

NILS HARTMANN

<https://nilshartmann.net>

Moderne Frontends

Single-

Page-

Anwendung

oder

HTMX?

# NILS HARTMANN

nils@nilshartmann.net

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



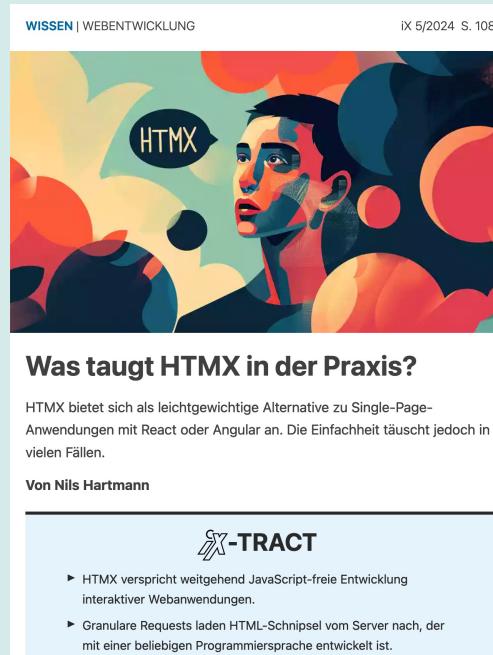
<https://graphql.schule/video-kurs>



<https://reactbuch.de>

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

# 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. A speech bubble above the head contains the text 'HTMX'. Below the image, the title 'Was taugt HTMX in der Praxis?' is displayed. The text explains that HTMX is a lightweight alternative to Single-Page-Anwendungen (SPAs) 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.

# Web- Anwendungen

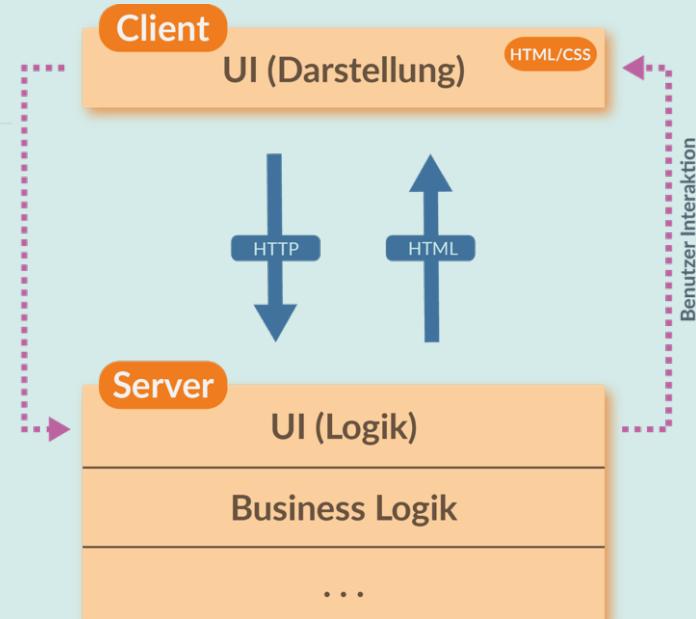
## "Klassische" Web-Anwendung (Web 1.0)

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 developer tools like Chrome DevTools or a specific IDE extension. The left sidebar displays a file tree for a project structure:

- Files
- main
- package.json
- tsconfig.json
- nextjs-workspace
- app
- shared
- api
- api-error.ts
- backend-queries.ts (highlighted)
- server-actions.ts
- types.ts
- blog
- components
- AppLink.tsx
- Button.tsx
- ButtonBar.tsx
- Card.tsx
- Heading.tsx
- Layout.tsx
- LoadingIndicator.module.css
- LoadingIndicator.tsx

The main content area shows the code for `backend-queries.ts`:

```
// ...
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 shows a search results panel for the word `fetch`, indicating 6 references. One reference is highlighted in yellow, corresponding to the line `r = await fetch('http://localhost:7002, ...'`.

# INTERAKTIONEN

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

The screenshot shows a user interface for managing posts. 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 dropdown contains two options: "Relevanteste zuerst" and "Neueste zuerst". It also includes a note: "Diese Auswahl wirkt sich nur auf Ihre aktuelle Feed-Ansicht auf diesem Gerät aus. Sie können Ihre Standardansicht in den Einstellungen ändern." A red arrow points from the left towards the "Neueste zuerst" option. At the bottom right 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.

# 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.
- An input field for "Ggf. Geburtsname (optional)" containing "Geburtsname", which is validated with a green checkmark and a red arrow pointing to it.

A red horizontal arrow points from the validation icons of the first and last fields to a red error message at the bottom of the form.

Bitte füllen Sie alle Pflichtfelder aus.

### (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

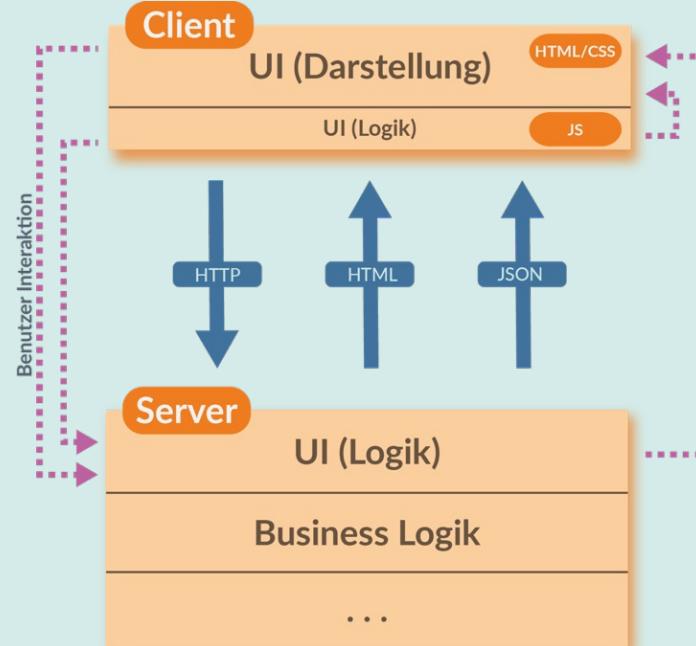
## "Klassische" Web-Anwendung (Web 2.0)

Mit JavaScript-Schnipseln (früher jQuery):

- Interaktion
- Validierung

Konsequenz:

- Unsaubere Architektur (was ist wo?)
- Wartungshölle

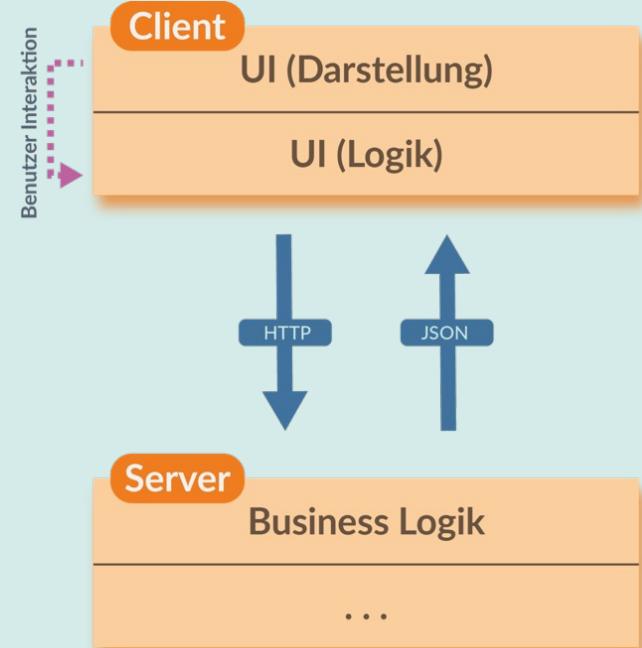


## Single-Page-Anwendungen (ca. seit 2010)

- 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/>

### 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

## 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")

```
<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:  
[localhost:8080/hello](http://localhost:8080/hello)

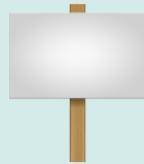
### HelloWorldController

- HTMX-Attribute:  
Replies with  
**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>
```

# Demo



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

## Beispiel-Anwendung

Source-Code (GitHub):

<https://react.schule/jug-oldenburg-htmx>

Nach dem Meetup checke ihr hier den geschriebenen Code ein  
(eigener Branch)

# 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

### # 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

- 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-Schnipsel 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!

Code: <https://react.schule/jug-oldenburg-htmx>

heise Artikel: <https://react.schule/heise-htmx>

Fragen & Kontakt: [nils@nilshartmann.net](mailto:nils@nilshartmann.net)

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