

React

wo soll das noch hinführen?

Eine Diskussion

Sebastian Springer | Nils Hartmann | und ihr

Go full-stack with a framework

React is a library. It lets you put components together, but it doesn't prescribe how to do routing and data fetching. To build an entire app with React, we recommend a full-stack React framework like [Next.js](#) or [Remix](#).

Can I use React without a framework?

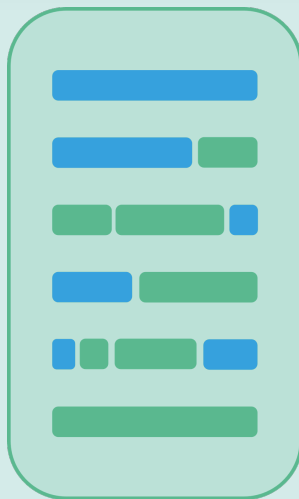
^ Hide Details

You can definitely use React without a framework—that's how you'd [use React for a part of your page](#). **However, if you're building a new app or a site fully with React, we recommend using a framework.**

Here's why.

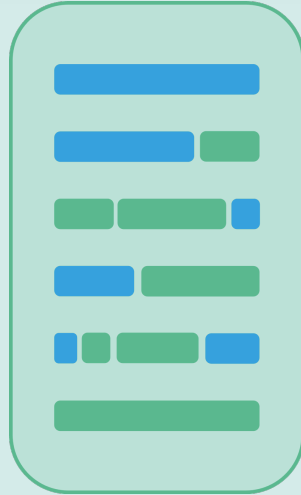
If your app has unusual constraints not served well by these frameworks, or you prefer to solve these problems yourself, you can roll your own custom setup with React. Grab `react` and `react-dom` from npm, set up your custom build process with a bundler like [Vite](#) or [Parcel](#), and add other tools as you need them for routing, static generation or server-side rendering, and more.

Vergangenheit: JavaScript-Schnipsel-Architektur



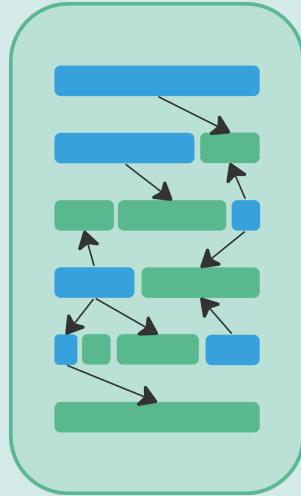
Typische Technik: Backend-Sprache, Template-Sprache, JavaScript, jQuery

Vergangenheit: JavaScript-Schnipsel-Architektur



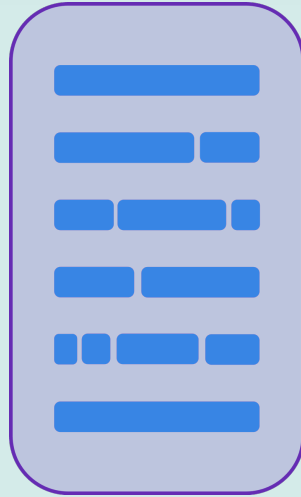
Gut: JavaScript nur für Interaktion

Vergangenheit: Schnipsel-Architektur



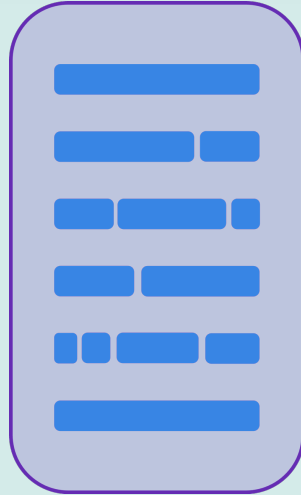
Schlecht: Schnell unübersichtlich, unwartbar, kaputt

Gegenwart: Single-Page-Anwendung



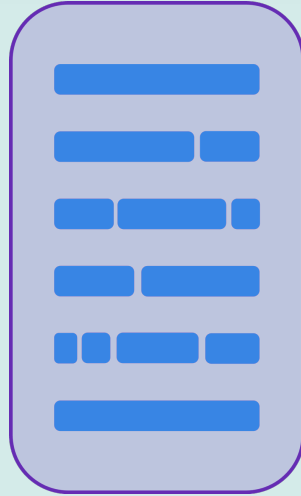
Alles in JavaScript

Gegenwart: Single-Page-Anwendung



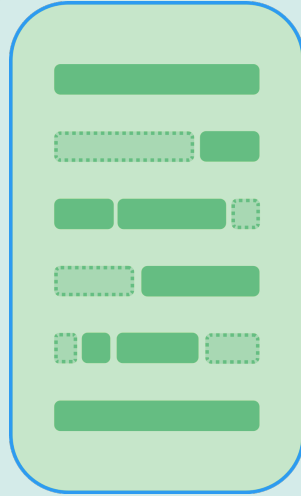
Gut: Klare Verantwortlichkeit (UI ist Client-Sache)

Gegenwart: Single-Page-Anwendung



Schlecht: JavaScript auch für statische UI (Performance!)

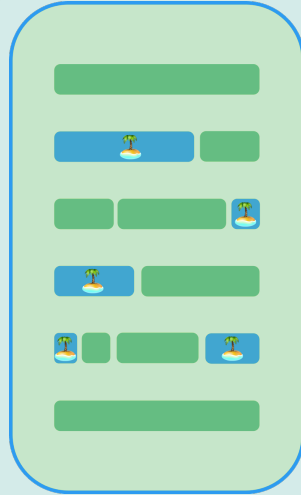
Fullstack-Anwendungen



Ebenfalls ganz in JavaScript

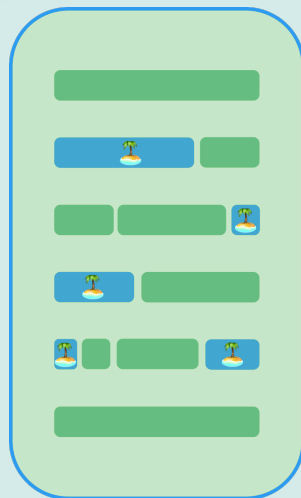
Laufen auf dem Server

Fullstack-Anwendungen



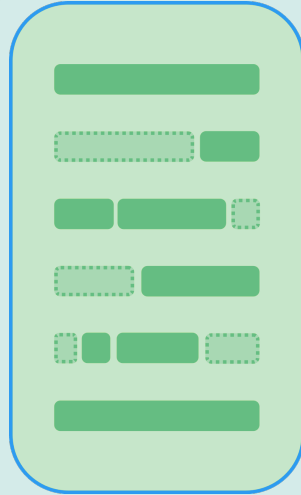
Gut: JavaScript **zur Laufzeit** nur für interaktive Teile im Client ("Islands")

Fullstack-Anwendungen



Schlecht: ?

Fullstack-Anwendungen




Zurück zur Schnipsel-Architektur

Aber: Auftrennung in Server/Client-Teil übernimmt aber das Framework für uns

...aber wo soll das alles hinführen?



- 
- Was bedeutet das für die Zukunft von SPAs?
 - Sind Fullstack-Frameworks überhaupt stabil?
 - oder doch lieber Vanilla React?
 - wie seht ihr das?
 - braucht ihr das?