

State of

React

2025

NILS HARTMANN

<https://nilshartmann.net>



NILS HARTMANN

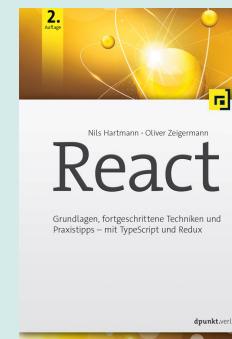
nils@nilshartmann.net

Freiberuflischer Software-Entwickler, –Architekt, Coach, Trainer

Java, React, TypeScript



<https://graphql.schule>



<https://react.schule>

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

Themen

Neues in React

Themen

Neues in React

Fullstack mit TanStack

Themen

Neues in React

Fullstack mit TanStack

Testen (fast) ohne Mocks

Neues in

React

Unsichtbare Komponenten... mit State

<Activity />

Elegante UI Übergänge mit Transitionen

<ViewTransition />

Unsichtbare Komponenten... mit State

<Activity />
Experimental!
<ViewTransition />

Elegante Übergänge mit Transitionen

React

Demo



Routing & Data Fetching

mit

TanStack

TECHNOLOGIE STACK

Data Fetching

TanStack Query

Routing

TanStack Router

TanStack Demo



- **TanStack Query: Vollständige Data-Fetching-Lösung**

- <https://tanstack.com/query/latest>
- ehemals bekannt als "React Query"
- Flexibler Cache für serverseitige Daten
- Hohe Typsicherheit
- Keine Notwendigkeit für useEffect mehr (für Data-Fetching)

- **zod: Laufzeitvalidierung**

- <https://zod.dev/>
- Typen- und Wertebereiche werden mit zod API beschrieben
- TypeScript-Typen werden aus der Beschreibung generiert
- zod validiert Objekte zur Laufzeit und leitet TypeScript-Typen ab
- Sehr gute Integration in die TanStack Bibliotheken
- Alternative: ArkType (<https://arktype.io/>)
 - Noch sehr neu
 - "Spezielle" API

TOOLSTACK ROUTING UND DATA FETCHING

- **ky: Alternative zu fetch und axios**

- <https://github.com/sindresorhus/ky>
- Kleiner Wrapper um die fetch API
- Spezialisierte Methoden für die einzelnen HTTP-Methoden
- Automatische JSON-Konvertierung inklusive zugehöriger Header
- Vereinfachte Verarbeitung des Payloads
- HTTP Status Codes, die Fehler anzeigen führen zu Errors

TOOLSTACK ROUTING UND DATA FETCHING

- **TanStack Router: Typsicheres Routing**

- <https://tanstack.com/router/latest>
- Routen werden aus Dateien und Verzeichnissen generiert
- Alle Routing-Informationen sind typsicher (inklusive Search Parameter)
- Effizientes Arbeiten mit Search Parametern
- Gute Integration mit TanStack Query
- Alternative: React Router (<https://reactrouter.com/>)
 - Überzeugt mich nicht (mehr)

Fullstack React

mit

TanStack

TECHNOLOGIE STACK

Data Fetching

TanStack Query

Routing

TanStack Router

TECHNOLOGIE STACK

Data Fetching

TanStack Query

Routing

TanStack Router

Fullstack

TanStack Start

TECHNOLOGIE STACK

Data Fetching

TanStack Query

Routing

TanStack Router

Fullstack
Experimental!
TanStack Start

Fullstack

Demo



- **TanStack Start: Fullstack-Framework für React**

- <https://tanstack.com/start/latest>
- Bietet SSR und Server Functions
- (Noch) kein Support für React Server Components (RSC)
- Basiert auf dem TanStack Router
- Gute Integration von TanStack Query (optional)
- Alternativen:
 - React Router (vergleichbarer Ansatz)
 - Next.js (eher Server-first Ansatz)

Experimental!

Testen

fast

ohne Mocks

TECHNOLOGIE STACK

Testen

Vitest Browser Mode

Netzwerk Mock

Mock Service Worker

TECHNOLOGIE STACK

Testen

Vitest Bro  *Experimental!* *Mode*

Netzwerk Mock

Mock Service Worker

Testen

Demo



Frontend und Backend ausschalten!

- **Vitest Browser Mode: Komponenten im Browser testen**

- <https://vitest.dev/guide/browser/>
- Kann einzelne (React-)Komponenten im echten Browser testen
- Browser-Zugriff über Playwright
- Web UI zur Ansicht und Ausführung der Tests
- Assertion API ähnlich wie React Testing Library
- Eingebaute Retries bei Assertions für stabile Tests bei asynchrone Use-Cases
- Alternative: Playwright Component Tests
 - <https://playwright.dev/docs/test-components>
 - Habe nicht den Eindruck, dass der Ansatz von Playwright weiterverfolgt wird

Experimental!

TECHNOLOGIE STACK TESTEN

- **Mock Service Worker: Netzwerk Requests "ganz unten" mocken**
 - <https://mswjs.io/>
 - Beschreibt den gewollten Netzwerk-Verkehr durch Abfangen der Requests
 - Framework- und Umgebungsunabhängig
 - Support für REST und GraphQL APIs

Vielen Dank

Code & Slides: <https://react.schule/jax2025-react>

Fragen und Kontakt:

nils@nilshartmann.net

<https://nilshartmann.net/kontakt>

