

The background of the slide features a close-up photograph of a yellow flower, likely a sunflower, with several prominent brown stamens extending from its center. The flower is set against a warm, blurred background of yellow and orange tones.

NILS HARTMANN

<https://nilshartmann.net>

Single- Page- Anwendung oder

HTMX ?

Wie doof kann Nils sein?

```
❯ rm -rf .git
```

```
❯ rm jughh24_spa_oder_htmx.pdf jughh24_spa_oder_htmx.pptx
```

4. April, 17:28

NILS HARTMANN

nils@nilshartmann.net

Freiberuflicher Entwickler, Architekt, Trainer aus Hamburg
Java, Spring, GraphQL, React, TypeScript



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

<https://reactbuch.de>

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

Frontend-Entwicklung: Nehmen wir das ernst?

Frontend-Entwicklung: Nehmen wir das ernst?



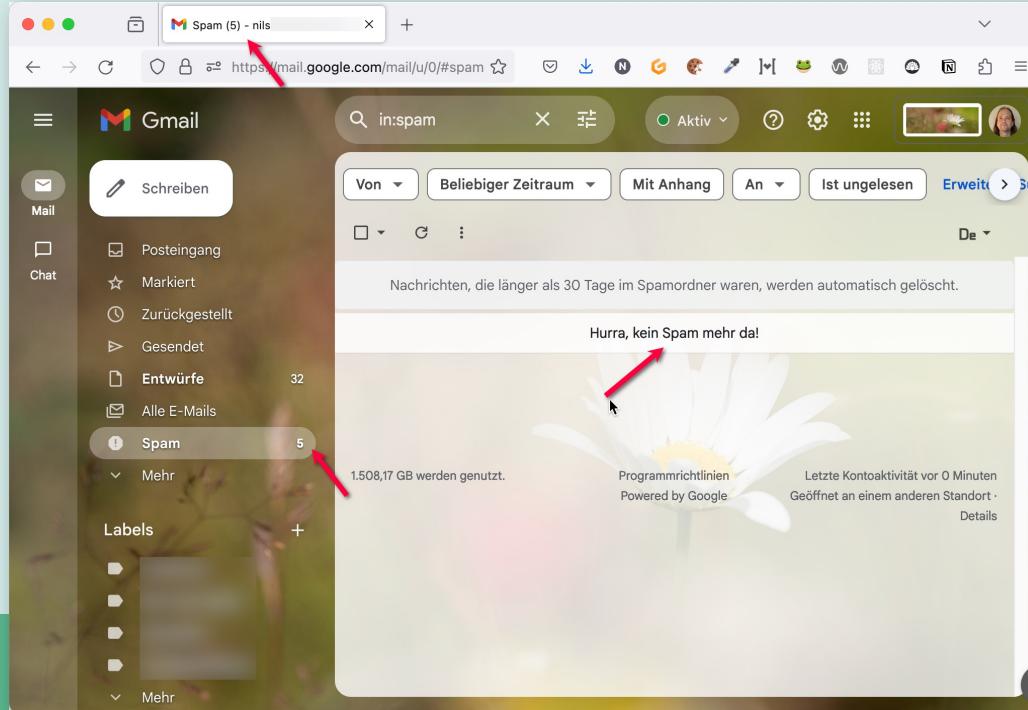
Das ist doch kein Problem,
bisschen HTML+CSS = fertig

Frontend-Entwicklung: Nehmen wir das ernst?



Was soll der ganze
SPA-Kram?

Frontend-Entwicklung: Was soll schon schief gehen?



Single-Page- Anwendungen

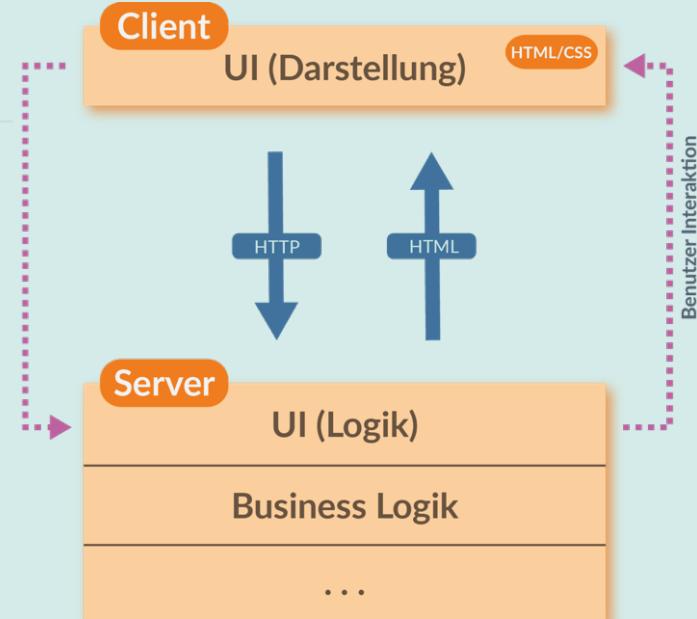
"Klassische" Web-Anwendung (Web 1.0)

Browser ist nur für die Darstellung zuständig

- UI wird auf dem Server erzeugt
(z.B. JSP/JSF, ASP, PHP)

Konsequenz:

- Roundtrips für jede Kleinigkeit
- Dynamik / Interaktion fast unmöglich



INTERAKTIONEN

(Mikro-)Interaktionen: Fast nie rein Seiten-basiert

The screenshot shows a browser-based code editor interface, likely from a GitHub repository or a local development environment. The main window displays the file `backend-queries.ts` from the `nextjs-intro/nextjs-workspace/app/shared/api` directory. The code is written in TypeScript and contains logic for fetching blog teaser lists from an external backend service using the `fetch` API.

```
// ...
type GetBlogTeaserListParams = {
  orderBy?: OrderBy;
  filter?: string;
};

export async function getBlogTeaserList({
  orderBy = "desc",
  filter,
}: GetBlogTeaserListParams) {
  console.log("Starting fetch to external backend service");

  const searchParams = new URLSearchParams();
  searchParams.set("order_by", orderBy);
  if (filter) {
    searchParams.set("filter", filter);
  }

  if (getBlogTeaserListSlowdown) {
    searchParams.set("slowDown", getBlogTeaserListSlowdown);
  }

  const url = `http://localhost:7002/posts?teaser=${searchParams.to
  console.log("Reading Teasers from", url);

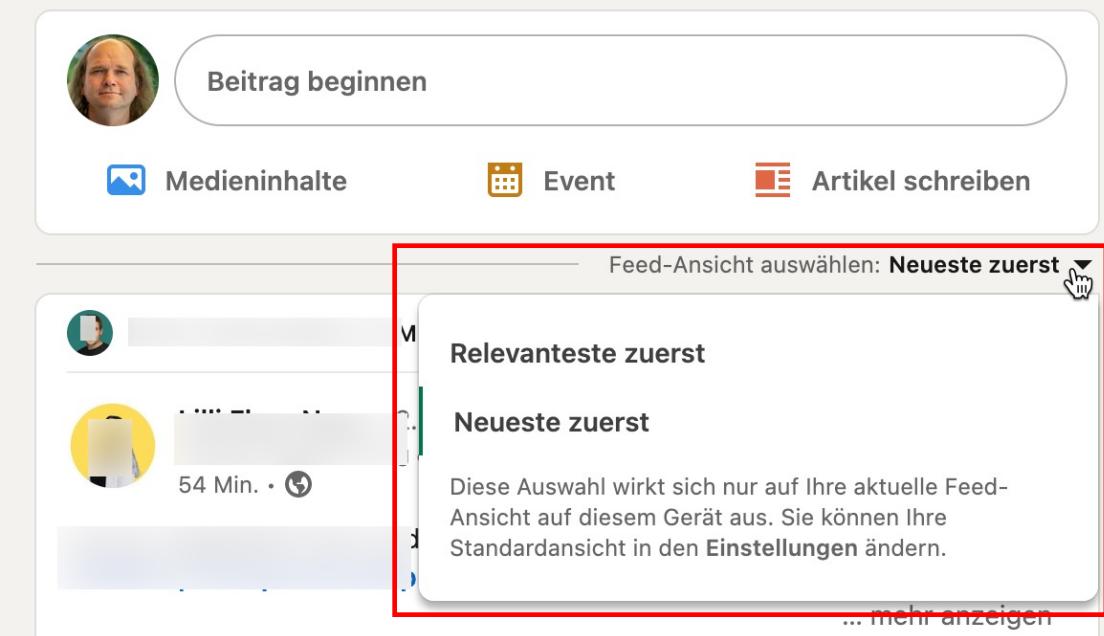
  const r = await fetch(url, {
    next: {
      tags: ["teaser"],
    },
  });

  console.log(
    "Fetch request to external backend service returned timestamp",
    r.headers.get("x-backend-started-at"),
  );
}
```

The right side of the interface features a sidebar with navigation and search functionality. It includes sections for "All Symbols", "References", and "Search". A search bar at the bottom right is set to "Search for this symbol".

INTERAKTIONEN

(Mikro-)Interaktionen: kommen schneller als man denkt?



The screenshot shows a user interface for managing posts or interactions. At the top, there's a profile picture of a person with blonde hair and a button labeled "Beitrag beginnen". Below this are three buttons: "Medieninhalte" (with a camera icon), "Event" (with a calendar icon), and "Artikel schreiben" (with a document icon). A red box highlights a dropdown menu titled "Feed-Ansicht auswählen: Neueste zuerst". This menu contains two options: "Relevanteste zuerst" and "Neueste zuerst". A red arrow points from the right towards the "Neueste zuerst" option. Below the menu, a note states: "Diese Auswahl wirkt sich nur auf Ihre aktuelle Feed-Ansicht auf diesem Gerät aus. Sie können Ihre Standardansicht in den Einstellungen ändern." At the bottom of the dropdown, there's a link "... mehr anzeigen". To the left of the dropdown, a portion of the main feed is visible, showing a post by a user named "M" and another post by a user with a yellow profile picture, dated "54 Min. ·".

INTERAKTIONEN

(Mikro-)Interaktionen: kommen schneller als man denkt?

The image shows a user interface with the following elements:

- A radio button group for gender selection, with "Frau" and "Herr" options.
- A dropdown menu labeled "Titel (optional)" with the placeholder "Bitte wählen".
- An input field for "Vorname" containing "Nils", which is validated with a green checkmark and a red arrow pointing to it.
- An input field for "Nachname" which is currently empty and highlighted with a red border, indicating it is a required field. A red arrow points to the error message below it.
- A red error message at the bottom of the "Nachname" field: "Bitte füllen Sie alle Pflichtfelder aus."
- An input field for "Ggf. Geburtsname (optional)" with a green checkmark and a red arrow pointing to it.

(Mikro-)Interaktionen: in der Regel 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

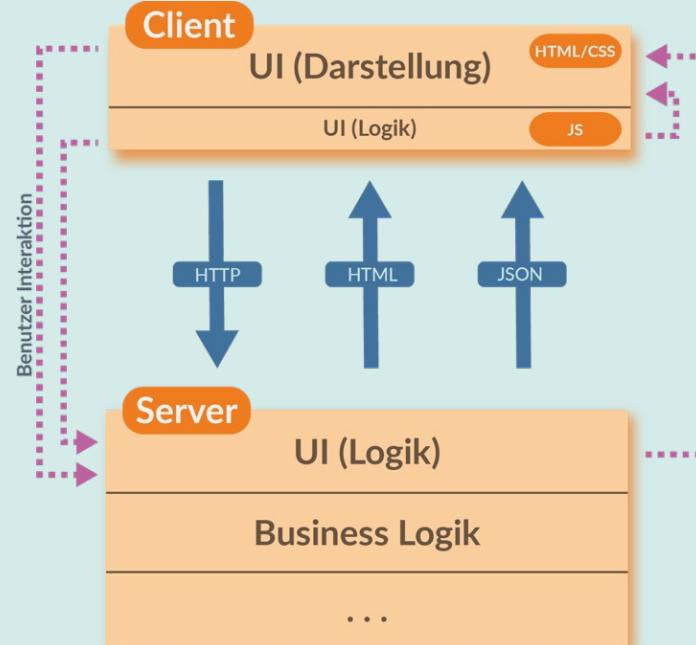
"Klassische" Web-Anwendung (Web 2.0)

Mit JavaScript-Schnipseln (früher jQuery):

- Interaktion
- Validierung

Konsequenz:

- Unsaubere Architektur (was ist wo?)
- Wartungshölle

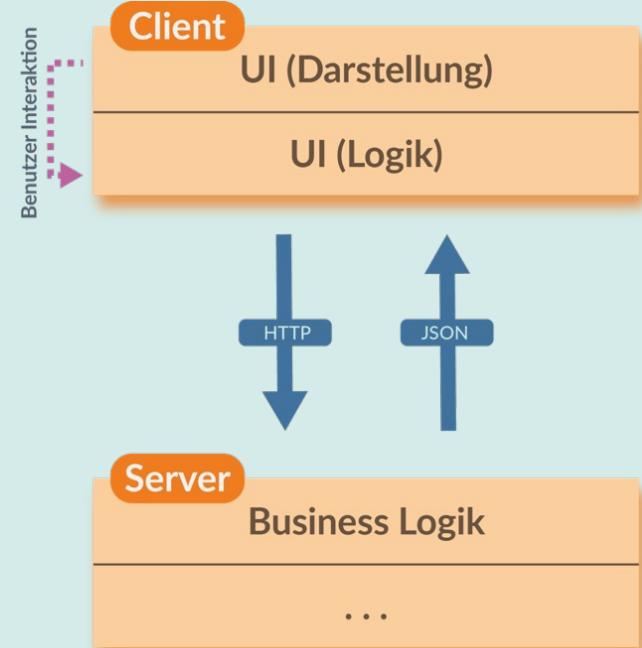


Single-Page-Anwendungen (ca. seit 2010)

- Client ist vollständig in JavaScript geschrieben
- Client läuft vollständig im Browser

Konsequenz:

- Klare Trennung zwischen Backend und Frontend
- Server "nur" für Geschäftslogik, Datenhaltung...
- Frontend "nur" für Darstellung und Interaktion
- Server muss APIs für Daten bereit stellen



SINGLE-PAGE-ANWENDUNGEN

Probleme von Single-Page-Anwendungen

Probleme von Single-Page-Anwendungen

- Was fällt euch ein? 🤔

Probleme von Single-Page-Anwendungen

- Was fällt euch ein? 🤔
- Gängige Vorurteile? 😡

Probleme von Single-Page-Anwendungen

- JavaScript total schlimm! ("undefined is not a function")
- SEO geht nicht / schlecht!
- Langsam!
- Viel JavaScript muss geladen werden!
- JS-Bibliotheken ändern sich ständig!
- Tooling funktioniert nicht bzw. ist komplex!

HTMX

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/>

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-Schnippel 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>
```

Demo: Hello World

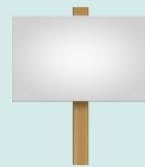
Single-Page-Anwendung oder HTMX?

- SPA ist ein **Pattern** oder eine **Architektur**, aber kein konkretes Framework
- Wir müssten bei einem Vergleich eigentlich ein SPA-Framework wählen
 - d.h. es müsste heißen "Angular vs. HTMX" oder "React vs. HTMX"
- Ich versuche eine Mischung aus "Konzept" / "Idee" und Umsetzung (React)

Single-Page-Anwendung oder HTMX?

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

Demo



Beispiel-Anwendung

Source-Code (GitHub):

<https://react.schule/jughh-htmx>

Nach dem Meetup checke ihr hier den geschriebenen Code ein

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

- Will ich eine "Hypertext"-Anwendung bauen
oder
- eine Anwendung, die die Anforderungen meiner Benutzer 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

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

FAZIT: SINGLE-PAGE-ANWENDUNG ODER HTMX?

HTMX

- Man braucht trotzdem JavaScript (!)
 - ...dann hat man plötzlich einen Technologie- und Sprachen-Zoo



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-Schnippsel liefern kann?
 - Habt ihr eine passende Template-Sprache?
 - Versteht man, was im Backend passiert?
- **Für Spring Boot gibt es eigene Bibliothek**

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

Vielen Dank!

Slides: <https://react.schule/jughh-htmx>

Fragen & Kontakt: nils@nilshartmann.net

Twitter: [@nilshartmann](https://twitter.com/nilshartmann)