

NILS HARTMANN

<https://nilshartmann.net>



Single- Page- Anwendung oder Fullstack HTMX?

NILS HARTMANN

nils@nilshartmann.net

Freiberuflischer Entwickler, Architekt, Trainer

Java, Spring, GraphQL, React, TypeScript



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

<https://reactbuch.de>

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

INTRO

Single-Page-Anwendung, Fullstack oder HTMX?

Single-Page-Anwendung, Fullstack oder HTMX?

- SPA und Fullstack-Apps sind **Pattern** oder **Architekturen**, keine konkreten Frameworks

Single-Page-Anwendung, Fullstack oder HTMX?

- SPA und Fullstack-Apps sind **Pattern** oder **Architekturen**, keine konkreten Frameworks
- Wir müssten bei einem Vergleich eigentlich ein SPA-Framework wählen
 - d.h. es müsste heißen "Angular vs. HTMX" oder "Next.js vs. HTMX"

Single-Page-Anwendung, Fullstack oder HTMX?

- SPA und Fullstack-Apps sind **Pattern** oder **Architekturen**, keine konkreten Frameworks
- Wir müssten bei einem Vergleich eigentlich ein SPA-Framework wählen
 - d.h. es müsste heißen "Angular vs. HTMX" oder "Next.js vs. HTMX"
- Ich versuche eine Mischung aus "Konzept" / "Idee" und Umsetzung
 - React und Next.js

Single-Page-Anwendung, Fullstack oder HTMX?

- HTMX ist **clientseitige** Bibliothek

Single-Page-Anwendung, Fullstack oder HTMX?

- HTMX ist **clientseitige** Bibliothek
- Bei einem realistischen Vergleich muss man auch die **Serverseite** betrachten

Single-Page-Anwendung, Fullstack oder HTMX?

- HTMX ist **clientseitige** Bibliothek
- Bei einem realistischen Vergleich muss man auch die **Serverseite** betrachten
- Hier haben wir ein ähnliches Problem wie bei SPA und Fullstack
 - es müsste heißen "React vs. HTMX mit Node.JS und Handlebars"

Single-Page-Anwendung, Fullstack oder HTMX?

- HTMX ist **clientseitige** Bibliothek
- Bei einem realistischen Vergleich muss man auch die **Serverseite** betrachten
- Hier haben wir ein ähnliches Problem wie bei SPA und Fullstack
 - es müsste heißen "React vs. HTMX mit Node.JS und Handlebars"
- Auch hier versuche ich einen Mittelweg

Agenda

Agenda

Teil 1: **HTMX Intro**

Agenda

Teil 1:
HTMX Intro

Teil 2:
Vergleich der Technologien

Teil 1

htmx

Intro

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

- 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 - 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
- **Andere Aktualisierungen der UI gehen mit HTMX nicht**

```
<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";
    }
}
```

htmx



in der Praxis

HTMX in der Praxis

- Ich zeige euch jetzt ein paar Code-Beispiele im Editor

HTMX in der Praxis

- Ich zeige euch jetzt ein paar Code-Beispiele im Editor
- Danach gebe ich euch ein Git-Repository mit einem Übungsworckspace

HTMX in der Praxis

- Ich zeige euch jetzt ein paar Code-Beispiele im Editor
- Danach gebe ich euch ein Git-Repository mit einem Übungsworckspace
- In dem Git-Repository macht ihr dann eine Übung
- Git-Repository: <https://github.com/nilshartmann/jsdays-htmx-intro>
- Branch mit Schritt-für-Schritt Live-Coding Ergebnis:
https://github.com/nilshartmann/jsdays-htmx-intro/tree/jstays_2025

HTMX in der Praxis

- <http://localhost:3000>

Single-Page-App

Fullstack

oder

htmx



TEIL 2

HTMX in der Praxis

- Repository mit Beispiel-Anwendung mit HTMX/Java, React (SPA) und Next.js (Fullstack):

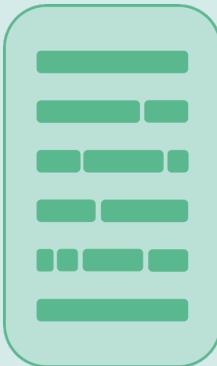
- <https://github.com/nilshartmann/spa-fullstack-htmx-example>

- Branch mit dem Live-Coding aus dem Workshop:

https://github.com/nilshartmann/spa-fullstack-htmx-example/tree/jsdays_2025

Architekturen für Webanwendungen

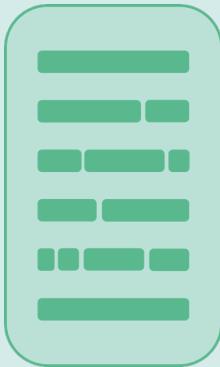
"Klassisch": Serverseitiges Rendern



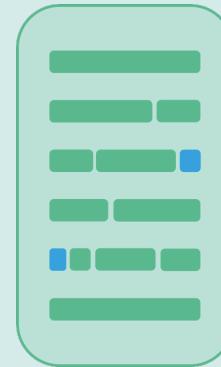
HTML-Seite

- 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 ("vanilla JS", jQuery, **htmx**)

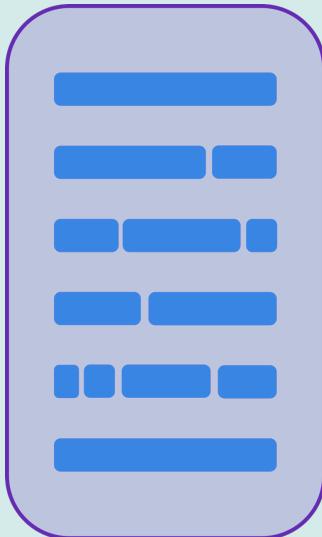
Ansatz 2: Single-Page-Anwendungen



JavaScript-Code

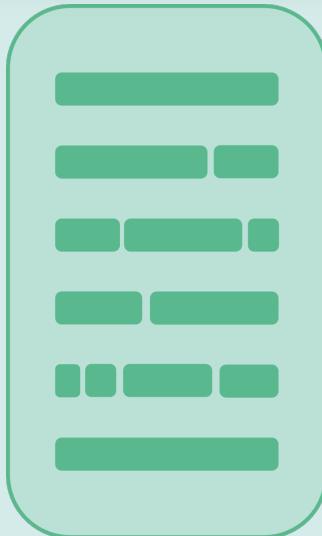
- Aus "Seiten" werden "Anwendungen"

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

Ansatz 3: "Fullstack"-Anwendungen (Meta-Frameworks)

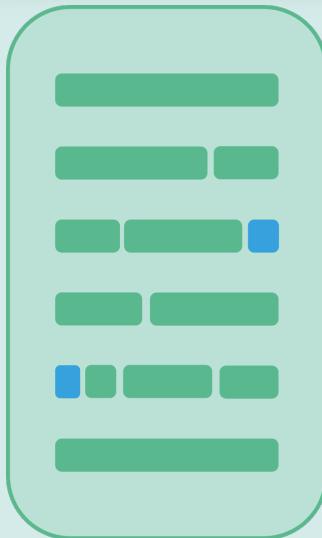


Nur Server (+JS)

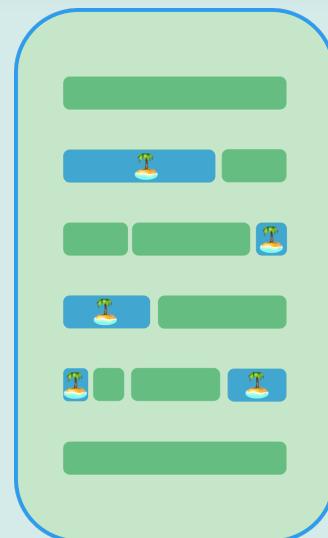


Nur Client (SPA)

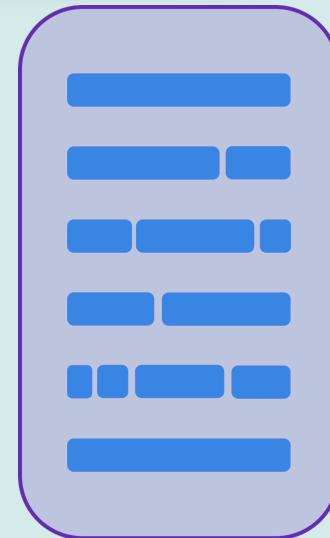
Ansatz 3: "Fullstack"-Anwendungen (Meta-Frameworks)



Nur Server (+JS)

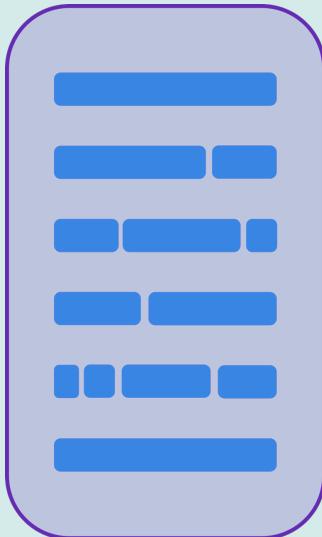


JS Fullstack Anwendung

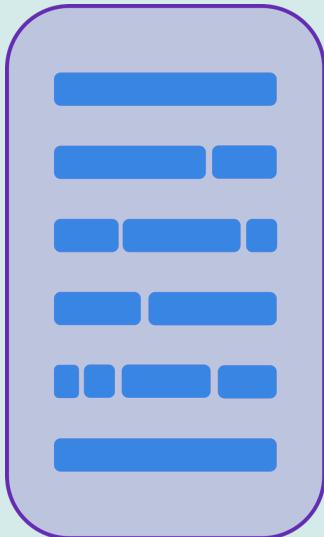


Nur Client (SPA)

Fullstack-Anwendungen (JavaScript)

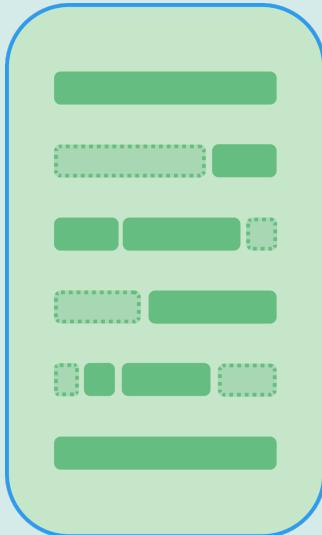


Fullstack-Anwendungen (JavaScript)



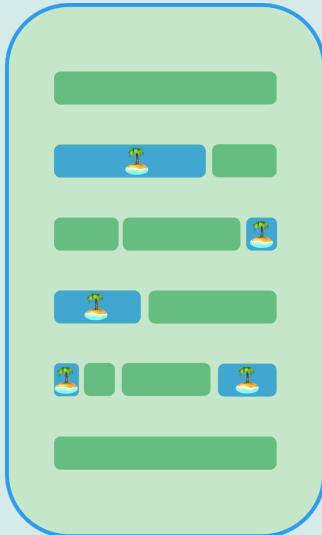
- Ebenfalls vollständig in **JavaScript** geschrieben

Fullstack-Anwendungen (JavaScript)



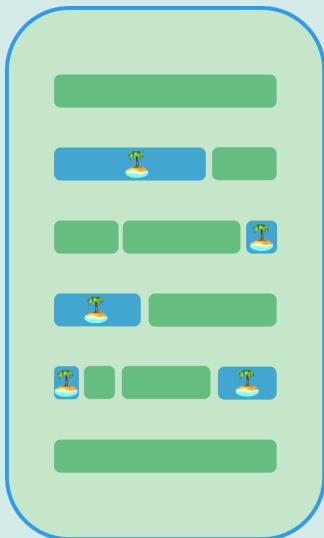
- Grundsätzliche Idee:
 1. **UI-Code** wird serverseitig vorgerendert
 2. **UI-Code** wird zum Browser gesendet und angezeigt

Fullstack-Anwendungen (JavaScript)



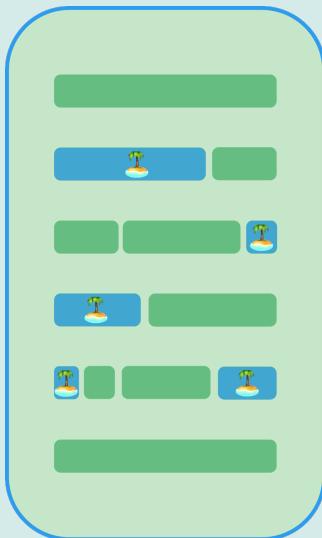
- Grundsätzliche Idee:
 1. **UI-Code** wird serverseitig vorgerendert
 2. **UI-Code** wird zum Browser gesendet und angezeigt
 3. Nur der JavaScript-Code ("Islands") **für Interaktionen** wird zum Browser geschickt

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

BEISPIEL-ANWENDUNG

- Eine Beispiel-Anwendung...
- <http://localhost:9099>

CODE-BEISPIELE

Anwendungsfälle: Im Editor!

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

A screenshot of a browser's developer tools element inspector. It shows a text input field with a blue border. Inside the field, the word "sal" is typed. To the right of the input field is a small "Clear" button with underlined text.

```
sal
```

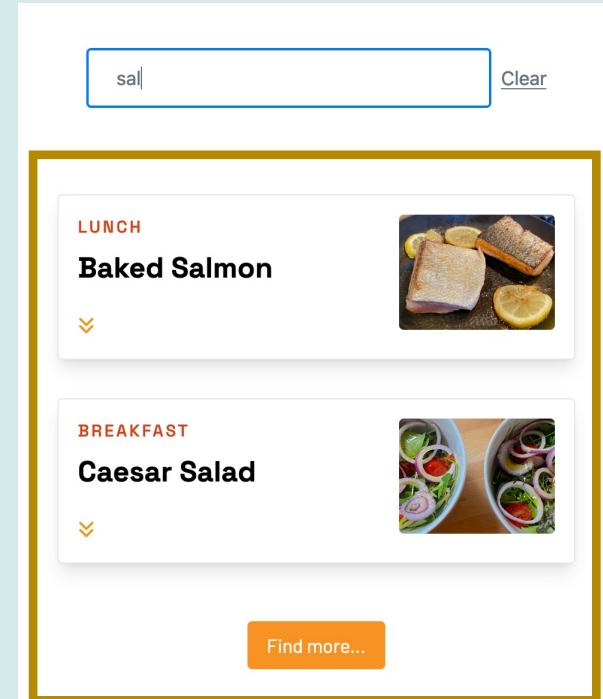
Clear

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

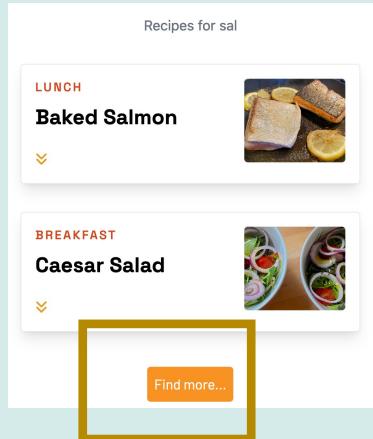
 [Clear](#)

GET /search



CODE-BEISPIELE

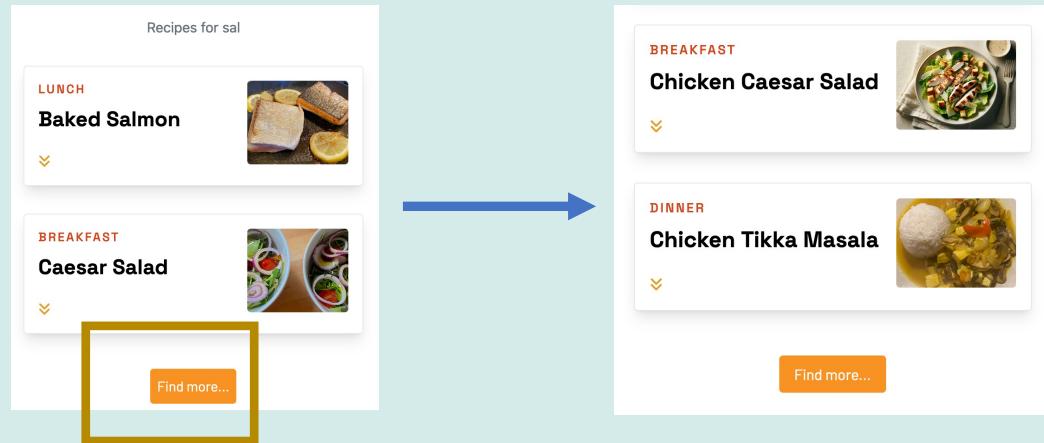
Beispiel: Bestehende Darstellung aktualisieren



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
▼ 

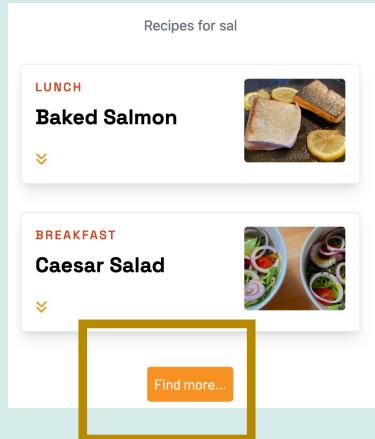
DINNER
Chicken Tikka Masala
▼ 

[Find more...](#)

GET /search?page=3

CODE-BEISPIELE

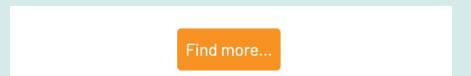
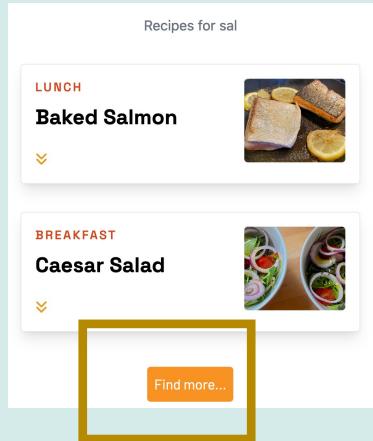
Beispiel: Bestehende Darstellung aktualisieren



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

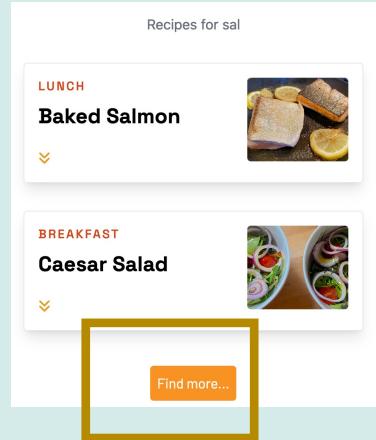
▼

BREAKFAST

Caesar Salad

▼

[Find more...](#)



HTMX Request

[Find more...](#)

`id="moreButton"`

`hx-swap-oob="true"`

BREAKFAST

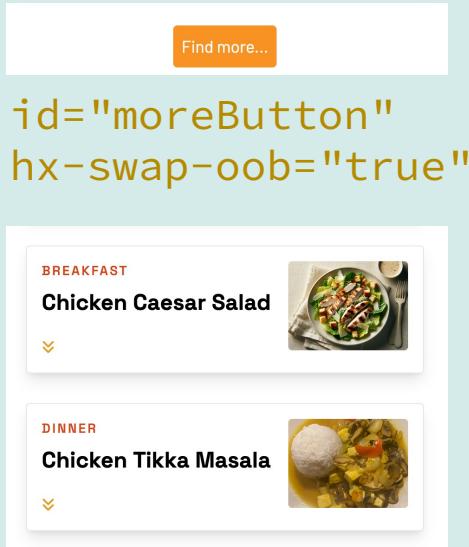
Chicken Caesar Salad

▼

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"`

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"



Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

Find more...

GET /search?page=3

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"`



Find more...

Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

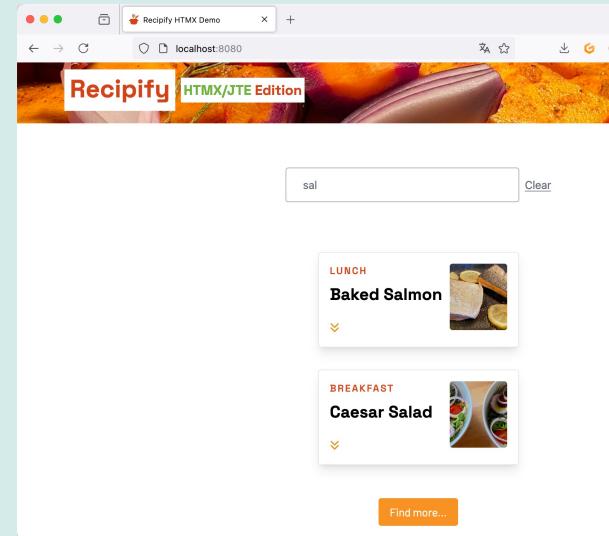
BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

GET /search?page=3

Beispiel: Parallel Requests

-  Netzwerk-Tab

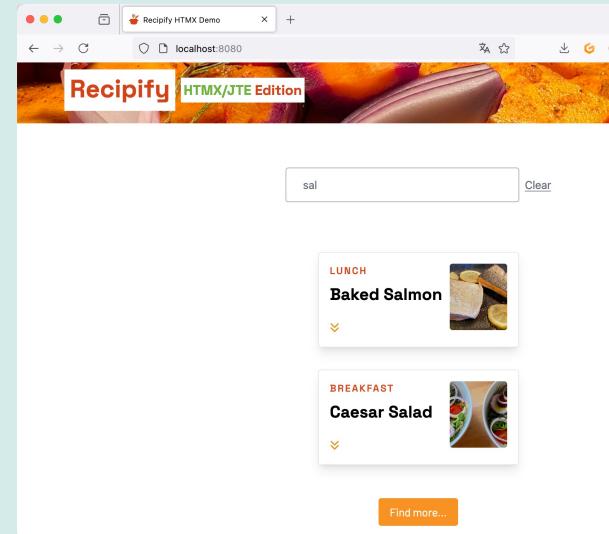


```
SearchPageContent.jte: slowdown=2000, dafür delay raus
```

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

Beispiel: Parallel Requests

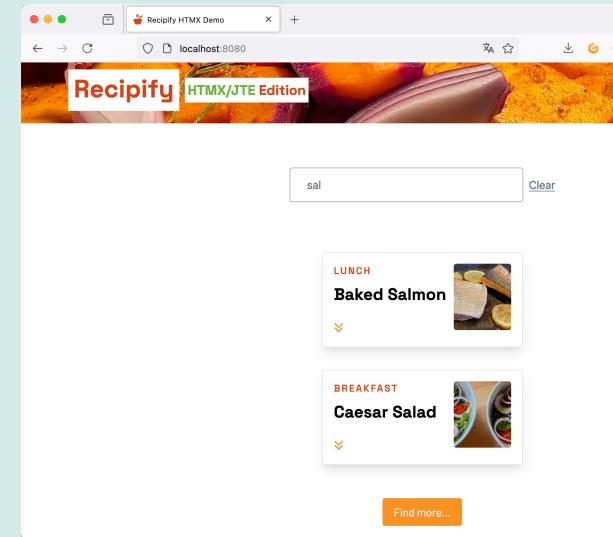
- 🕵️ Netzwerk-Tab
- HTMX führt Requests nacheinander aus
- In der SPA können wir das selbst entscheiden



Beispiel: Parallel Requests

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

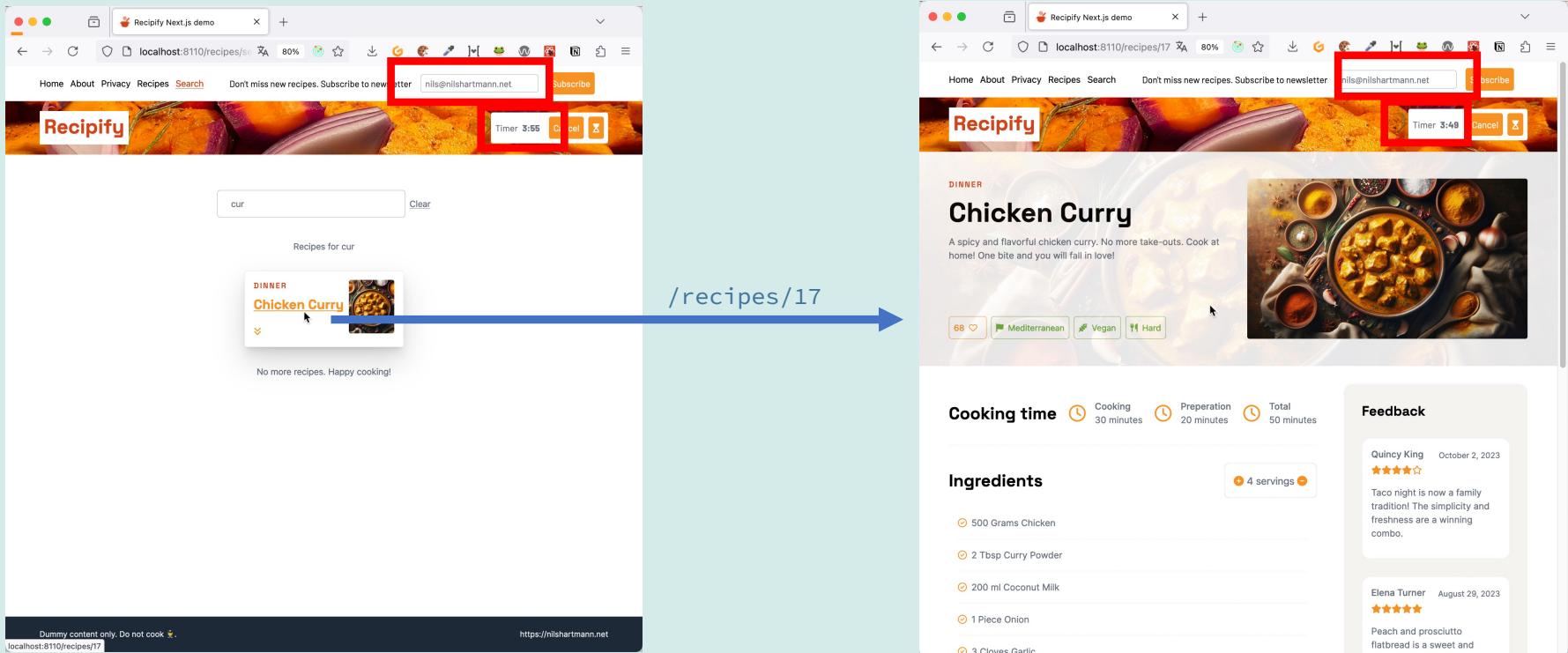
SearchPageContent.jte: slowdown



CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel

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



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

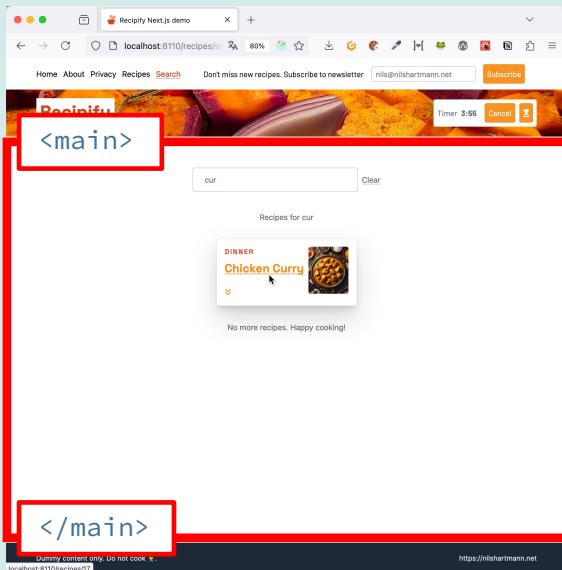
- Für Klick ohne JavaScript
- Für Klick mit JavaScript
- Aktualisieren der URL im Browser

RecipeSummaryCard.jte

Beispiel: SPA-like Seitenwechsel in HTMX

Variante 1

RecipeSummaryCard.jte

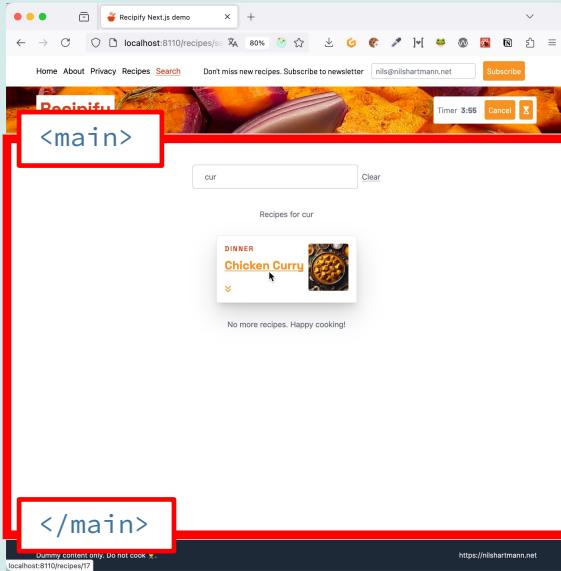


CODE-BEISPIELE

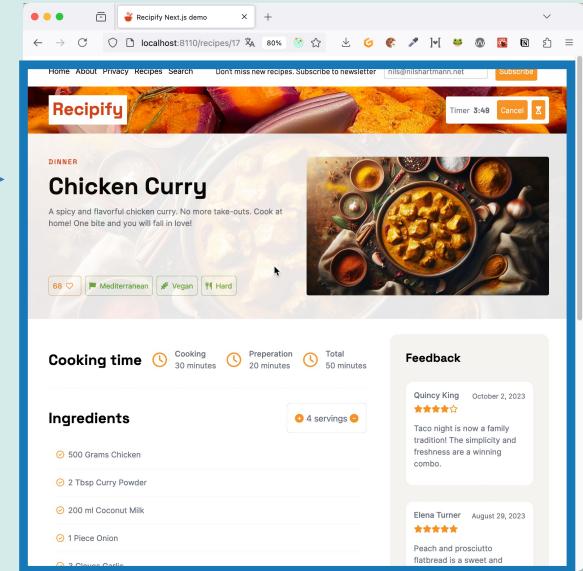
Beispiel: SPA-like Seitenwechsel in HTMX

Variante 1

RecipeSummaryCard.jte



/recipes/17

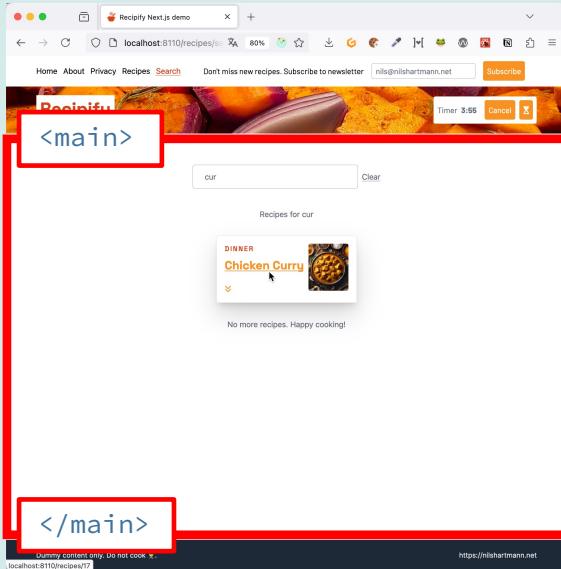


CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

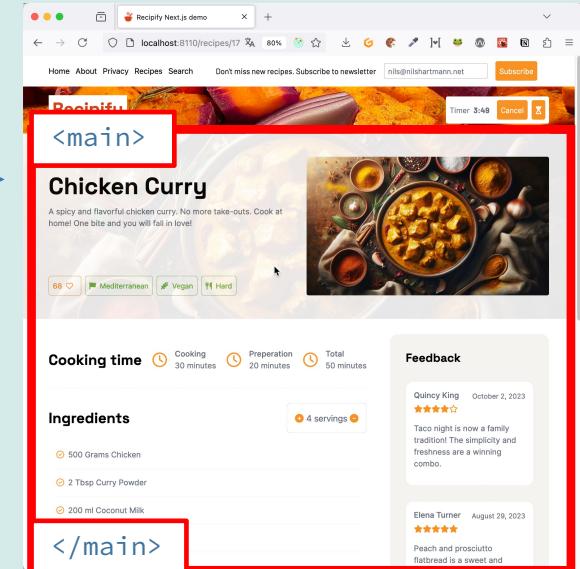
Variante 1

RecipeSummaryCard.jte



/recipes/17

hx-select="main"

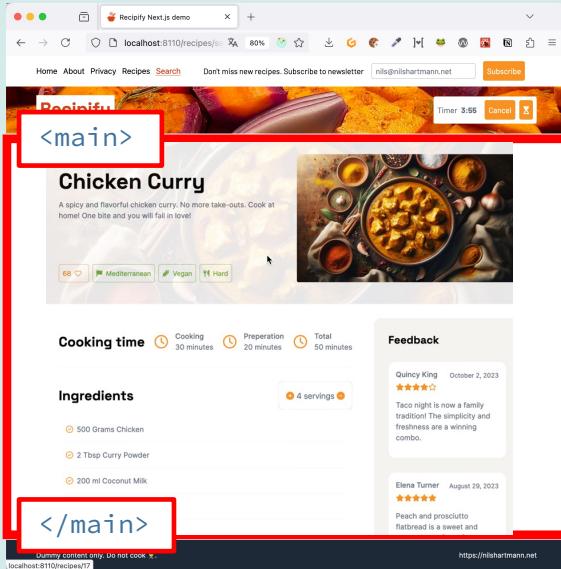


CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

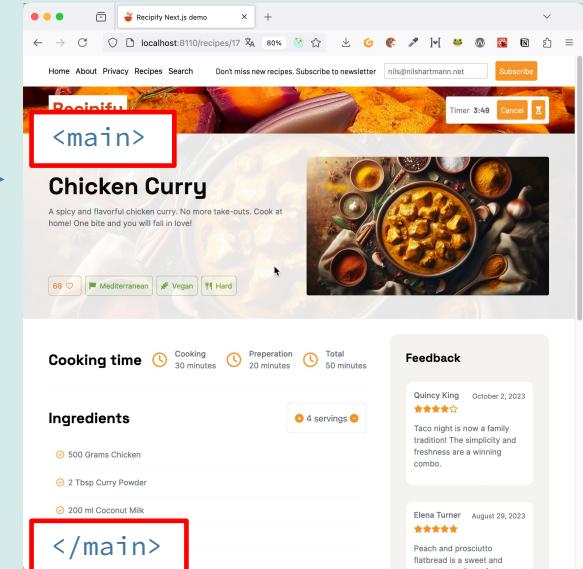
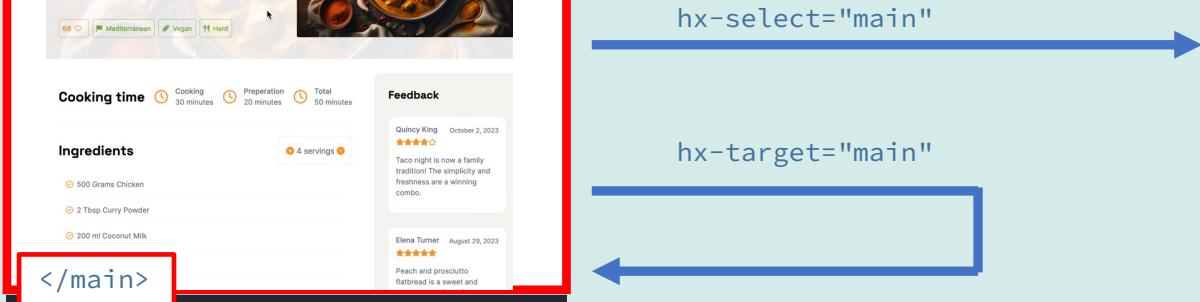
Variante 1

RecipeSummaryCard.jte



/recipes/17

hx-select="main"



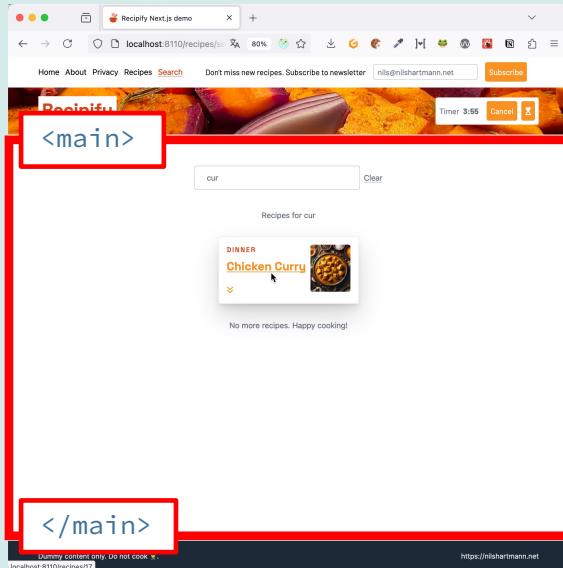
Variante 1: HTMX

- **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

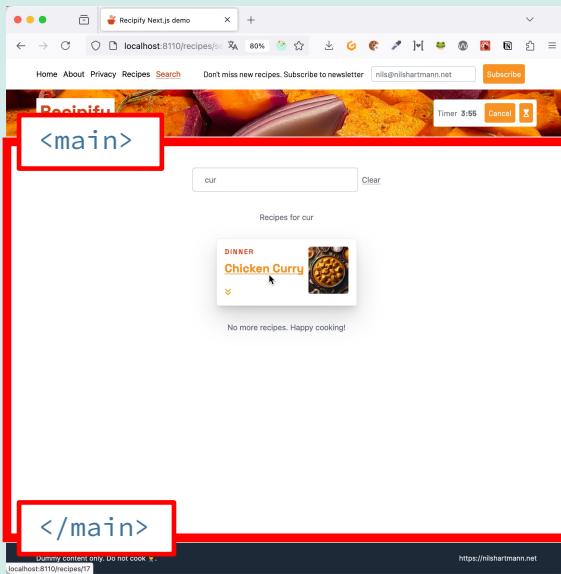
Variante 2: Eigener Endpunkt



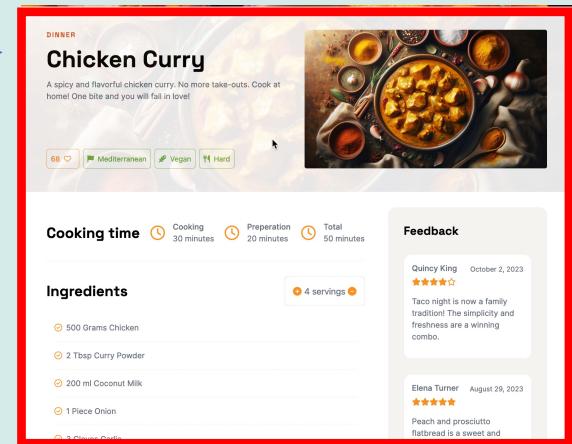
CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

Variante 2: Eigener Endpunkt



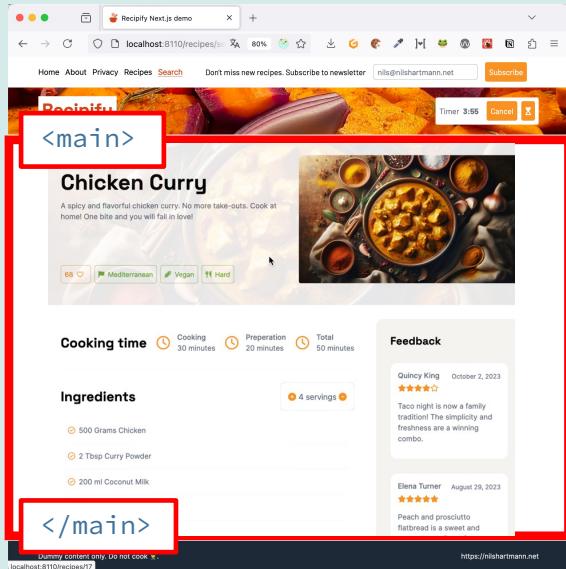
/recipes/17/details



CODE-BEISPIELE

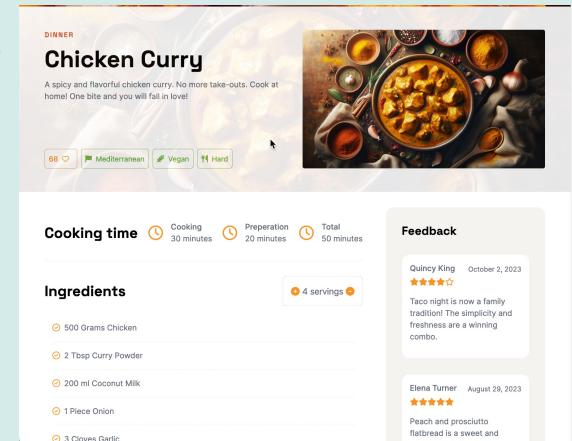
Beispiel: SPA-like Seitenwechsel in HTMX

Variante 2



/recipes/17/details

hx-target="main"



Variante 2: Eigener Endpunkt

- **Link** muss wissen:

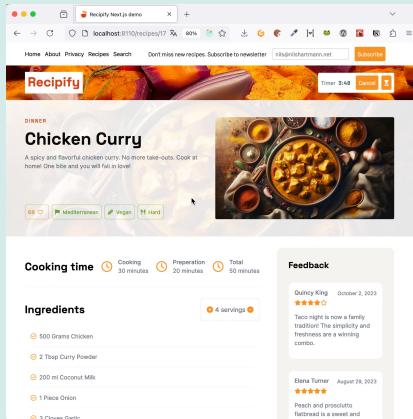
- welcher Endpunkt
- wo muss Ergebnis auf der aktuellen Seite eingebaut werden

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 the Recipify homepage with a featured recipe for "Chicken Curry". The page includes a header with navigation links like Home, About, Privacy, Recipes, Search, and a newsletter sign-up. Below the header is a banner for "Recipify" with a photo of a dish. The main content area displays the "Chicken Curry" recipe card. The card features a large image of the dish, the title "Chicken Curry", a brief description, cooking time (30 minutes), preparation time (20 minutes), and total time (50 minutes). It also includes categories like Mediterranean, Vegan, and Hard. The "Ingredients" section lists: 500 Grams Chicken, 2 Tbsp Curry Powder, 200 ml Coconut Milk, 1 Peice Onion, and 3 Cloves Garlic. The "Feedback" section contains two reviews: one from "Quincy King" dated October 2, 2023, and another from "Elena Turner" dated August 29, 2023.

/recipes/17

The screenshot shows the Recipify details page for the same "Chicken Curry" recipe. The layout is identical to the homepage version, featuring the "Chicken Curry" card with its image, title, and detailed description. The "Cooking time", "Preparation time", and "Total time" are listed as 30 minutes, 20 minutes, and 50 minutes respectively. The "Feedback" section again shows reviews from Quincy King and Elena Turner. The "Ingredients" section is identical to the homepage version.

/recipes/17/details

CODE-BEISPIELE

Beispiel: Seite priorisieren

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

- Header:** The logo "Recipify" is at the top left, followed by a large image of the salad.
- Catagory:** "BREAKFAST" is listed under the title.
- Title:** "Classic Caesar Salad" is prominently displayed.
- Description:** A brief description follows: "Crispy romaine lettuce with creamy Caesar dressing. Bon Appétit! Food that feels like home."
- 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 "Your rating:" with five stars.
- Instructions:** A section for step-by-step cooking instructions.
- Feedback:** A section titled "Feedback" with the placeholder text "Loading feedback..." and three circular progress indicators.
- Opinion:** A section titled "Your opinion?" 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

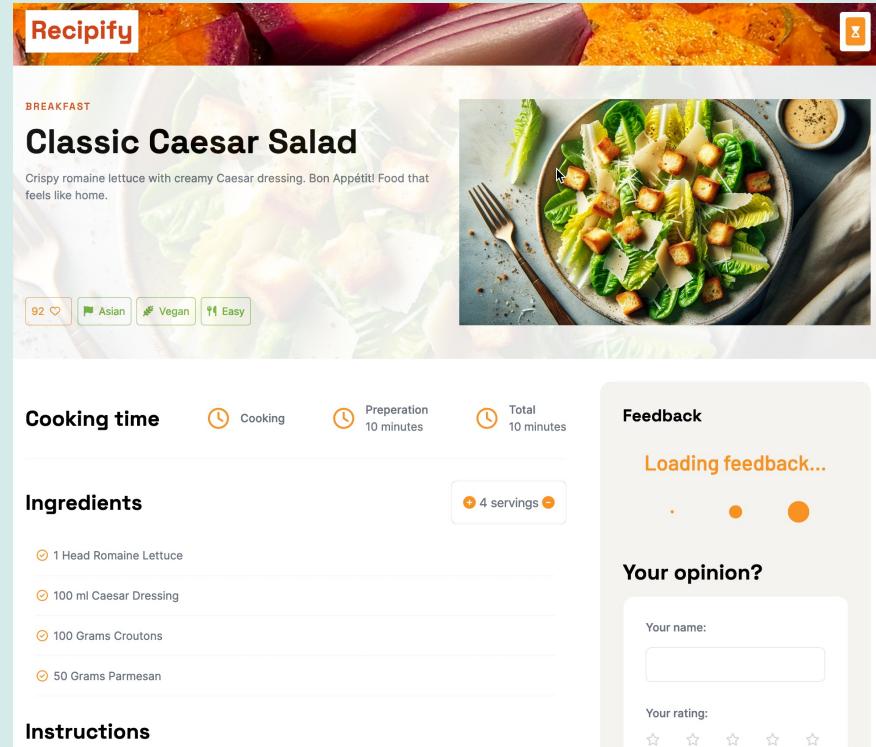
慢速请求慢速请求
slowDown_GetRecipe=2000

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 card.
- 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 image of the salad is on the right side.
- Metrics:** Below the description are four small boxes: "92 ❤️", "Asian", "Vegan", and "Easy".
- Time:** "Cooking time" is listed with a clock icon, followed by "Preparation 10 minutes" and "Total 10 minutes".
- Ingredients:** A list of ingredients includes "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.
- Feedback:** On the right, there's a section for "Feedback" with the placeholder text "Loading feedback...". It also features a rating scale with three orange dots.
- Your opinion?**: A form on the right allows users to enter their name and rating. It includes fields for "Your name:" and "Your rating:" with five 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" category. 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 4 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.

Single-Page-App

Fullstack

oder

htmx



Single Page-App

Fazit

fullstack

oder

htmx



Single Page-App

Fazit

Eure
Erkenntnisse?

itmX



NILS HARTMANN
<https://nilshartmann.net>

Vielen Dank!

Fragen und Kontakt

nils@nilshartmann.net

<https://nilshartmann.net/kontakt>