

NILS HARTMANN

<https://nilshartmann.net>

Frontend-Entwicklung

HTMX
oder

Single-
Page-
Anwendung ?

NILS HARTMANN

<https://nilshartmann.net>

Frontend-Entwicklung

HTMX

oder

Single-
Page-

Anwendung

?

oder

JS-Fullstack-Anwenddung

NILS HARTMANN

nils@nilshartmann.net

Freiberuflicher Entwickler, Architekt, Trainer

Java, Spring, GraphQL, React, TypeScript



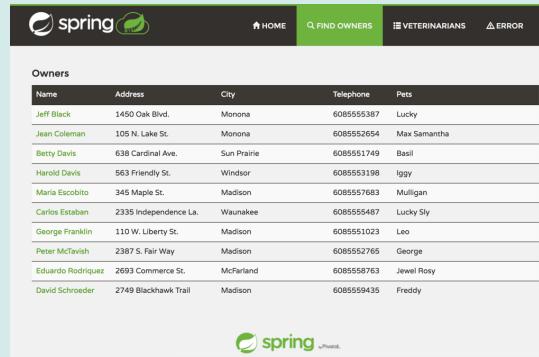
<https://graphql.schule/video-kurs>

<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 table of pet owners. The table has columns for Name, Address, City, Telephone, and Pets. There are 12 rows of data.

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

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 various individuals with their pets. The table includes columns for Name, Address, City, Telephone, and Pets. Notable entries include Jeff Black with his dog Lucky, Jean Coleman with Max Samantha, and Harold Davis with Iggy.

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 recent activities.
- 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 of 108 km/h, and a connection to Hamburg-Altona.
- Hibernate**: A project management tool showing a list of tasks and projects related to "Hibernate ORM".
- Microsoft Teams**: A communication platform showing a chat history and a search bar.
- 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 platform for organizing events, showing a list of attendees for a "Frontend for Backend: HTMX oder Single-Page-Anwendung?" event.

WEBANWENDUNGEN

The image displays a collage of various web application screenshots, illustrating different 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 and file sharing interface.
- Spring**: A developer-oriented platform showing a list of owners with details like name, address, city, phone, and pets.

WEBANWENDUNGEN

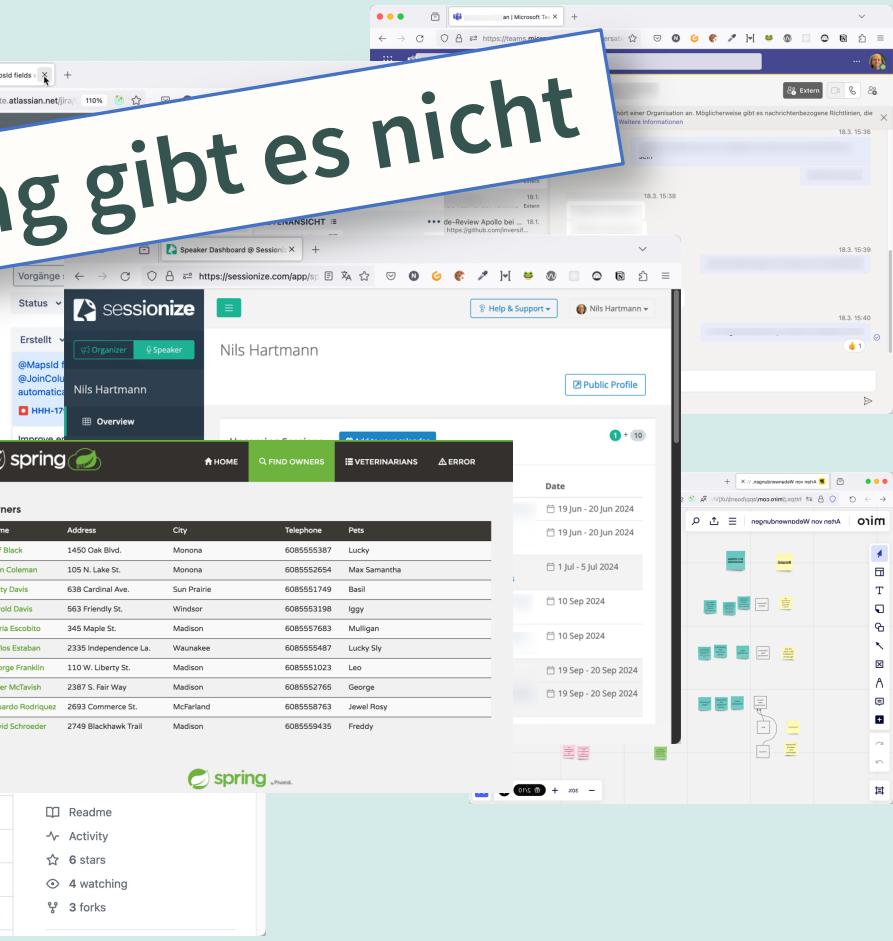
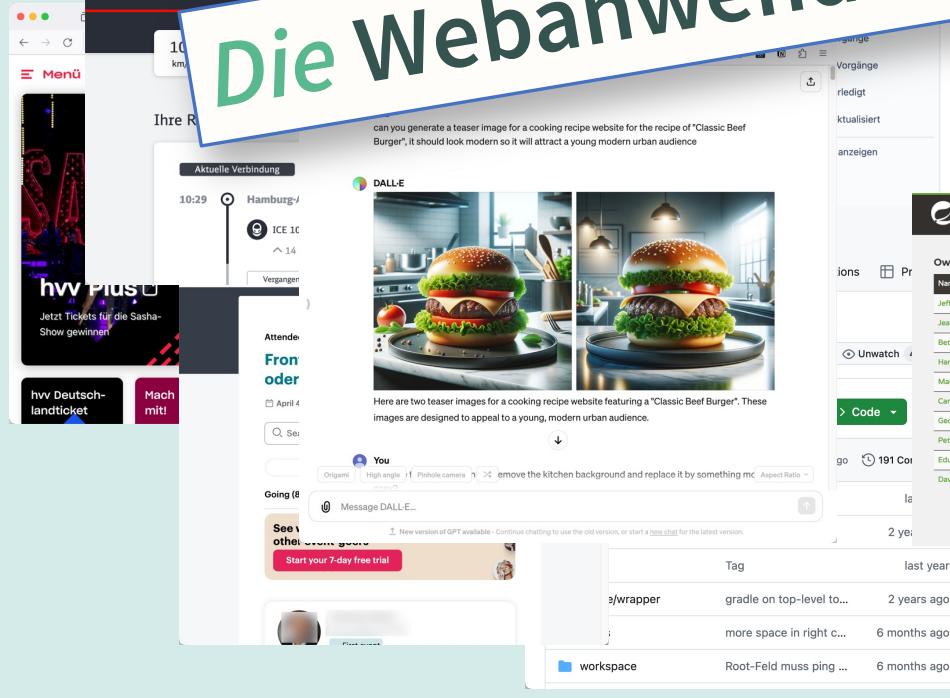
Die Webanwendung gibt es nicht

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

Die Webanwendung gibt es nicht



WEBANWENDUNGEN

Die Webanwendung gibt es nicht

...aber: (fast) alle brauchen JavaScript



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

- A central window titled "Booking.com" displays "Reiseinformationen für den ICE 109".
- To the left, a window titled "ICE Portal" shows "Nächster Halt in 6 min" and "Gleis 2".
- At the bottom, a window titled "DALL-E" shows an AI-generated image of a classic beef burger.
- On the right, a window titled "Speaker Dashboard @ Sessionize" shows a profile for "Nils Hartmann".
- At the bottom, a GitHub repository page for "springframework/spring" is visible.

A screenshot of a Microsoft Teams interface:

- A large blue banner across the top reads "Die Webanwendung gibt es nicht".
- The main area shows a "Vorgänge" (Tasks) list for "Nils Hartmann".
- To the right, a "Kalender" (Calendar) shows events for June and July 2024.

A screenshot of a GitHub repository page for "springframework/spring".

Key sections visible:

- Code**: Shows a list of commits and pull requests, including:
 - Commit 6085512654 by Max Samantha: "remove the kitchen background and replace it by something more modern"
 - Commit 6085511749 by Basil: "more space in right column"
 - Commit 6085557683 by Mulligan: "fixing a bug"
 - Commit 6085554837 by Lucky Sly: "improving API documentation"
 - Commit 6085551023 by Leo: "fixing a bug"
 - Commit 6085552765 by George: "fixing a bug"
 - Commit 6085558763 by Jewel Rosy: "fixing a bug"
 - Commit 6085559435 by Freddy: "fixing a bug"
- Readme**: A file listing project details.
- Activity**: A section showing recent activity.
- 6 stars**: A rating section.
- 4 watching**: A section showing who is watching the repository.
- 3 forks**: A section showing who has forked the repository.

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 the Booking.com homepage. Below it is the ICE Portal showing travel information for the ICE 109. To the right 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 the DALL-E interface, showing two images of a beef burger and a message from DALL-E. To the right of DALL-E is a screenshot of the Sessionize speaker dashboard for Nils Hartmann. At the bottom right is a screenshot of the Spring website showing a list of owners with columns for Name, Address, City, Telephone, and Pets. The bottom left shows a portion of a calendar application with a grid of tasks.

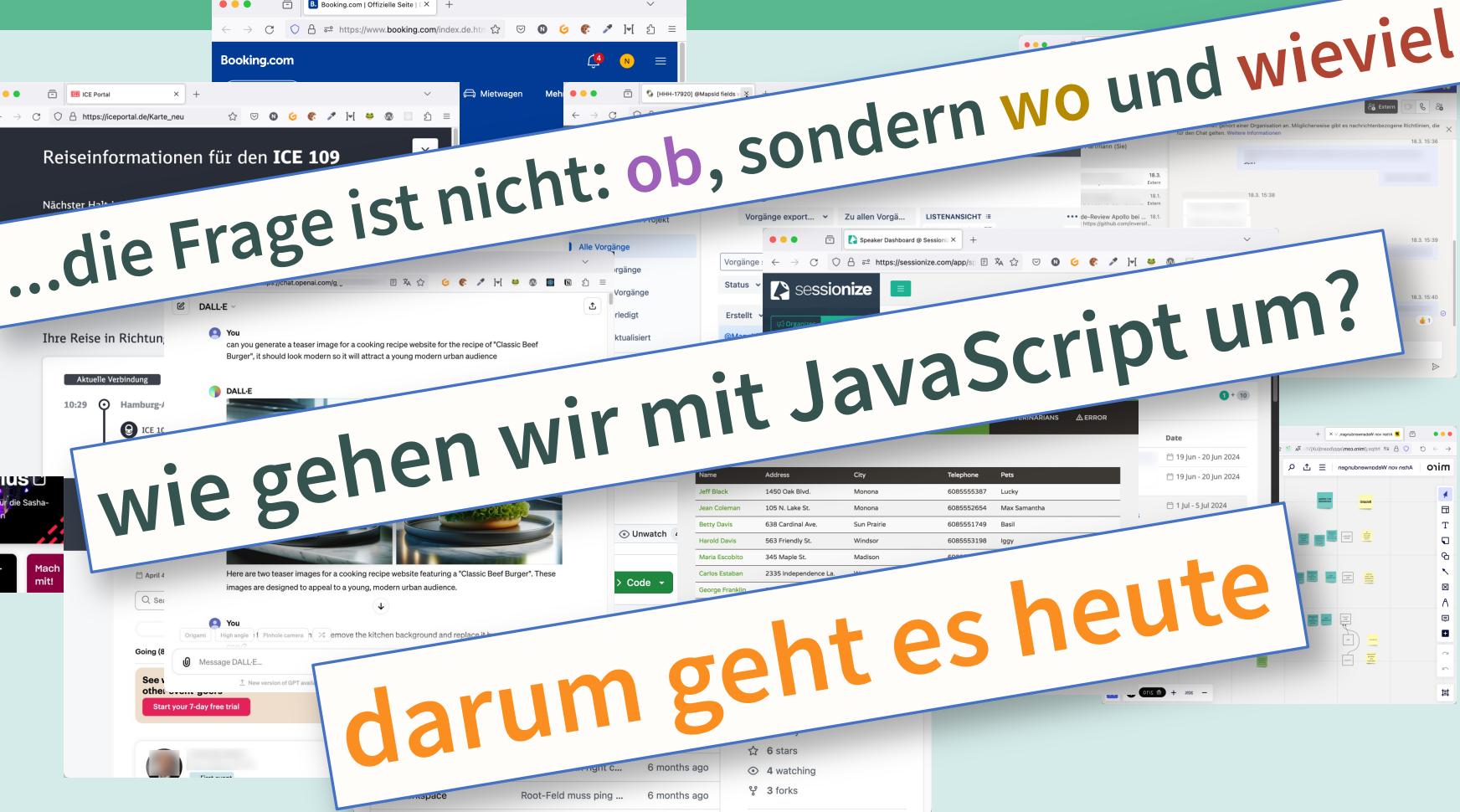
WEBANWENDUNGEN

...die Frage ist nicht: ob, sondern wo und wieviel wie gehen wir mit JavaScript um?

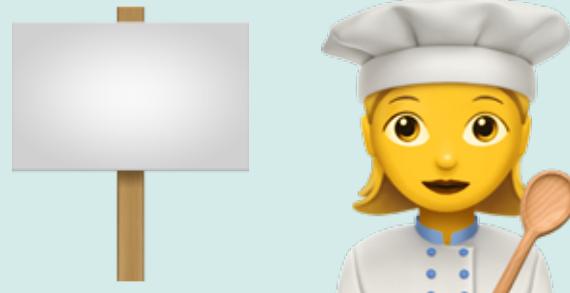
The image is a collage of several screenshots of different web applications, illustrating various types of web-based tools and interfaces:

- A screenshot of the Booking.com homepage.
- A screenshot of the ICE Portal showing travel information for the ICE 109.
- A screenshot of the DALL-E interface, showing a conversation with AI about generating images for a cooking recipe website.
- A screenshot of the sessionize.com Speaker Dashboard.
- A screenshot of a GitHub Issues page for a project named "Mappid fields".
- A screenshot of a calendar application showing multiple events and tasks.

WEBANWENDUNGEN



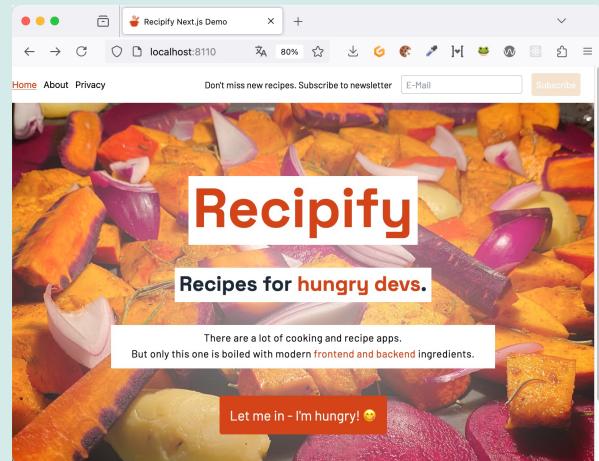
Demo



BEISPIELE

Eine Webanwendung...

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

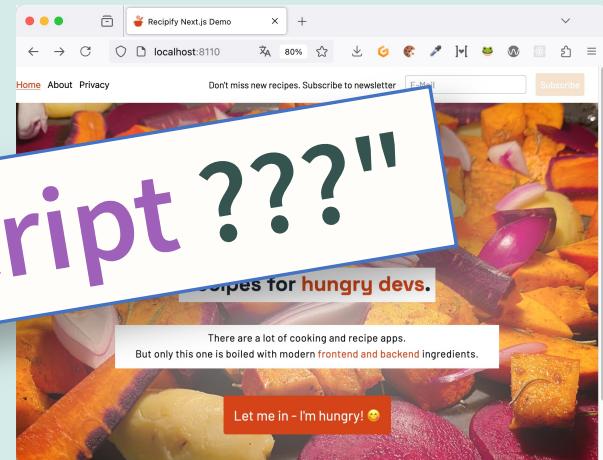


<http://localhost:8110>

BEISPIELE

Eine Webanwendung...

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

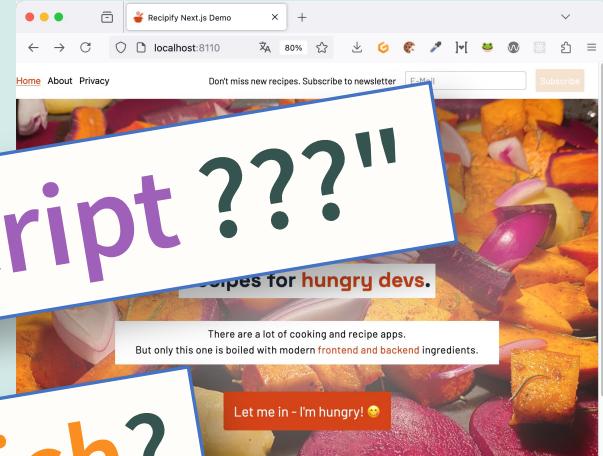


<http://localhost:8110>

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

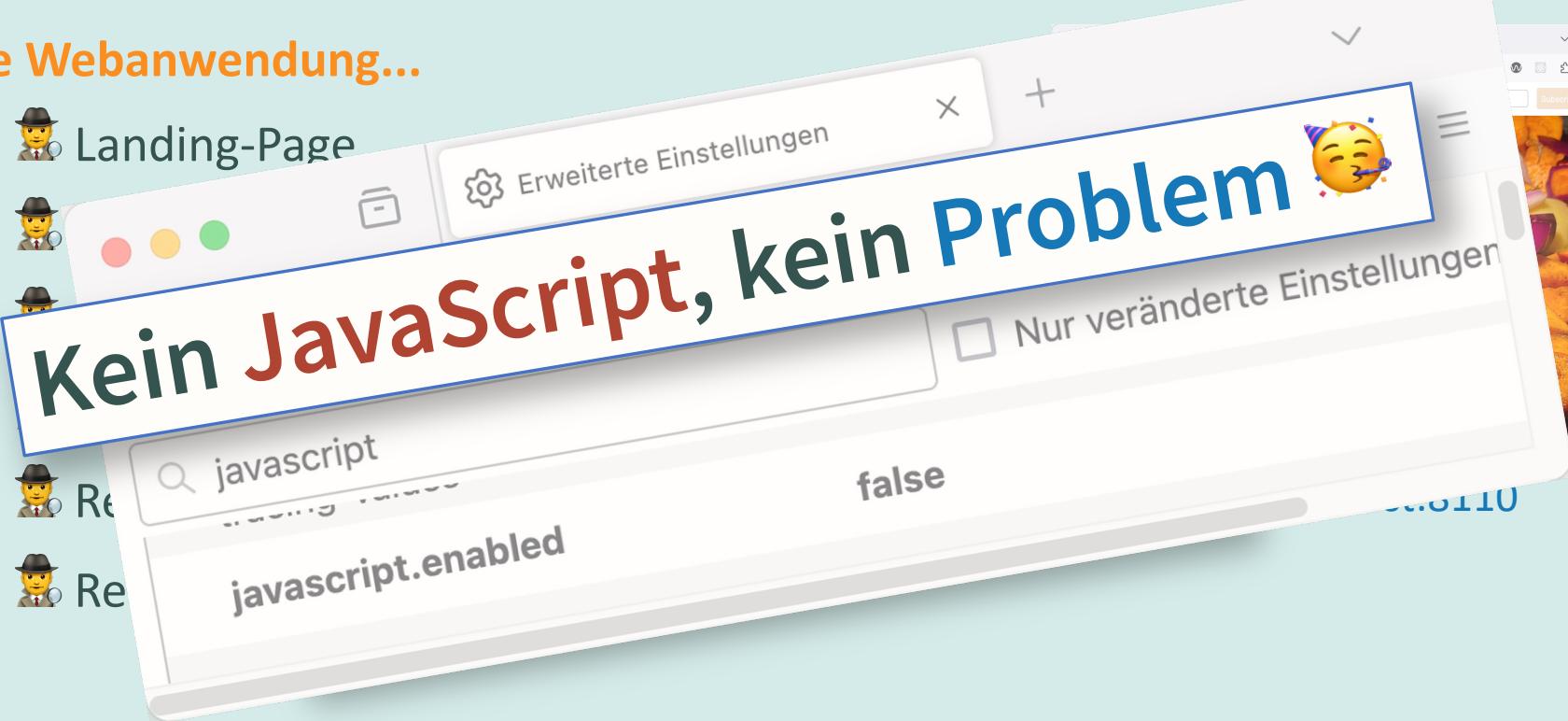
5. 🕵️ Re

6. 🕵️ Re



BEISPIELE

Eine Webanwendung...

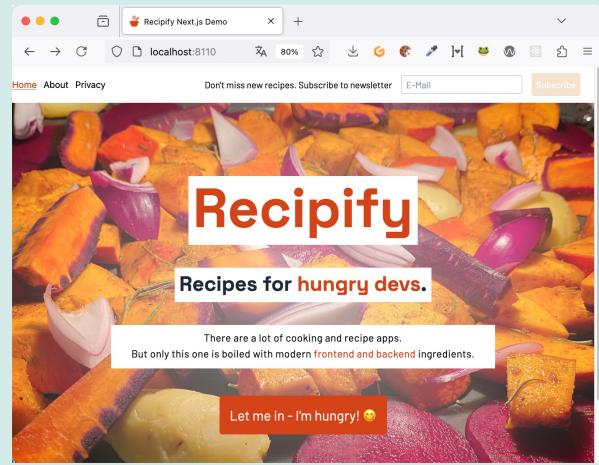
1. 🕵️ Landing-Page
 2. 🕵️
 3. 🕵️ **Kein JavaScript, kein Problem** 😊
 - 4.
 5. 🕵️ Re
 6. 🕵️ Re
- 

Eine Webanwendung...

1. 🕵️ Landing-Page
 2. 🕵️
 3. 🕵️
 4. 🕵️
 5. 🕵️ Re
 6. 🕵️ Re
-
- Kein JavaScript, kein Problem
- Feierabend machen?

BEISPIELE

Offenbar geht's doch auch ohne

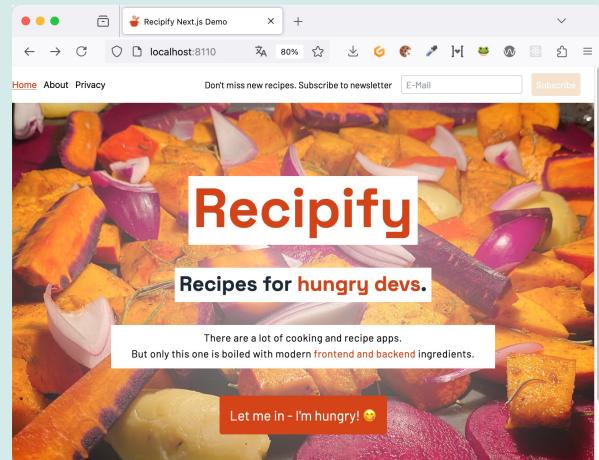


<http://localhost:8110>

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 😢

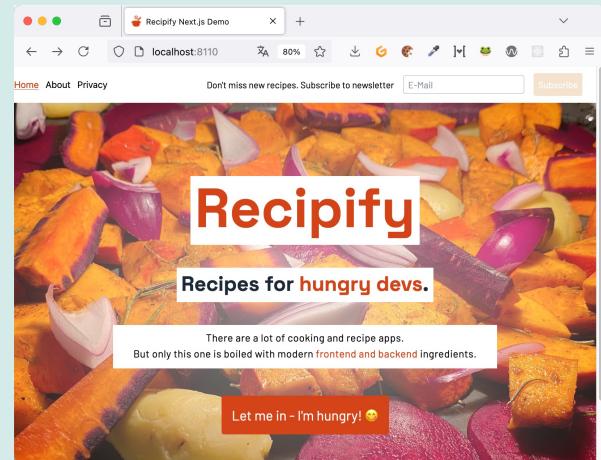


<http://localhost:8110>

Wie sieht es denn mit JavaScript aus?

`demo_config Suspense einschalten!`

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
- JS braucht man vielleicht schneller/früher als man häufig denkt

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
- 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**, PHP, dotNET, Node.js

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 Resultat von localhost:8080/hello

hx-trigger, hx-get und hx-target

- HelloWorldController

("Hypermedia")

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

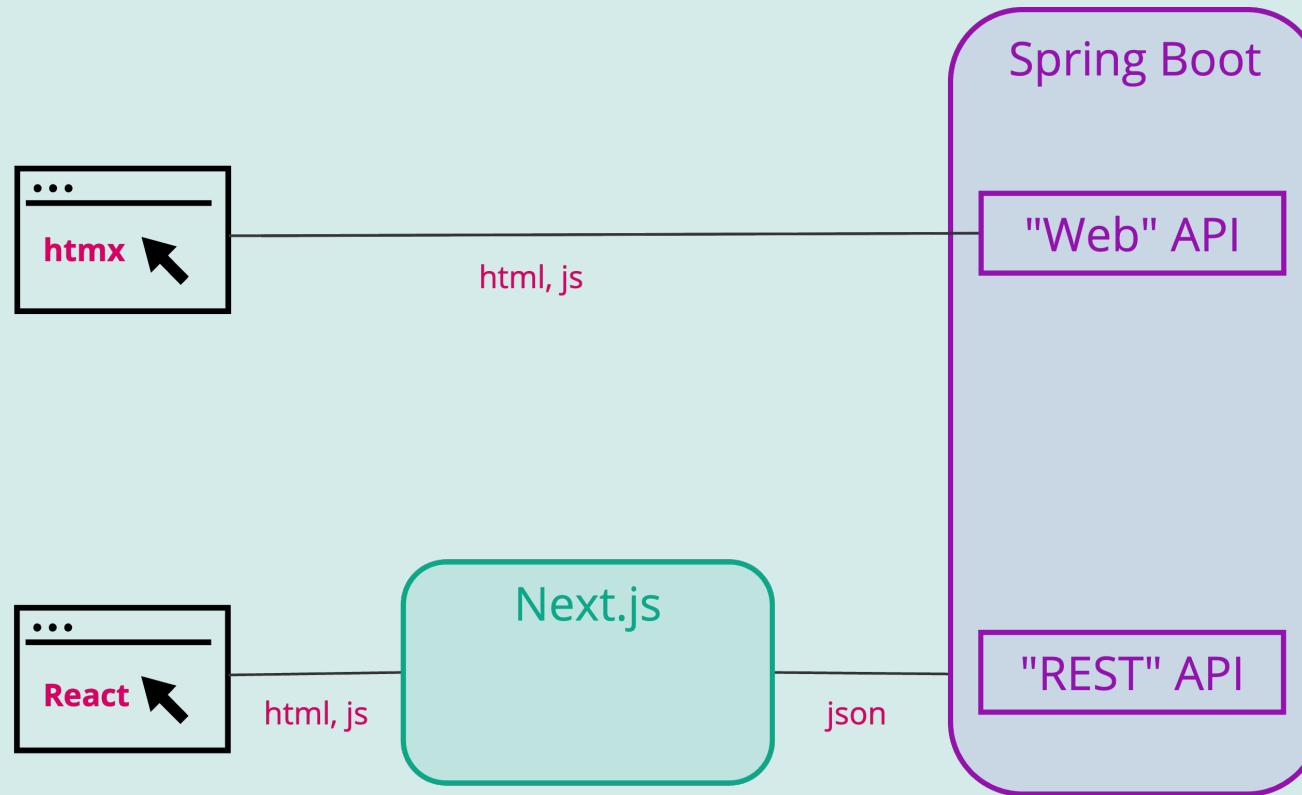
    <div id="result"></div>

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

Code Beispiele



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

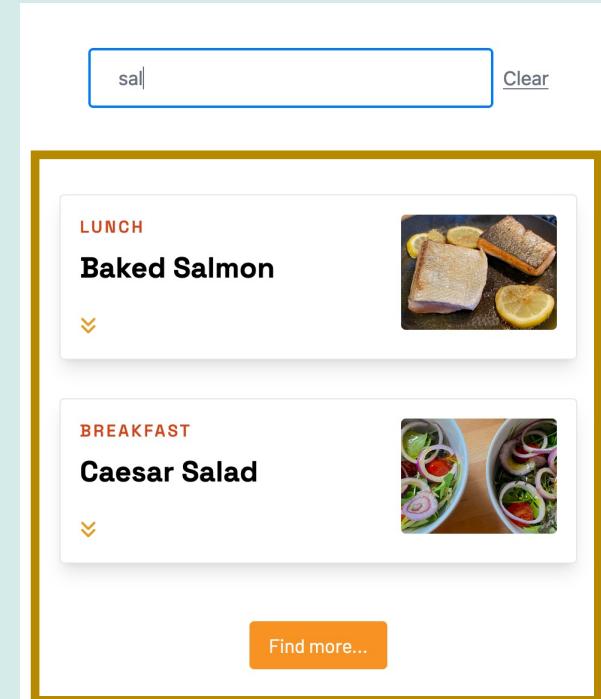
 [Clear](#)

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

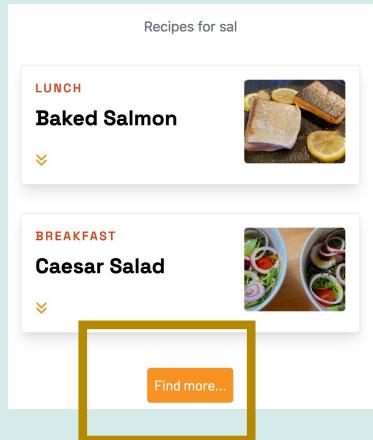
 [Clear](#)

GET /search



CODE-BEISPIELE

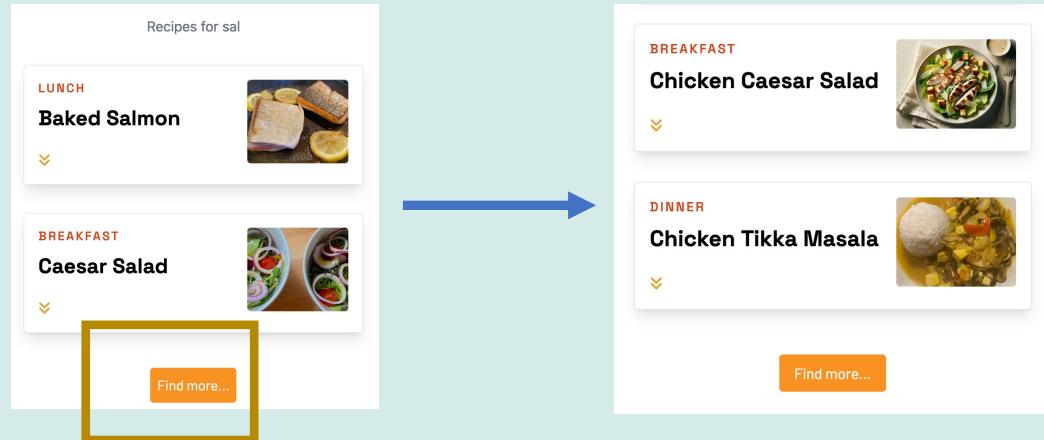
Beispiel: Bestehende Darstellung aktualisieren



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...](#)



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...](#)

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...](#)

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...](#)

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...](#)

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...](#)

BREAKFAST
Chicken Caesar Salad
▼ 

DINNER
Chicken Tikka Masala
▼ 

[Find more...](#)

GET /search?page=2

Chicken Caesar Salad

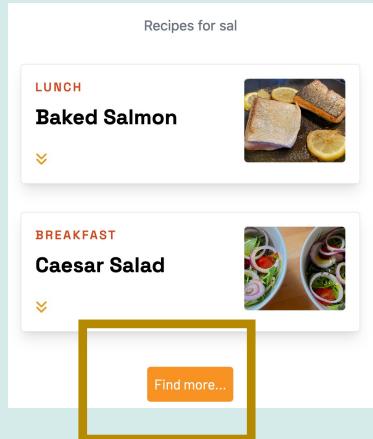
Chicken Tikka Masala

GET /search?page=3

Find more...

CODE-BEISPIELE

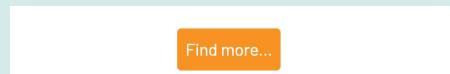
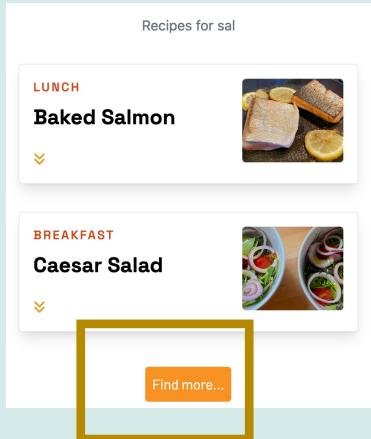
Beispiel: Bestehende Darstellung aktualisieren



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 

▼

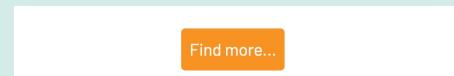
BREAKFAST

Caesar Salad 

▼

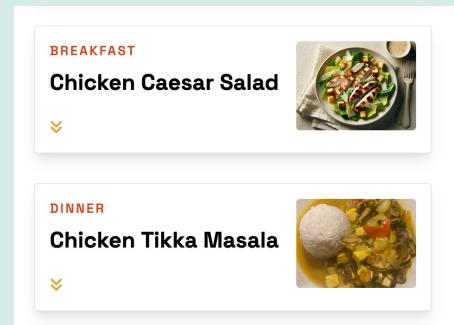
[Find more...](#)

GET /search?page=2



Find more...

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



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

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

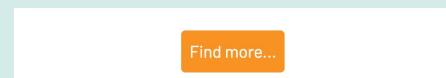
Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

[Find more...](#)

GET /search?page=2



`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

CODE-BEISPIELE

Beispiel: Bestehende Darstellung aktualisieren

Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

[Find more...](#)

GET /search?page=2

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
▼ 

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"`



Recipes for sal

LUNCH
Baked Salmon
▼ 

BREAKFAST
Caesar Salad
▼ 

BREAKFAST
Chicken Caesar Salad
▼ 

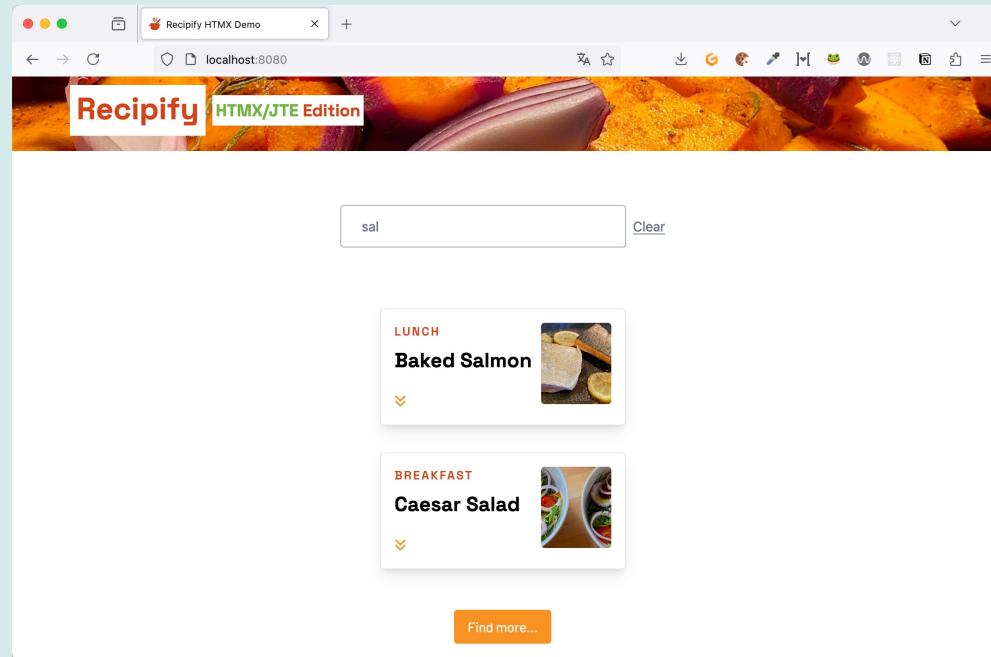
DINNER
Chicken Tikka Masala
▼ 

[Find more...](#)

GET /search?page=3

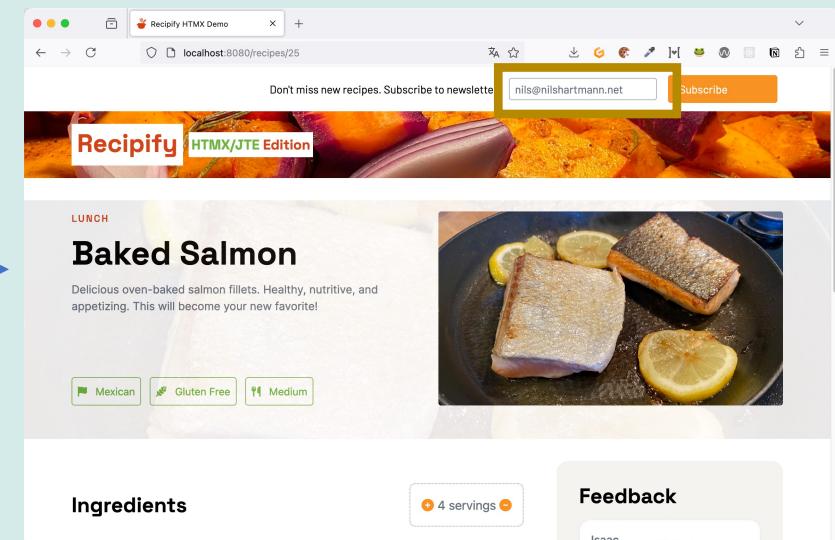
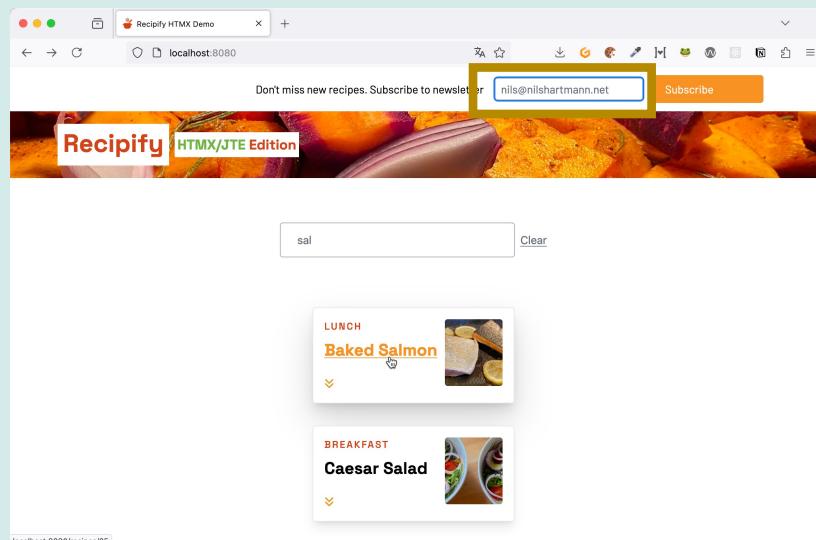
Beispiel: Parallel Requests

-  Netzwerk-Tab



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

CODE-BEISPIELE

Beispiel: SPA-like Seitenwechsel in HTMX

Beispiel: SPA-like Seitenwechsel in HTMX

- Link muss entsprechend ausgezeichnet werden
 - mit und ohne JS
 - Ganze Seite vs. nur Content
 - Aktualisieren der URL

```
RecipeSummaryCard.jte
```

Beispiel: SPA-like Seitenwechsel in HTMX

- Link muss entsprechend ausgezeichnet werden

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

```
RecipeSummaryCard.jte
```

- "Controller": Eventuell zwei Endpunkte
 - ganze Seite vs. Teil-Update

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 bowl of Caesar salad with croutons and parmesan cheese.
- Metrics:** 92 ❤️, Asian, Vegan, Easy
- Cooking time:** Cooking 10 minutes, Preparation 10 minutes, Total 10 minutes
- Feedback:** Loading feedback... (with three orange dots)
- Ingredients:**
 - 1 Head Romaine Lettuce
 - 100 ml Caesar Dressing
 - 100 Grams Croutons
 - 50 Grams Parmesan
- Instructions:** (instructions are currently empty)
- Feedback Form:** Your name: _____, Your rating: ★ ★ ★ ★ ★

CODE-BEISPIELE

Beispiel: Seite priorisieren in Next.js

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

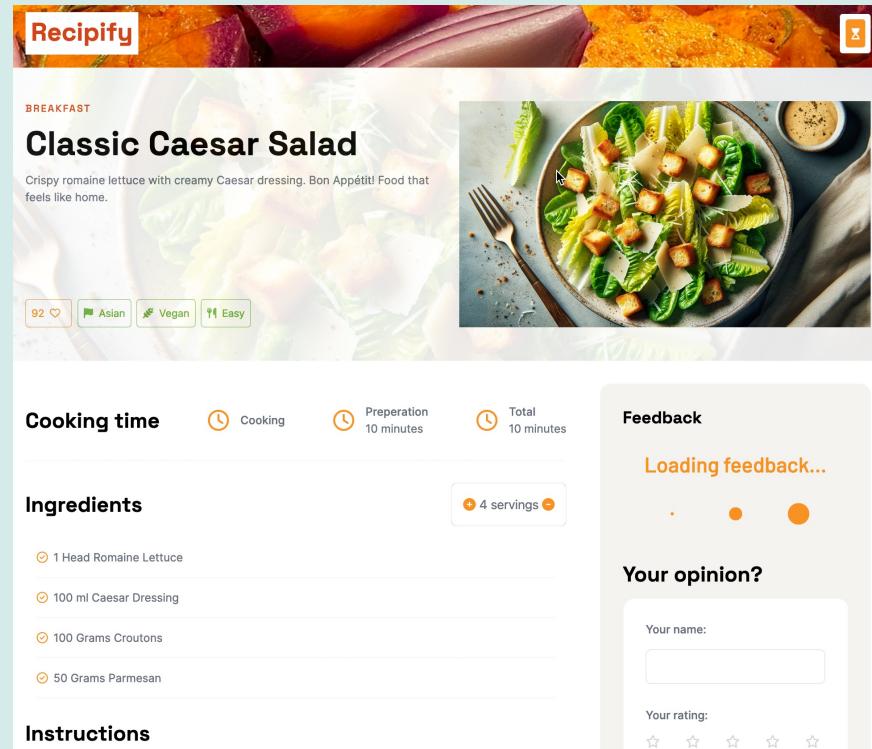
慢速请求食谱慢速请求食谱
slowDown_GetRecipe=2000

Spring Controller Debug

慢速请求网络Tab总时长
Netzwerk-Tab Gesamtdauer

慢速请求食谱ID页
[recipeId]/page.tsx

慢速请求食谱页内容
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 includes the following details:

- CATEGORY:** BREAKFAST
- IMAGE:** A large image of a bowl of Caesar salad with lettuce, croutons, and dressing.
- DESCRIPTION:** Crispy romaine lettuce with creamy Caesar dressing. Bon Appétit! Food that feels like home.
- STATS:** 92 ❤️, Asian, Vegan, Easy
- TIME:** Cooking 10 minutes, Preparation 10 minutes, Total 10 minutes
- SERVINGS:** 4 servings
- 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)
- OPINION:** Your opinion? (with fields for name and rating)

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 website Recipify. At the top, there's a banner with a close button. Below it, the title "Classic Caesar Salad" is displayed in bold. A description follows: "Crispy romaine lettuce with creamy Caesar dressing. Bon Appétit! Food that feels like home." To the left of the text, there's a small image of a salad. To the right, there's a large, appetizing photo of the finished dish in a bowl with croutons and dressing. Below the title, there are four small cards: "92 ❤️", "Asian", "Vegan", and "Easy". Under the photo, there are three time-related sections: "Cooking time" (with a clock icon), "Preparation" (10 minutes), and "Total" (10 minutes). To the right of these, there's a "Feedback" section with a placeholder "Loading feedback...". Below the cooking time, there's a "Ingredients" section with a list of items: "1 Head Romaine Lettuce", "100 ml Caesar Dressing", "100 Grams Croutons", and "50 Grams Parmesan". To the right of the ingredients, there's a "4 servings" indicator. At the bottom, there's an "Instructions" section and a "Your opinion?" section with fields for "Your name:" and "Your rating:" (with five star icons).

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

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

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

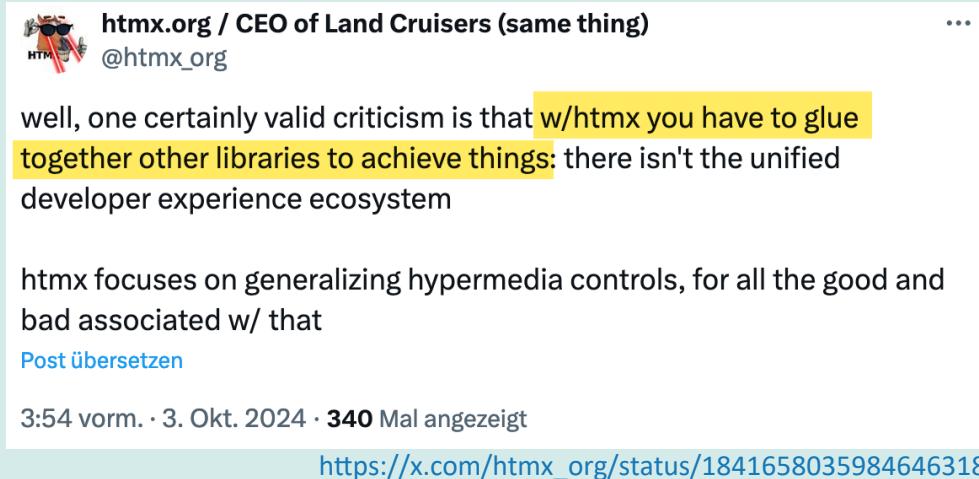
Single-Page-Anwendung oder HTMX

- 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)

<https://htmx.org/essays/a-real-world-react-to-htmx-port/>

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 spread from 'WISSEN | WEBENTWICKLUNG' issue IX 5/2024, page S. 108. The top half features a colorful abstract illustration of a person's face surrounded by large, stylized, overlapping shapes in shades of red, orange, yellow, and blue. A speech bubble above the head contains the text 'HTMX'. The bottom half contains the following text:

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.

NILS HARTMANN
<https://nilshartmann.net>



vielen Dank!

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

Fragen und Kontakt

nils@nilshartmann.net

<https://nilshartmann.net/kontakt>

