

NILS HARTMANN

<https://nilshartmann.net>

Frontend-Entwicklung

HTMX
oder

Single-
Page-
Anwendung ?

NILS HARTMANN

nils@nilshartmann.net

Freiberuflicher Entwickler, Architekt, Trainer

Java, Spring, GraphQL, React, TypeScript



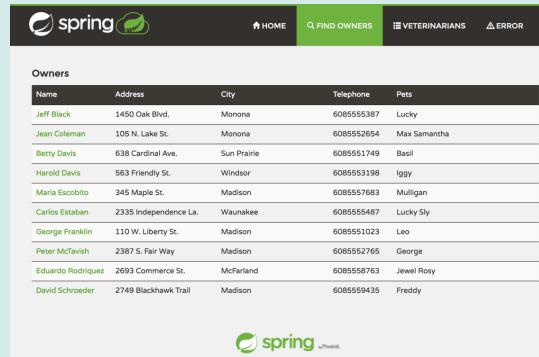
<https://graphql.schule/video-kurs>

<https://reactbuch.de>

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

Webanwendungen...

WEBANWENDUNGEN



The screenshot displays a web application interface for managing pet owners. At the top, there is a navigation bar with the Spring logo, followed by links for HOME, FIND OWNERS (highlighted in green), VETERINARIANS, and ERROR.

The main content area is titled "Owners" and contains a table with columns: Name, Address, City, Telephone, and Pets. The table lists 15 entries, each with a small thumbnail image of a pet next to its name.

Name	Address	City	Telephone	Pets
Jeff Black	1450 Oak Blvd.	Monona	6085555387	Lucky
Jean Coleman	105 N. Lake St.	Monona	6085552654	Mae Samantha
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085511749	Basil
Harold Davis	563 Friendly St.	Windsor	6085553198	Iggy
Maria Escobito	345 Maple St.	Madison	608557683	Mulligan
Carlos Esteban	2335 Independence La.	Wauakee	6085555487	Lucky Sly
George Franklin	110 W. Liberty St.	Madison	6085511023	Leo
Peter McTavish	2387 S. Fair Way	Madison	6085552765	George
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy

At the bottom of the page, there is a footer with the Spring logo and the word "powered".

WEBANWENDUNGEN

The screenshot shows the ICE Portal website interface. At the top, it displays "Reiseinformationen für den ICE 109". Below this, it says "Nächster Halt in 6 min" and "Gleis 2". The train's current speed is 108 km/h, and the destination is Osnabrück Hbf at 12:35, with a red digital clock showing 12:46. A warning icon is present. On the left, there's a sidebar with "Menü" and a "hw plus" advertisement for Sasha Show tickets. The main content area shows the route from Hamburg-Altona to Basel SBB, listing stops like Hamburg-Altona, ICE 109 nach Basel SBB, and Gl. 10. It includes buttons for "Vergangene Halte zeigen", "Ab", "An", "Suchen", "Erweiterte Suche", and "Aktuelle Meldungen". At the bottom, there are links for "hvv Deutschlandticket", "Mach mit!", "Schüler*innen", and "Schnelles Internet".

The screenshot shows the Spring website with a navigation bar featuring "HOME", "FIND OWNERS" (which is highlighted in green), "VETERINARIANS", and "ERROR". The main content area is titled "Owners" and displays a table of pet owners with their details:

Name	Address	City	Telephone	Pets
Jeff Black	1450 Oak Blvd.	Monona	6085555387	Lucky
Jean Coleman	105 N. Lake St.	Monona	6085552654	Max Samantha
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085511749	Basil
Harold Davis	563 Friendly St.	Windsor	6085553198	Iggy
Maria Escobito	345 Maple St.	Madison	608557683	Mulligan
Carlos Esteban	2335 Independence La.	Waunakee	6085555487	Lucky Sly
George Franklin	110 W. Liberty St.	Madison	6085511023	Leo
Peter McTavish	2387 S. Fair Way	Madison	6085552765	George
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy

At the bottom, there's a logo for "spring" and "powered by JBoss Seam".

WEBANWENDUNGEN

The image displays four distinct web applications arranged horizontally:

- Booking.com**: A travel booking website showing flight information for an ICE 109 train. It includes details like the next stop at Osnabrück Hbf (Gleis 2) in 6 minutes, current speed (108 km/h), and a Genius Prämien offer for Nils.
- ICE Portal**: A real-time train status application for the ICE 109. It shows the train's current position at Hamburg-Altona, its destination (Basel SBB), and the number of stops remaining (14). It also features a speed indicator (108 km/h) and a "Schnelles Internet" button.
- Reiseinformationen für den ICE 109**: A separate view of the ICE 109's route, showing stops from Hamburg-Altona to Basel SBB, including the number of stops (14) and the current time (10:29).
- Spring**: A pet adoption platform. The header includes links for HOME, FIND OWNERS, VETERINARIANS, and ERROR. The main content is a table titled "Owners" listing pet adopters with their contact information and pets' names.

WEBANWENDUNGEN

The image shows a Mac desktop with five browser tabs open:

- Booking.com**: A flight search results page for the ICE 109 from Osnabrück Hbf to Basel SBB.
- ICE Portal**: A travel information page for the ICE 109, showing the next stop at Gleis 2 in 6 minutes.
- HIBERNATE**: A Jira project management interface for the "Hibernate ORM" project, showing a list of tasks and their status.
- sessionize**: A speaker dashboard for Nils Hartmann, showing session details and a public profile.
- spring**: A GitHub repository for "g-graphql-training" by Nils Hartmann, displaying code, issues, and pull requests.

WEBANWENDUNGEN

The image displays a collage of several web application interfaces:

- Booking.com**: A travel booking website showing flight information for the ICE 109 from Osnabrück Hbf to Basel SBB.
- ICE Portal**: A real-time train status board showing the next stop at Gleis 2 in 6 minutes, speed (108 km/h), and a "Schnelles Internet" connection indicator.
- Hibernate**: A project management tool showing a list of tasks and projects related to "Hibernate ORM".
- Microsoft Teams**: A communication platform displaying a chat history between users.
- Sessionize**: An event management tool showing a speaker dashboard for Nils Hartmann.
- GitHub**: A code repository for a "spring-graphql-training" project, listing pull requests and issues.
- Eventbrite**: A screenshot of an event page for a "Frontend for Backend: HTMX oder Single-Page-Anwendung?" event.

WEBANWENDUNGEN

The image displays a collage of several web application screenshots, illustrating various types of web-based tools and services:

- Booking.com**: A travel booking website showing flight information for the ICE 109.
- ICE Portal**: A mobile-style interface showing travel details, including "Nächster Halt in 6 min" (Next stop in 6 min) and "Gleis 2" (Platform 2).
- ChatGPT - DALLE**: A collaboration between AI chat and image generation, showing a request for a cooking recipe website and two generated images of a beef burger.
- HIBERNATE**: A project management tool showing a list of projects and tasks, with a focus on "Hibernate ORM".
- sessionize**: A platform for organizing events, showing a speaker dashboard for Nils Hartmann.
- Microsoft Teams**: A communication and collaboration tool showing a chat history with multiple messages and users.
- Spring**: A developer-oriented platform showing a list of owners with details like name, address, city, phone, and pets.

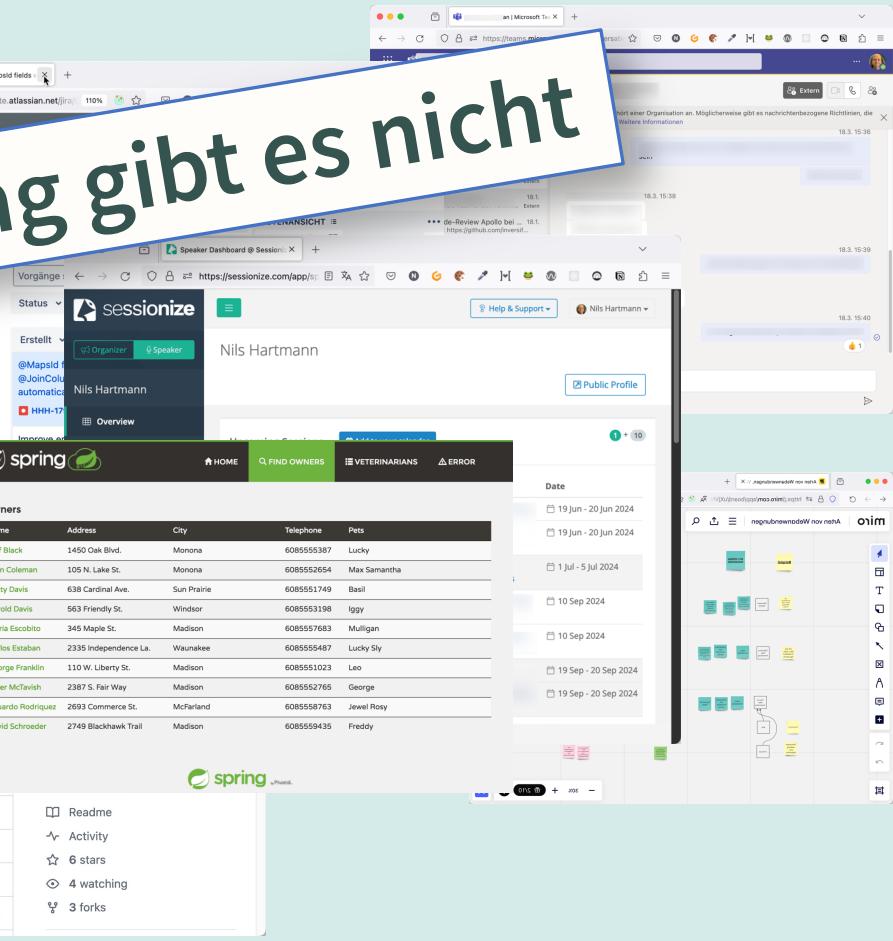
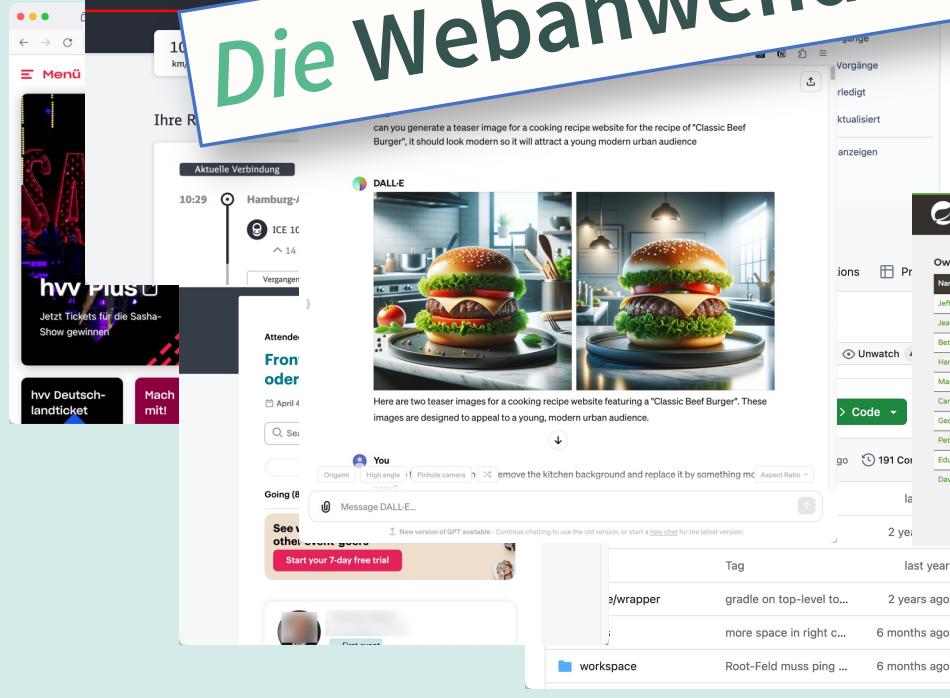
WEBANWENDUNGEN

Die Webanwendung gibt es nicht

The image is a collage of several screenshots from various web applications, demonstrating different types of web-based tools and interfaces:

- Booking.com**: A travel booking website showing flight and hotel search results.
- ICE Portal**: A travel-related portal showing flight status (108 km/h) and a map.
- ChatGPT - DALL-E**: A screenshot of a conversation with AI models, showing two generated images of a "Classic Beef Burger".
- Hibernate ORM**: A screenshot of a project management interface for the Hibernate ORM software.
- Vorgänge**: A screenshot of a task or event management system.
- sessionize**: A screenshot of a speaker dashboard for sessionize.com.
- spring**: A screenshot of a web application for managing pet owners, showing a table of owners with columns for Name, Address, City, Telephone, and Pets.
- Google Sheets**: A screenshot of a Google Sheets spreadsheet with multiple tabs and data.

WEBANWENDUNGEN



WEBANWENDUNGEN

Die Webanwendung gibt es nicht

...aber: (fast) alle brauchen JavaScript



A screenshot of a Mac OS X desktop environment showing several open windows:

- Booking.com**: A travel booking website.
- ICE Portal**: A train information portal showing the next stop in 6 minutes.
- Hibernate Atlassian.net Jira**: A project management tool.
- Microsoft Teams**: A communication platform.
- Sessionize Speaker Dashboard**: A speaker dashboard for a conference.
- DALL-E**: An AI image generation interface.
- GitHub Issues**: A list of issues for a Java project.
- Google Sheets**: A spreadsheet application.



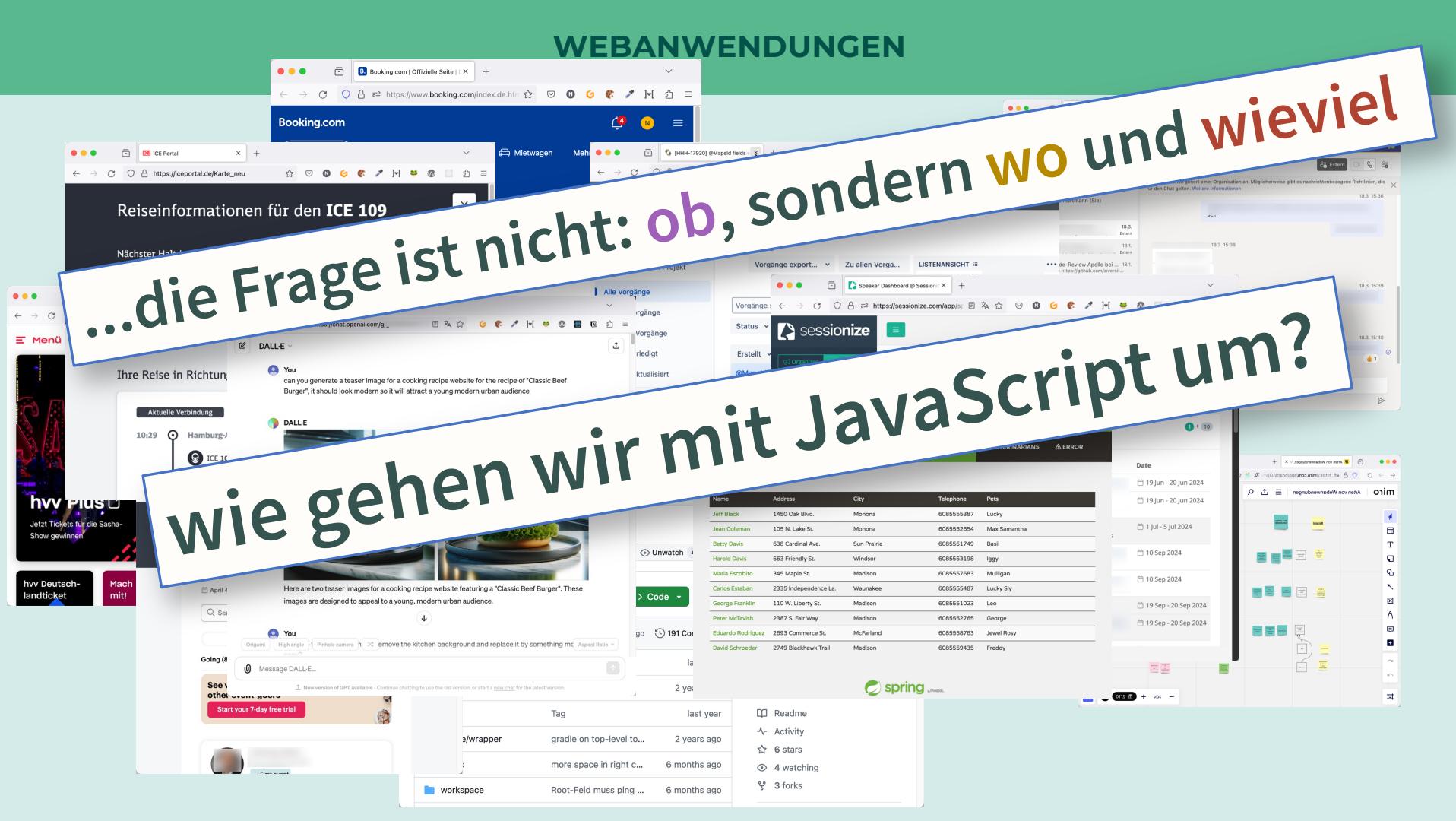
- Readme
- Activity
- 6 stars
- 4 watching
- 3 forks

WEBANWENDUNGEN

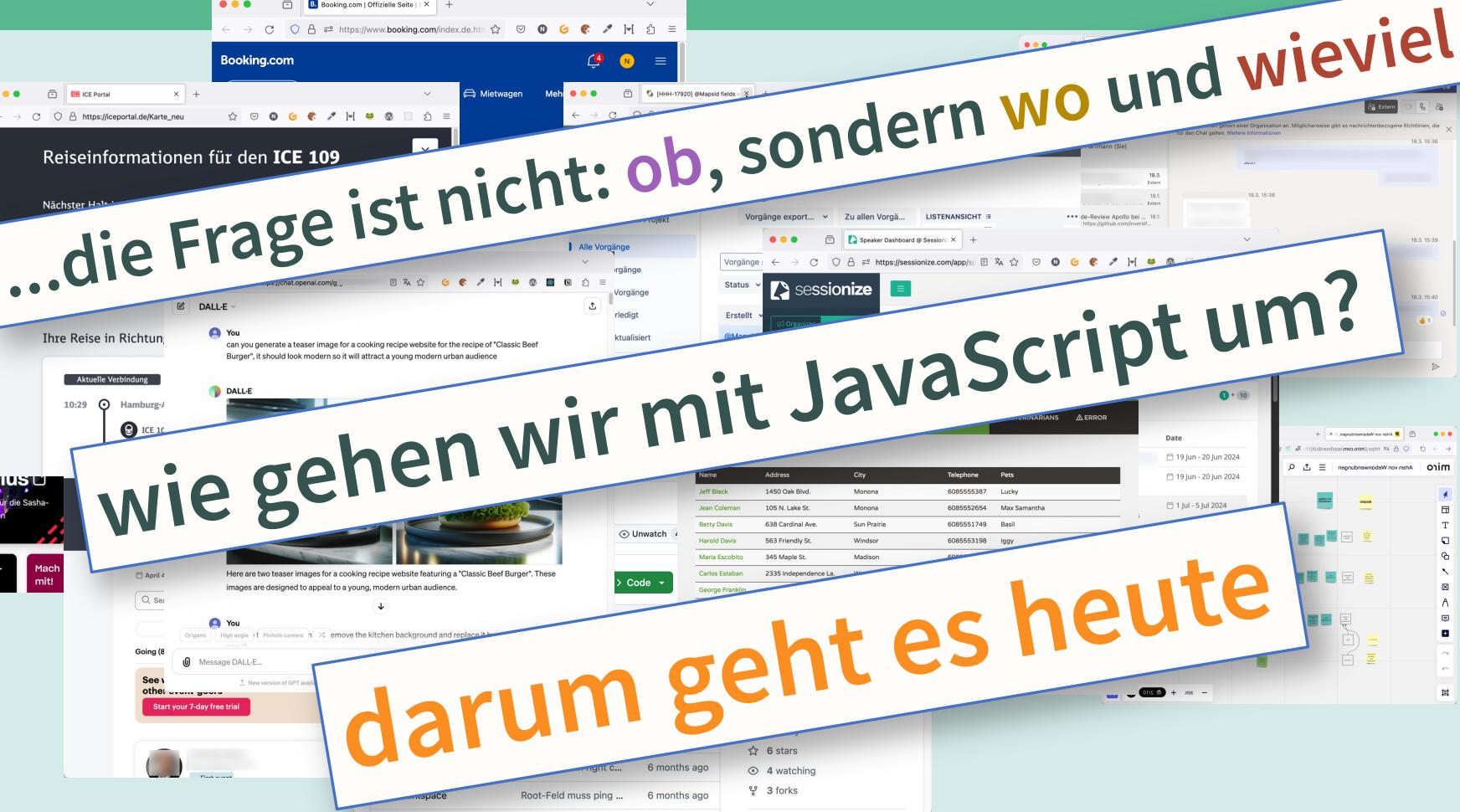
...die Frage ist nicht: ob, sondern wo und wieviel

The image is a collage of several web application screenshots, each showing a different interface or feature. At the top left is a screenshot of the Booking.com homepage. Below it is a screenshot of the ICE Portal showing travel information for the ICE 109. To the right of the ICE Portal is a screenshot of a browser tab titled 'Mietwagen' with a URL like 'https://www.booking.com/index.de.htm'. In the center is a screenshot of a DALL-E interface where a user has requested two images of a classic beef burger. To the right of the DALL-E interface is a screenshot of the Sessionize speaker dashboard for Nils Hartmann. Below the Sessionize dashboard is a screenshot of the Spring website showing a list of owners with columns for Name, Address, City, Telephone, and Pets. At the bottom right is a screenshot of a calendar application showing multiple events and tasks. The overall theme of the collage is the variety and complexity of modern web-based tools and services.

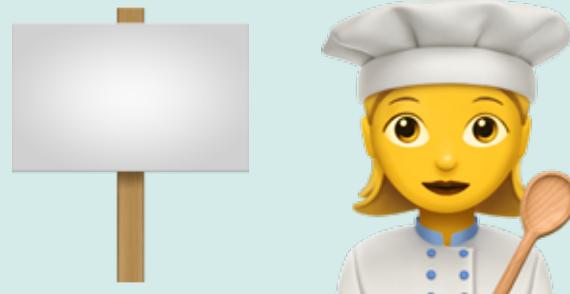
WEBANWENDUNGEN



WEBANWENDUNGEN



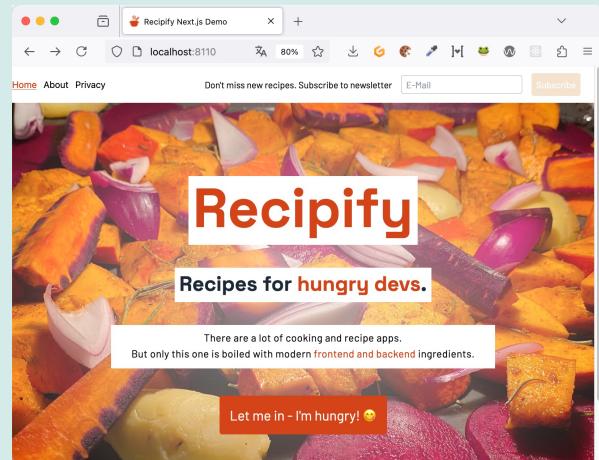
Demo



BEISPIELE

Eine Webanwendung...

1. 🕵️ Landing-Page
2. 🕵️ Recipes
3. 🕵️ Link mit Sortierung
4. 🕵️ Like-Button
5. 🕵️ Recipe-Seite
6. 🕵️ Recipe-Seite Kommentar schreiben

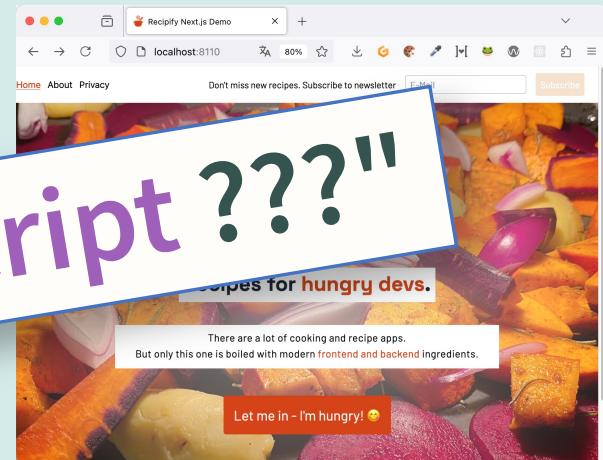


<http://localhost:8110>

BEISPIELE

Eine Webanwendung...

1. 🕵️ Landing-Page
2. 🕵️ Recipes
3. 🕵️ Recipe
4. 🕵️ Recipe-Seite
5. 🕵️ Recipe-Seite Kommentar schreiben

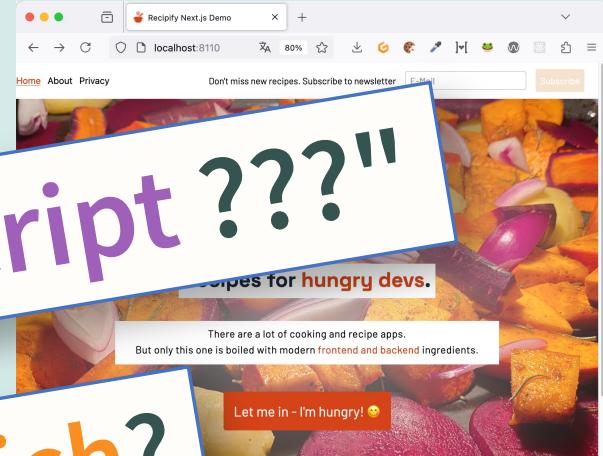


<http://localhost:8110>

BEISPIELE

Eine Webanwendung...

1. 🕵️ Landing-Page
2. 🕵️ Recipes
3. 🕵️ Recipe
4. 🕵️ Recipe-Seite
5. 🕵️ Recipe-Seite
6. 🕵️ Recipe-Seite



diese auch? wirklich?

Was meint ihr? 🤔

BEISPIELE

Eine Webanwendung...

1. 🕵️ Landing-Page

2. 🕵️

3. 🕵️

4. 🕵️ L

5. 🕵️ Re

6. 🕵️ Re

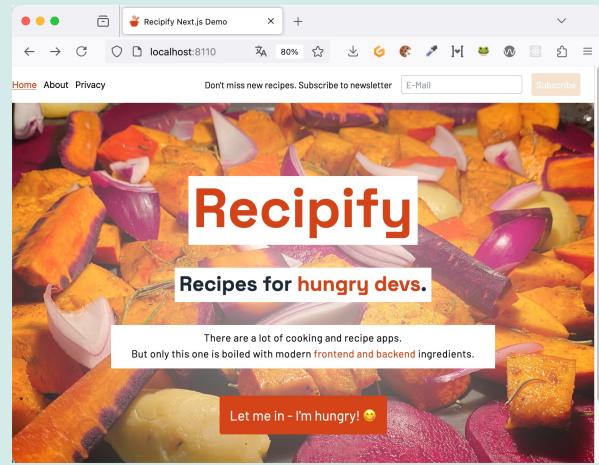


Eine Webanwendung...

1. 🕵️ Landing-Page
 2. 🕵️
 3. 🕵️
 4. 🕵️
 5. 🕵️ Re
 6. 🕵️ Re
-
- Erweiterte Einstellungen
- Nur veränderte Einstellungen
- javascript
- javascript.enabled
- false
- Kein JavaScript, kein Problem
- Feierabend machen?

BEISPIELE

Offenbar geht's doch auch ohne

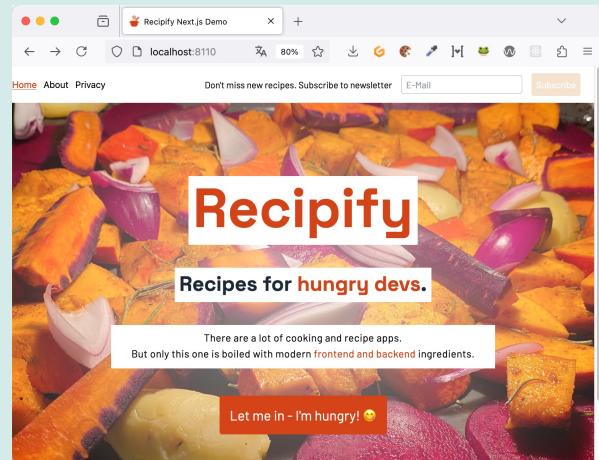


<http://localhost:8110>

BEISPIELE

Offenbar geht's doch auch ohne – ABER:

1. 🕵️ Feedback Like-Button 😢
2. 🕵️ Feedback Seitenwechsel zu Recipe 😢
3. 🕵️ Timer 😢
4. 🕵️ Newsletter-Feld Inhalt Seitenwechsel 😢
5. 🕵️ Neuer Kommentar: Zeichenzähler 😢
6. 🕵️ Portionszähler 😢
7. 🕵️ Suche bei Tastendruck 😢



<http://localhost:8110>

BEISPIELE

Wie sieht es denn mit JavaScript aus?

```
demo_config: enableSuspenseForFeedback = true
```

1. 🕵️ Feedback Like-Button 😊
2. 🕵️ Feedback Seitenwechsel zu Recipe 😊
3. 🕵️ Timer 😊
4. 🕵️ Newsletter-Feld Inhalt Seitenwechsel 😊
5. 🕵️ Neuer Kommentar: Zeichenzähler 😊
6. 🕵️ Portionszähler 😊
7. 🕵️ Suche bei Tastendruck 😊



<http://localhost:8110>

Beste UI/UX: oft nur mit JavaScript umsetzbar

- Es gibt Fortschritte in CSS und HTML, die JS an mehr Stellen verzichtbar machen, aber nicht an allen

Beste UI/UX: oft nur mit JavaScript umsetzbar

- Es gibt Fortschritte in CSS und HTML, die JS an mehr Stellen verzichtbar machen, aber nicht an allen

Man sollte den "Kleinkram" nicht unterschätzen 🙏

Architekturen

für

Webanwendungen

Ansatz 1: "Klassisch": Serverseitiges Rendern + JS-Schnipsel

- Server erzeugt fertige HTML-Seite
- Für Interaktionen werden Links und Formulare genommen
- Jede Interaktion erfolgt über Server-Roundtrip

Ansatz 1: "Klassisch": Serverseitiges Rendern + JS-Schnipsel

- Server erzeugt fertige HTML-Seite
- Für Interaktionen werden Links und Formulare genommen
- Jede Interaktion erfolgt über Server-Roundtrip
- Für Interaktionen wird JavaScript geschrieben

Ansatz 1: "Klassisch": Serverseitiges Rendern + JS-Schnipsel

- Server erzeugt fertige HTML-Seite
- Für Interaktionen werden Links und Formulare genommen
- Jede Interaktion erfolgt über Server-Roundtrip
- Für Interaktionen wird JavaScript geschrieben
- **Typische Vertreter:** **Spring MVC**, JEE

Ansatz 2: Single-Page-Anwendung (SPA)

- Darstellung erfolgt komplett über JavaScript im Browser
- HTML spielt untergeordnete Rolle
- Backend liefert **Daten** (z.B. REST/GraphQL), kein HTML

Ansatz 2: Single-Page-Anwendung (SPA)

- Darstellung erfolgt komplett über JavaScript im Browser
- HTML spielt untergeordnete Rolle
- Backend liefert **Daten** (z.B. REST/GraphQL), kein HTML
- **Typische Vertreter:** **React**, Angular, Vue

Ansatz 3: Fullstack-Anwendung (mit JavaScript)

- Anwendung ist vollständig in JavaScript geschrieben
- Server läuft mit JavaScript
- Server liefert HTML-Code aus
- Server liefert für interaktive Teile JavaScript-Code aus
- Zum Beispiel als "Backend for Frontend"

Ansatz 3: Fullstack-Anwendung (mit JavaScript)

- Anwendung ist vollständig in JavaScript geschrieben
- Server läuft mit JavaScript
- Server liefert HTML-Code aus
- Server liefert für interaktive Teile JavaScript-Code aus
- Zum Beispiel als "Backend for Frontend"
- **Typische Vertreter:** **Next.js**, SvelteKit, Nuxt, Astro, Qvik

...und 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" 🤔)

Passt zur Variante 1 (serverseitige Anwendungen)

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

- 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 frei

```
<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

Demo: Hello World

- Das Beispiel besteht aus:
 - Rechte Seite: <localhost:8080/hello>
 - Werkzeugleiste:
 - **hello-world.jte**
 - **HelloWorldController**
 - **hello-world-response.jte**
- Der Name ist eine Abkürzung für "Hypermedia"

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

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

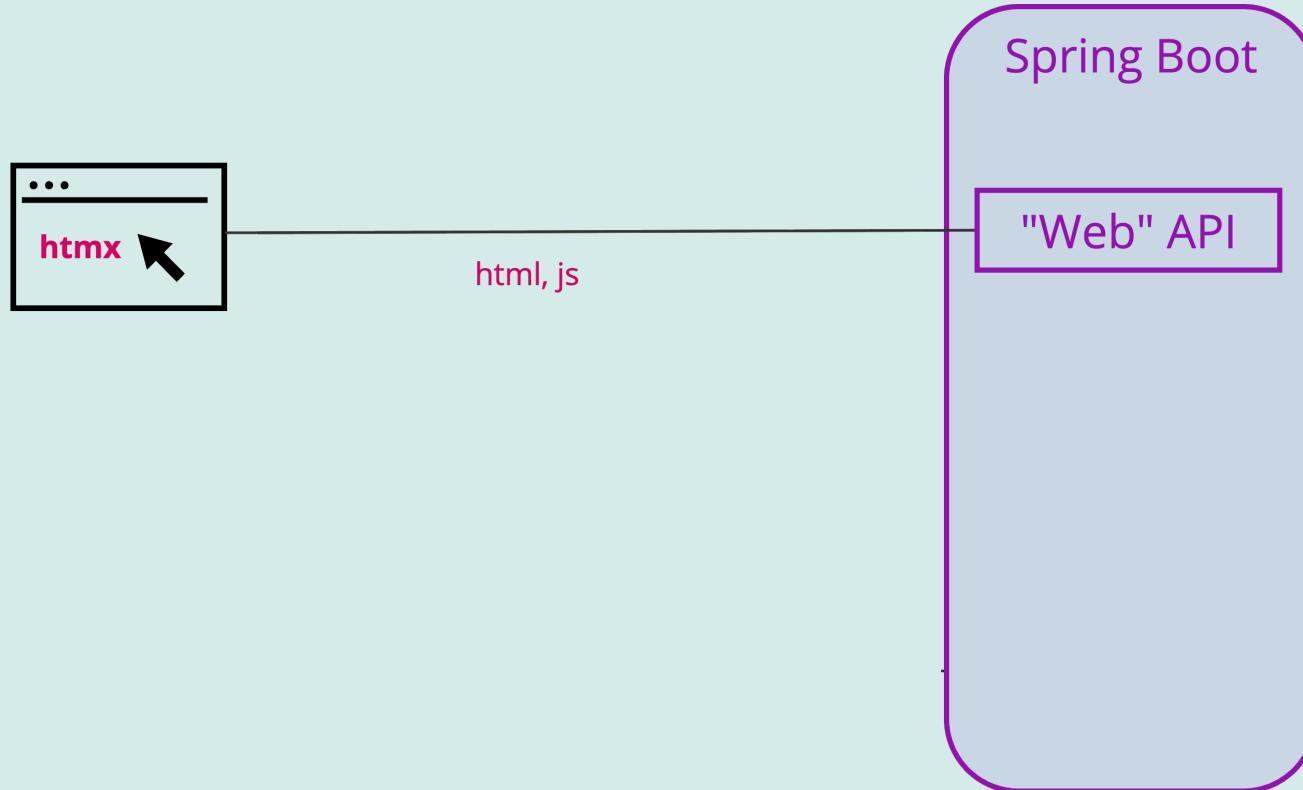
    <script
      type="text/javascript"
      th:src="@{/htmx/htmx-1.9.10.min.js}"
    ></script>
  </body>
</html>
```

Code Beispiele

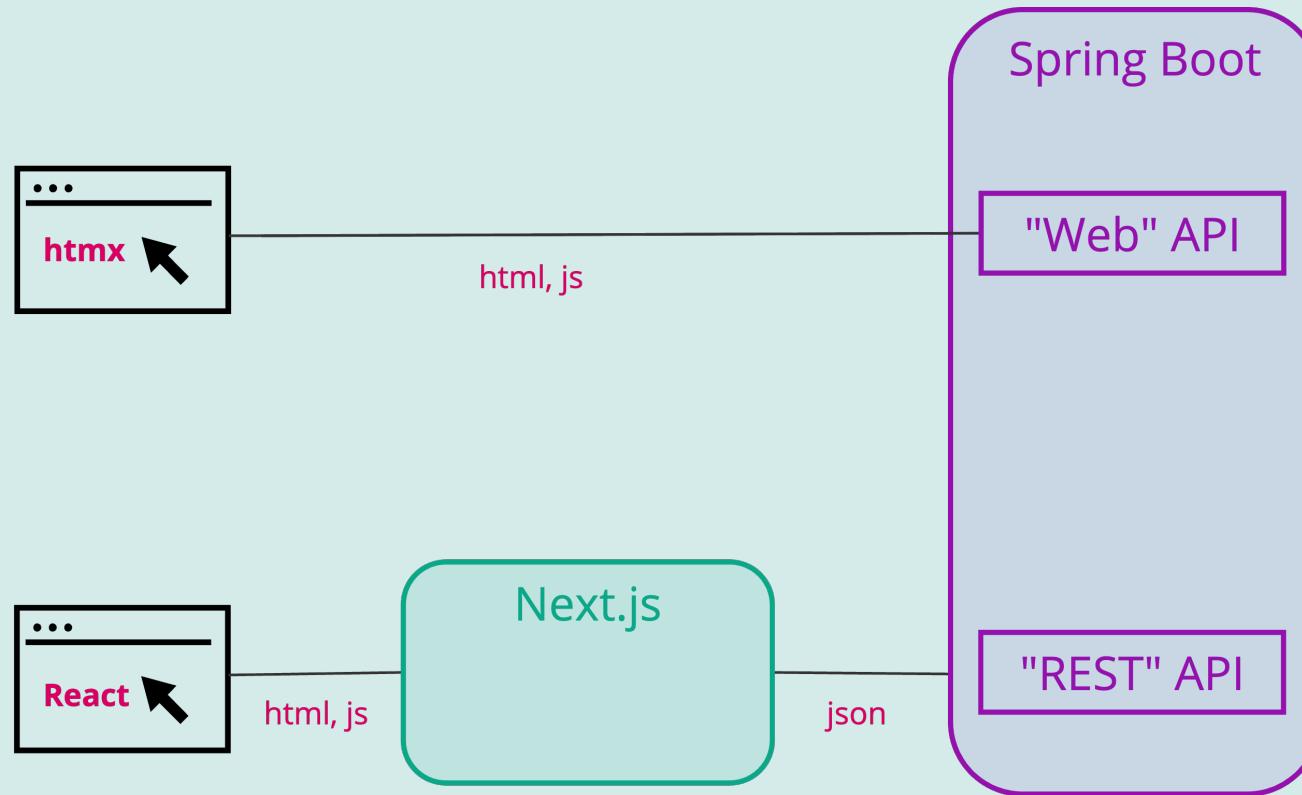


[HTTP://LOCALHOST:8080](http://localhost:8080)

Architektur und Techstack Recipify



Architektur und Techstack Recipify



Beispiel: Formulare

- Wie werden die Daten gesendet?
- Was kommt als Antwort?
- Wie disablen wir das Formular?
- Wie setzen wir eine (Fehler-)Meldung zurück
- 🧑‍💻 NewsletterRegistration.jte / .tsx

Subscribe to newsletter

E-Mail

Subscribe

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

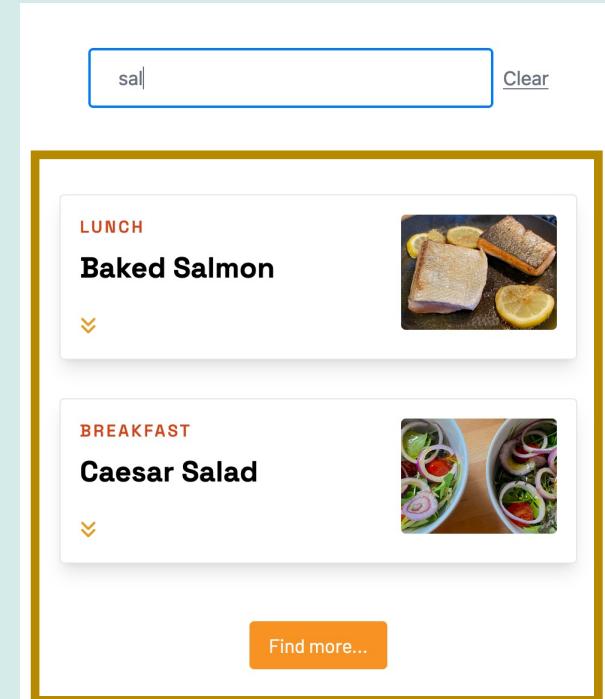
 [Clear](#)

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

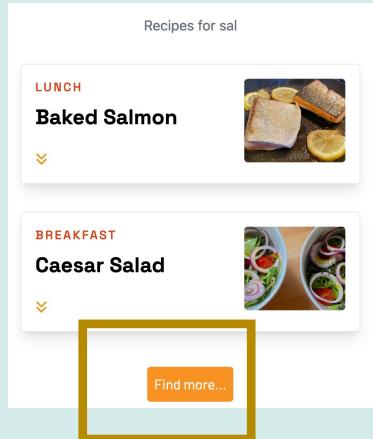
 [Clear](#)

GET /search



CODE-BEISPIELE

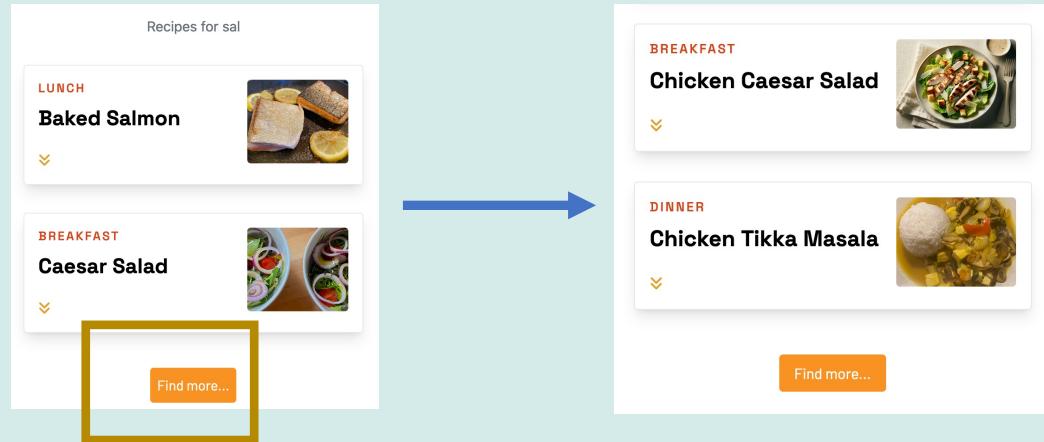
Beispiel: Bestehende Darstellung aktualisieren



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

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

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
▼ 

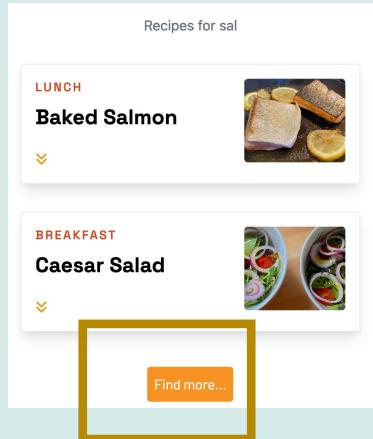
DINNER
Chicken Tikka Masala
▼ 

[Find more...](#)

GET /search?page=3

CODE-BEISPIELE

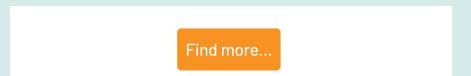
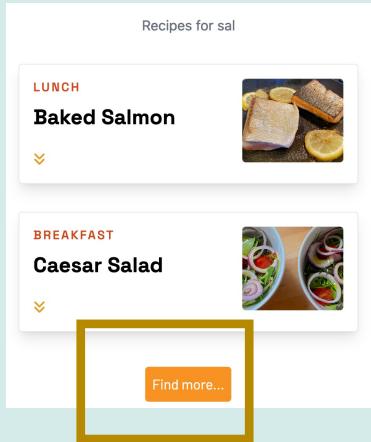
Beispiel: Bestehende Darstellung aktualisieren



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"`



GET /search?page=2

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...

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"`



Find more...

Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

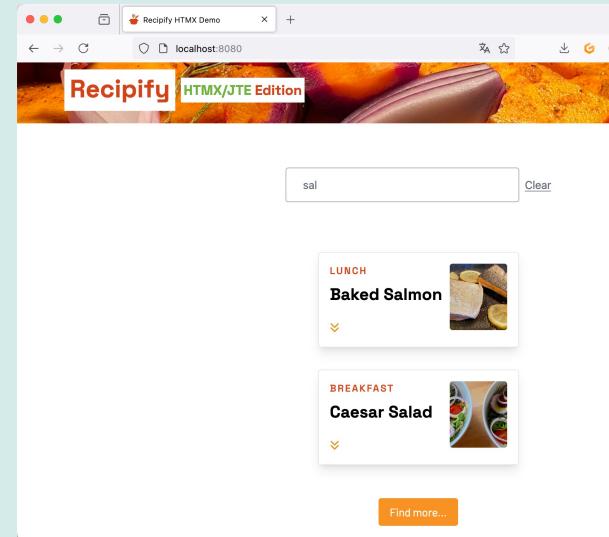
BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

GET /search?page=3

Beispiel: Parallel Requests

-  Netzwerk-Tab

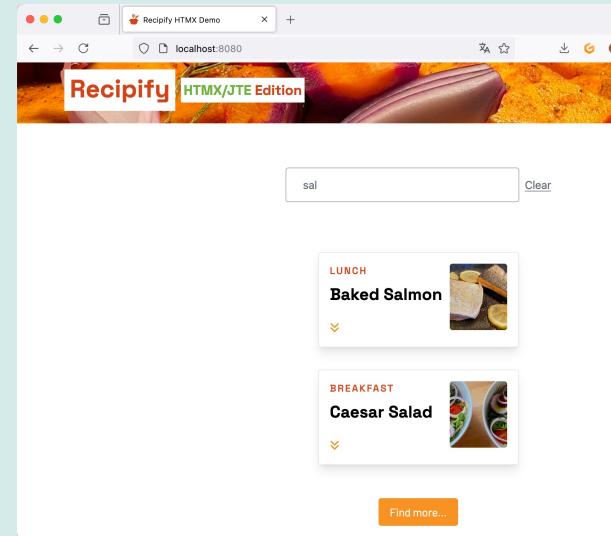


```
SearchPageContent.jte: slowdown=2000, dafür delay raus
```

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

Beispiel: Parallel Requests

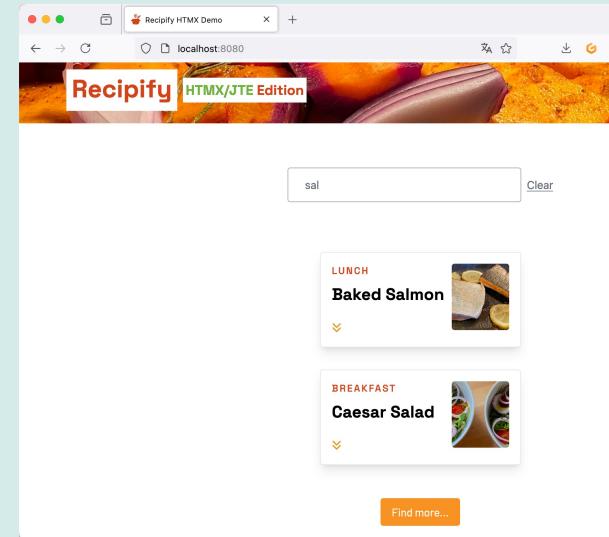
-  Netzwerk-Tab
- HTMX führt Requests nacheinander aus
- In der SPA können wir das selbst entscheiden



Beispiel: Parallel Requests

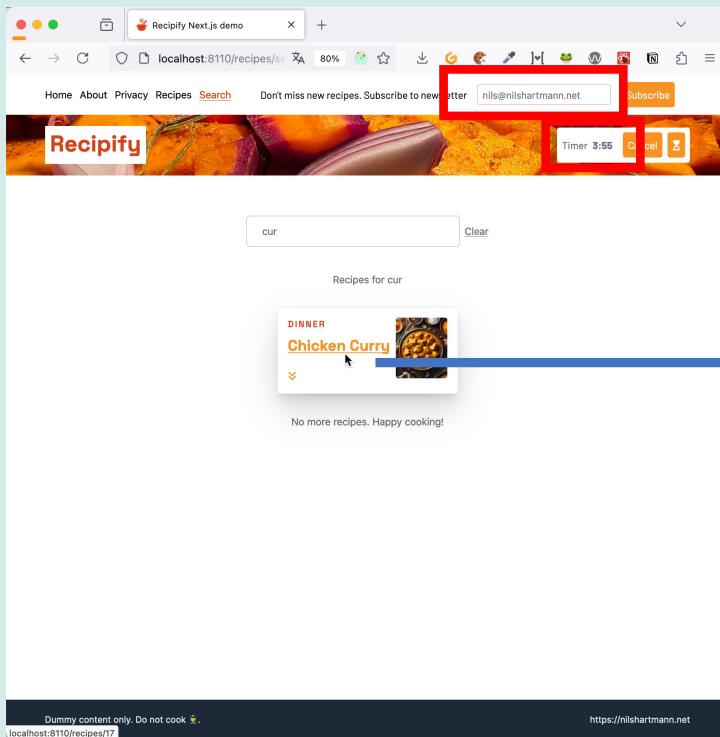
- 🕵️ Netzwerk-Tab
- HTMX führt Requests nacheinander aus
- "Alte" Request werden nicht verworfen
- In der SPA können wir das selbst entscheiden

SearchPageContent.jte: slowdown

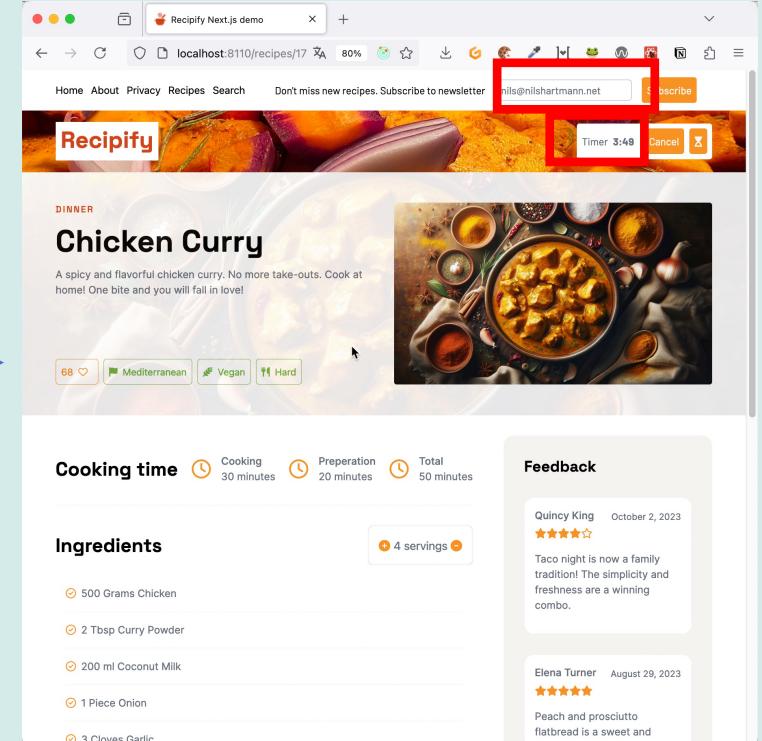


Beispiel: SPA-like Seitenwechsel

- 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

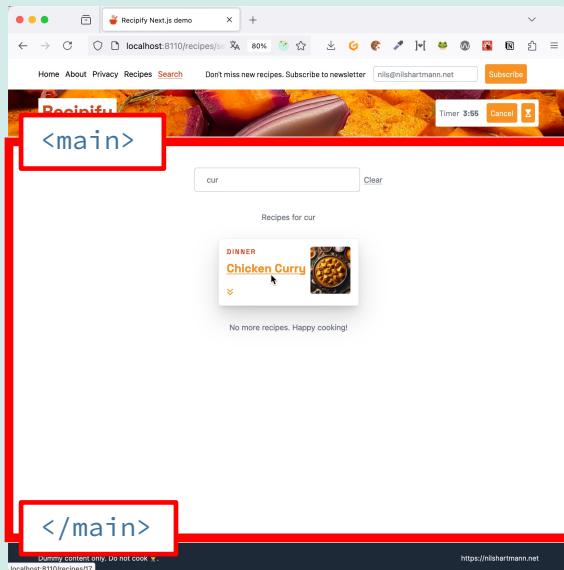
- Link muss entsprechend ausgezeichnet werden

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

RecipeSummaryCard.jte

Beispiel: SPA-like Seitenwechsel in HTMX

Variante 1



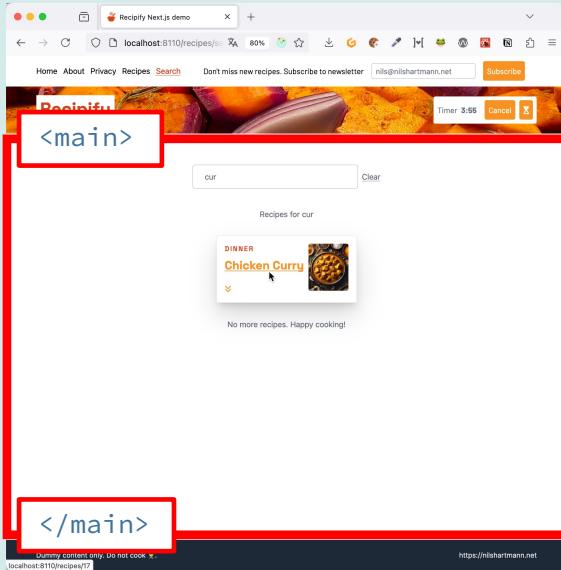
RecipeSummaryCard.jte

CODE-BEISPIELE

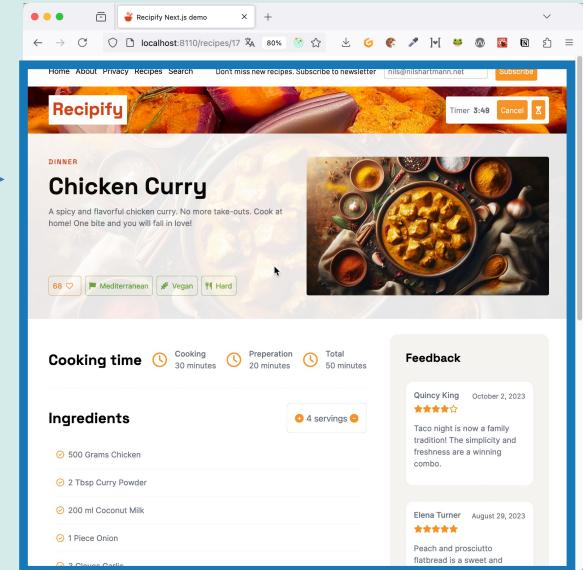
Beispiel: SPA-like Seitenwechsel in HTMX

Variante 1

RecipeSummaryCard.jte



/recipes/17

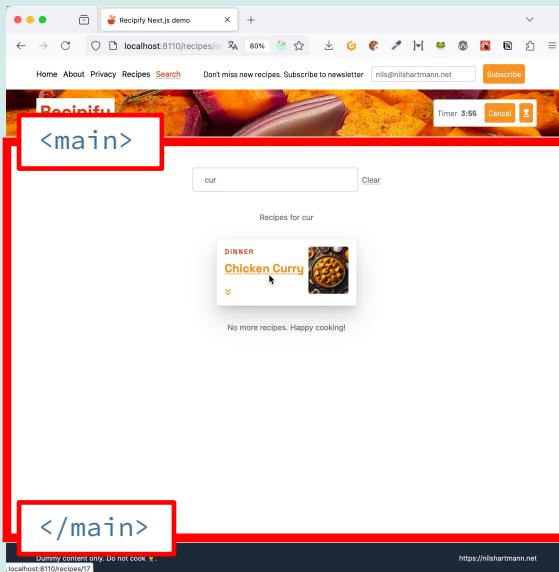


CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

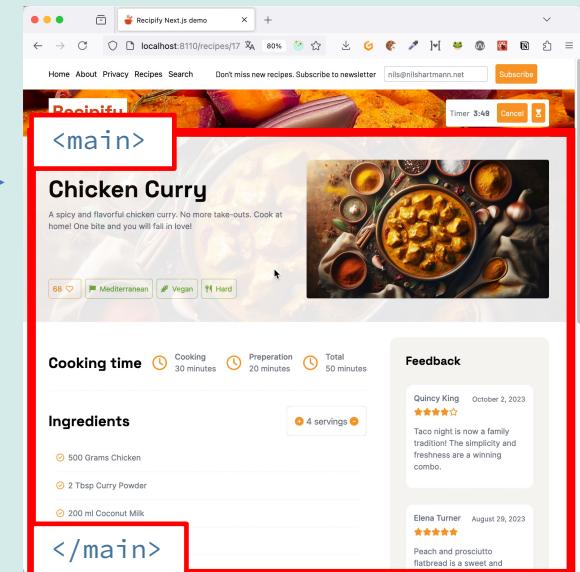
Variante 1

RecipeSummaryCard.jte



/recipes/17

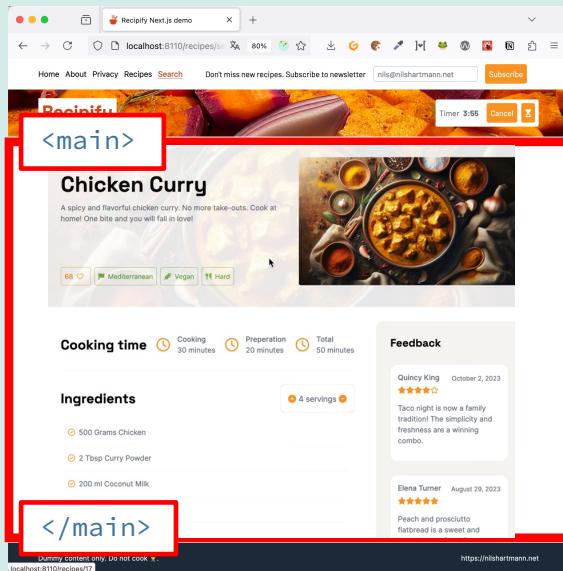
hx-select="main"



CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

Variante 1

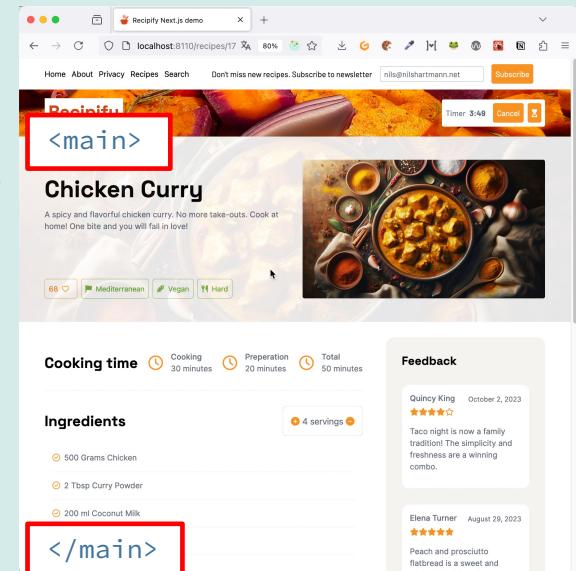


/recipes/17

hx-select="main"

hx-target="main"

RecipeSummaryCard.jte



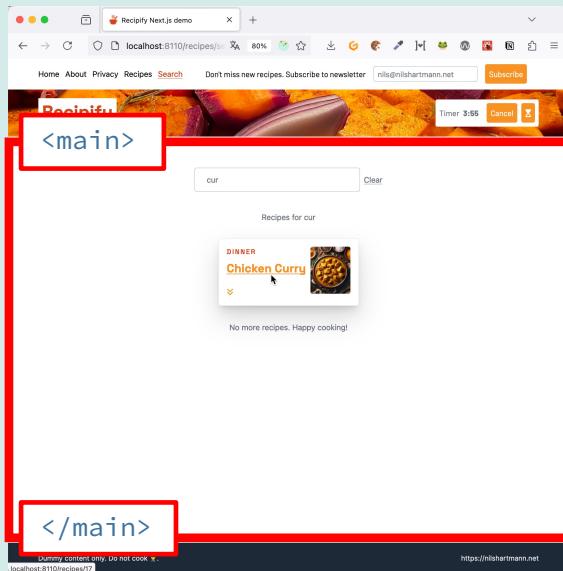
Variante 1: HTMX

- **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

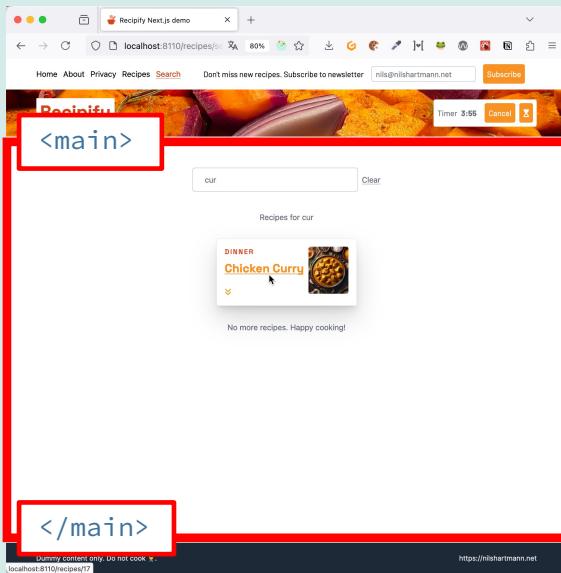
Variante 2: Eigener Endpunkt



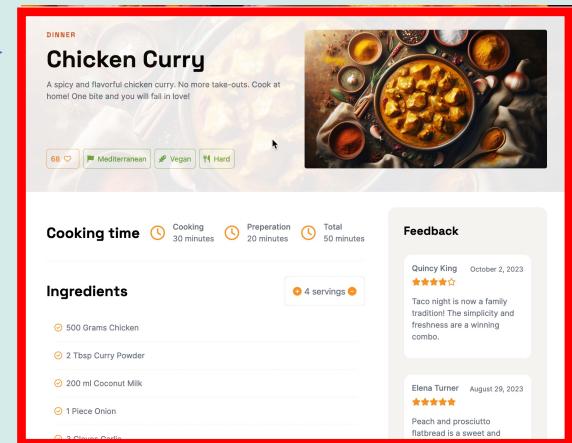
CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

Variante 2: Eigener Endpunkt



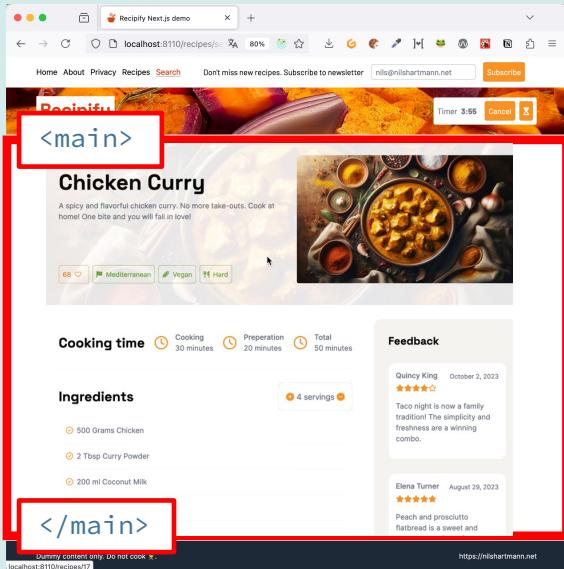
/recipes/17/details



CODE-BEISPIELE

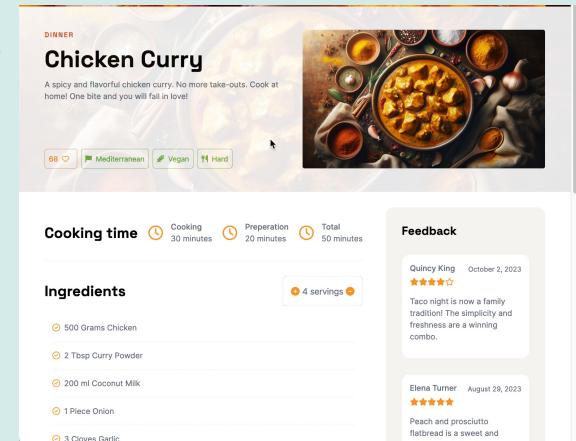
Beispiel: SPA-like Seitenwechsel in HTMX

Variante 2



/recipes/17/details

hx-target="main"



Variante 2: Eigener Endpunkt

- **Link** muss wissen:

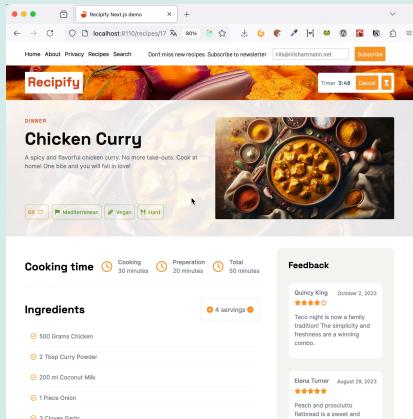
- welcher Endpunkt
- wo muss Ergebnis auf der aktuellen Seite eingebaut werden

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`

CODE-BEISPIELE

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, and cooking time information (Cooking 30 minutes, Preparation 20 minutes, Total 50 minutes). It also includes a "Feedback" section with reviews from users like Quincy King and Elena Turner, and a "Ingredients" section listing items such as 500 Grams Chicken, 2 Tbsp Curry Powder, 200 ml Coconut Milk, 1 Peice Onion, and 3 Cloves Garlic.

/recipes/17

The screenshot shows the Recipify details page for the same "Chicken Curry" recipe. This view is specifically for the "details" endpoint. It includes the title "Chicken Curry", a brief description, cooking time (Cooking 30 minutes, Preparation 20 minutes, Total 50 minutes), a "Feedback" section with reviews, and an "Ingredients" section. The ingredients listed are identical to the full page: 500 Grams Chicken, 2 Tbsp Curry Powder, 200 ml Coconut Milk, 1 Peice Onion, and 3 Cloves Garlic.

/recipes/17/details

CODE-BEISPIELE

Beispiel: Seite priorisieren

The screenshot shows a recipe card for "Classic Caesar Salad" on the Recipify platform. The card includes the following details:

- Category:** BREAKFAST
- Title:** Classic Caesar Salad
- Description:** Crispy romaine lettuce with creamy Caesar dressing. Bon Appétit! Food that feels like home.
- Image:** A large, vibrant photograph of a Caesar salad in a bowl, garnished with croutons and parmesan cheese.
- Metrics:** 92 likes, Asian cuisine, 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:** (Listed below the ingredients)
- Feedback:** Loading feedback... (with three orange dots).
- Your opinion?** (Form fields for name and rating).

CODE-BEISPIELE

Beispiel: Seite priorisieren in Next.js

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

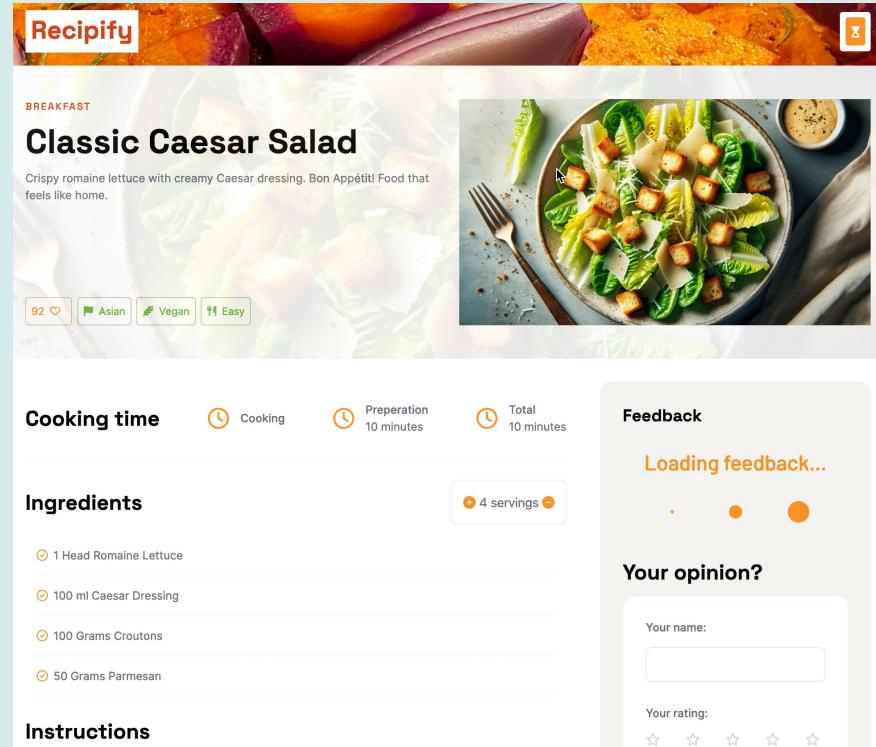
慢速请求慢速请求
slowDown_GetRecipe=2000

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 includes 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" 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.

Fazit

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

Single-Page-Anwendung oder HTMX

- It depends (natürlich 😐)

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

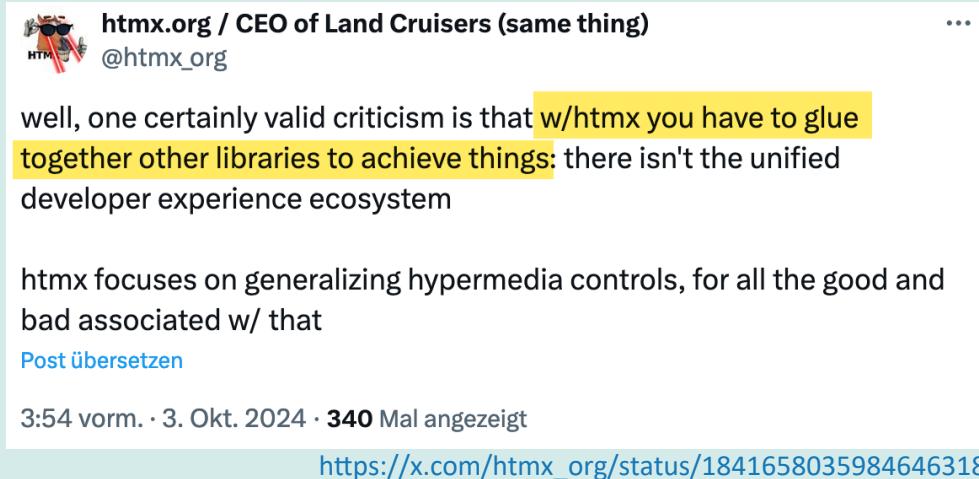
Single-Page-Anwendung oder HTMX

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

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

Single-Page-Anwendung 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
- Früher oder später braucht man doch JavaScript



htmx.org / CEO of Land Cruisers (same thing) @htmx_org ...

well, one certainly valid criticism is that w/htmx you have to glue together other libraries to achieve things: there isn't the unified developer experience ecosystem

htmx focuses on generalizing hypermedia controls, for all the good and bad associated w/ that

[Post übersetzen](#)

3:54 vorm. · 3. Okt. 2024 · 340 Mal angezeigt

https://x.com/htmx_org/status/1841658035984646318

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,

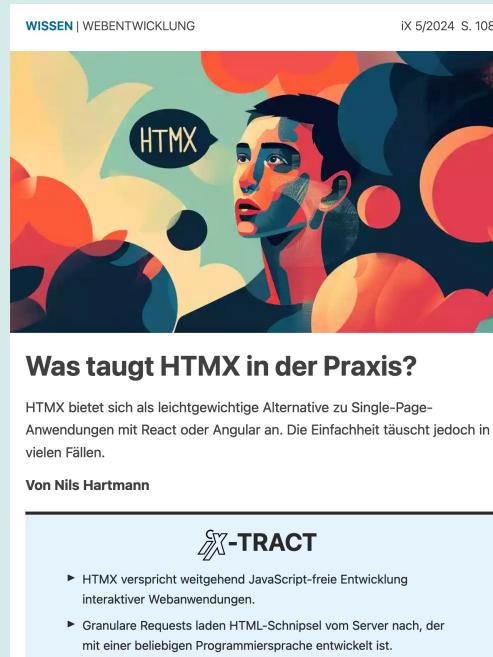
dann ist man vielleicht falsch in der Frontend-Entwicklung



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)

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.

Beispiel-Anwendung

Source-Code und Slides:

[**https://react.schule/wjax-2024-htm**](https://react.schule/wjax-2024-htm)

Nach dem Vortrag checke ich den geschriebenen "Live-Code" auf einem neuen
Branch ein

NILS HARTMANN
<https://nilshartmann.net>



vielen Dank!

Slides & Code: <https://react.schule/wjax-2024-htm>

Fragen und Kontakt

nils@nilshartmann.net

<https://nilshartmann.net/kontakt>

