

NILS HARTMANN
<https://nilshartmann.net>



Nächste Generation

Webframeworks

Alles

Fullstack ?
oder was ?

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)

Kurze Werbung!



React 19 Day

- Online-Konferenz der EnterJS
- Donnerstag, 17. Oktober
- Alle Infos: <https://enterjs.de/react.php>

Kurze Werbung!



React 19 Day

- Online-Konferenz der EnterJS
- Donnerstag, 17. Oktober
- Alle Infos: <https://enterjs.de/react.php>

20% Rabatt: REACT_2024_20_R4VA

Kurze Werbung!

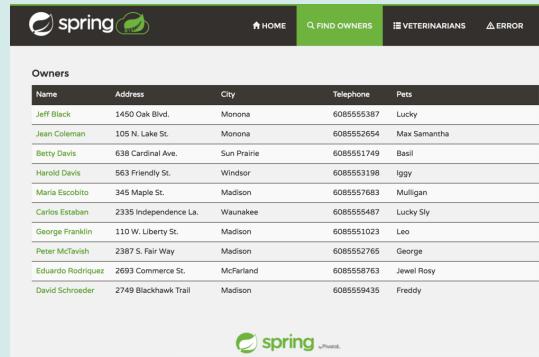


"Kurz und knackig – dein Update auf React 19"

- Online **Workshop** mit Nils und Basti Springer
- Mittwoch, 20. November
- Alle Infos: <https://enterjs.de/react.php>

Webanwendungen...

WEBANWENDUNGEN



The screenshot displays a web application interface for managing pet owners. At the top, there is a navigation bar with the Spring logo, followed by links for HOME, FIND OWNERS (highlighted in green), VETERINARIANS, and ERROR.

The main content area is titled "Owners" and contains a table with columns: Name, Address, City, Telephone, and Pets. The table lists 15 entries, each with a small thumbnail image of a pet next to its name.

Name	Address	City	Telephone	Pets
Jeff Black	1450 Oak Blvd.	Monona	6085555387	Lucky
Jean Coleman	105 N. Lake St.	Monona	6085552654	Mae Samantha
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085511749	Basil
Harold Davis	563 Friendly St.	Windsor	6085553198	Iggy
Maria Escobito	345 Maple St.	Madison	608557683	Mulligan
Carlos Esteban	2335 Independence La.	Wauakee	6085555487	Lucky Sly
George Franklin	110 W. Liberty St.	Madison	6085511023	Leo
Peter McTavish	2387 S. Fair Way	Madison	6085552765	George
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy

At the bottom of the page, there is a footer with the Spring logo and the word "powered".

WEBANWENDUNGEN

The screenshot shows a web browser window titled "ICE Portal" displaying travel information for ICE 109. At the top, it says "Reiseinformationen für den ICE 109". It indicates the next stop in 6 minutes at Gleis 2. The train is currently at Osnabrück Hbf, time 12:35, speed 108 km/h. A red bar highlights the speed. Below this, there's a map icon and a "Schnelles Internet" button. The main content area shows the route from Hamburg-Altona to Basel SBB, with stops like Hamburg-Altona, ICE 109 nach Basel SBB, and Gl. 10. Buttons for "Vergangene Halte zeigen", "Ab", "An", and "Suchen" are visible. The bottom of the page features promotional banners for "hw plus" and "hvv Deutschlandticket", along with social media links for "Mach mit!" and "Schüler*innen".

The screenshot shows a web application for "spring". The header includes a logo, navigation links for "HOME", "FIND OWNERS", "VETERINARIANS", and "ERROR", and a search bar. The main content is a table titled "Owners" listing pet owners with their addresses, city, telephone number, and pets. The table has columns for Name, Address, City, Telephone, and Pets.

Name	Address	City	Telephone	Pets
Jeff Black	1450 Oak Blvd.	Monona	6085555387	Lucky
Jean Coleman	105 N. Lake St.	Monona	6085552654	Max Samantha
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085511749	Basil
Harold Davis	563 Friendly St.	Windsor	6085553198	Iggy
Maria Escobito	345 Maple St.	Madison	608557683	Mulligan
Carlos Esteban	2335 Independence La.	Wauakee	6085555487	Lucky Sly
George Franklin	110 W. Liberty St.	Madison	6085511023	Leo
Peter McTavish	2387 S. Fair Way	Madison	6085552765	George
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy

At the bottom, there's a "spring" logo and the word "invest".

WEBANWENDUNGEN

The image displays four distinct web applications arranged horizontally:

- Booking.com**: A travel booking website showing flight information for flight 108 from Osnabrück Hbf to Basel SBB. It includes details like speed (108 km/h), route (Hamburg-Altona to Basel SBB), and arrival time (12:46). A promotional overlay for "Genius, Nils?" Genius Prämien! is visible.
- ICE Portal**: A real-time train information system for the ICE 109. It shows the next stop at Gleis 2 in 6 minutes, current speed (108 km/h), and a map icon. A "Schnelles Internet" (fast internet) button is also present.
- Eventbrite**: An event management platform showing an attendee list for an event on April 4, 2024. The list includes columns for name, address, city, telephone, and pets. Notable entries include Jeff Black (Monona, Lucky), Jean Coleman (Monona, Max Samantha), Betty Davis (Sun Prairie, Basil), Harold Davis (Windsor, Iggy), Maria Escobito (Madison, Mulligan), Carlos Esteban (Waunakee, Lucky Sly), George Franklin (Madison, Leo), Peter McTavish (Madison, George), Eduardo Rodriguez (McFarland, Jewel Rosy), and David Schroeder (Madison, Freddy).
- Spring**: A pet adoption platform listing owners and their pets. The table includes columns for Name, Address, City, Telephone, and Pets. Owners listed are Jeff Black, Jean Coleman, Betty Davis, Harold Davis, Maria Escobito, Carlos Esteban, George Franklin, Peter McTavish, Eduardo Rodriguez, and David Schroeder, each with their respective pet names and contact information.

WEBANWENDUNGEN

The image shows a Mac desktop with several open windows illustrating various web applications:

- Booking.com**: A travel search results page for a flight from Osnabrück Hbf to Basel SBB.
- ICE Portal**: A train status board for the ICE 109, showing the next stop at Gleis 2 in 6 minutes.
- Hibernate ORM**: A Jira project management interface for the "Hibernate ORM" software project.
- sessionize**: A speaker dashboard for Nils Hartmann, showing his profile and session details.
- g-graphql-training**: A GitHub repository for a GraphQL training project, showing pull requests and code snippets.
- hvhplus**: An event registration page for a Sasha-Show ticketing event.

WEBANWENDUNGEN

The image displays a collage of several web application interfaces:

- Booking.com**: A travel booking website showing flight information for the ICE 109 from Osnabrück Hbf to Basel SBB.
- ICE Portal**: A real-time train status board showing the next stop at Gleis 2 in 6 minutes, speed (108 km/h), and a "Schnelles Internet" connection indicator.
- Hibernate**: A project management tool showing a list of tasks and projects related to "Hibernate ORM".
- Microsoft Teams**: A communication platform displaying a chat history between users.
- Sessionize**: An event management tool showing a speaker dashboard for Nils Hartmann.
- GitHub**: A code repository for a "spring-graphql-training" project, listing pull requests and issues.
- Eventbrite**: A screenshot of an event page for a "Frontend for Backend: HTMX oder Single-Page-Anwendung?" event.

WEBANWENDUNGEN

The image displays a collage of several web application screenshots, illustrating various types of web-based tools and services:

- Booking.com**: A travel booking website showing flight information for the ICE 109.
- ICE Portal**: A mobile-style interface showing travel details, including "Nächster Halt in 6 min" (Next stop in 6 min) and "Gleis 2" (Platform 2).
- ChatGPT - DALLE**: A collaboration between AI chat and image generation, showing a request for a cooking recipe website and two generated images of a beef burger.
- HIBERNATE**: A project management tool showing a list of projects and tasks, with a focus on "Hibernate ORM".
- sessionize**: A platform for organizing events, showing a speaker dashboard for Nils Hartmann.
- Microsoft Teams**: A communication and collaboration tool showing a chat history and file sharing interface.
- Spring**: A developer-oriented platform showing a list of owners with details like name, address, city, phone, and pets.

WEBANWENDUNGEN

Die Webanwendung gibt es nicht

The image is a collage of several screenshots from various web applications, illustrating the concept of "Die Webanwendung gibt es nicht" (The web application does not exist). The applications shown include:

- A travel booking site (Booking.com) showing flight and hotel search results.
- A navigation or travel app (ICE Portal) showing a map and travel details.
- A cooking recipe website featuring a "Classic Beef Burger" recipe.
- A project management tool (Hibernate ORM) showing a list of tasks and versions.
- A session tracking tool (sessionize) showing user activity and profiles.
- A pet adoption platform (spring) listing owners and their pets.
- A calendar application showing multiple events and schedules.
- A messaging or communication interface (ChatGPT) showing a conversation with DALL-E.
- A ticketing or event platform (hvv Plus) showing event details and ticket purchases.

WEBANWENDUNGEN

Die Webanwendung gibt es nicht

...aber: (fast) alle brauchen JavaScript

The screenshot shows a Mac OS X desktop with a windowed browser. The browser has several tabs open:

- ICE Portal
- Booking.com | Offizielle Seite
- ChatGPT - DALL-E
- DALL-E
- hvg plus

The main content area displays two images of a beef burger.

hvg plus content:
Jetzt Tickets für die Sasha-Show gewinnen!
hvg Deutschlandticket
Mach mit!

ChatGPT content:
Here are two teaser images for a cooking recipe website featuring a "Classic Beef Burger". These images are designed to appeal to a young, modern urban audience.

DALL-E content:
Message DALL-E...
Start your 7-day free trial!

The screenshot shows a desktop environment with a Springboard application window open. The application has several panes:

- Code editor pane showing a Java file with annotations like `@SpringBootApp`, `@Entity`, and `@Table`.
- Owners table pane showing a list of owners with columns: Name, Address, City, Telephone, and Pets.
- Github repository pane showing a list of commits and pull requests.
- Bottom right pane showing a diagram or flowchart.

GitHub repository details:

Tag	last year
<code>@wrapper</code>	gradle on top-level to...
	2 years ago
	more space in right c...
	6 months ago
<code>workspace</code>	Root-Feld muss ping ...
	6 months ago

Bottom right pane details:

- Readme
- Activity
- 6 stars
- 4 watching
- 3 forks

WEBANWENDUNGEN

Die Webanwendung gibt es nicht

...aber: (fast) alle brauchen JavaScript

...die Frage ist nicht: ob, sondern wo und wieviel



WEBANWENDUNGEN

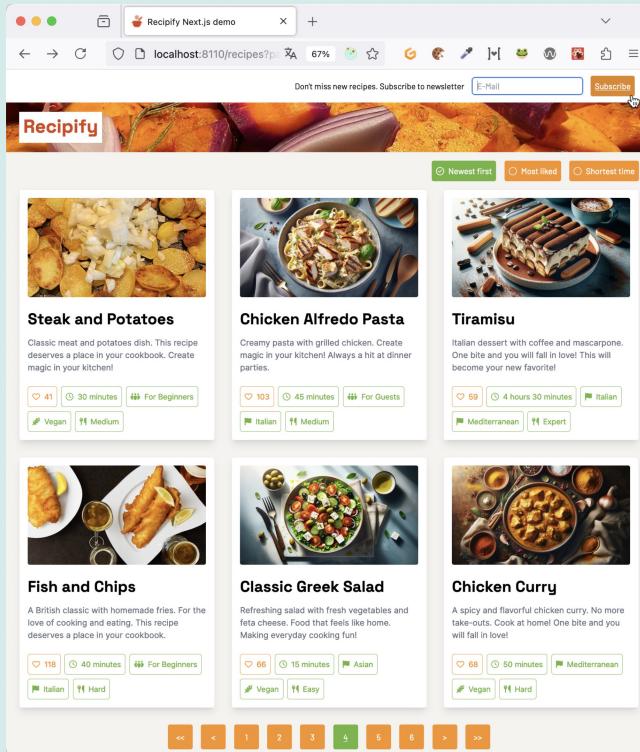
Die Webanwendung gibt es nicht

...aber: (fast) alle brauchen JavaScript

...die Frage ist nicht: ob, sondern wo und wieviel

...und in welcher Form

BEISPIEL-ANWENDUNG



• <http://localhost:8110>

BEISPIEL-CODE

Ihr findet das "Recipify"-Beispiel hier:

Next.JS

<https://github.com/nilshartmann/alles-fullstack>

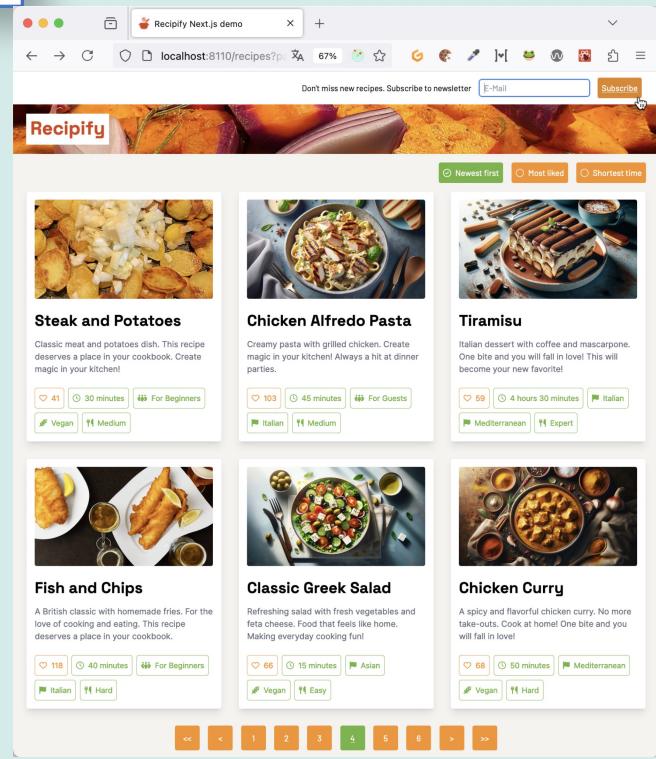
Klassische serverseitige Anwendung...

Klassische serverseitige Anwendung...

- Typische Vertreter zum Beispiel Spring MVC, PHP, .NET, Node.JS
- Oft Model-View-Controller (MVC) Pattern
- Templatesprache (z.B. Thymeleaf, Handlebars)

Unser Server erzeugt eine HTML-Seite...

```
@Controller  
public class RecipeController {  
  
    private final RecipeRepository repository;  
  
    RecipeController(RecipeRepository repository) {  
        this.repository = repository;  
    }  
  
    @GetMapping("/")  
    String getRecipesList(Pageable pageable, Model model) {  
  
        var recipes = this.repository.findAll(pageable);  
  
        // Model mit allen Informationen erzeugen, die  
        // das Template zum rendern braucht  
        model.addAttribute("recipes", recipes);  
        model.addAttribute("pageable", pageable);  
  
        // Name des Templates, das den HTML-Code für die  
        // Seite erzeugt  
        return "index";  
    }  
}
```



WEBANWENDUNGEN

...die ist aber komplett statisch

The screenshot shows a web browser window displaying a static website for a recipe platform. The header features the word "Recipify" in a white, rounded font. Below the header is a navigation bar with links like "Home", "About", "Contact", and "Logout". A search bar is present, along with a "Cart" icon showing a count of 3. The main content area displays a grid of six recipe cards:

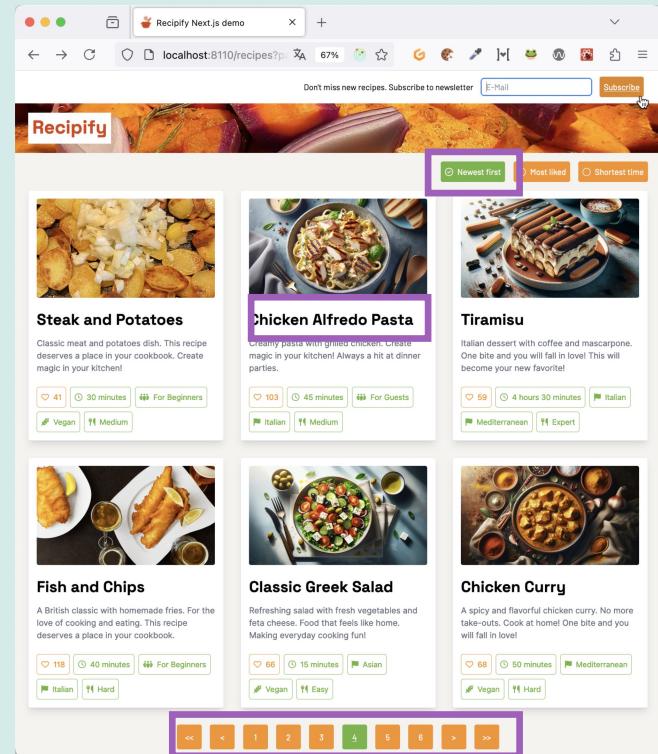
- Steak and Potatoes**: A dish of meat and potatoes. Details: 41 likes, 30 minutes, For Beginners, Vegan, Medium.
- Chicken Alfredo Pasta**: A bowl of pasta with chicken. Details: 103 likes, 45 minutes, For Guests, Italian, Medium.
- Tiramisu**: A dessert with layers of cake and coffee. Details: 59 likes, 30 minutes, Italian, Expert.
- Fish and Chips**: A plate of fish and chips. Details: 118 likes, 40 minutes, For Beginners, Italian, Hard.
- Classic Greek Salad**: A salad with various toppings. Details: 66 likes, 15 minutes, Asian, Vegan, Easy.
- Chicken Curry**: A bowl of curry. Details: 68 likes, 50 minutes, Mediterranean, Vegan, Hard.

At the bottom of the page is a navigation bar with icons for back, forward, and search, followed by a series of numbered buttons from 1 to 6.

WEBANWENDUNGEN

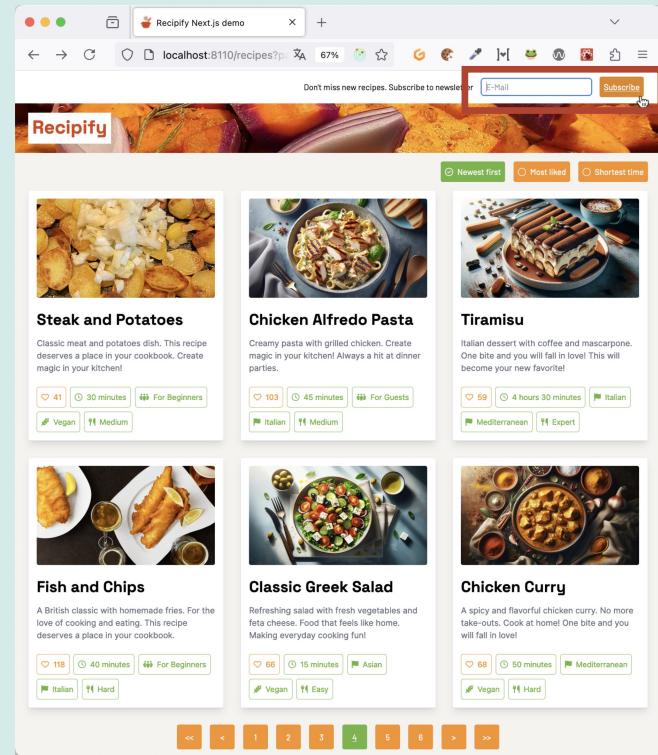
...die ist aber komplett statisch

- Interaktion per **Link**



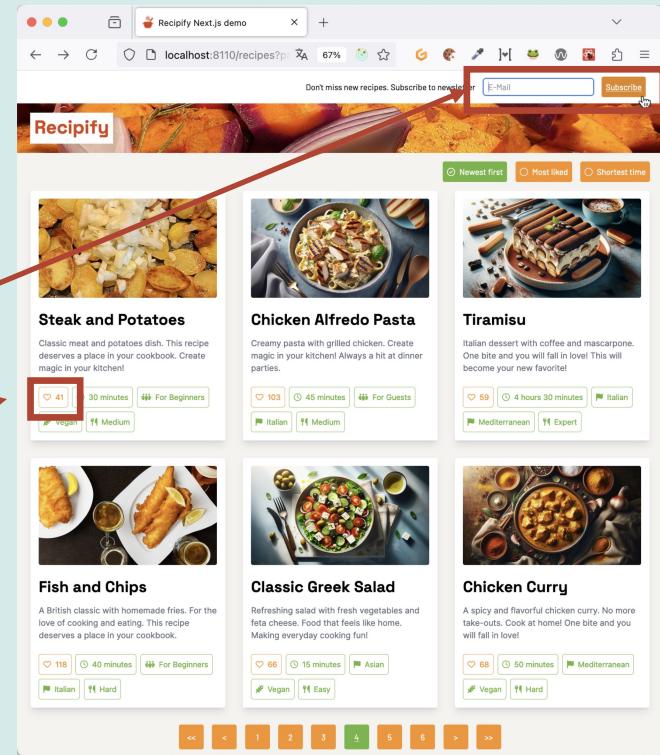
...die ist aber komplett statisch

- Interaktion per **Link**
- oder **Formular**



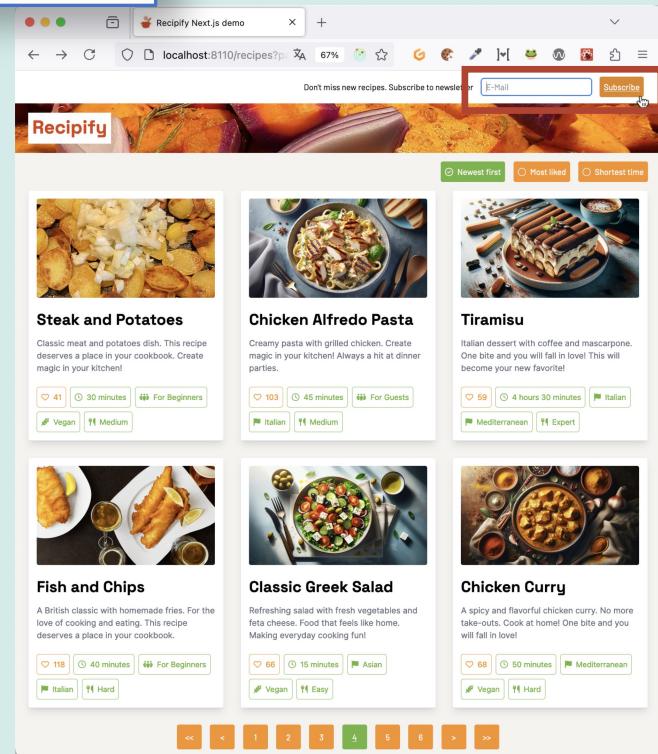
...die ist aber komplett statisch

- Interaktion per **Link**
- oder **Formular**
- in jedem Fall wird eine **komplett neue** Seite abgefragt



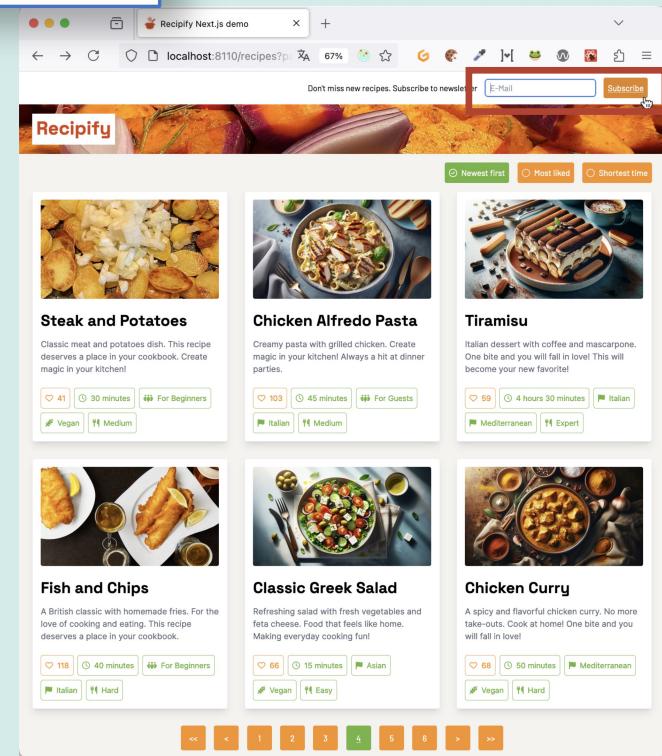
Bei jeder Interaktion: ein Server-Roundtrip...

- ist das ein Problem? 🤔
- was hat das für Konsequenzen? 🤔



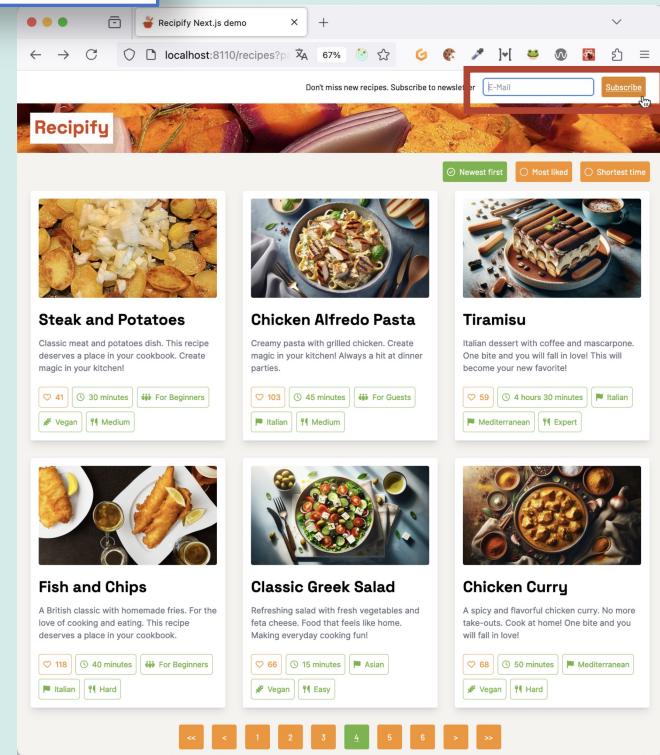
Bei jeder Interaktion: ein Server-Roundtrip...

- Menge an Daten, Latenz
- Zustand in Eingabefeldern geht verloren
- Animationen etc. laufen nicht weiter



Bei jeder Interaktion: ein Server-Roundtrip...

- Menge an Daten, Latenz
 - Zustand in Eingabefeldern geht verloren
 - Animationen etc. laufen nicht weiter
-
-  Newsletter
 -  Like-Button
 -  Timer



Weitere Interaktionen...

- Feedback direkt beim Tippen
- z.B. Zeichen-Zähler

Your opinion?

Your name:

Nils

Your rating:



Your comment:

Lovely!

7/500 characters

Submit Rating

Weitere Interaktionen...

- Feedback direkt beim Tippen
- z.B. Zeichen-Zähler
- Doppeltes Submit verhindern

Your opinion?

Your name:

Nils

Your rating:



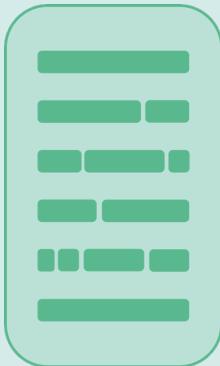
Your comment:

Lecker...!

10/500 characters

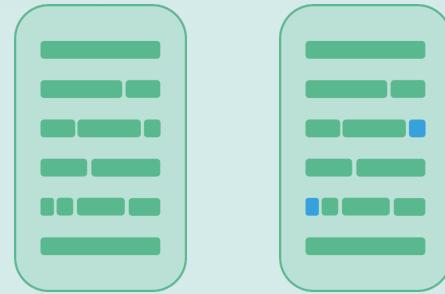
Submit Rating

"Können wir nicht hier und da, ad-hoc JavaScript hinzufügen?"



HTML-Seite

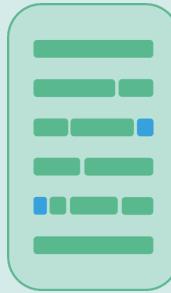
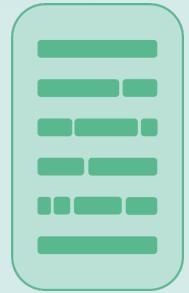
Ja, das geht!



HTML-Seite

- JavaScript Schnipsel einstreuen ("vanilla JS" oder zum Beispiel jQuery)

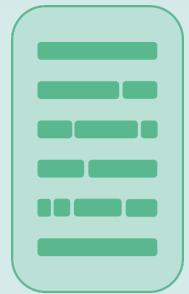
Ja, das geht!



HTML-Seite

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

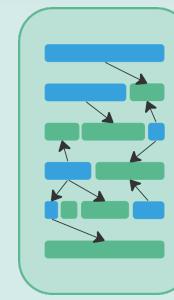
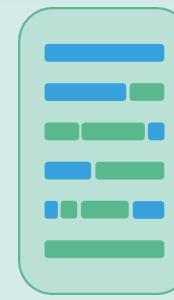
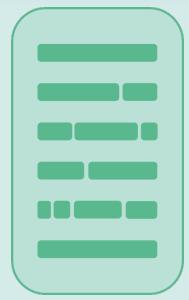
"...hier und da müsste auch noch schnell was interaktives hin..."



HTML-Seite

- Wir schreiben also noch etwas mehr JavaScript Schnipsel

"...und hier... und hier ... und hier... und ..." au weia!



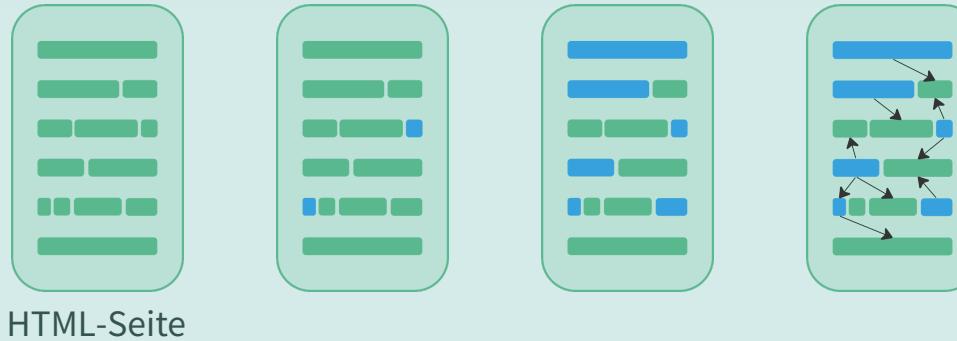
HTML-Seite

Das Problem von Schnipsel-Architektur



- Bunter Strauß an Server- und Client-Technologien (Backend-Sprache, Template-Sprache, JavaScript)
- Verantwortlichkeit willkürlich auf Frontend und Backend aufgeteilt

Das Problem von Schnipsel-Architektur



- Die Probleme "schleichen sich ein"
- Spätestens wenn es um Interaktionen geht, die nicht nur lokal sind
- Plötzlich hat unser Code nicht "ein paar Probleme" sondern ist **kaputt**

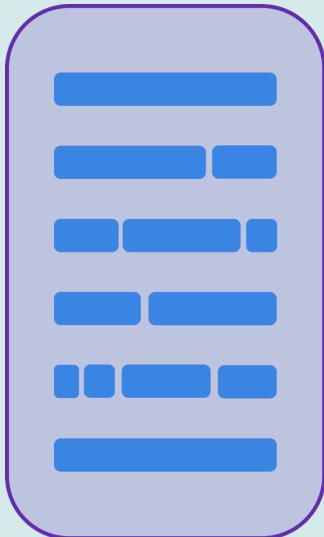
Dann besser alles in JavaScript? Single-Page-Anwendungen



JavaScript-Code

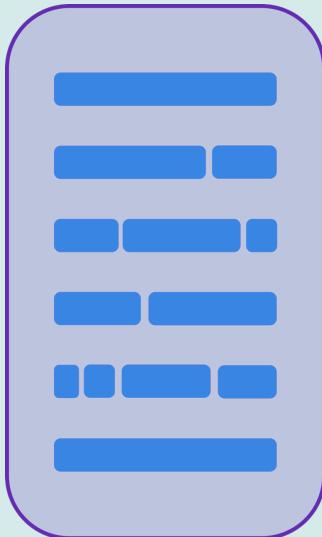
- Aus "Seiten" werden jetzt "Anwendungen"
- Klare Verantwortlichkeit: Server für Logik und Daten, Browser für UI
- Es gibt stabile und verbreitete Frameworks für jeden Geschmack

Single-Page-Anwendungen



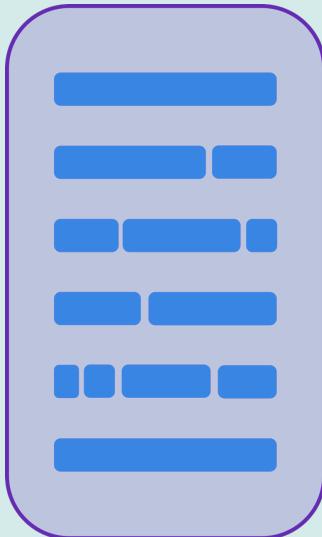
- Darstellung erfolgt vollständig mit JavaScript
- Statisches HTML spielt (fast) keine Rolle
- Die Anwendung kommuniziert mit dem Backend über API
- Ausgetauscht werden Daten, aber keine UI
- Vertreter: Angular, React, Svelte, Vue

Single-Page-Anwendungen



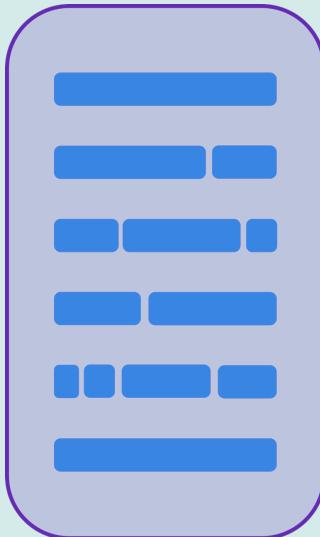
- Konsequenz viel JavaScript zur Laufzeit
- Auch für statische Inhalte

Single-Page-Anwendungen



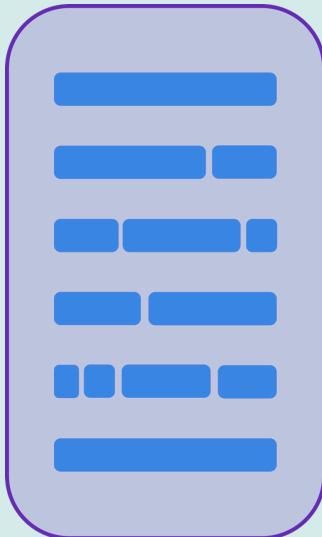
- JavaScript-Code:
 1. muss zum Browser gesendet werden
 2. muss vom Browser ausgeführt werden
 3. kann dann die darzustellenden Daten lesen
 4. kann dann erst die Daten anzeigen
 5. erst dann ist die Anwendung einsatzbereit
 6. Mit jedem Feature wird es mehr

Single-Page-Anwendungen



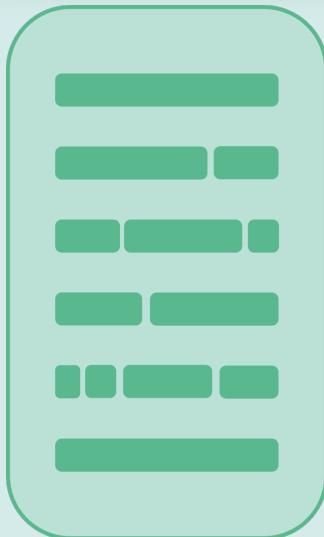
- Das hat Auswirkungen auf die **Laufzeit**-Performance
- Insbesondere beim Starten

Single-Page-Anwendungen



- Das hat Auswirkungen auf die **Laufzeit**-Performance
- Insbesondere beim Starten
- Ob das ein Problem für die eigene Anwendung ist, muss man von Fall zu Fall entscheiden

"Gibt es denn nichts dazwischen"?



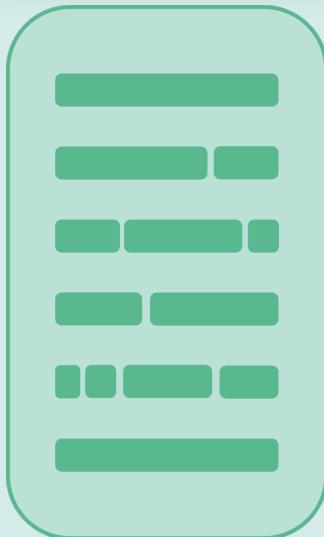
Nur Server (+JS)



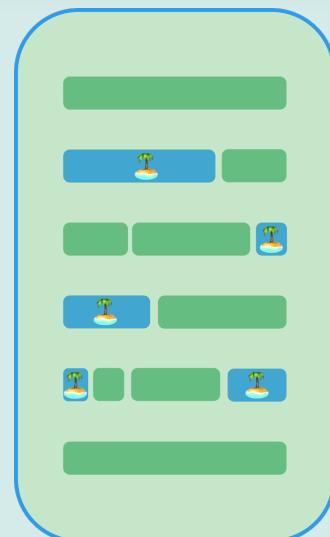
Nur Client (SPA)

WEBANWENDUNGEN

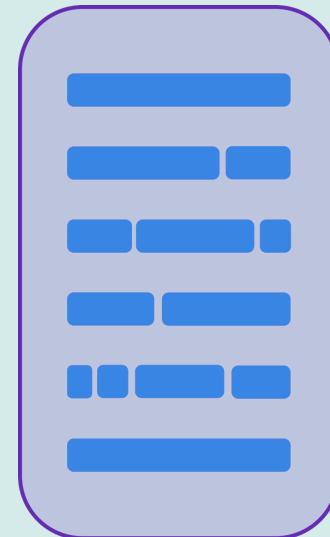
"Doch – Fullstack-Anwendungen 😊"



Nur Server (+JS)



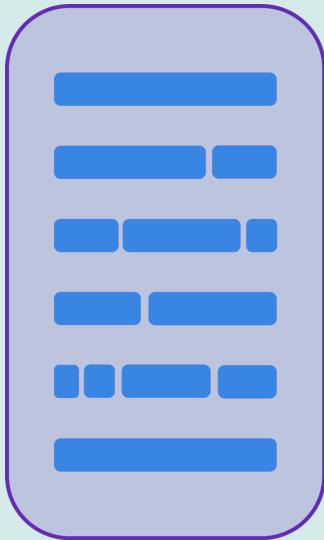
JS Fullstack Anwendung



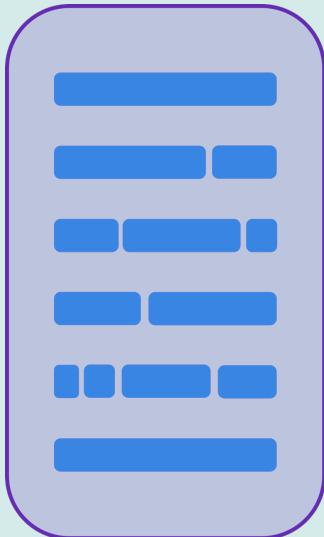
Nur Client (SPA)

FULLSTACK-ANWENDUNGEN (JAVASCRIPT)

Fullstack-Anwendungen (JavaScript)

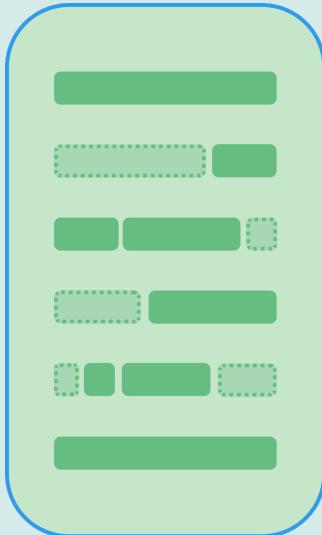


Fullstack-Anwendungen (JavaScript)



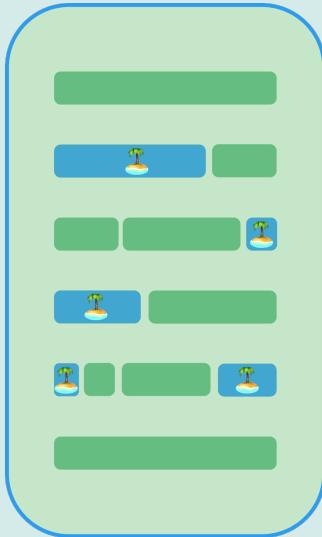
- Ebenfalls vollständig in **JavaScript** geschrieben

Fullstack-Anwendungen (JavaScript)



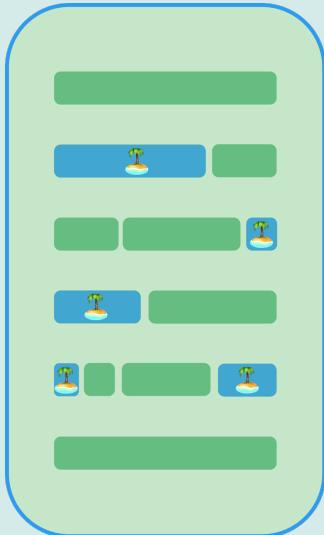
- Grundsätzliche Idee:
 1. **UI** wird serverseitig vorgerendert
 2. **UI** wird zum Browser gesendet und angezeigt

Fullstack-Anwendungen (JavaScript)



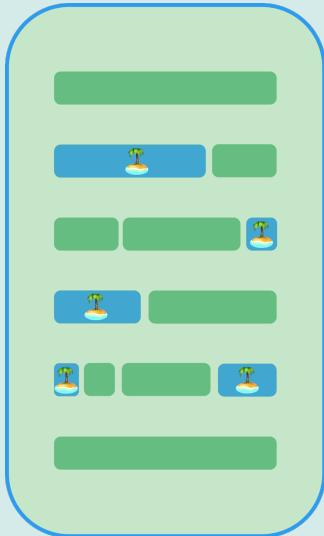
- Grundsätzliche Idee:
 1. **UI** wird serverseitig vorgerendert
 2. **UI** wird zum Browser gesendet und angezeigt
 3. Nur der JavaScript-Code ("Islands") für **Interaktionen** wird zum Browser geschickt

Fullstack-Anwendungen (JavaScript)



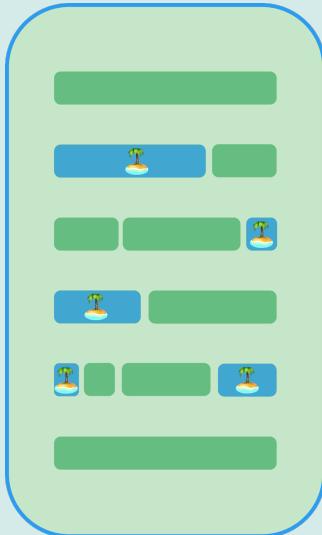
- Anwendung startet schneller:
 1. Browser bekommt fertige **UI** zur Darstellung

Fullstack-Anwendungen (JavaScript)



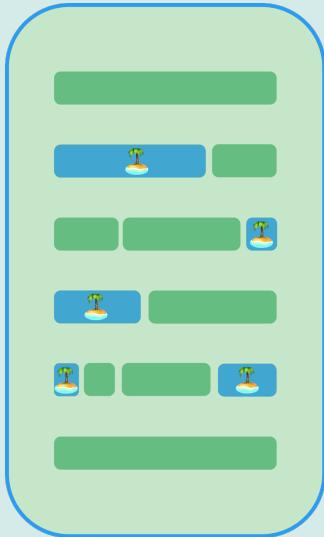
- Anwendung startet schneller:
 1. Browser bekommt fertige **UI** zur Darstellung
 2. Anwendung ist **sichtbar** ✓

Fullstack-Anwendungen (JavaScript)



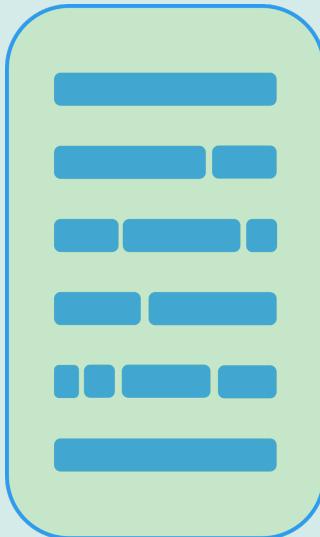
- Anwendung startet schneller:
 1. Browser bekommt fertige **UI** zur Darstellung
 2. Anwendung ist **sichtbar** ✓
 3. Der **notwendige JavaScript-Code** wird nachgeladen

Fullstack-Anwendungen (JavaScript)



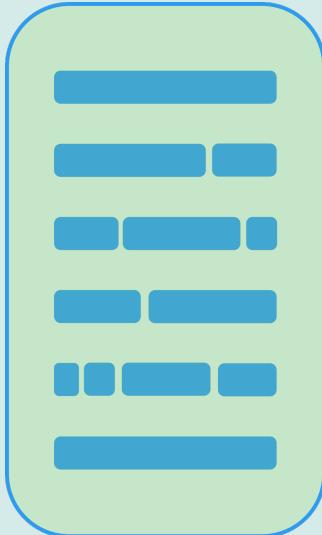
- Anwendung startet schneller:
 1. Browser bekommt fertige **UI** zur Darstellung
 2. Anwendung ist **sichtbar** ✓
 3. Der **notwendige JavaScript-Code** wird nachgeladen
 4. Anwendung jetzt **interaktiv** ✓

Fullstack-Anwendungen (JavaScript)



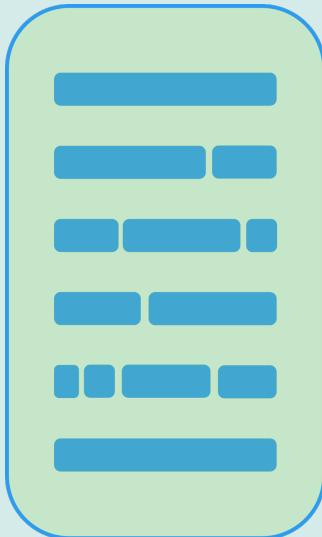
- Wir sind zurück zur **JavaScript-Schnipsel-Architektur** 😊

Fullstack-Anwendungen (JavaScript)



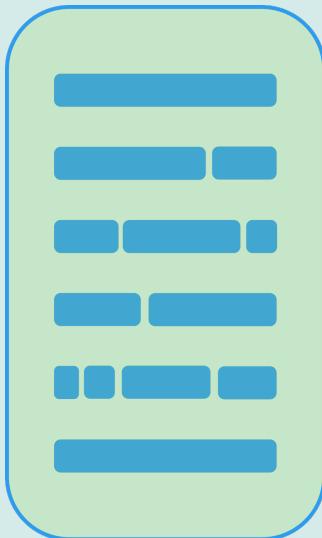
- Wir sind zurück zur **JavaScript-Schnipsel-Architektur** 😊
- Alles beim Alten also? Nein!

Fullstack-Anwendungen (JavaScript)



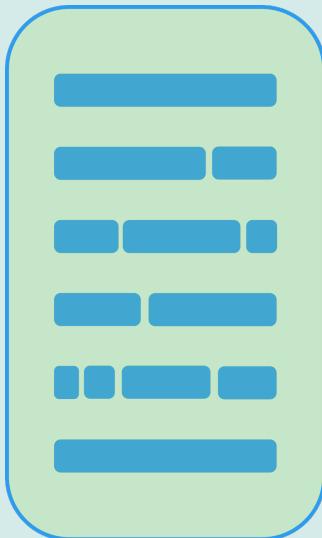
- Wir sind zurück zur **JavaScript-Schnipsel-Architektur** 😊
- Alles beim Alten also? Nein!
 - Die Schnipsel werden **automatisch** vom Framework erzeugt

Fullstack-Anwendungen (JavaScript)



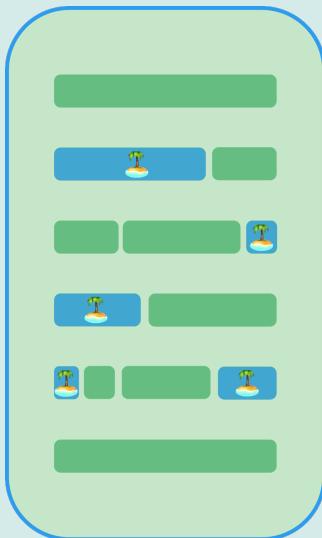
- Wir sind zurück zur **JavaScript-Schnipsel-Architektur** 😊
- Alles beim Alten also? Nein!
 - Die Schnipsel werden **automatisch** vom Framework erzeugt
 - Sie existieren nur zur **Laufzeit**

Fullstack-Anwendungen (JavaScript)



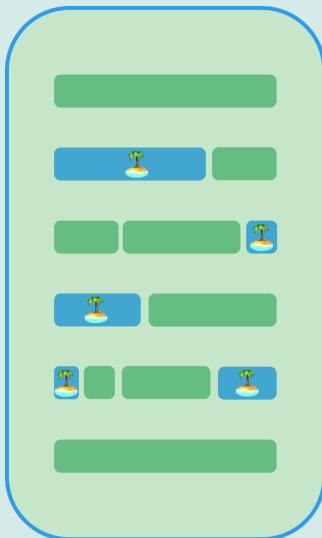
- Wir sind zurück zur **JavaScript-Schnipsel-Architektur** 😊
- Alles beim Alten also? Nein!
 - Die Schnipsel werden **automatisch** vom Framework erzeugt
 - Sie existieren nur zur **Laufzeit**
- In der **Entwicklung** ist "unser" Code aus "einem Guss"
- keine Aufteilung in unterschiedliche **Technologien**

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)



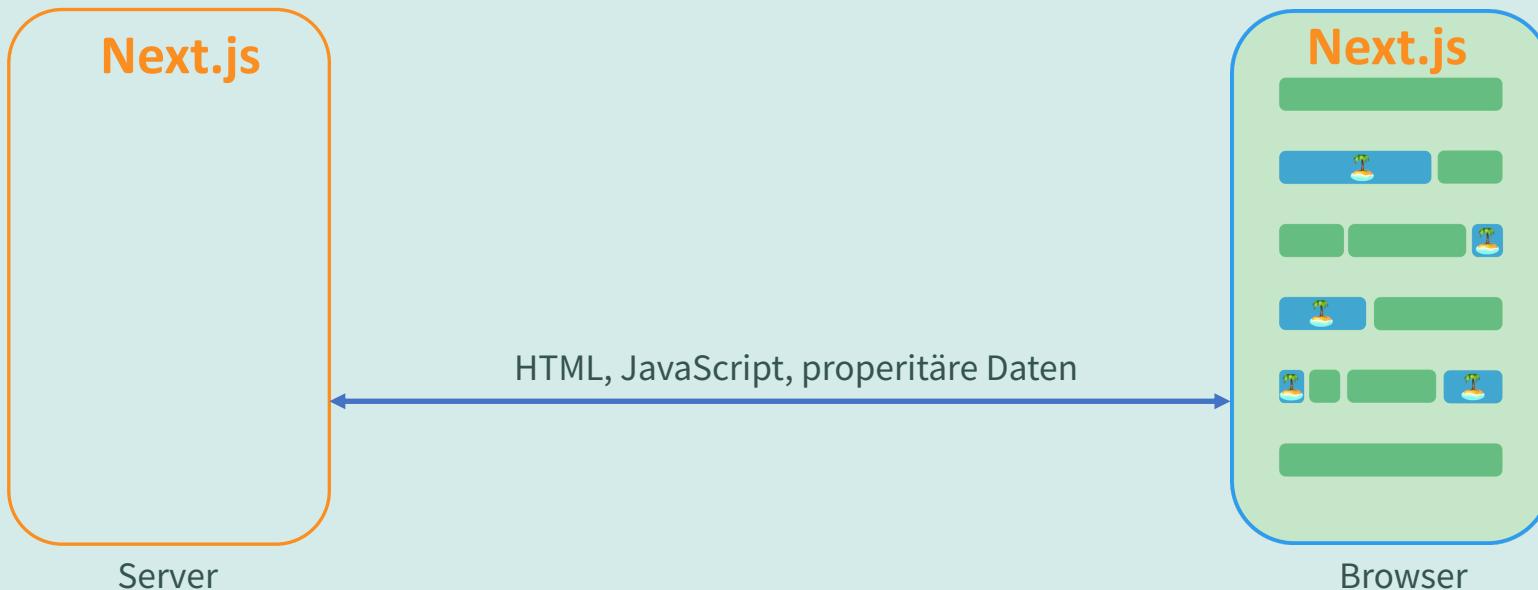
- Bekannte Vertreter:
 1. Next.js (React)
 2. SvelteKit (Svelte)
 3. Nuxt.js (Vue)
 4. Astro (eigenes Framework + Support für alle SPAs)
 5. Qwik (eigenes Framework)
- Funktionalität und Herangehensweise unterschiedlich

Beispiel: Next.js

EIN PAAR BEISPIELE...

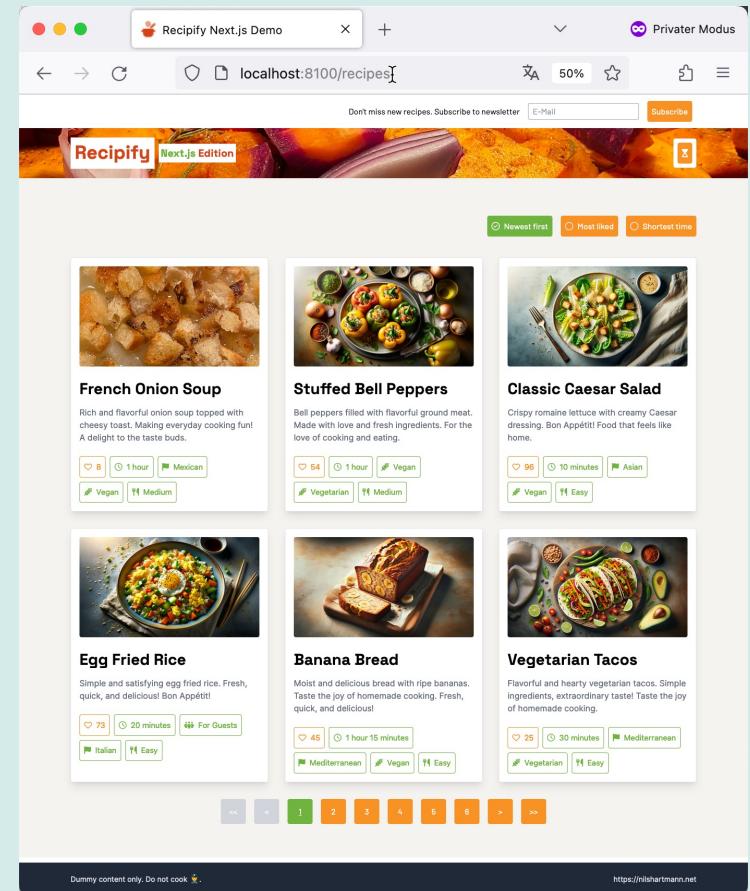
Next.js

- Basiert auf React
- Läuft im Client und im Server



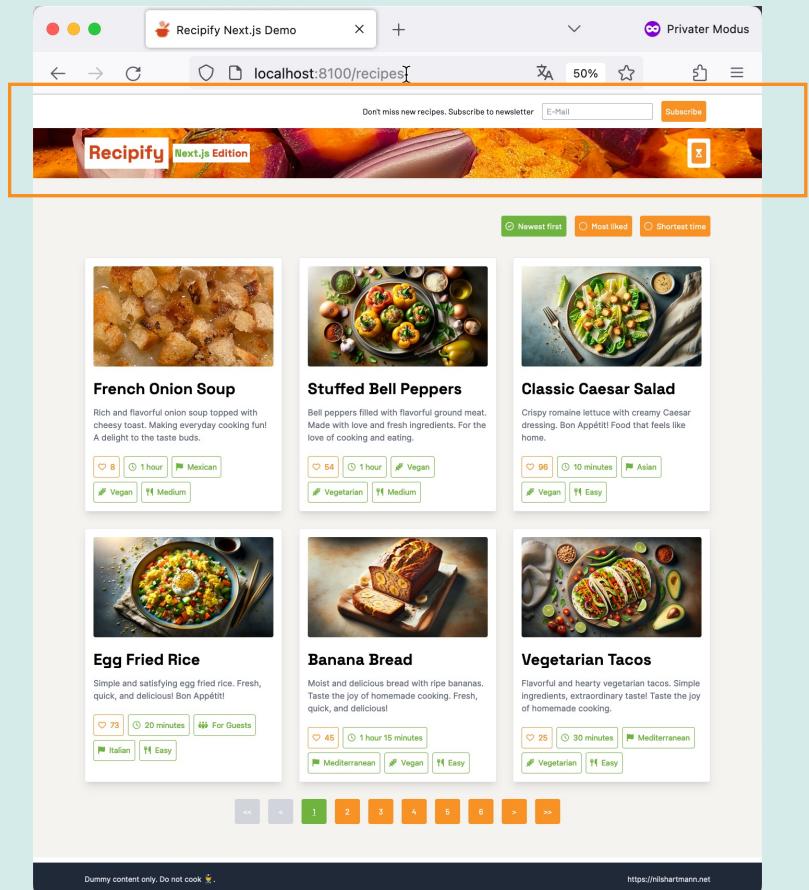
BEISPIEL: INITIALER SEITENAUFRUF

- **Beispiel: Seitenaufrufe**
- Demo: localhost:8100
- React-Komponente statt Controller
 - Aufgaben sind vergleichbar
- Seite kommt als HTML zurück
 - 🧑‍🍳 Vergleichbar mit klassisch serverseitig
 - 🧑‍🍳 Wo wird die Komponente ausgeführt?
 - 🧑‍🍳 Seitenwechsel mit PaginationBar
 - 🧑‍🍳 Open in new Tab
 - 🧑‍🍳 React Dev Tools
 - 🧑‍🍳 Was passiert ohne JavaScript?



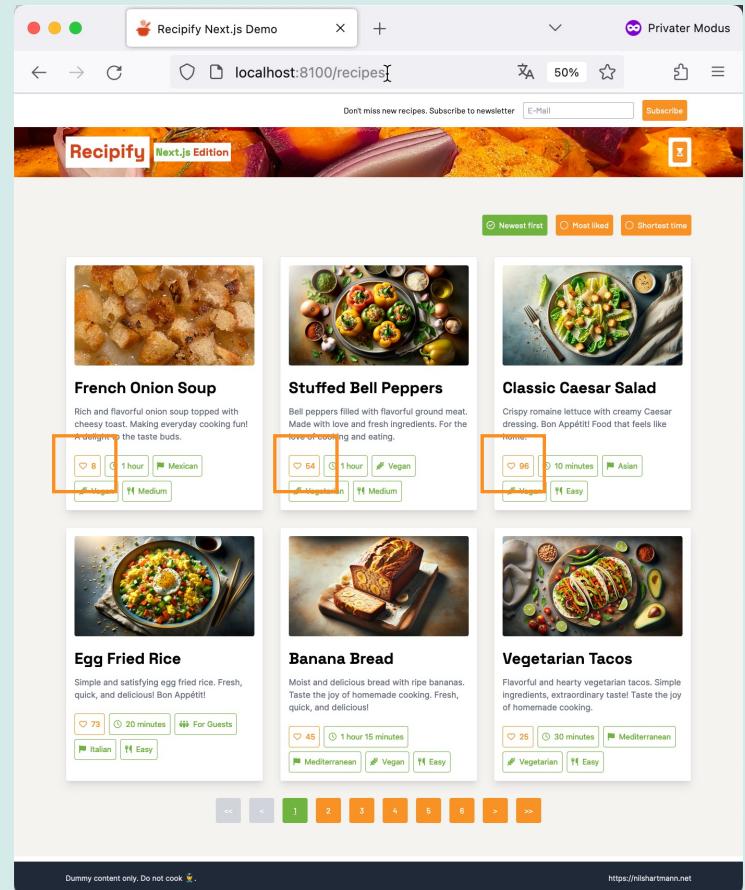
BEISPIEL: SEITENWECHSEL

- **Beispiel: Layout**
- Wiederverwendbare Rahmen
 - 🧑‍🍳 20_layout/recipes/layout.tsx kopieren
 - 🕒 Timer-Komponente im Header
 - 🤔 Was passiert beim Seitenwechsel
 - 🤔 Wie würden wir das "klassisch" machen?



BEISPIEL: INTERAKTIONEN

- **Beispiel: Interaktion**
- Zustand bleibt erhalten
 - 🧑 LikeWidget.tsx
 - 🧑 Server-Function (impliziter Endpunkt)
 - 🤔 Was passiert beim Submit mit dem Timer?
 - 🤔 Was passiert ohne JS?



BEISPIEL: INTERAKTIONEN

- **Beispiel: Interaktion #2**

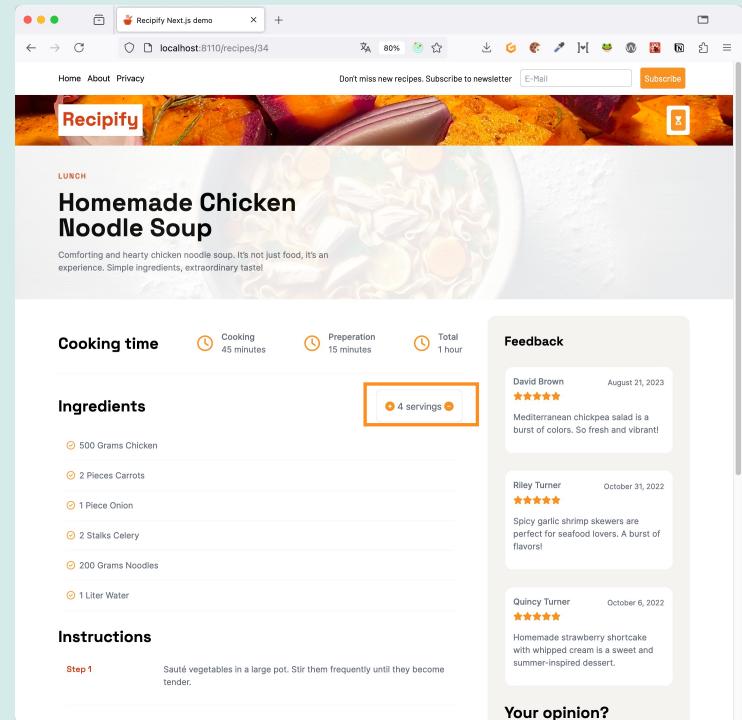
- Portionszähler

-  Reine Client-Information,
Berechnung soll nur auf dem Client
stattfinden

(RecipePageContent.tsx)

-  Ähnlich wie bei serverseitigem Ansatz:
 - wir müssen keine API für Daten bereitstellen
 - Kommunikation übers Netz übernimmt Framework

-  Ähnlich wie bei Single-Page:
 - Logik findet auf dem Client statt
 - Es gibt aber keinen Technologie-Bruch



BEISPIEL: PRIORISIERUNG VON INHALTEN

- Beispiel: Priorisierungen
- Feedback anzeigen
- 🧑‍🍳 was passiert, wenn das Laden des Feedbacks länger dauert?

The screenshot shows a recipe card for "Homemade Chicken Noodle Soup" on a website called Recipify. The card includes the following details:

- Cooking time:** Cooking 45 minutes, Preparation 15 minutes, Total 1 hour.
- Ingredients:** 500 Grams Chicken, 2 Pieces Carrots, 1 Piece Onion, 2 Stalks Celery, 200 Grams Noodles, 1 Liter Water.
- Instructions:** Step 1: Sauté vegetables in a large pot. Stir them frequently until they become tender.
- Feedback:** A sidebar displays three reviews:
 - David Brown (August 21, 2023): ★★★★☆ Mediterranean chickpea salad is a burst of colors. So fresh and vibrant!
 - Riley Turner (October 31, 2022): ★★★★☆ Spicy garlic shrimp skewers are perfect for seafood lovers. A burst of flavors!
 - Quincy Turner (October 6, 2022): ★★★★☆ Homemade strawberry shortcake with whipped cream is a sweet and summer-inspired dessert.
- Your opinion?**: A placeholder for user feedback.

"Alles Fullstack, oder was?"

Fazit

EIN PAAR BEISPIELE...

FAZIT

- Alles Fullstack oder was?

FAZIT

- Alles Fullstack oder was?
- Nö.

FAZIT

- Alles Fullstack oder was?
- Nö.
- Aber interessante Alternative

FAZIT

- **Alles Fullstack oder was?**
- Nö.
- Aber interessante Alternative
- JavaScript zur Laufzeit wird gegenüber SPA reduziert

FAZIT

- **Alles Fullstack oder was?**
- Nö.
- Aber interessante Alternative
- JavaScript zur Laufzeit wird gegenüber SPA reduziert
- JavaScript zur Entwicklungszeit wird gegenüber SPA eher mehr

FAZIT

- **Alles Fullstack oder was?**
- Nö.
- Aber interessante Alternative
- JavaScript zur Laufzeit wird gegenüber SPA reduziert
- JavaScript zur Entwicklungszeit wird gegenüber SPA eher mehr
- Wie immer: es kommt auf die Anforderung drauf an 😎

NILS HARTMANN
<https://nilshartmann.net>



vielen Dank!

Slides: <https://react.schule/tnt-fullstack-oder-was>

Fragen & Kontakt: nils@nilshartmann.net

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

