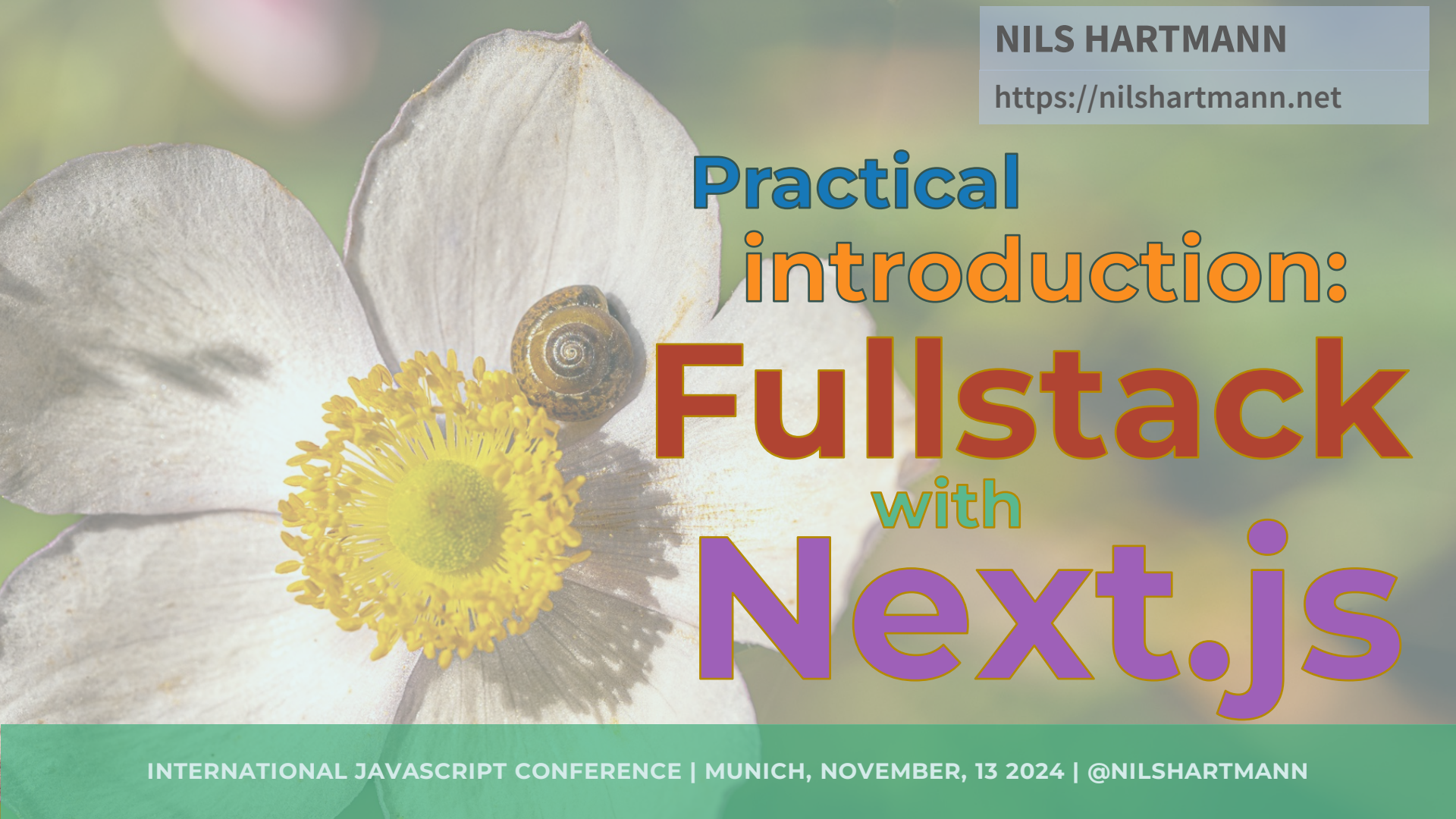
NILS HARTMANN
https://nilshartmann.net



Practical introduction: Fullstack with Next.js

INTERNATIONAL JAVASCRIPT CONFERENCE | MUNICH, NOVEMBER, 13 2024 | @NILSHARTMANN

NILS HARTMANN

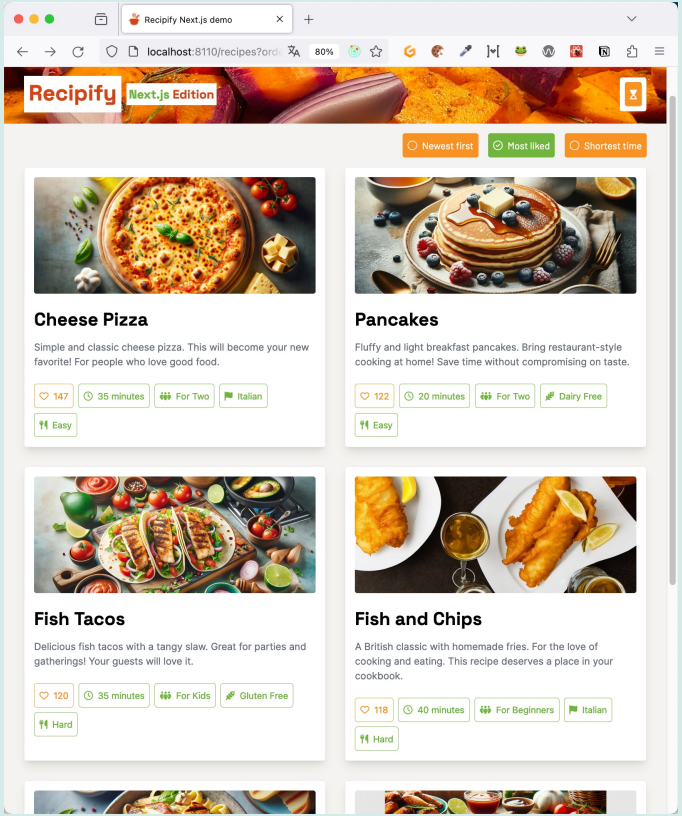
nils@nilshartmann.net

Freelance Software Developer, Architect and Trainer from Hamburg

Java, Spring, GraphQL, React, TypeScript

https://graphql.schule/video-kurs

https://reactbuch.de



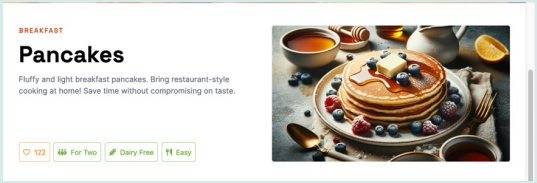
http://localhost:8110

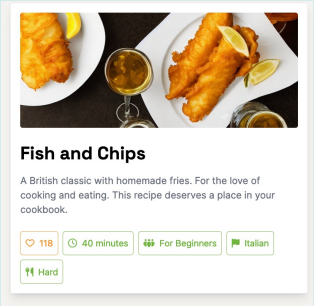
AN EXAMPLE...

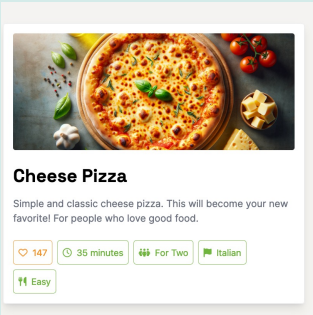
What makes up the sample application?

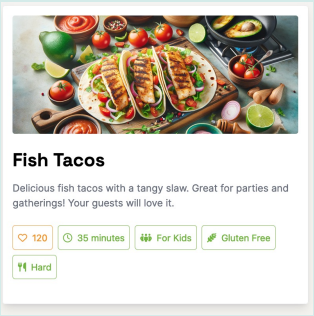
What makes up the sample application?

- Lots of food 😋



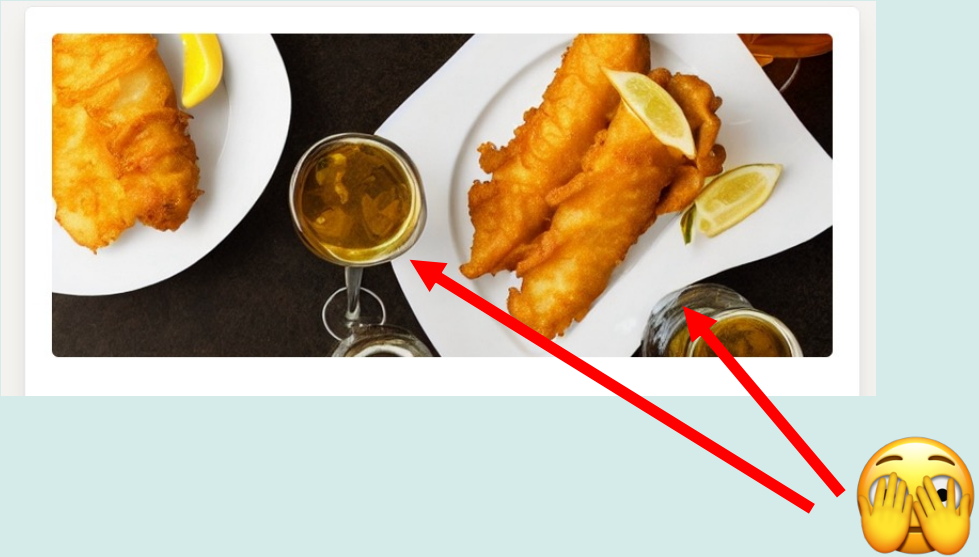






What makes up the sample application?

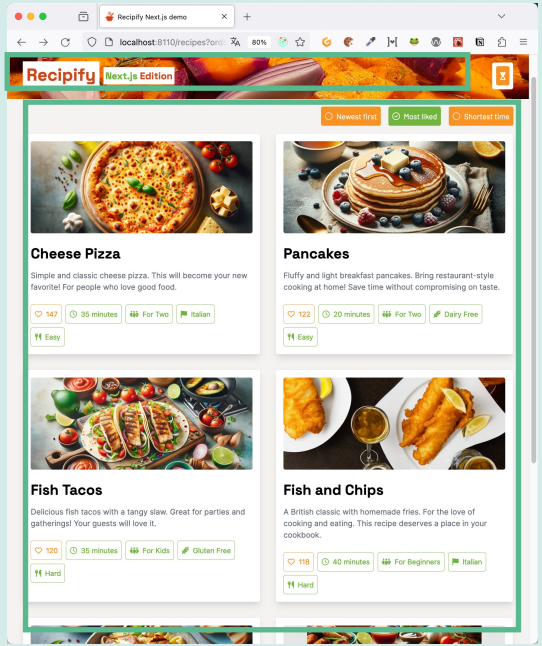
- Some bad AI-generated images

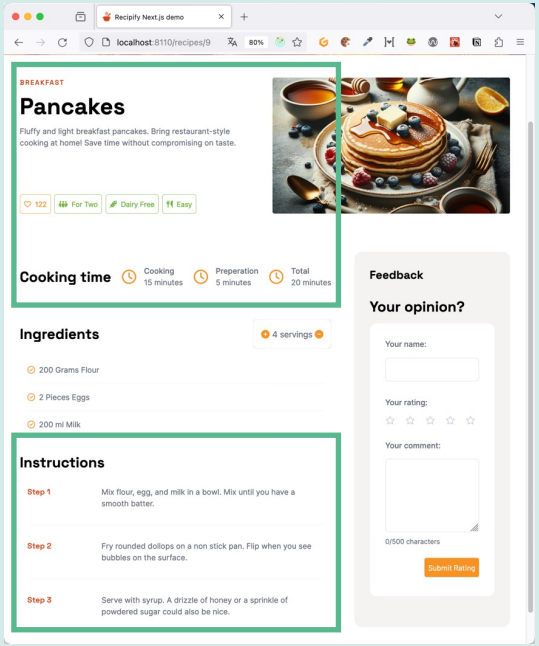


AN EXAMPLE...

What makes up the sample application... technically?

- Lots of static content ☺️

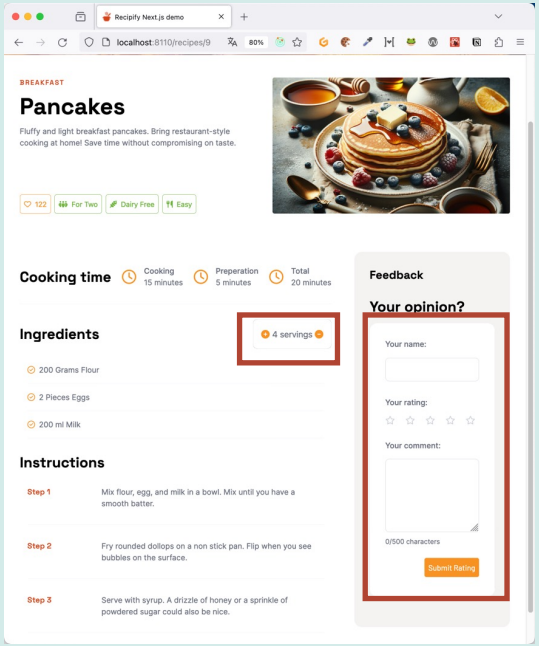




AN EXAMPLE...

What makes up the sample application... technically?

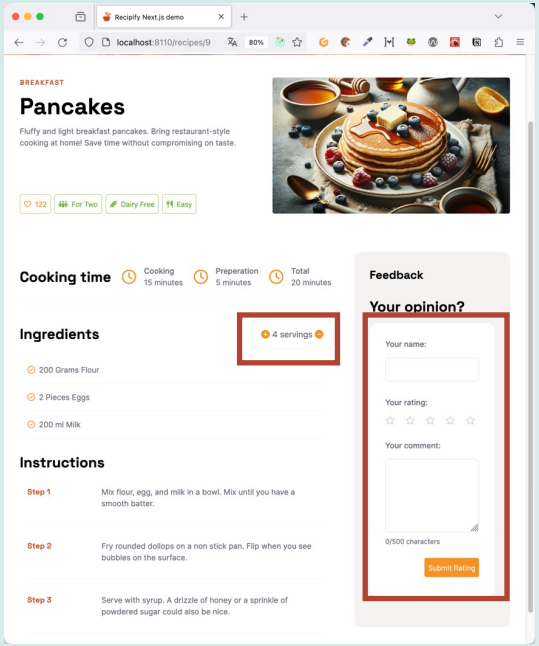
- Lots of static content ☺️
- ... only a few interactions 🦥



AN EXAMPLE...

What makes up the sample application... technically?

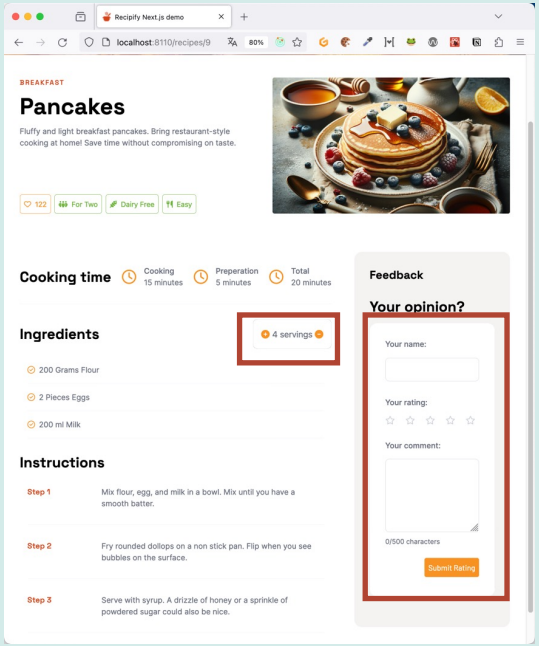
- Lots of static content ☺️
- ... only a few interactions 🦥
- ... **does that justify a Single-Page-Application?** 🤔



AN EXAMPLE...

What makes up the sample application... technically?

- Lots of static content ☺️
- ... only a few interactions 🦥
- ... **does that justify a Single-Page-Application?** 🤔
- ...huge amount of JS in the browser?
- ...no caching of content?
- ...not best possible SEO?



React

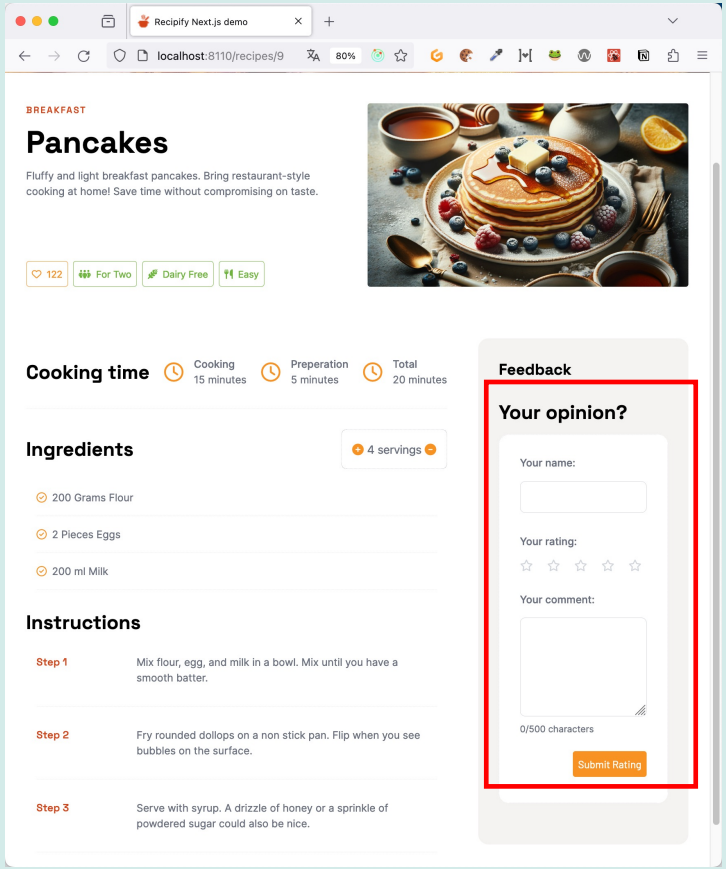
Server Components

Kinds of components

DIFFERENT KINDS OF COMPONENTS

Client components (as known before)

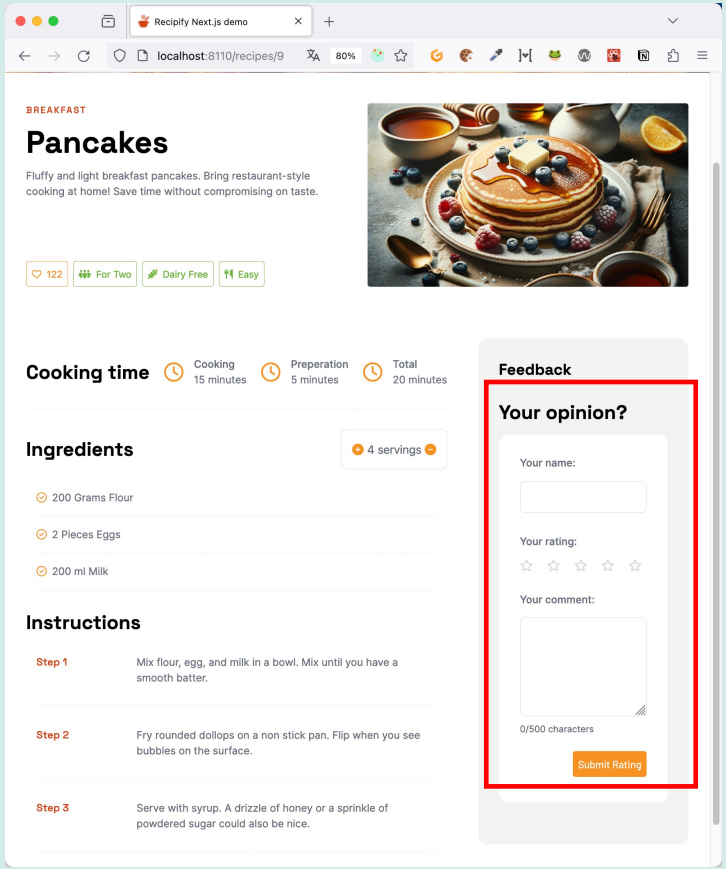
- Rendered on the Client



DIFFERENT KINDS OF COMPONENTS

Client componenten (as known before)

- Rendered on the Client
- or on the Server 🙄 (SSR)

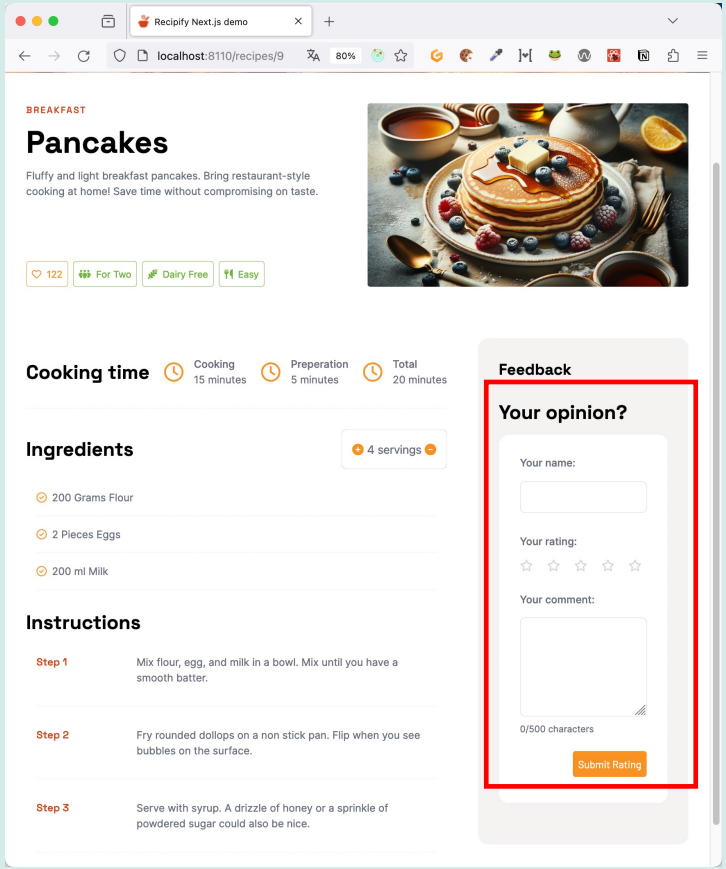


DIFFERENT KINDS OF COMPONENTS

Client componenten (as known before)

- Rendered on the Client
- or on the Server 🙄 (SSR)

- JavaScript-Code is sent to the browser
- They can be interactive and can held state



New: React Server Components

DIFFERENT KINDS OF COMPONENTS

New: React Server Components

- are rendered on the Server

DIFFERENT KINDS OF COMPONENTS

New: React Server Components

- are rendered on the Server
- or during Build 🙄

DIFFERENT KINDS OF COMPONENTS

New: React Server Components

- are rendered on the Server
- or during Build 🙄

- returning UI (!) to the browser (but no JavaScript-Code)

DIFFERENT KINDS OF COMPONENTS

Different kinds can be used together

- From JSX perspective they look the same

Next.js

Next.js

- sits on top of React

Next.js

- sits on top of React
- Some features are React standard, mainly React Server Components

Next.js

- sits on top of React
- Some features are React standard, mainly React Server Components
- Those could be implemented by other frameworks too

Next.js

- sits on top of React
- Some features are React standard, mainly React Server Components
- Those could be implemented by other frameworks too
- in the demo we will see Next.js and React features

Next.js

- sits on top of React
- Some features are React standard, mainly React Server Components
- Those could be implemented by other frameworks too
- in the demo we will see Next.js and React features
- This features are available in the App Router (Next.js 14+)

Next.js

- You need to run Next.js in production on a server

Next.js

- You need to run Next.js in production on a server
- that's a big difference to single-page-applications

Next.js

- You need to run Next.js in production on a server
- that's a big difference to single-page-applications
- Both Next.js and React are also running in the browser

Demo

DEMO

Server Components #1

- /recipes with fetchRecipes `rlp`
- where is the component rendered?
  - Console.log
  - React Dev Tools
- how can we order and filter the list? we do not have state here!
- what is refetched when we re-order the list? -> Network tab!
- what is fetched when we re-load the page -> Network tab!

DEMO

Next.js: Layout component

- /recipe/layout.tsx `la`
- Look at the timer!

DEMO

Server Components #2: Suspense

- /recipe/[recipeId]/page.tsx. `rp`
- notFound
- slow_down in demo-config.ts
- Add loading.tsx

Server Components #2: Avoid waterfalls

- In RecipePageContent, Feedbacks are fetched
- slowdown_feedback
- Add Suspense Boundary
- What do we have now: Request takes Recipe-Request + Feedback-Request

Server Components #2: Avoid waterfalls

- Move fetchFeedback to [recipeId]/index.tsx
- Pass feedbackPromise to RecipePageContent
- Adjust <FeedbackList ...> in RecipePageContent
- Adjust FeedbackList to await promise

DEMO

Server Actions: A very simple form

- LikeWidget "use server"
- This works without JavaScript!
- Note: with the new React 19 Hooks useActionState, useFormStatus, useOptimistic we could also build "real" forms that are progressively enhanced

DEMO

Client Components: Now we need JavaScript in the Browser

- Show desired result in working application
- RecipePageContent: replace IngredientsSection with ConfigurableIS
- Note:
  - we pass props "as normal" from server to client components
  - that even works for promises!
  - we can use Server Functions to execute code on the server (no endpoint required)

NILS HARTMANN
https://nilshartmann.net

Thank you very much!

Slides & Code: https://react.schule/ijs-nextjs

Questions and contact:

nils@nilshartmann.net

https://nilshartmann.net/contact