Get Greeting
    </div>

    <div id="result"></div>

    <script
      type="text/javascript"
      th:src="@{/htmx/htmx-1.9.10.min.js}"
    ></script>
  </body>
</html>
```

HTMX - Grundlagen

- HTML-Elemente werden mit HTMX-Attributen ergänzt
- Damit wird beschrieben, welche Server Requests bei einem "Trigger" ausgeführt werden sollen
- HTMX kümmert sich um die Ausführung des Requests und die Verarbeitung der Antwort

```
<html lang="en">  
  <body>  
    <div hx-get="/hello-world"  
         hx-trigger="click"  
         hx-target="#result">  
      Get Greeting  
    </div>  
  
    <div id="result"></div>  
  
    <script  
        type="text/javascript"  
        th:src="@{/htmx/htmx-1.9.10.min.js}"  
    ></script>  
  </body>  
</html>
```

HTMX – Der Server: Anforderungen

- Der Server nimmt (Ajax-)Requests entgegen

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class HelloWorldController {

    @GetMapping(value = "/hello-world",
                headers = "HX-Request")
    String helloWorldResponse() {
        return "hello-world-response";
    }
}
```

HTMX – Der Server: Anforderungen

- Der Server nimmt (Ajax-)Requests entgegen
- HTMX-Requests sind per HTTP Header zu identifizieren

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class HelloWorldController {

    @GetMapping(value = "/hello-world",
                headers = "HX-Request")
    String helloWorldResponse() {
        return "hello-world-response";
    }
}
```

HTMX – Der Server: Anforderungen

- Der Server nimmt (Ajax-)Requests entgegen
- HTMX-Requests sind per HTTP Header zu identifizieren
- Der Server liefert dann HTML-Schnipsel

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class HelloWorldController {

    @GetMapping(value = "/hello-world",
                headers = "HX-Request")
    String helloWorldResponse() {
        return "hello-world-response";
    }
}
```

HTMX – Der Server: Anforderungen

- Der Server nimmt (Ajax-)Requests entgegen
- HTMX-Requests sind per HTTP Header zu identifizieren
- Der Server liefert dann HTML-Schnipsel
- Wie die Schnipsel erzeugt werden, entscheidet der Server bzw. ihr

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class HelloWorldController {

    @GetMapping(value = "/hello-world",
                headers = "HX-Request")
    String helloWorldResponse() {
        return "hello-world-response";
    }
}
```

HTMX – Der Server: Anforderungen

- Der Server nimmt (Ajax-)Requests entgegen
- HTMX-Requests sind per HTTP Header zu identifizieren
- Der Server liefert dann HTML-Schnipsel
- Wie die Schnipsel erzeugt werden, entscheidet der Server bzw. ihr
- Zum Beispiel mit **Template-Sprache**

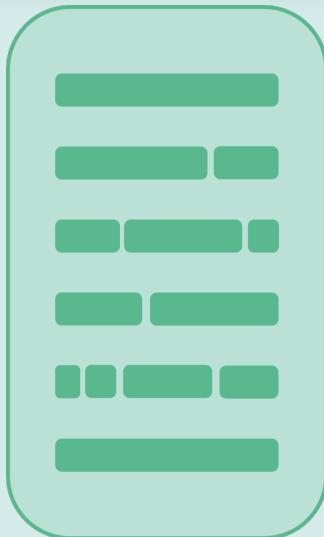
```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class HelloWorldController {

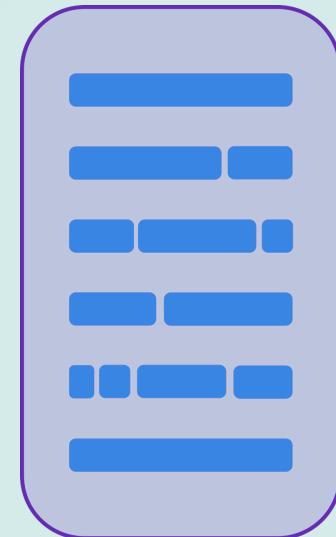
    @GetMapping(value = "/hello-world",
                headers = "HX-Request")
    String helloWorldResponse() {
        return "hello-world-response";
    }
}
```

"Fullstack"-Anwendungen (Meta-Frameworks)

"Fullstack"-Anwendungen (Meta-Frameworks)

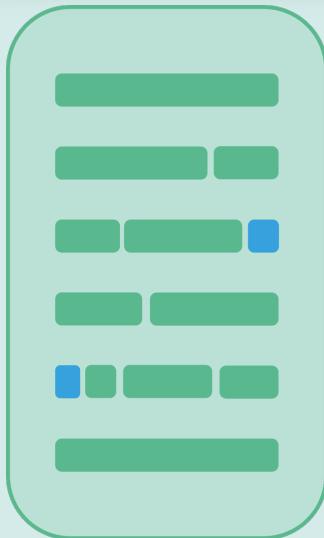


Nur Server (+JS)

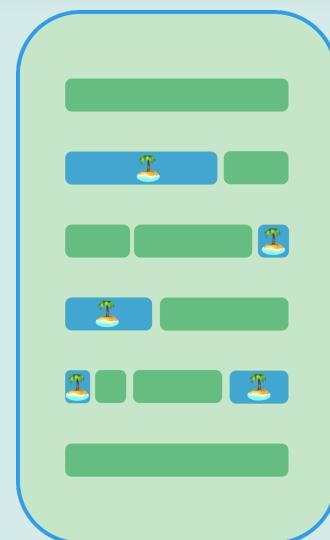


Nur Client (SPA)

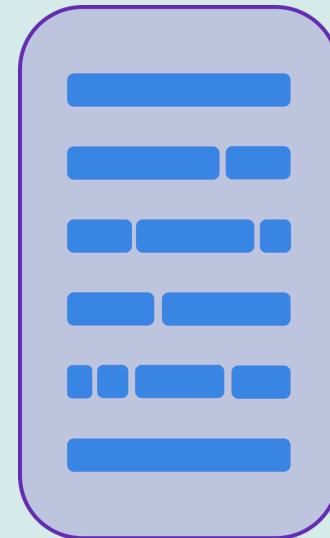
"Fullstack"-Anwendungen (Meta-Frameworks)



Nur Server (+JS)



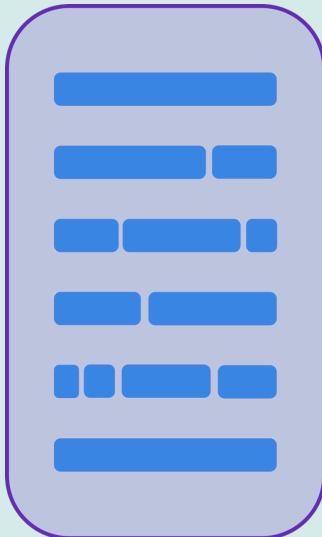
JS Fullstack Anwendung



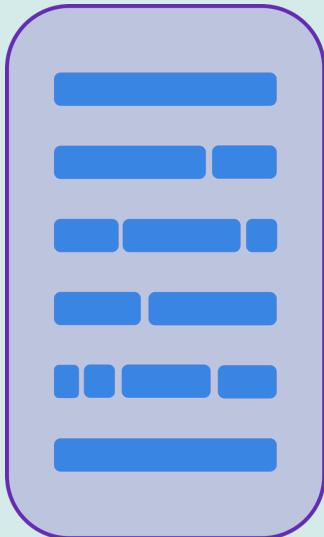
Nur Client (SPA)

FULLSTACK-ANWENDUNGEN (JAVASCRIPT)

Fullstack-Anwendungen (JavaScript)

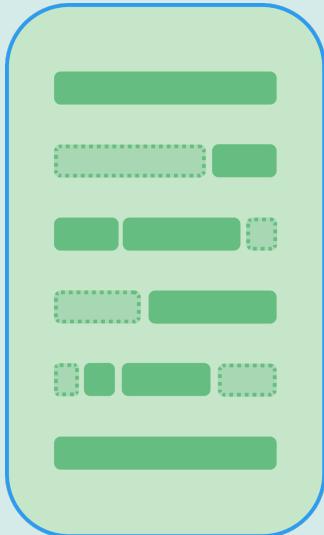


Fullstack-Anwendungen (JavaScript)



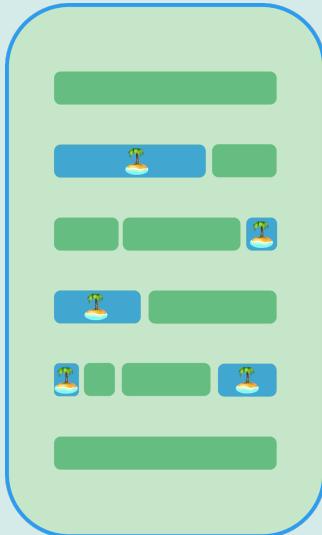
- Ebenfalls vollständig in **JavaScript** geschrieben

Fullstack-Anwendungen (JavaScript)



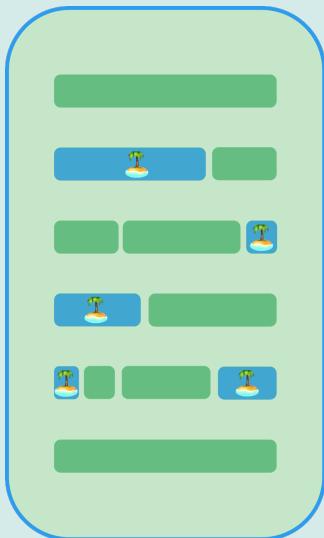
- Grundsätzliche Idee:
 1. **UI-Code** wird serverseitig vorgerendert
 2. **UI-Code** wird zum Browser gesendet und angezeigt

Fullstack-Anwendungen (JavaScript)



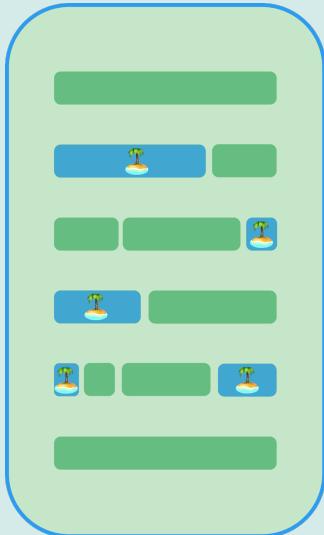
- Grundsätzliche Idee:
 1. **UI-Code** wird serverseitig vorgerendert
 2. **UI-Code** wird zum Browser gesendet und angezeigt
 3. Nur der JavaScript-Code ("Islands") **für Interaktionen** wird zum Browser geschickt

Fullstack-Anwendungen (JavaScript)



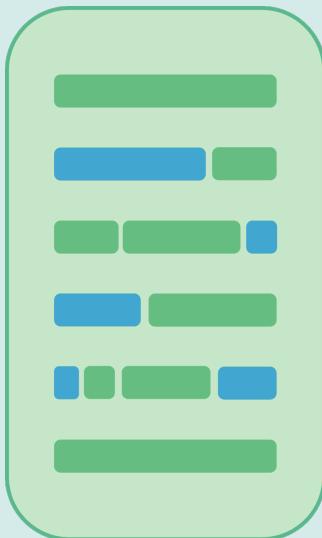
- Bekannte Vertreter:
 1. **Next.js** (React)
 2. SvelteKit (Svelte)
 3. Nuxt.js (Vue)
 4. Astro (eigenes Framework + Support für alle SPAs)
 5. Qwik (eigenes Framework)

Fullstack-Anwendungen (JavaScript)



- Bekannte Vertreter:
 1. **Next.js** (React)
 2. SvelteKit (Svelte)
 3. Nuxt.js (Vue)
 4. Astro (eigenes Framework + Support für alle SPAs)
 5. Qwik (eigenes Framework)
- Funktionalität und Herangehensweise unterschiedlich

Fullstack-Anwendungen (JavaScript)



- Wir sind zurück zur **JavaScript-Schnipsel**-Architektur
 - aber: die Schnipsel werden **automatisch** vom Framework erzeugt
 - die Schnipsel existieren nur zur **Laufzeit**

Fullstack

oder

htmx



BEISPIEL-ANWENDUNG

- Eine Beispiel-Anwendung...
- <http://localhost:8090>

CODE-BEISPIELE



Code-Beispiele im Editor!

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

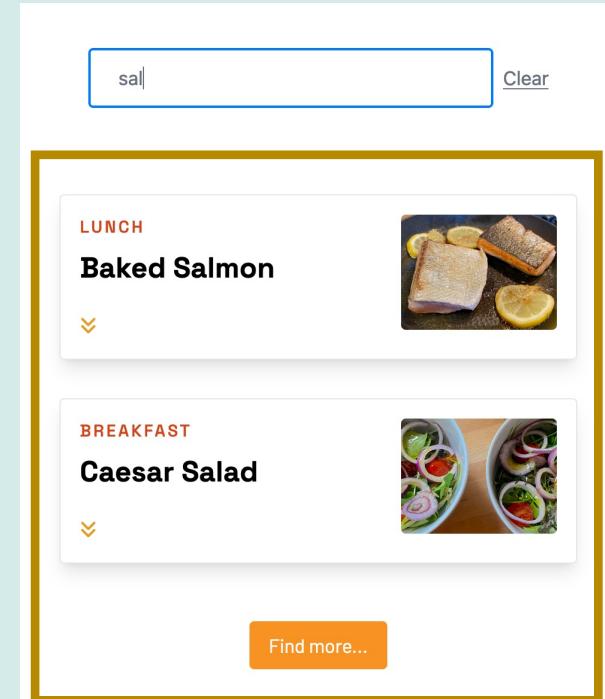
 [Clear](#)

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

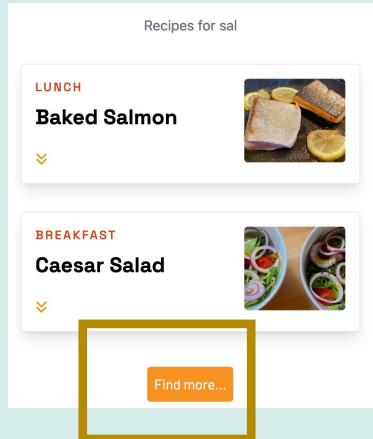
 [Clear](#)

GET /search



CODE-BEISPIELE

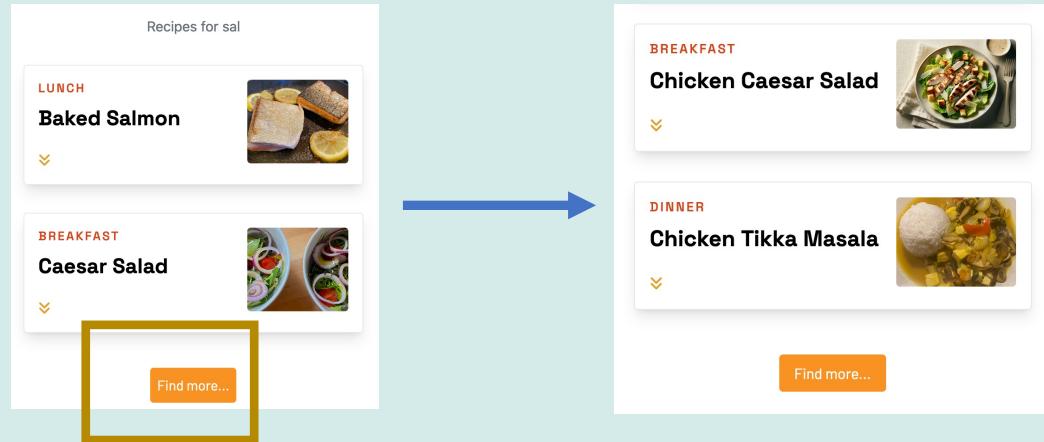
Beispiel: Bestehende Darstellung aktualisieren



GET /search?page=2

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren



GET /search?page=2

hx-swap="beforeend"

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

Recipes for sal

LUNCH

Baked Salmon

▼



[Find more...](#)

BREAKFAST

Caesar Salad

▼



HTMX Request

BREAKFAST

Chicken Caesar Salad

▼



DINNER

Chicken Tikka Masala

▼



[Find more...](#)

GET /search?page=2

hx-swap="beforeend"

Recipes for sal

LUNCH

Baked Salmon

▼



[Find more...](#)

BREAKFAST

Caesar Salad

▼



[Find more...](#)

BREAKFAST

Chicken Caesar Salad

▼



DINNER

Chicken Tikka Masala

▼



[Find more...](#)

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

Recipes for sal

LUNCH
Baked Salmon

BREAKFAST
Caesar Salad

[Find more...](#)

HTMX Request

BREAKFAST
Chicken Caesar Salad

DINNER
Chicken Tikka Masala

[Find more...](#)

GET /search?page=2

hx-swap="beforeend"



Recipes for sal

LUNCH
Baked Salmon

BREAKFAST
Caesar Salad

[Find more...](#)

BREAKFAST
Chicken Caesar Salad

DINNER
Chicken Tikka Masala

[Find more...](#)

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

[Find more...](#)

HTMX Request

Recipes for sal

BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

[Find more...](#)

GET /search?page=2

hx-swap="beforeend"



Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

[Find more...](#)

GET /search?page=2

BREAKFAST
Chicken Caesar Salad
▼ 

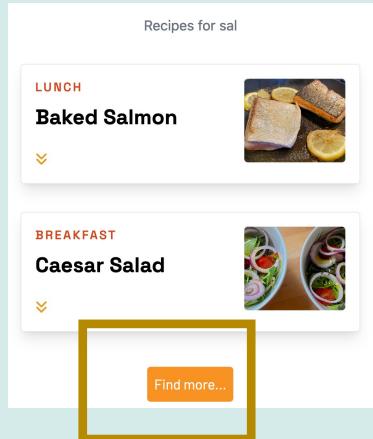
DINNER
Chicken Tikka Masala
▼ 

[Find more...](#)

GET /search?page=3

CODE-BEISPIELE

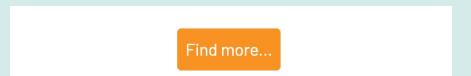
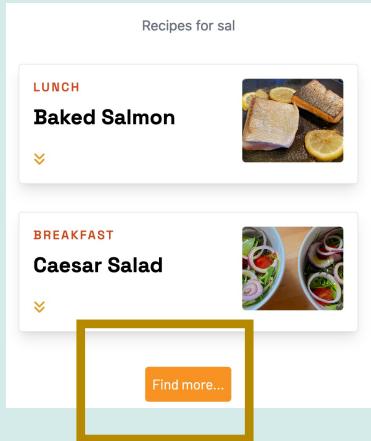
Beispiel: Bestehende Darstellung aktualisieren



GET /search?page=2

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren



```
id="moreButton"  
hx-swap-oob="true"
```

GET /search?page=2

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

Recipes for sal

LUNCH

Baked Salmon

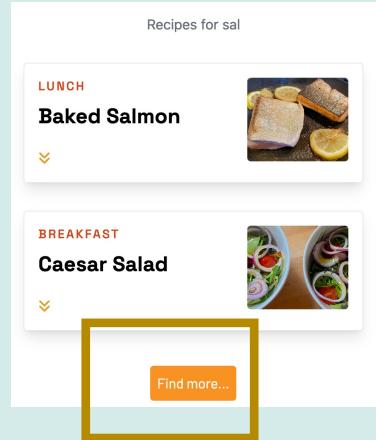
▼

BREAKFAST

Caesar Salad

▼

[Find more...](#)



HTMX Request

[Find more...](#)

`id="moreButton"`

`hx-swap-oob="true"`

BREAKFAST

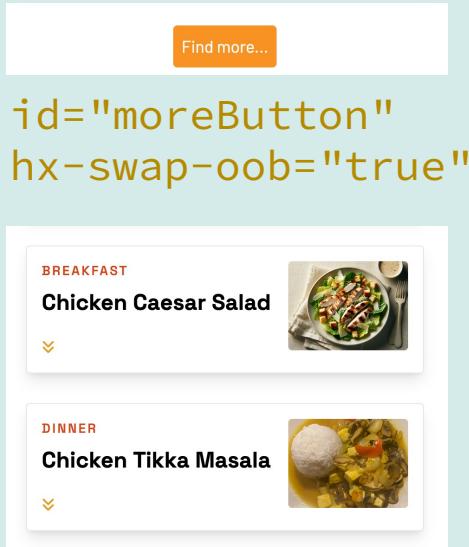
Chicken Caesar Salad

▼

DINNER

Chicken Tikka Masala

▼



`hx-target="result"`

`hx-swap="beforeend"`

GET /search?page=2

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

Recipes for sal

- LUNCH
Baked Salmon
▼ 
- BREAKFAST
Caesar Salad
▼ 

Find more...

HTMX Request

`id="moreButton"
hx-swap-oob="true"`

Find more...

- BREAKFAST
Chicken Caesar Salad
▼ 
- DINNER
Chicken Tikka Masala
▼ 

`hx-target="result"
hx-swap="beforeend"`



Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

Find more...

GET /search?page=2

GET /search?page=3

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

Find more...

HTMX Request

GET /search?page=2

id="moreButton"
hx-swap-oob="true"

BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

hx-target="result"
hx-swap="beforeend"



Find more...

GET /search?page=3

Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

[Find more...](#)

GET /search?page=2

FindMoreButton.jte

RecipeSearch.tsx

Find more...

`id="moreButton"
hx-swap-oob="true"`

BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

`hx-target="result"
hx-swap="beforeend"`



Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

BREAKFAST
Chicken Caesar Salad
▼ 

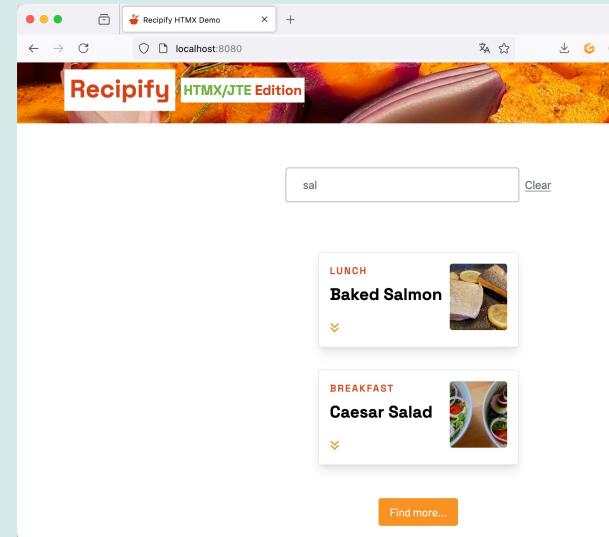
DINNER
Chicken Tikka Masala
▼ 

[Find more...](#)

GET /search?page=3

Beispiel: Parallel Requests

-  Netzwerk-Tab

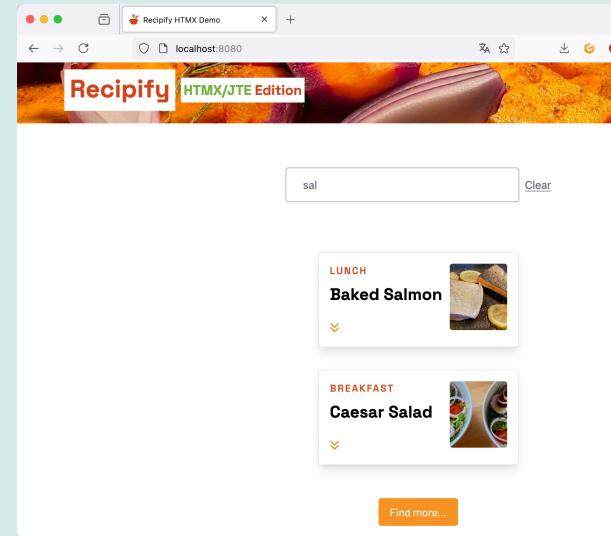


```
SearchPageContent.jte: slowdown=2000, dafür delay raus
```

```
demo-config: slowDown_search=2000, debounce_search=0
```

Beispiel: Parallel Requests

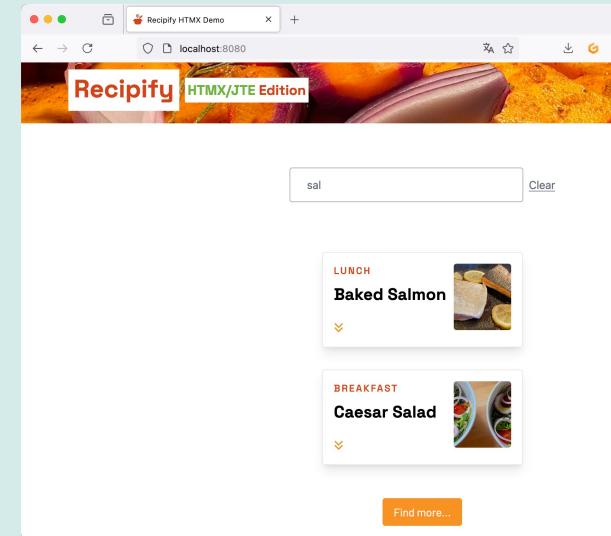
- 🕵️ Netzwerk-Tab
- HTMX führt Requests nacheinander aus
- In der SPA können wir das selbst entscheiden



Beispiel: Parallel Requests

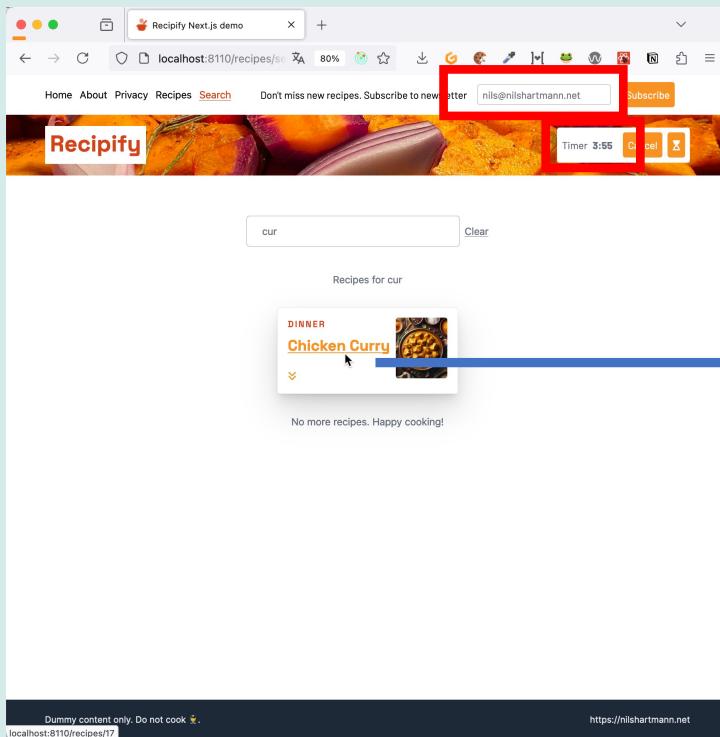
- 🕵️ Netzwerk-Tab
- HTMX führt Requests nacheinander aus
- "Alte" Request werden nicht verworfen
- In der SPA können wir das selbst entscheiden

SearchPageContent.jte: slowdown

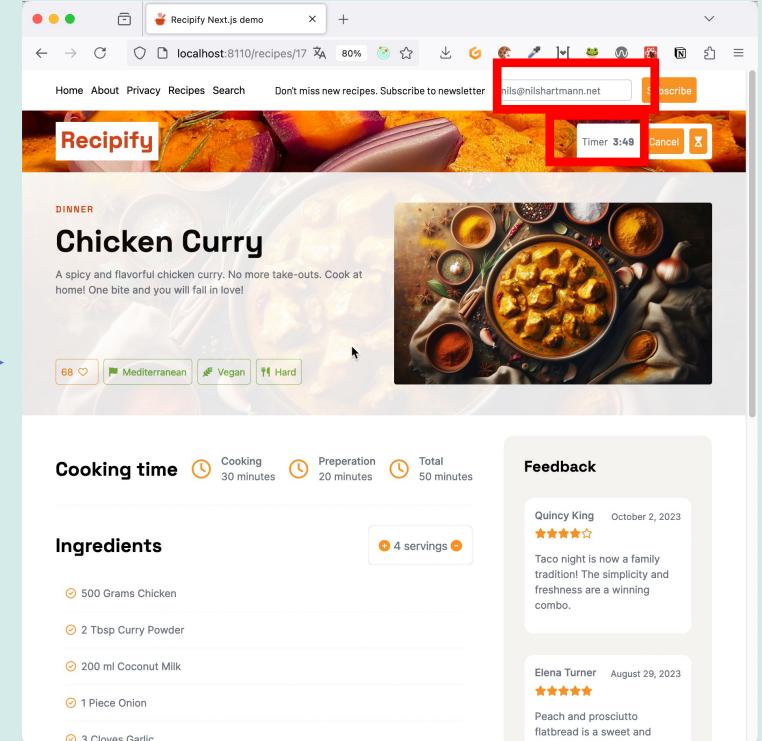


Beispiel: SPA-like Seitenwechsel

- Beim Aufruf der Rezept-Seite soll der Rahmen unverändert bleiben



/recipes/17



Beispiel: SPA-like Seitenwechsel in Next.js

- **Link** verhält sich automatisch richtig:

- mit und ohne JS
- Ganze Seite vs. nur Content
- Aktualisieren der URL

```
<Link  
  href={`/recipes/${recipe.id}`}  
>  
  {recipe.title}  
</Link>
```

(RecipeSummaryCard.tsx)

Beispiel: SPA-like Seitenwechsel in Next.js

- Link verhält sich automatisch richtig:

- mit und ohne JS
- Ganze Seite vs. nur Content
- Aktualisieren der URL

```
<Link  
  href={`/recipes/${recipe.id}`}  
>  
  {recipe.title}  
</Link>
```

(RecipeSummaryCard.tsx)

- "Controller": Identischer Code
 - Fullpage Request
 - Incremental Update

[recipeId]/page.tsx

recipes/layout.tsx

Beispiel: SPA-like Seitenwechsel in Next.js

- Link verhält sich automatisch richtig:

- mit und ohne JS
- Ganze Seite vs. nur Content
- Aktualisieren der URL

```
<Link  
  href={`/recipes/${recipe.id}`}  
>  
  {recipe.title}  
</Link>
```

(RecipeSummaryCard.tsx)

- "Controller": Identischer Code

- Fullpage Request
- Incremental Update

[recipeId]/page.tsx

recipes/layout.tsx

- Das funktioniert **clientseitig** mit allen SPA-Routern

Beispiel: SPA-like Seitenwechsel in HTMX

- Link muss entsprechend ausgezeichnet werden

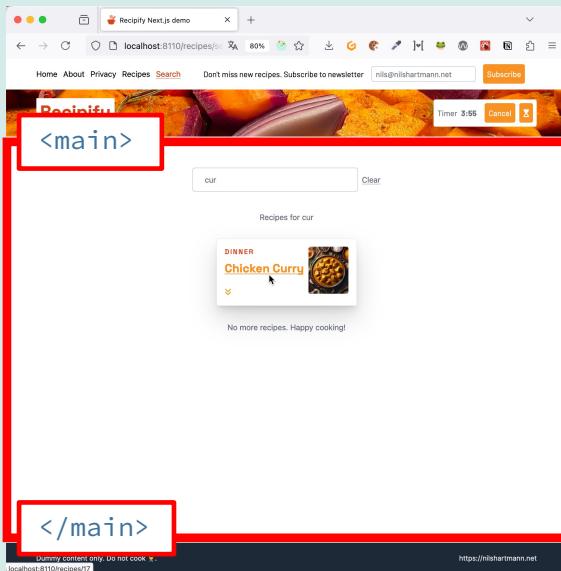
- Für Klick ohne JavaScript
- Für Klick mit JavaScript
- Aktualisieren der URL im Browser

RecipeSummaryCard.jte

Beispiel: SPA-like Seitenwechsel in HTMX

Variante 1

RecipeSummaryCard.jte

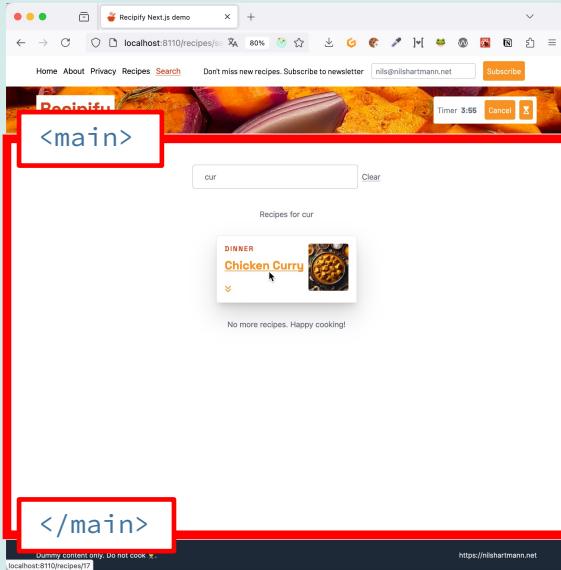


CODE-BEISPIELE

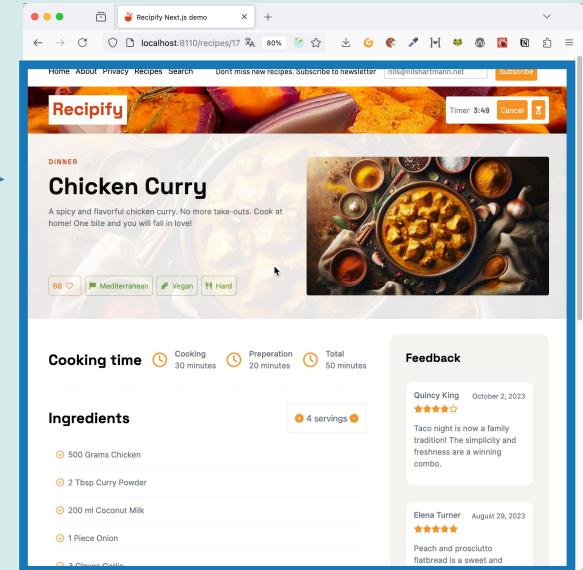
Beispiel: SPA-like Seitenwechsel in HTMX

Variante 1

RecipeSummaryCard.jte



/recipes/17

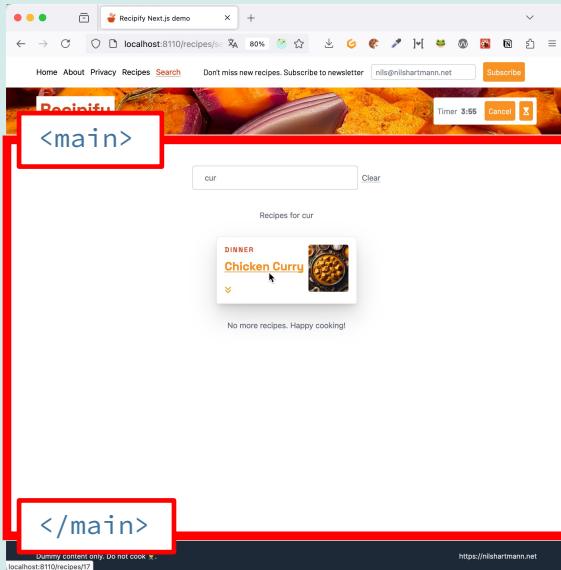


CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

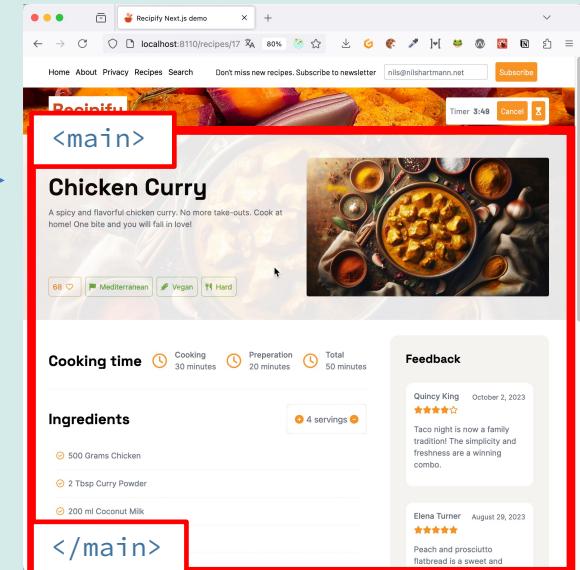
Variante 1

RecipeSummaryCard.jte



/recipes/17

hx-select="main"

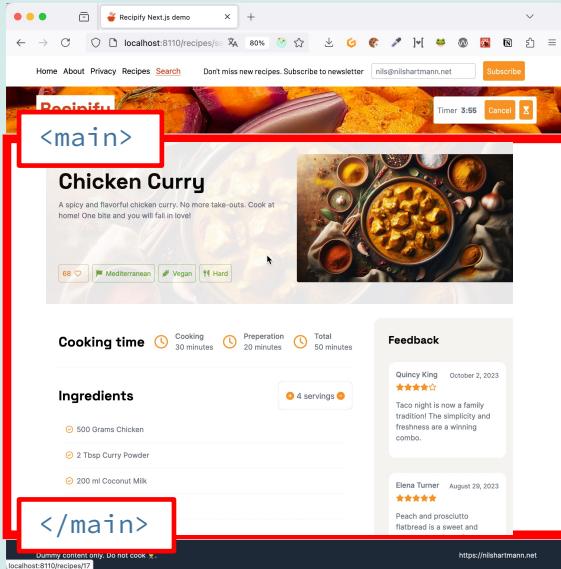


CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

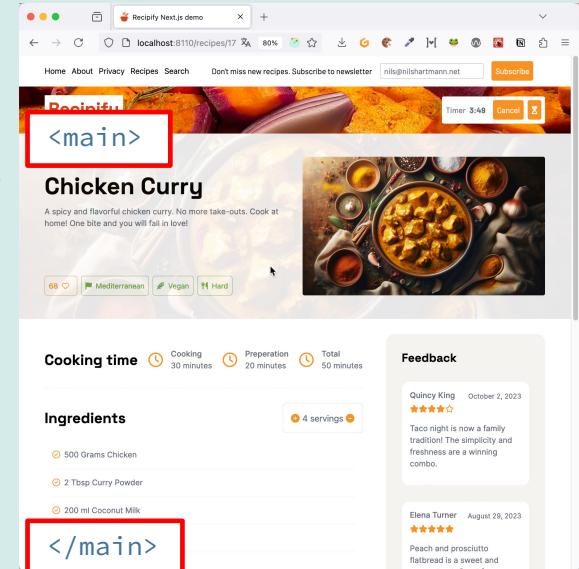
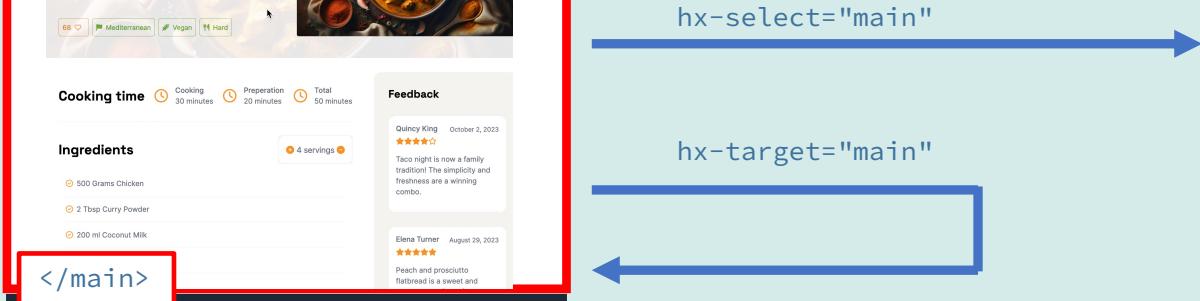
Variante 1

RecipeSummaryCard.jte



/recipes/17

hx-select="main"



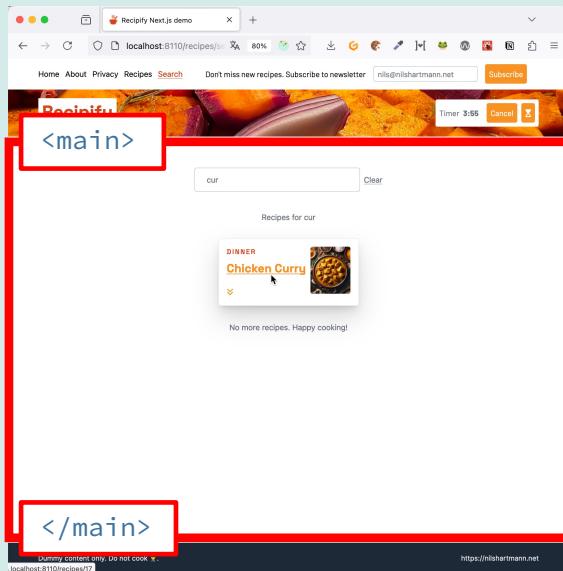
Variante 1: HTMX

- **Link** muss wissen:

- wie sieht das Ergebnis aus
- welches Element brauche ich daraus
- wo muss das auf der aktuellen Seite eingebaut werden

Beispiel: SPA-like Seitenwechsel in HTMX

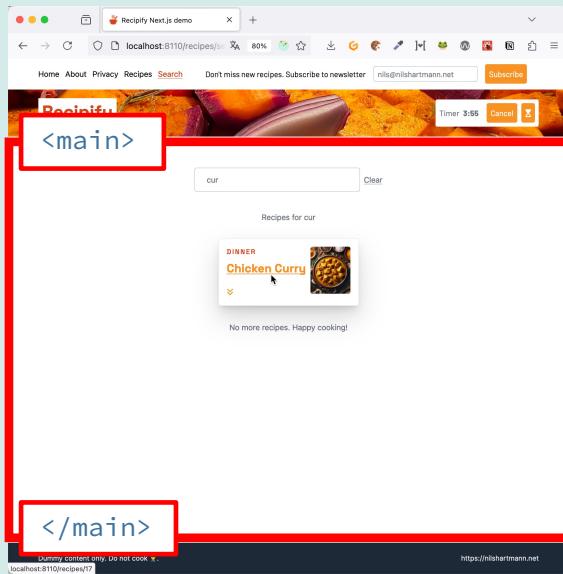
Variante 2: Eigener Endpunkt



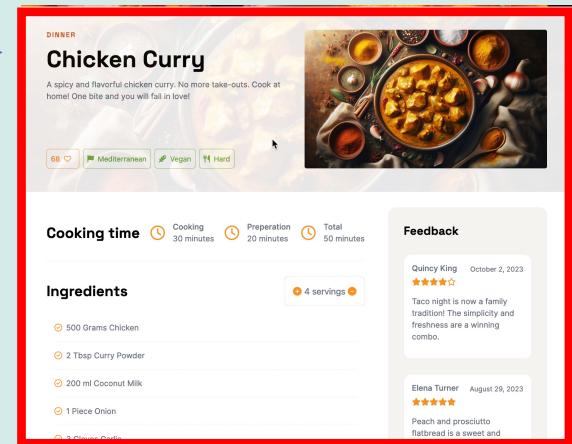
CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

Variante 2: Eigener Endpunkt



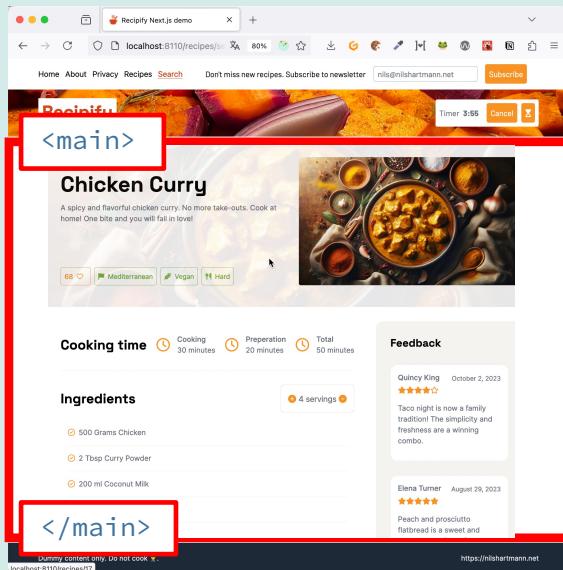
/recipes/17/details



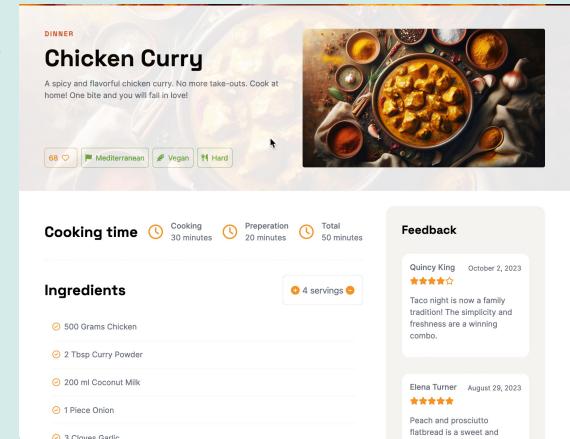
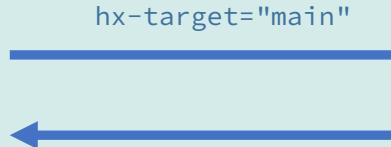
CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

Variante 2



/recipes/17/details



Variante 2: Eigener Endpunkt

- **Link** muss wissen:

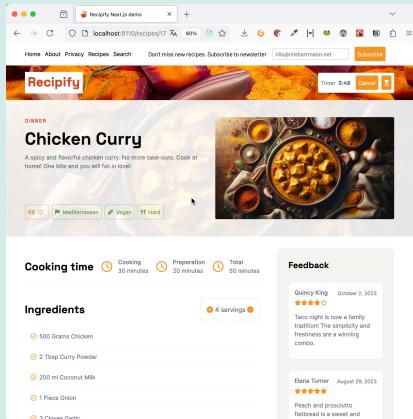
- welcher Endpunkt
- wo muss Ergebnis auf der aktuellen Seite eingebaut werden

Variante 2: Eigener Endpunkt

- **Backend** muss zwei Endpunkte zur Verfügung stellen:

Variante 2: Eigener Endpunkt

- **Backend** muss zwei Endpunkte zur Verfügung stellen:
 - Ganze Seite



`/recipes/17`

Variante 2: Eigener Endpunkt

- Backend muss zwei Endpunkte zur Verfügung stellen:
 - Ganze Seite
 - Nur den Mittelteil

The screenshot shows a web browser window with the title 'Recipify Nextjs demo'. The main content is a recipe for 'Chicken Curry'. At the top, there's a navigation bar with links for Home, About, Privacy, Recipes, Search, and a newsletter sign-up. Below the navigation is a banner for 'Recipify' with a photo of a dish. The main section has a heading 'DINNER' and 'Chicken Curry'. A sub-headline says 'A spicy and flavorful chicken curry for more take-outs. Cook at home! Once bite and you will fall in love!' Below this is a large image of the dish. To the right of the image are buttons for '68', 'Mediterranean', 'Vegan', and 'Hard'. Under the heading, there are sections for 'Cooking time' (30 minutes), 'Preparation' (20 minutes), and 'Total' (50 minutes). A 'Feedback' section contains two reviews: one from 'Quincy King' (Oct 02, 2023) and one from 'Elana Turner' (Aug 29, 2023). At the bottom, there's a 'Ingredients' section listing: 500 Grams Chicken, 2 Tbsp Curry Powder, 200 ml Coconut Milk, 1 Peice Onion, and 3 Cloves Garlic. It also indicates '4 servings'.

/recipes/17

This screenshot shows the same recipe page but with a different URL: '/recipes/17/details'. The layout is identical to the first screenshot, featuring the 'Recipify' banner, the 'Chicken Curry' recipe card with its details, cooking time, preparation time, total time, and feedback section. The ingredients list is also present at the bottom.

/recipes/17/details

CODE-BEISPIELE

Beispiel: Seite priorisieren

The screenshot shows a recipe card for "Classic Caesar Salad" on a website called Recipify. The card is structured with priority levels:

- Header:** "Recipify" logo at the top right, above the title.
- Title:** "Classic Caesar Salad" in large bold text.
- Catagory:** "BREAKFAST" in small text above the title.
- Description:** "Crispy romaine lettuce with creamy Caesar dressing. Bon Appétit! Food that feels like home."
- Image:** A large, appetizing photo of the salad.
- Meta Data:** Buttons for "92 ❤️", "Asian", "Vegan", and "Easy".
- Time:** "Cooking time" (clock icon), "Preparation 10 minutes" (clock icon), and "Total 10 minutes" (clock icon).
- Ingredients:** A list of ingredients: "1 Head Romaine Lettuce", "100 ml Caesar Dressing", "100 Grams Croutons", and "50 Grams Parmesan".
- Instructions:** "Instructions" section below the ingredients.
- Feedback:** "Feedback" section with "Loading feedback..." and a progress bar.
- User Input:** "Your opinion?" section with fields for "Your name:" and "Your rating:" (with five star icons).

CODE-BEISPIELE

Beispiel: Seite priorisieren in Next.js

慢速请求反馈慢速请求反馈
slowDown_GetFeedbacks=3000

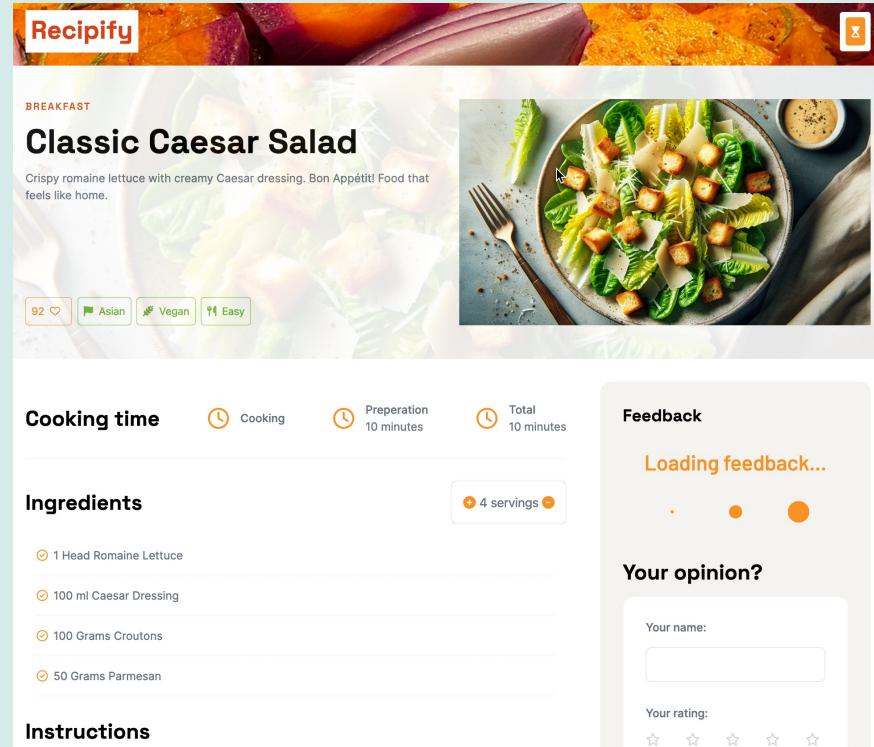
慢速请求慢速请求
slowDown_GetRecipe=2000

慢速请求慢速请求
Spring Controller Debug

慢速请求慢速请求
Network-Tab Total Duration

慢速请求慢速请求
[recipeId]/page.tsx

慢速请求慢速请求
RecipePageContent.tsx



CODE-BEISPIELE

Beispiel: Seite priorisieren in HTMX

The screenshot shows a recipe card for "Classic Caesar Salad" on a website called Recipify. The card is divided into several sections:

- Header:** The word "Recipify" is in the top right corner of the image.
- Catagory:** "BREAKFAST" is listed under the title.
- Title:** "Classic Caesar Salad" is the main title.
- Description:** A brief description follows: "Crispy romaine lettuce with creamy Caesar dressing. Bon Appétit! Food that feels like home."
- Image:** A large, appetizing photo of the salad is on the right side.
- Meta Data:** Below the description are four small boxes: "92 ❤️", "Asian", "Vegan", and "Easy".
- Time & Servings:** "Cooking time" (clock icon), "Preparation 10 minutes" (clock icon), "Total 10 minutes" (clock icon), and "4 servings" (clock icon).
- Ingredients:** A list of ingredients with circular icons: "1 Head Romaine Lettuce", "100 ml Caesar Dressing", "100 Grams Croutons", and "50 Grams Parmesan". To the right is a box for "4 servings".
- Instructions:** The word "Instructions" is at the bottom left of the card.
- Feedback:** A section on the right titled "Feedback" contains the text "Loading feedback..." and three orange dots.
- Your opinion?**: A section on the far right with fields for "Your name:" (empty input field) and "Your rating:" (five empty star icons).

CODE-BEISPIELE

Beispiel: Seite priorisieren in HTMX



Feedback.jte



<http://localhost:8080/recipes/25?slowdown=2000>



Netzwerk-Tab

The screenshot shows a recipe card for "Classic Caesar Salad" from the "BREKFAST" section. The card includes a thumbnail image of the salad, cooking time (10 minutes), preparation time (10 minutes), ingredients (Romaine Lettuce, Caesar Dressing, Croutons, Parmesan), instructions, and a feedback section with a rating of 3 stars.

Recipify

BREAKFAST

Classic Caesar Salad

Crispy romaine lettuce with creamy Caesar dressing. Bon Appétit! Food that feels like home.

92 ❤️ Asian Vegan Easy

Cooking time Cooking 10 minutes Preparation 10 minutes Total 10 minutes

Ingredients

- 1 Head Romaine Lettuce
- 100 ml Caesar Dressing
- 100 Grams Croutons
- 50 Grams Parmesan

Instructions

Feedback

Loading feedback...

Your opinion?

Your name:

Your rating:

Beispiel: Clientseitige Interaktion

- Clientseitige Interaktion

Ingredients

+ 4 servings -

✓ 1.0 Piece Chicken Breast

✓ 1.0 Head Romaine Lettuce

✓ 100.0 ml Caesar Dressing

✓ 100.0 Grams Croutons

✓ 50.0 Grams Parmesan

Beispiel: Clientseitige Interaktion in Next.js



IngredientsSection.tsx

Ingredients

+ 4 servings -

✓ 1.0 Piece Chicken Breast

✓ 1.0 Head Romaine Lettuce

✓ 100.0 ml Caesar Dressing

✓ 100.0 Grams Croutons

✓ 50.0 Grams Parmesan

Beispiel: Clientseitige Interaktion in HTMX

- Wir brauchen spätestens hier JavaScript



Ingredients.jte

Ingredients

+ 4 servings -

✓ 1.0 Piece Chicken Breast

✓ 1.0 Head Romaine Lettuce

✓ 100.0 ml Caesar Dressing

✓ 100.0 Grams Croutons

✓ 50.0 Grams Parmesan

Beispiel: Clientseitige Interaktion in HTMX

- Wir brauchen spätestens hier JavaScript



SearchPageContent.jte

The screenshot shows a search interface. At the top right is a small icon of a person wearing a hat and coat. To its right is the text "SearchPageContent.jte". Below this is a search bar with a placeholder text "Type three letters to start search" enclosed in a yellow rectangular border. To the right of the search bar is a "Clear" button.

Webanwendungen

JavaScript

oder

htmx



Webanwendungen

Fazit

JavaScript

oder

htmx



SPA/Fullstack oder HTMX

- It depends (natürlich 😐)

SPA/Fullstack oder HTMX

- It depends (natürlich 😐)
- Will ich eine "Hypertext"-Anwendung bauen
oder
- eine Anwendung, die die Anforderungen meiner Benutzer:innen erfüllt? 😊

SPA/Fullstack oder HTMX

- Für mich ist "wir machen nur HTML plus bisschen JS" zu kurz gesprungen
- Das sieht man auch an der Vielzahl der HTMX-Attribute

Executive Summary

- The effort took about 2 months (with a 21K LOC code base, mostly JavaScript)
- No reduction in the application's user experience (UX)
- They reduced the code base size by 67% (21,500 LOC to 7200 LOC)
- They **increased python code by 140%** (500 LOC to 1200 LOC), a good thing if you prefer python to JS
- They reduced their total JS dependencies by 96% (255 to 9)
- They reduced their web build time by 88% (40 seconds to 5)

Executive Summary

These:

Wenn "if you prefer YOUR-BACKEND-LANGUAGE-HERE to JS" zutrifft,

dann ist man vielleicht falsch in der Frontend-Entwicklung



reduced python code by 140% (500 LOC to 1200 LOC), a good thing if
you prefer python to JS

- They reduced their total JS dependencies by 96% (255 to 9)
- They reduced their web build time by 88% (40 seconds to 5)

Webanwendung

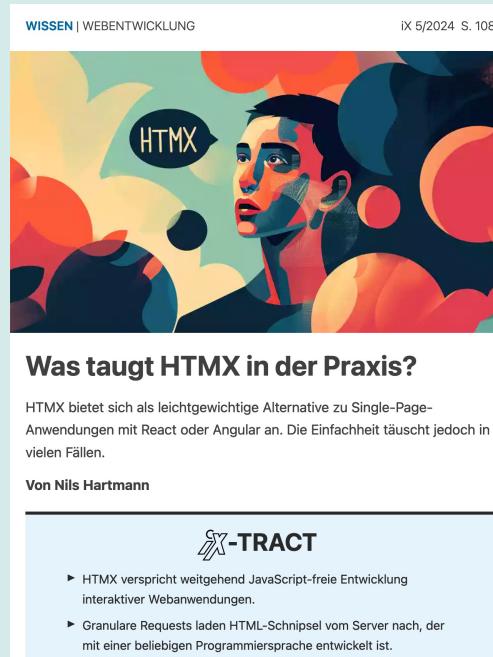
Fazit

Eure
Erkenntnisse?

itmX



Mein Artikel (heise / iX) zum Thema: <https://react.schule/heise-htmx>



The screenshot shows a magazine page from 'WISSEN | WEBENTWICKLUNG' issue 'iX 5/2024 S. 108'. The main image is a colorful illustration of a person's face surrounded by abstract shapes, with a speech bubble containing the word 'HTMX'. Below the image, the title 'Was taugt HTMX in der Praxis?' is displayed. A short text explains that HTMX is a lightweight alternative to Single-Page-Anwendungen like React or Angular, noting that its simplicity can be misleading. The author is listed as 'Von Nils Hartmann'. At the bottom, there is a section titled 'iX-TRACT' with two bullet points: 'HTMX verspricht weitgehend JavaScript-freie Entwicklung interaktiver Webanwendungen.' and 'Granulare Requests laden HTML-Schnipsel vom Server nach, der mit einer beliebigen Programmiersprache entwickelt ist.'

WISSEN | WEBENTWICKLUNG
iX 5/2024 S. 108

Was taugt HTMX in der Praxis?

HTMX bietet sich als leichtgewichtige Alternative zu Single-Page-Anwendungen mit React oder Angular an. Die Einfachheit täuscht jedoch in vielen Fällen.

Von Nils Hartmann

iX-TRACT

- HTMX verspricht weitgehend JavaScript-freie Entwicklung interaktiver Webanwendungen.
- Granulare Requests laden HTML-Schnipsel vom Server nach, der mit einer beliebigen Programmiersprache entwickelt ist.

Vielen Dank!

Code: <https://react.schule/jugms-htmx>

Fragen und Kontakt:

nils@nilshartmann.net

<https://nilshartmann.net/kontakt>

