



NILS HARTMANN

<https://nilshartmann.net>

Fullstack Webanwendungen im Praxis Check mit Next.js

MAGDEBURG DEVELOPER DAYS | 14. MAI 2025 | @NILSHARTMANN

NILS HARTMANN

nils@nilshartmann.net

Freiberuflicher Software-Entwickler, –Architekt, Coach, Trainer

Java, Spring, GraphQL, React, TypeScript



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



<https://reactbuch.de>

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

EIN BEISPIEL...

http://localhost:8110

Recipify Next.js demo

localhost:8110/recipes


Don't miss latest news. Subscribe to newsletter

E-Mail

Subscribe

Recipify

+ Newest dishes to impress at you



French Onion Soup

Rich and flavorful onion soup topped with cheesy toast. Making everyday cooking fun! A delight to the taste buds.


5

1 hour

Mexican

Vegan

Medium



Stuffed Bell Peppers

Bell peppers filled with flavorful ground meat. Made with love and fresh ingredients. For the love of cooking and eating.


65

1 hour

Vegan

Vegetarian

Medium



Classic Caesar Salad

Crispy romaine lettuce with creamy Caesar dressing. Bon Appétit! Food that feels like home.

92

10 minutes

Asian

Vegan

Easy

<<

<

1

2

3

4

5

6

>

>>

Dummy content only. Do not cook 🍴.

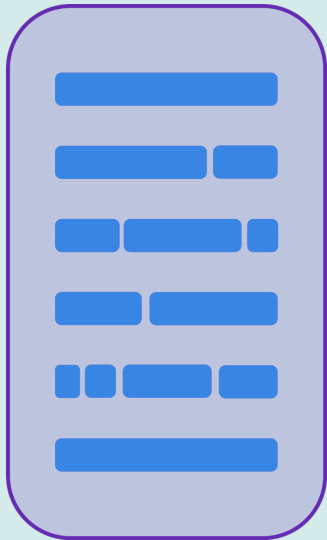
https://nilshartmann.net

Ein Beispiel...

- Brauchen wir dafür JavaScript im Browser? React?

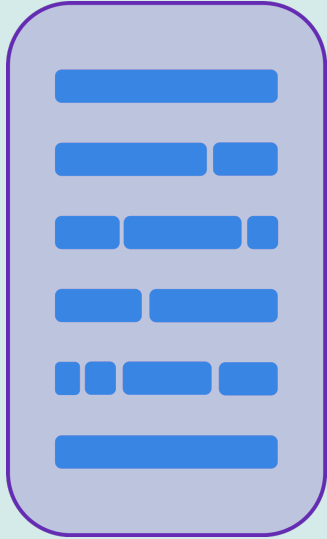
Architekturen für Webanwendungen

Single-Page-Anwendungen



- Darstellung erfolgt vollständig mit JavaScript
- Statisches HTML spielt (fast) keine Rolle
- Die Anwendung kommuniziert mit dem Backend über API
- Ausgetauscht werden Daten, aber keine UI
- Vertreter: Angular, **React**, Svelte, Vue

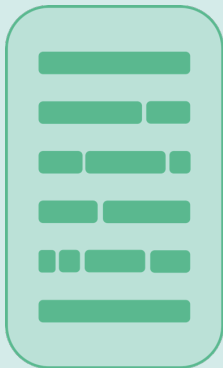
Single-Page-Anwendungen: Eure Meinung!



- Wo machen SPA Sinn? 🤔
- Welche Probleme gibt es damit? 🙄
- Gängige Vorurteile? 🤬

Alternativen zur SPA

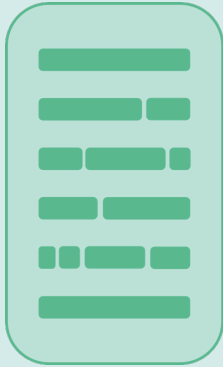
"Klassisch": Serverseitiges Rendern



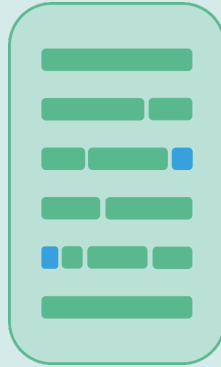
HTML-Seite

- Server liefert fertige HTML-Seite zum Client
- Für jede Interaktion dann Server-Roundtrip (Link, Formular)
- Typische Vertreter: PHP, Java, dotNET, ...

"Klassisch": Serverseitiges Rendern plus JavaScript "Schnipsel"



HTML-Seite



...mit JavaScript-Schnipseln

- Für feingranulare Updates wird JavaScript eingestreut

"Klassisch": Serverseitiges Rendern plus JavaScript "Schnipsel"

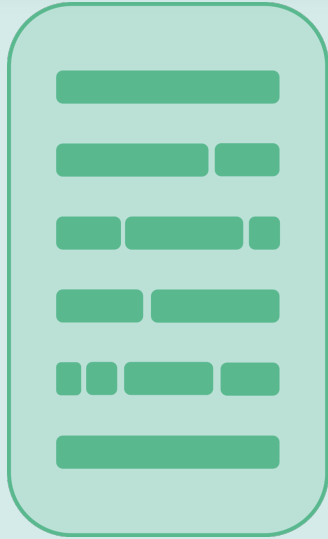


HTML-Seite

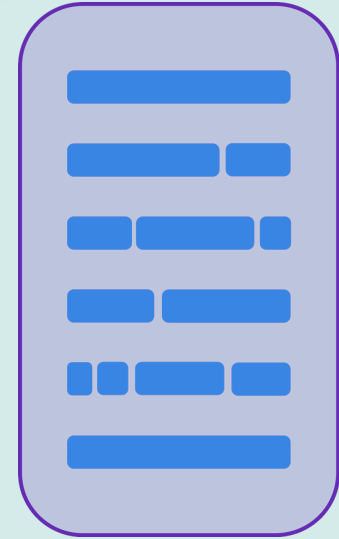
- Eigentlich optimal:
 - wir haben **JavaScript** nur da, wo wir es **wirklich** brauchen, für Interaktivität
 - alles andere kann statisches HTML und CSS sein ❤️

"Fullstack"-Anwendungen (Meta-Frameworks)

"Fullstack"-Anwendungen (Meta-Frameworks)

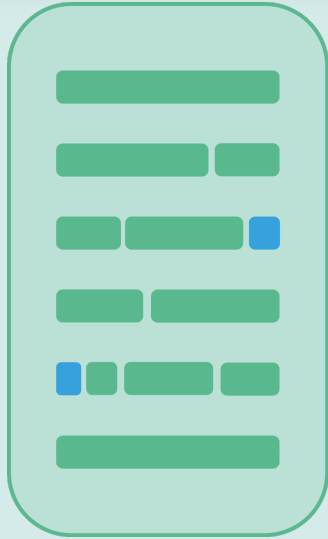


Nur Server (+JS)

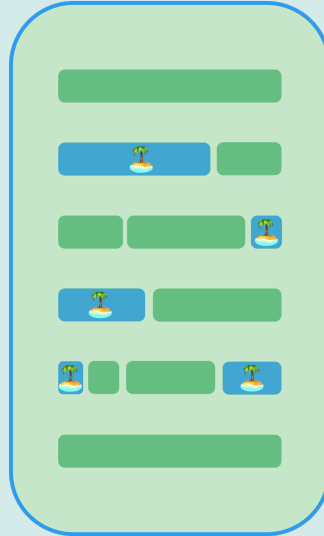


Nur Client (SPA)

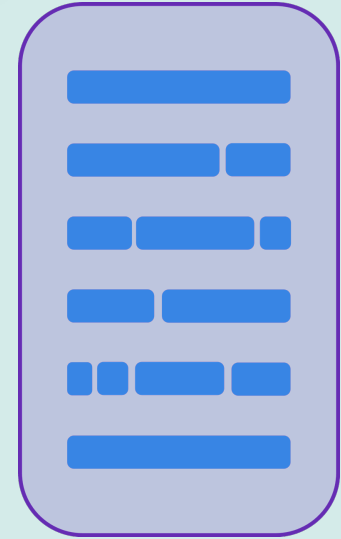
"Fullstack"-Anwendungen (Meta-Frameworks)



Nur Server (+JS)

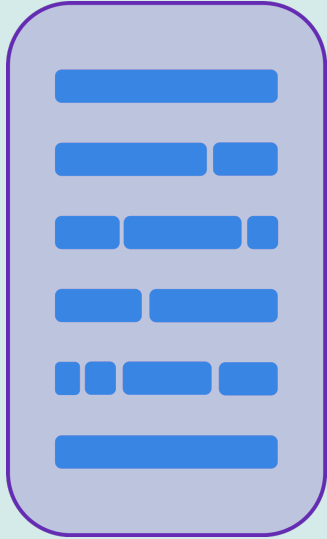


JS Fullstack Anwendung

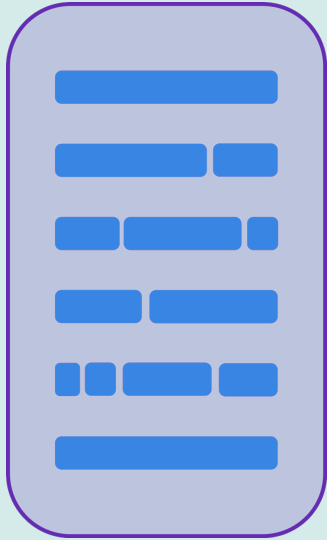


Nur Client (SPA)

Fullstack-Anwendungen (JavaScript)

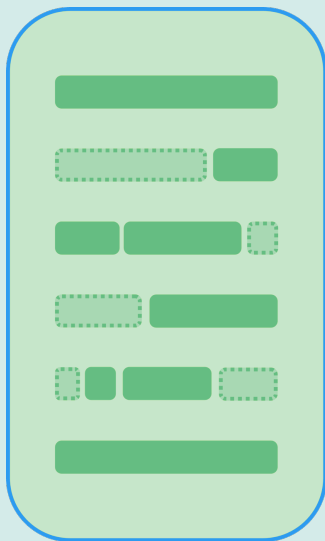


Fullstack-Anwendungen (JavaScript)



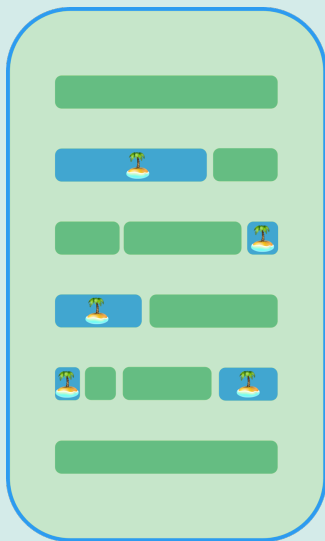
- Ebenfalls vollständig in **JavaScript** geschrieben

Fullstack-Anwendungen (JavaScript)



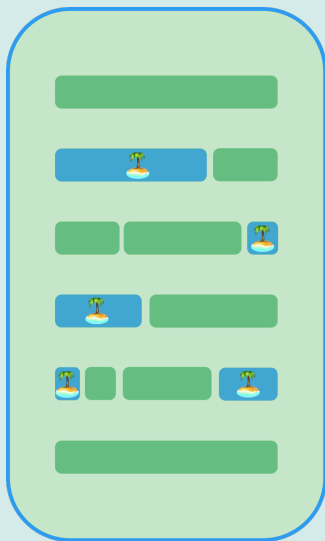
- Grundsätzliche Idee:
 1. **UI-Code** wird serverseitig vorgerendert
 2. **UI-Code** wird zum Browser gesendet und angezeigt

Fullstack-Anwendungen (JavaScript)



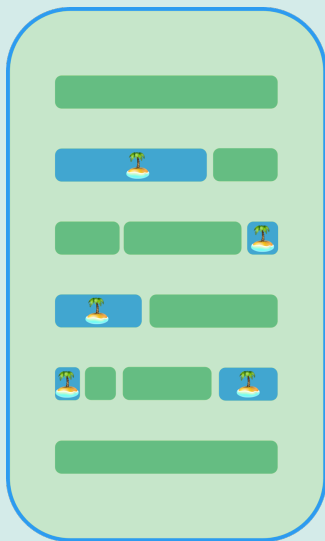
- Grundsätzliche Idee:
 1. **UI-Code** wird serverseitig vorgerendert
 2. **UI-Code** wird zum Browser gesendet und angezeigt
 3. Nur der JavaScript-Code ("Islands") für **Interaktionen** wird zum Browser geschickt

Fullstack-Anwendungen (JavaScript)



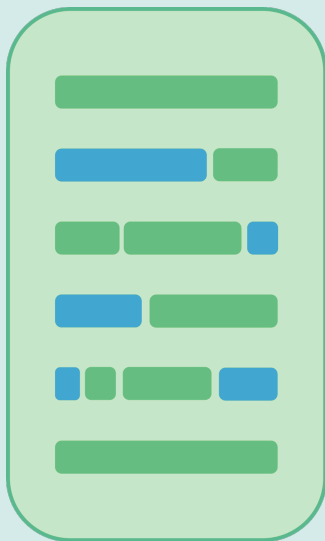
- Bekannte Vertreter:
 1. **Next.js** (React)
 2. SvelteKit (Svelte)
 3. Nuxt.js (Vue)
 4. Astro (eigenes Framework + Support für alle SPAs)
 5. Qwik (eigenes Framework)

Fullstack-Anwendungen (JavaScript)



- Bekannte Vertreter:
 1. **Next.js** (React)
 2. SvelteKit (Svelte)
 3. Nuxt.js (Vue)
 4. Astro (eigenes Framework + Support für alle SPAs)
 5. Qwik (eigenes Framework)
- Funktionalität und Herangehensweise unterschiedlich

Fullstack-Anwendungen (JavaScript)



- Wir sind zurück zur **JavaScript-Schnipsel**-Architektur
 - aber: die Schnipsel werden **automatisch** vom Framework erzeugt
 - die Schnipsel existieren nur zur **Laufzeit**

Live

Demo



[HTTP://LOCALHOST:8100](http://localhost:8100)

<https://nextjs.org>

<https://nextjs.org>

- basiert auf React

<https://nextjs.org>

- basiert auf React
- fügt serverseitiges Rendern und Routing hinzu

<https://nextjs.org>

- basiert auf React
- fügt serverseitiges Rendern und Routing hinzu
- ähnlich wie Spring Boot, das z.B. Webserver + JSP/Thymeleaf integriert

<https://nextjs.org>

- basiert auf React
- fügt serverseitiges Rendern und Routing hinzu
- ähnlich wie Spring Boot, das z.B. Webserver + JSP/Thymeleaf integriert
- "Meta-Framework"

React in zwei Minuten

- Mit React bauen wir Komponenten
- Eine Komponente ist eine JavaScript/TypeScript-Funktion
- Für die UI können wir HTML-artigen Code in JS schreiben

NILS HARTMANN

<https://nilshartmann.net>

Vielen Dank

Code: <https://react.schule/md-dev-days-nextjs>

Fragen und Kontakt:

nils@nilshartmann.net

<https://nilshartmann.net/kontakt>

