

A close-up photograph of tulip petals in shades of orange, yellow, and red, serving as the background for the title text.

Fullstack React: Server-first oder Client-first?

NILS HARTMANN

<https://nilshartmann.net>

NILS HARTMANN

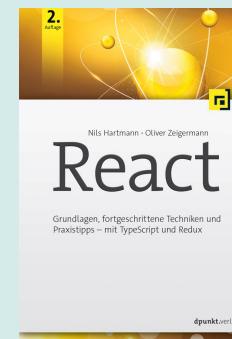
nils@nilshartmann.net

Freiberuflischer Software-Entwickler, –Architekt, Coach, Trainer

Java, React, TypeScript



<https://graphql.schule>



<https://react.schule>

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

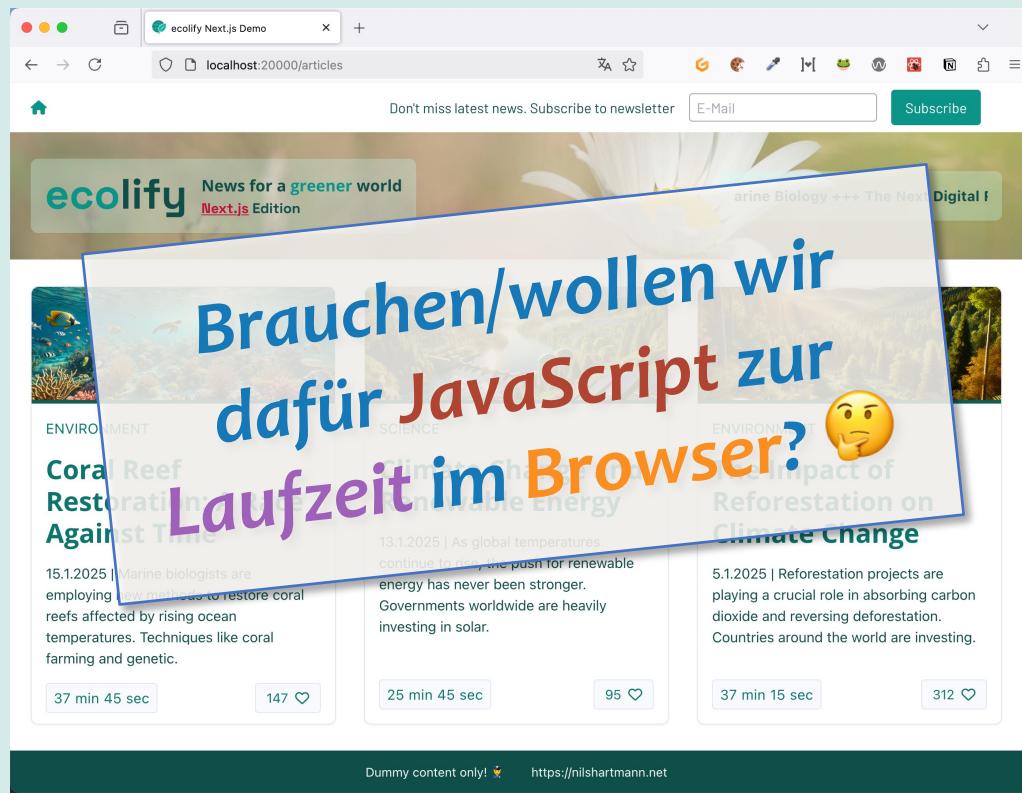
The screenshot shows a web browser window with the title "ecolify Next.js Demo" and the URL "localhost:20000/articles". The page features a header with a house icon, a newsletter subscription form ("Don't miss latest news. Subscribe to newsletter"), and a "Subscribe" button. Below the header is a banner with the text "ecolify News for a greener world Next.js Edition" and "Marine Biology +++ The Next Digital I". The main content area displays three news cards:

- ENVIRONMENT**
Coral Reef Restoration: A Race Against Time
15.1.2025 | Marine biologists are employing new methods to restore coral reefs affected by rising ocean temperatures. Techniques like coral farming and genetic.
37 min 45 sec 147 ❤️
- SCIENCE**
Climate Change and Renewable Energy
13.1.2025 | As global temperatures continue to rise, the push for renewable energy has never been stronger. Governments worldwide are heavily investing in solar.
25 min 45 sec 95 ❤️
- ENVIRONMENT**
The Impact of Reforestation on Climate Change
5.1.2025 | Reforestation projects are playing a crucial role in absorbing carbon dioxide and reversing deforestation. Countries around the world are investing.
37 min 15 sec 312 ❤️

At the bottom of the page, there is a dark footer bar with the text "Dummy content only! 🌎" and a link "https://nilshartmann.net".

<http://localhost:20000>

EINE ANWENDUNG...



<http://localhost:20000>

EINE ANWENDUNG...

Fullstack React:

Was ist das eigentlich?



Fullstack React (meine Definition)

- Eine Anwendung besteht aus **Server**- und **Client**-Seite
- Beide Seiten werden mit **einem** Framework entwickelt
- Betrifft z.B. Rendering, Data Fetching, Routing

Ansatz 1

Client-first

mit

TanStack

TanStack

<https://tanstack.com/>



TANSTACK

**High-quality open-source
software for web developers.**

Headless, type-safe, & powerful utilities for Web Applications, Routing, State Management, Data Visualization, Datagrids/Tables, and more.

Data Fetching

TanStack Query

TECHNOLOGIE STACK

Data Fetching

TanStack Query

Routing

TanStack Router

TECHNOLOGIE STACK

Data Fetching

TanStack Query

Routing

TanStack Router

Fullstack

TanStack Start

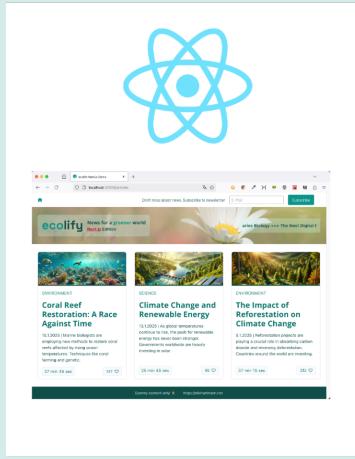
Beta-Version!

Client-first

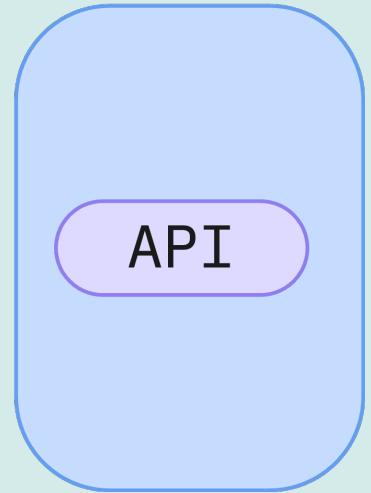
in der Praxis



Browser

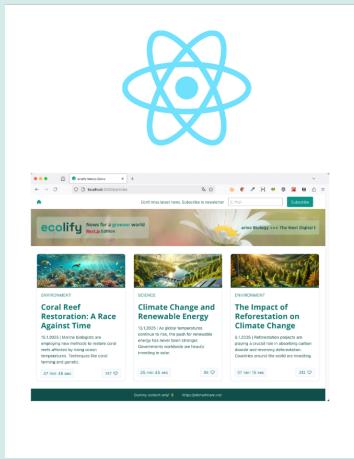


Backend (API)

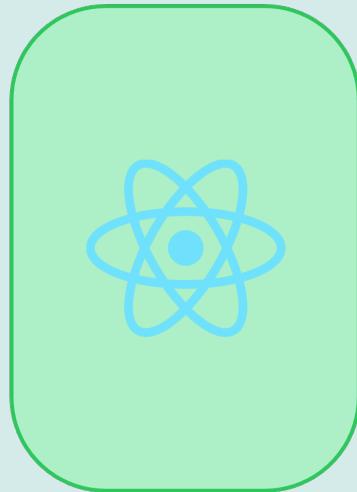


SINGLE-PAGE-ANWENDUNG

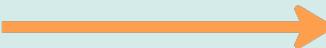
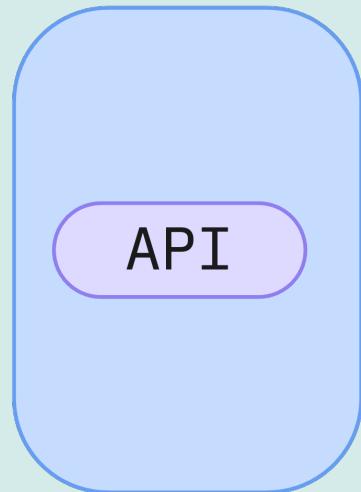
Browser



**React Framework
("Backend for Frontend")**



**Backend
(API)**



???



JSON



FULLSTACK ANWENDUNG

Ansatz 2

Server-first

mit

Next.js

NEXT.JS

<https://nextjs.org>

<https://nextjs.org>

- von Vercel
- basiert auf React

<https://nextjs.org>

- von Vercel
- basiert auf React
- fügt serverseitiges Rendern und Routing hinzu

<https://nextjs.org>

- von Vercel
- basiert auf React
- fügt serverseitiges Rendern und Routing hinzu
- entspricht der React "Architektur Vision"

Server-first

in der Praxis



Fazit

Client-first

oder

Server-first

Fazit

Client-first

Eure Meinung? 

Server-first

Client-first oder Server-first?

- Erst klären: braucht ihr Fullstack überhaupt?
 - Bringt neue Komplexität bei Entwicklung, Test, Debugging, Betrieb, ...

FAZIT

Client-first:

- Wenn ihr euch noch unsicher seid
 - oder eine bestehende SPA habt

Client-first:

- Wenn ihr euch noch unsicher seid
 - oder eine bestehende SPA habt
- Ihr könnt mit SPA anfangen, später auf Fullstack wechseln
 - gilt auch für React Router (ehem. Remix)
 - RSC werden später wohl auch unterstützt werden

FAZIT

Server-first:

- Faktisch zurzeit nur Next.js
 - Weniger Community-Projekt als TanStack (und teilweise React Router)
 - Sehr "meinungsstark"

FAZIT

Server-first:

- Faktisch zurzeit nur Next.js
 - Weniger Community-Projekt als TanStack (und teilweise React Router)
 - Sehr "meinungsstark"
- Kein Migrationspfad von bestehenden Anwendungen
 - "Static export" (SPA-Modus) von Next.js oft nutzlos

FAZIT

Server-first:

- Faktisch zurzeit nur Next.js
 - Weniger Community-Projekt als TanStack (und teilweise React Router)
 - Sehr "meinungsstark"
- Kein Migrationspfad von bestehenden Anwendungen
 - "Static export" (SPA-Modus) von Next.js oft nutzlos
- Konsequenter Ansatz für Anwendungen mit hohem Serveranteil

FAZIT

Client-first oder Server-first?

- Je statischer die Anwendung, desto eher Richtung Server

Vielen Dank

Code & Slides: <https://react.schule/tkdevtalk-fullstack>

Fragen und Kontakt:

nils@nilshartmann.net

<https://nilshartmann.net/kontakt>

