

NILS HARTMANN

<https://nilshartmann.net>

Frontend-Entwicklung

Single-

Page-

Anwendung

oder

HTMX?

NILS HARTMANN

nils@nilshartmann.net

Freiberuflischer Entwickler, Architekt, Trainer

Java, Spring, GraphQL, React, TypeScript



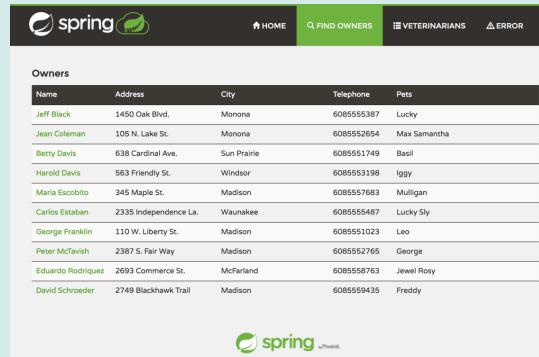
<https://graphql.schule/video-kurs>

<https://reactbuch.de>

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

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 the following data:

Name	Address	City	Telephone	Pets
Jeff Black	1450 Oak Blvd.	Monona	6085555387	Lucky
Jean Coleman	105 N. Lake St.	Monona	6085552654	Mox 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 a web browser window titled "ICE Portal" displaying travel information for ICE 109. At the top, it says "Reiseinformationen für den ICE 109". It indicates the next stop in 6 minutes at Gleis 2. The train is currently at Osnabrück Hbf, time 12:35, speed 108 km/h. A red bar highlights the speed. Below this, there's a map icon and a "Schnelles Internet" button. The main content area shows the route from Hamburg-Altona to Basel SBB, with stops like Hamburg-Altona, ICE 109 nach Basel SBB, and Gl. 10. Buttons for "Vergangene Halte zeigen", "Ab", "An", and "Suchen" are visible. The bottom of the page features promotional banners for "hw plus" and "hvv Deutschlandticket", along with social media links for "Mach mit!" and "Schüler*innen".

The screenshot shows a web application for "spring". The header includes a logo, navigation links for "HOME", "FIND OWNERS", "VETERINARIANS", and "ERROR", and a search bar. The main content is a table titled "Owners" listing pet owners with their addresses, city, telephone number, and pets. The table has columns for Name, Address, City, Telephone, and Pets.

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 "spring" logo and the word "invest".

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 page for the ICE 109 showing arrival information at Osnabrück Hbf at 12:35 (12:46) and a Genius Prämien offer for Nils.
- Spring**: A platform for pet owners. The interface includes a navigation bar with HOME, FIND OWNERS, VETERINARIANS, and ERROR links. Below is 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	608551749	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	608551023	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 a Mac desktop environment with several open windows illustrating various web applications:

- Booking.com**: A travel search results page for a flight from Osnabrück Hbf to Basel SBB.
- ICE Portal**: A train status board showing the next stop for ICE 109 at Gleis 2, arriving at Osnabrück Hbf at 12:46.
- Hibernate ORM**: A Jira project management interface for the "Hibernate ORM" software project, showing a list of tasks and their status.
- sessionize**: A speaker dashboard for Nils Hartmann, showing session details and a public profile.
- g-graphql-training**: A GitHub repository for a GraphQL training project, showing pull requests and code snippets.
- Attendee list for Frontend for Backend: HTMX oder Single-Page-Anwendung?**: An event attendee list from Ichiba GmbH, showing 80 attendees.

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 link to 'Schnelles Internet'.
- 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 with Nils Hartmann and other team members.
- 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 real-time train status application showing the next stop for the ICE 109 at Gleis 2.
- ChatGPT - DALLE**: An AI collaboration interface where ChatGPT generates a cooking recipe for a "Classic Beef Burger" and DALLE creates two images of the burger.
- HIBERNATE**: A project management tool for the Hibernate ORM software project.
- sessionize**: A speaker dashboard for Nils Hartmann.
- spring**: A developer-oriented website showing a table of owners with columns for Name, Address, City, Telephone, and Pets.
- Microsoft Teams**: A communication and collaboration platform showing a chat history and a grid of active conversations.

WEBANWENDUNGEN

Die Webanwendung gibt es nicht

The image is a collage of several screenshots from various web applications, demonstrating a wide range of software tools and platforms:

- ICE Portal:** A travel-related portal showing a map and flight information. It includes a speedometer icon indicating 108 km/h and a button for "Karte".
- Booking.com:** A screenshot of the Booking.com homepage.
- ChatGPT - DALL-E:** A conversation with AI models. One message shows two images of a beef burger, with the caption: "Here are two teaser images for a cooking recipe website featuring a "Classic Beef Burger". These images are designed to appeal to a young, modern urban audience."
- Hibernate ORM:** A screenshot of a project management interface for the Hibernate ORM software.
- sessionize:** A user profile page for Nils Hartmann on sessionize.com, showing basic details like name and a public profile link.
- Spring Framework:** Two screenshots of the Spring framework's "Owners" database table. The first table shows 10 entries with columns: Name, Address, City, Telephone, and Pets. The second table shows 12 entries with similar columns.
- Google Sheets:** A screenshot of a Google Sheets spreadsheet with multiple tabs and data entries.
- Other Applications:** Various other browser tabs and windows are visible, including what appears to be a GitHub repository and some messaging or collaboration tools.

WEBANWENDUNGEN

Die Webanwendung gibt es nicht
...aber: (fast) alle brauchen JavaScript

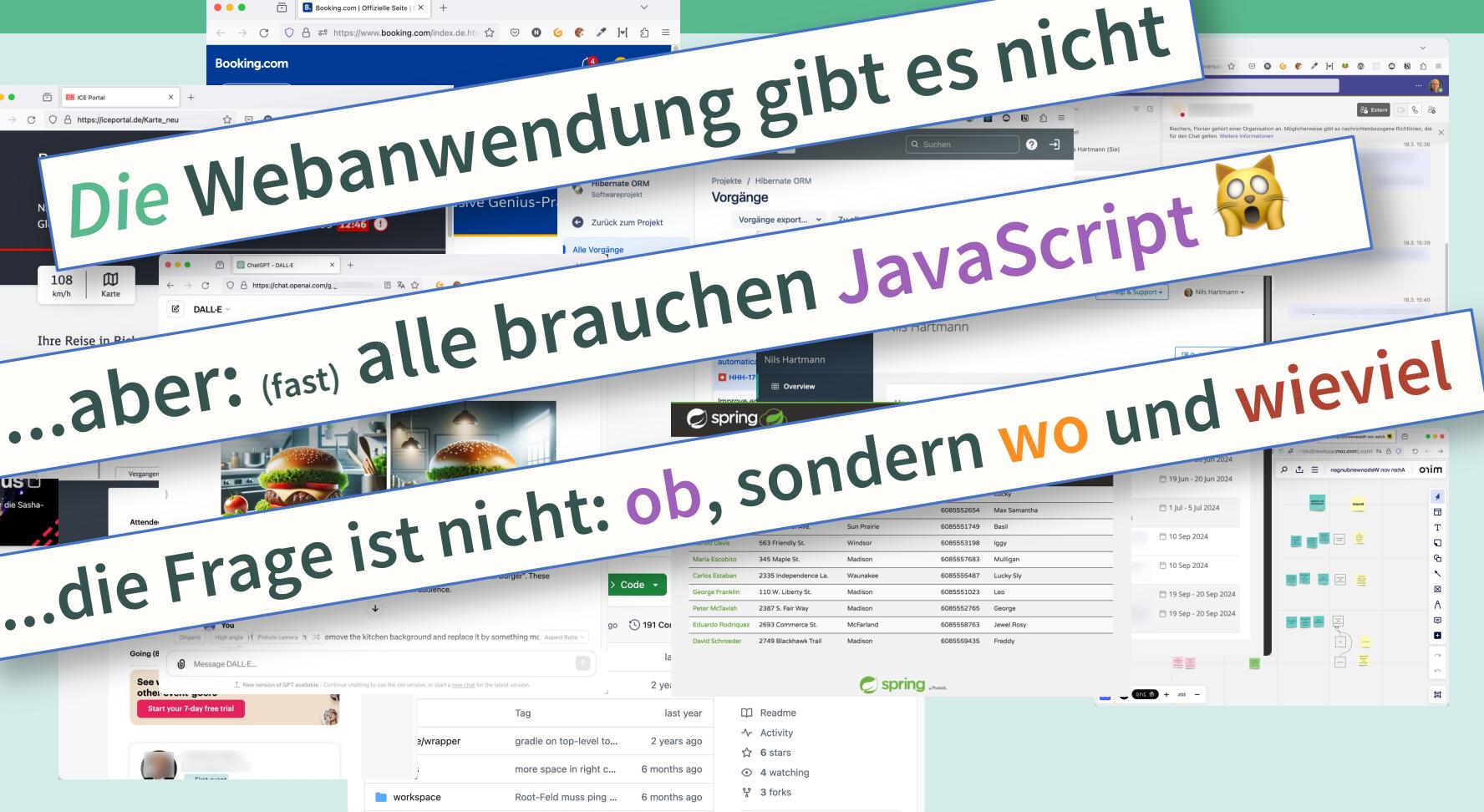
The screenshot shows a Mac OS X desktop environment with a Dock at the bottom. On the Dock, there are icons for 'ICE Portal', 'Booking.com', 'ChatGPT - DALL-E', and 'hvg plus'. Above the Dock, four browser tabs are open:

- ICE Portal**: A dark-themed portal with a speedometer showing 108 km/h.
- Booking.com**: The official website.
- ChatGPT - DALL-E**: A tab showing two images of a classic beef burger.
- hvg plus**: A tab for buying tickets for the Sasha Show.

The screenshot displays several web application interfaces:

- Hibernate ORM**: A software project management tool showing tasks like "Serve Genius-Pr..." and "Zurück zum Projekt".
- ChatGPT - DALL-E**: A tab showing two images of a classic beef burger.
- Spring Boot**: A Java framework interface showing a list of owners with columns for Name, Address, City, Telephone, and Pets.
- ChatGPT-openai.com**: A tab showing a message from DALL-E asking to remove the kitchen background from a burger image.
- ICE Portal**: A tab showing a speed limit of 108 km/h.
- Booking.com**: The official website.
- hvg plus**: A tab for buying tickets for the Sasha Show.

WEBANWENDUNGEN



WEBANWENDUNGEN

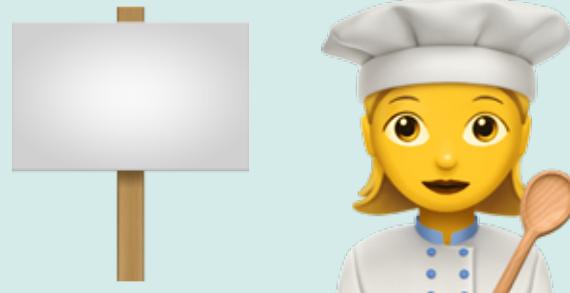
Die Webanwendung gibt es nicht

...aber: (fast) alle brauchen JavaScript

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

(...und wer es schreibt oder erzeugt 😈)

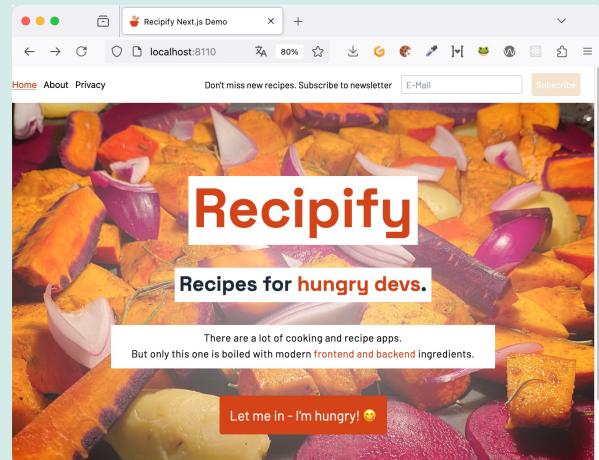
Demo



BEISPIELE

Eine Webanwendung...

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

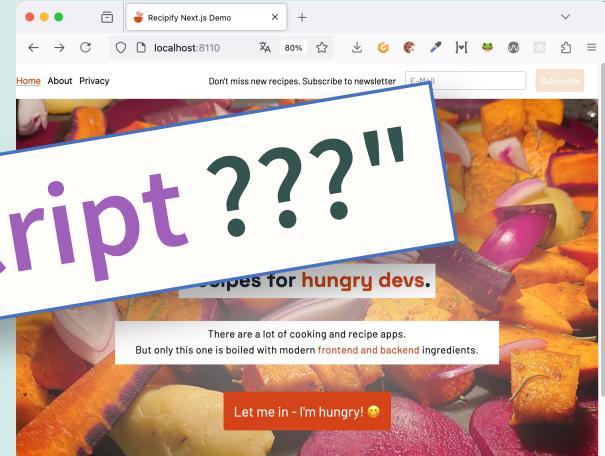


<http://localhost:8110>

BEISPIELE

Eine Webanwendung...

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

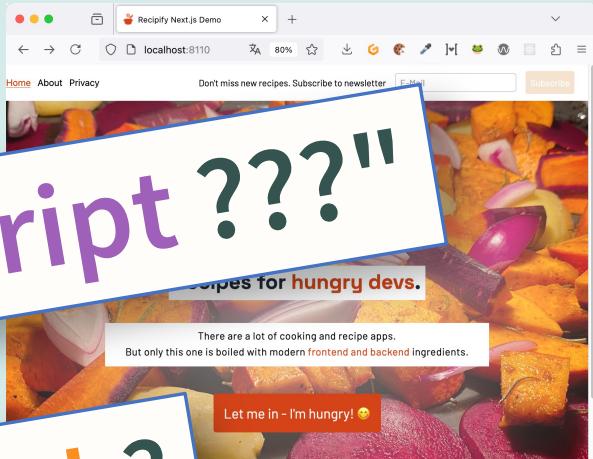


<http://localhost:8110>

BEISPIELE

Eine Webanwendung...

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



diese auch? wirklich?

BEISPIELE

Eine Webanwendung...

1. 🕵️ Landing-Page



- 2.

- 3.

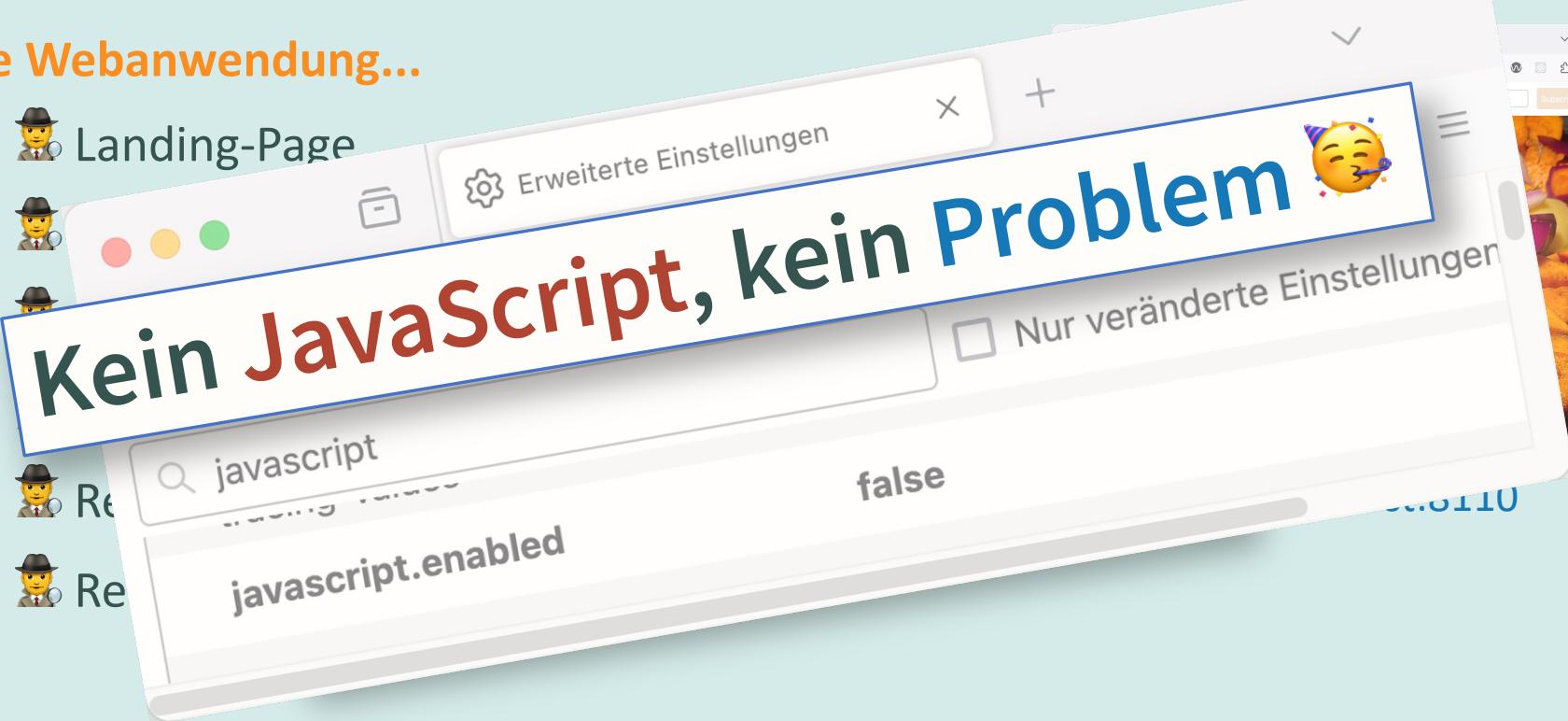
- 4.

- 5.

- 6.

BEISPIELE

Eine Webanwendung...

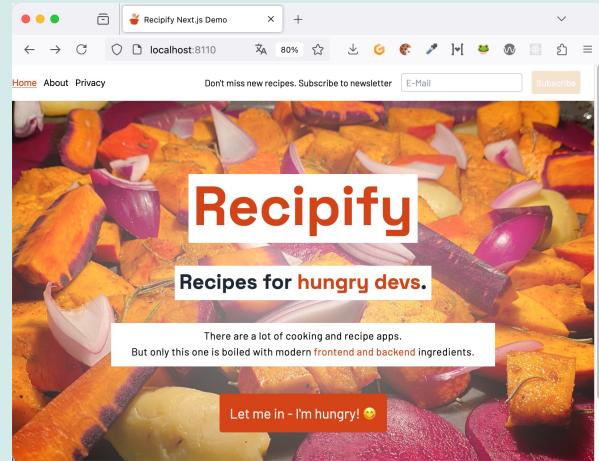
1. 🕵️ Landing-Page
 2. 🕵️
 3. 🕵️ **Kein JavaScript, kein Problem** 😊
 - 4.
 5. 🕵️ Re
 6. 🕵️ Re
- 

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

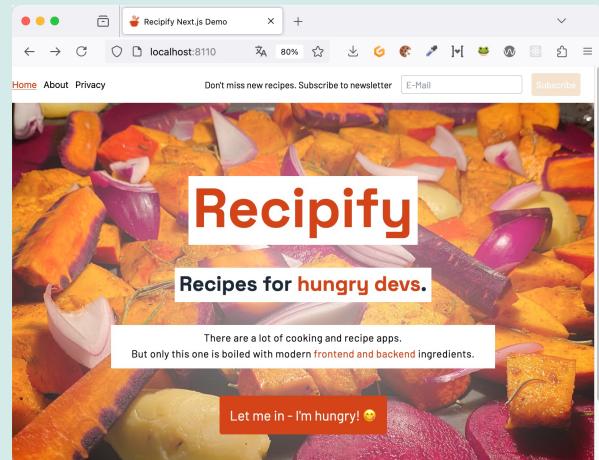


<http://localhost:8110>

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 😢

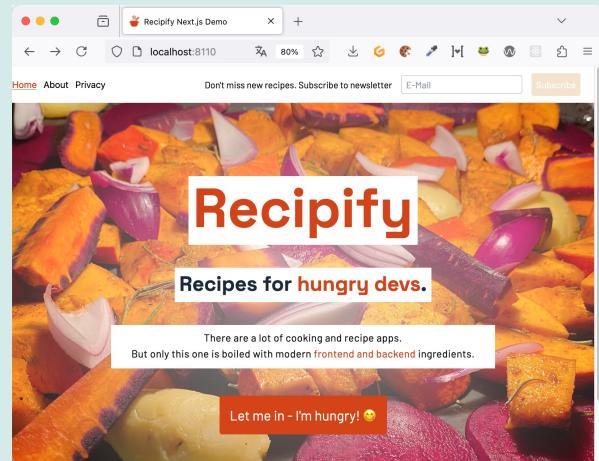


<http://localhost:8110>

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

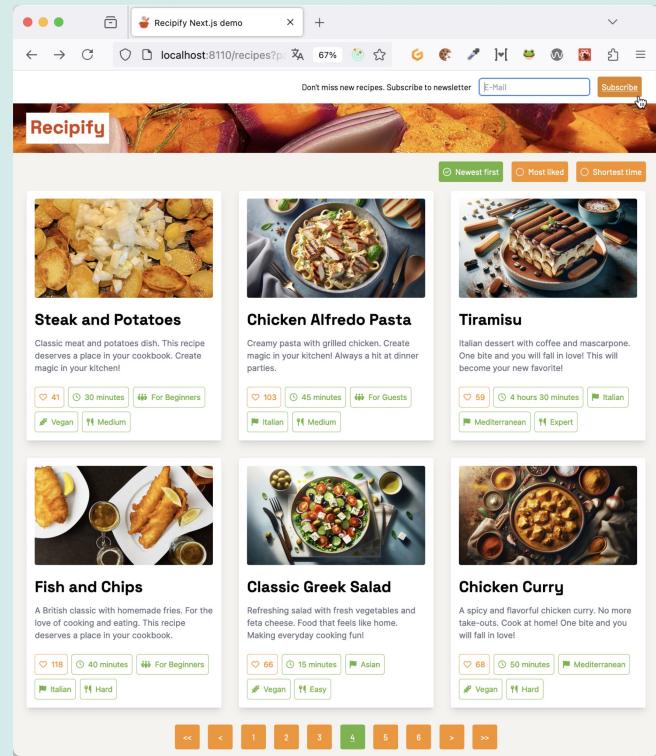
Webanwendungen

HTMX

"Klassisch"

"Klassisch": Serverseitiges Rendern

```
@Controller  
public class RecipeController {  
  
    private final RecipeRepository repository;  
  
    RecipeController(RecipeRepository repository) {  
        this.repository = repository;  
    }  
  
    @GetMapping("/")  
    String getRecipesList(Pageable pageable, Model model) {  
  
        var recipes = this.repository.findAll(pageable);  
  
        // Model mit allen Informationen erzeugen, die  
        // das Template zum rendern braucht  
        model.addAttribute("recipes", recipes);  
        model.addAttribute("pageable", pageable);  
  
        // Name des Templates, das den HTML-Code für die  
        // Seite erzeugt  
        return "index";  
    }  
}
```



Ansatz 1: "Klassisch": Serverseitiges Rendern

- 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

Technologie Stack zur Laufzeit

- Backend-Framework zum Rendern von HTML
- JavaScript-Schnipsel im Browser
- JavaScript ist eventuell optional (progressive enhancement)

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

Technologie Stack zur Entwicklungszeit

- Backend-Sprache (für Frontend-Anteil)
- Template-Sprache
- JavaScript
- Evtl. JavaScript-Bibliothek (Alpine, jQuery, ...)

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)

Technologie Stack zur Laufzeit

- JavaScript, JavaScript, JavaScript (im Browser)
- Backend für API

Ansatz 2: Single-Page-Anwendung (SPA)

Technologie Stack zur Laufzeit

- JavaScript, JavaScript, JavaScript (im Browser)
- Backend für API
- Unterschied zu Ansatz #1:
 - JavaScript wird für alles verwendet, auch statische Darstellungen
 - Kann sich negativ auf die initiale Performance auswirken

Ansatz 2: Single-Page-Anwendung (SPA)

Technologie Stack zur Entwicklungszeit

- Backend-Sprache (für APIs)
- JavaScript (für Frontend-Anteil)
- Template-Sprache bzw. SPA-Framework (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

Ansatz 3: Fullstack-Anwendung (mit JavaScript)

Technologie Stack zur Entwicklungszeit

- JavaScript (für Backend und Frontend)
- Fullstack-Framework (Next, Nuxt, SvelteKit, Astro, ...)

Ansatz 3: Fullstack-Anwendung (mit JavaScript)

Technologie Stack zur Entwicklungszeit

- Unterschied zu Ansatz 1:
 - Die Anwendung wird mit einer Sprache und einem Framework gebaut
 - Keine Aufteilung in Backend-Sprache und JavaScript

Ansatz 3: Fullstack-Anwendung (mit JavaScript)

Technologie Stack zur Laufzeit

- JavaScript auf dem Server (z.B. Node.js)
- JavaScript im Browser
 - Im Browser evt. optional (progressive enhancement)

Ansatz 3: Fullstack-Anwendung (*mit JavaScript*)

Technologie Stack zur Laufzeit

- Vergleichbar mit Ansatz #1:
 - JavaScript im Browser nur dort, wo notwendig 😊

...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
- Der Server muss HTML-Schnipsel liefern ("Hypermedia")
- Protokoll ist sehr einfach, kann jeder Server

```
<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 Ergebnis:
Reponse: localhost:8080/hello

HelloWorldController

- HTMX-Attribute:
Request: hx-trigger und hx-get

- Der Server muss HTML-Schnipsel liefern
("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>
```

HTMX - Grundlagen

- HTMX macht nur Server Requests
- 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>
```

Ansätze im Detail



[HTTP://LOCALHOST:8090](http://localhost:8090)

Beispiel 1: Daten nachladen

- Daten erst auf dem Server laden
 - (in HTMX: UI)
- Lade-Animation
- Paginierung

Feedback

Bryson Turner 20. September 2023

Baked garlic butter shrimp is a quick and flavorful dish. Perfect for seafood lovers!

Ivy Turner 10. Juli 2023

Family favorite! Everyone asked for seconds. This recipe is now in our regular rotation.



Beispiel 2: Einfaches Formular

- Wie werden die Daten gesendet?
- Was kommt als Antwort?
- Wie disablen wir das Formular?
- Wie setzen wir eine (Fehler-)Meldung zurück

Subscribe to newsletter

E-Mail

Subscribe

Beispiel 3: Suchfeld

- Wie starten wir einen Server-Request automatisch?
- Wie erweitern wir eine bestehende Liste?
- Wie können wir die Suche zurücksetzen?
- Wie können wir eine Information ein- und ausblenden?
- Mehrere Stellen in einer Seite aktualisieren

Clear

Recipes for sal

LUNCH

Baked Salmon



▼

BREAKFAST

Caesar Salad



▼

Find more...

Beispiel 4: 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

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

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)

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

HTMX

- HTMX hat eine eigene (dogmatische) Sichtweise auf die Welt
- Ob die zu einer Anwendung passt, hängt von der Anwendung ab
- Design und API erinnern an jQuery
- Wollen wir Anwendungen von heute mit Technik von vor zwanzig Jahren bauen?

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

HTMX

- Über das Backend und den Entwicklungsprozess sagt HTMX (fast) nichts
 - Wie sieht es mit Testen, Debuggen, Typsicherheit aus?
 - Habt ihr ein Backend, das HTML-Schnipsel liefern kann?
 - Habt ihr eine passende Template-Sprache?
- **Für Spring Boot gibt es mittlerweile built-in Support**

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

HTMX

- Für Kleinigkeiten muss ein Server Request für neue UI gemacht werden
 - Ist das ein Entwicklungs- und Architekturmuster das dir gefällt?
 - Wie sieht es mit Latenz und Datenmenge aus?
 - Wie viele Stellen hast du, an denen das nicht funktioniert?
 - Zum Beispiel Eingabefeld leermachen, Zeichenzähler...
- Wie gehst du mit Stellen um, an denen du doch JavaScript braucht?

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

Single-Page-Anwendung

- Adressieren das Problem "echter" Anwendungen im Browser
- Verwenden die Sprache der Ziel-Platform

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

Single-Page-Anwendung

- Ausgereifte Technologie
 - ...mit allen, was wir aus dem Backend kennen (automatische Tests, Typsicherheit, Debugger, CI-Pipelines)
- Stand der Technik nicht mit 2010 vergleichbar
 - Vieles hat sich konsolidiert und stabilisiert
 - Trotzdem noch viel Wandel und Innovation
- Als Alternative erscheinen mir Fullstack-Frameworks konsequenter als HTMX

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>

