E-Portfolio: NEXT.JS

Kevin Wiesner

Lecture: Software Engineering

Date: 16.11.2022

Agenda

Introduction

Core features of Next.js

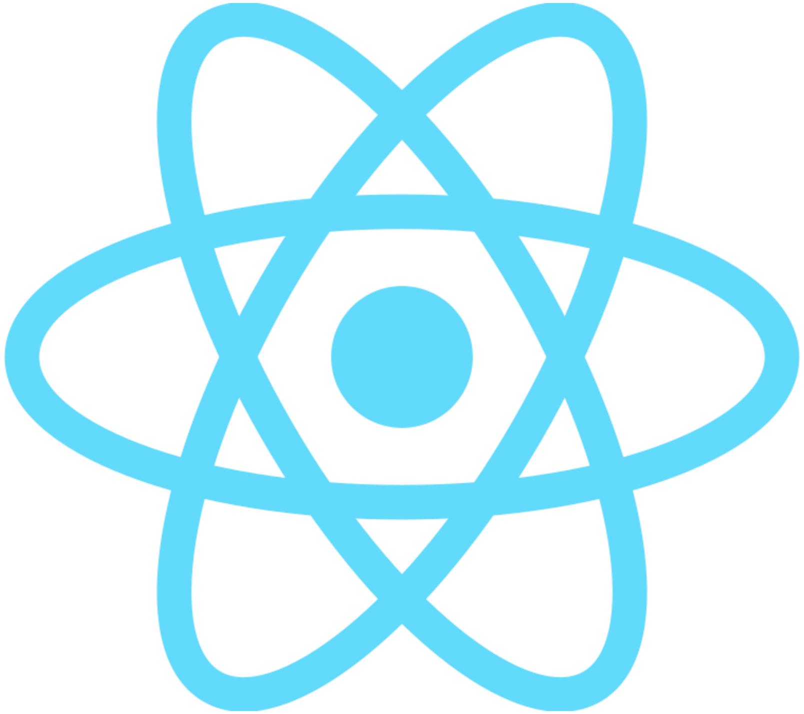
Case study

Summary

Introduction

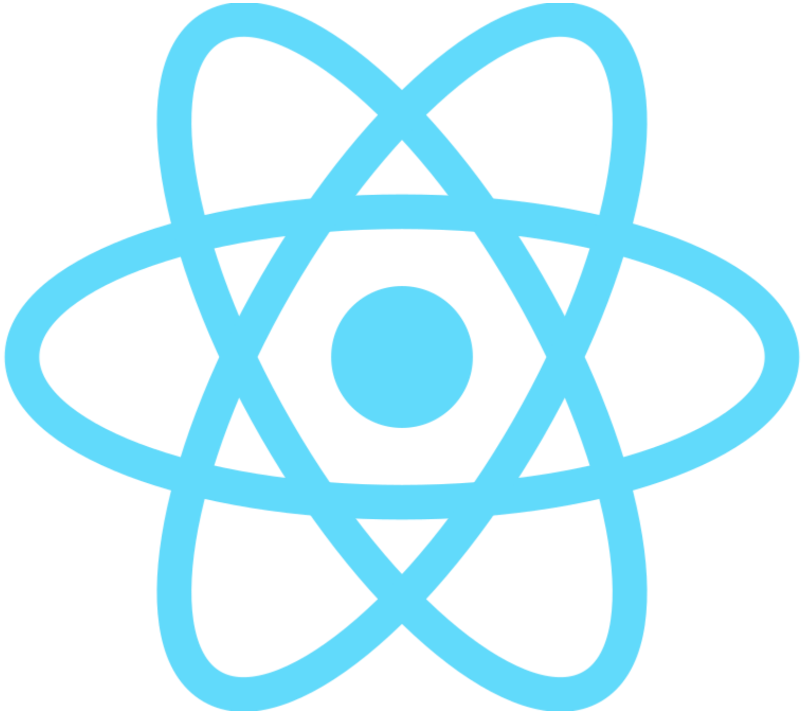
What is Next.js and why do we need it?

Let’s talk about React



> “ A JavaScript library for building user interfaces ”

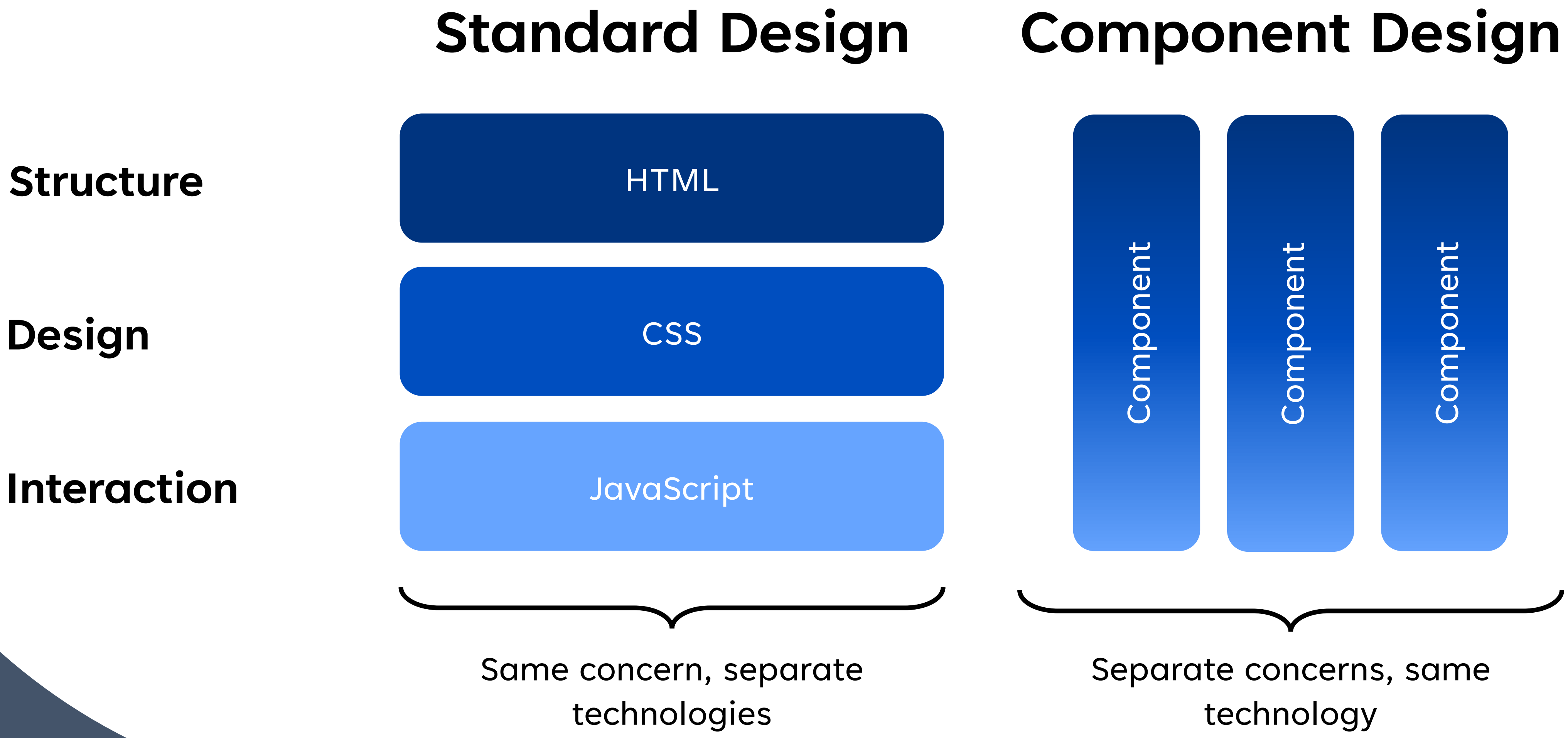
Let’s talk about React



Core features:

- Declarative
- Component-Based
- Learn Once, Write Anywhere

Let’s talk about React



Standard Design

Structure: HTML

Design: CSS

Interaction: JavaScript

Same concern, separate technologies

Component Design

Component

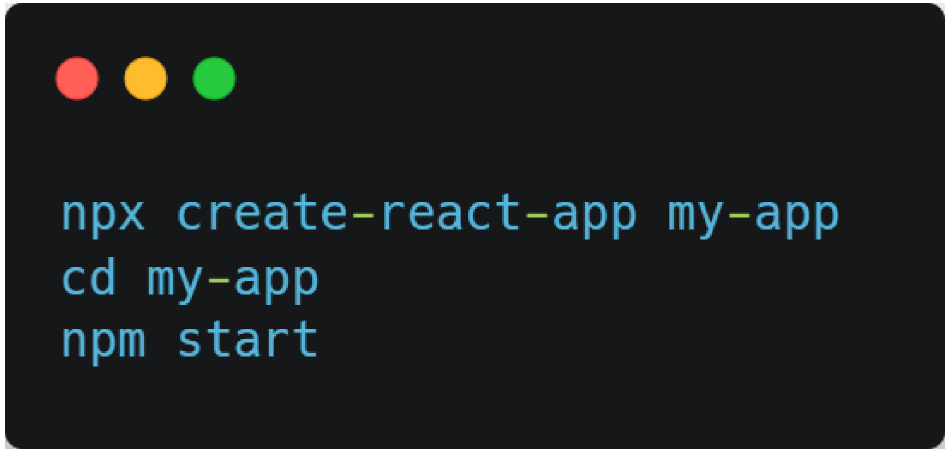
Component

Component

Separate concerns, same technology

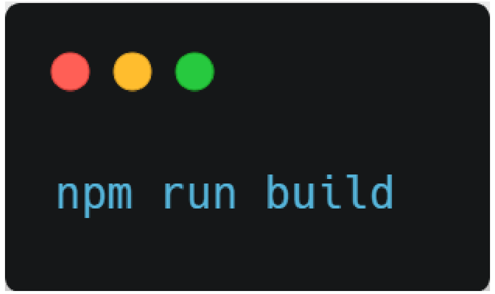
How does it look like?

Development



```
npx create-react-app my-app
cd my-app
npm start
```

Build



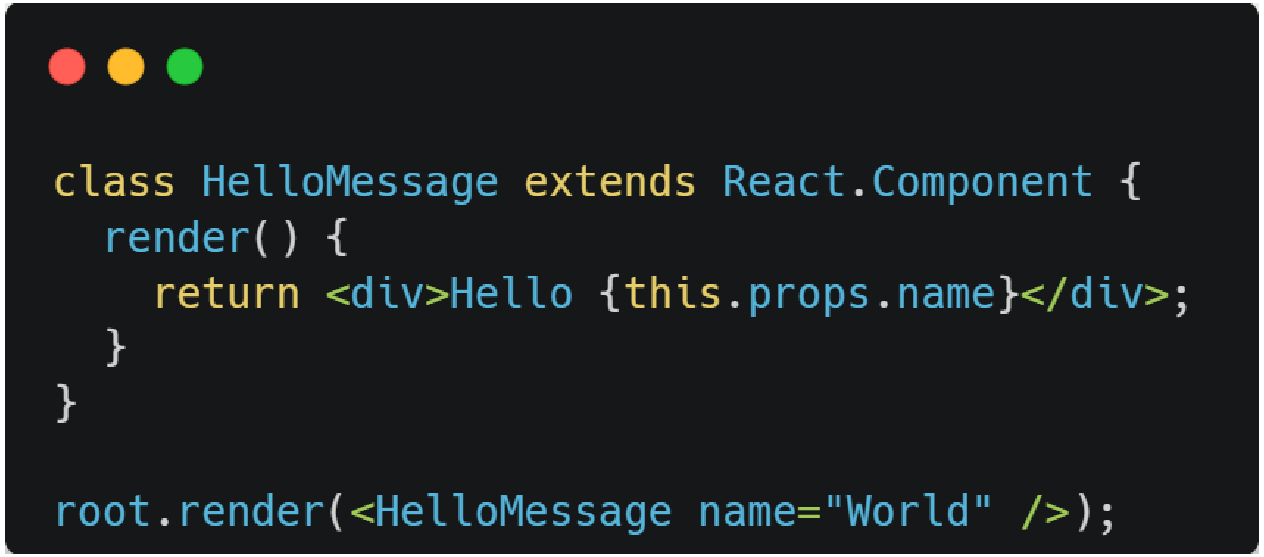
```
npm run build
```

→ **Single-Page-Applications**

How does it look like?

Developer Experience:

- JSX files
  - JavaScript Syntax Extension
  - Using “HTML” in JS
- Light-weight
- Less opinionated



```
class HelloMessage extends React.Component {
  render() {
    return <div>Hello {this.props.name}</div>;
  }
}

root.render(<HelloMessage name="World" />);
```

Is this enough?

Well… no

Is this enough?

Problems with current toolchain:

- What about routing?
- What about SSR?
- What about database integration?
- What about deployment?

Is this enough?

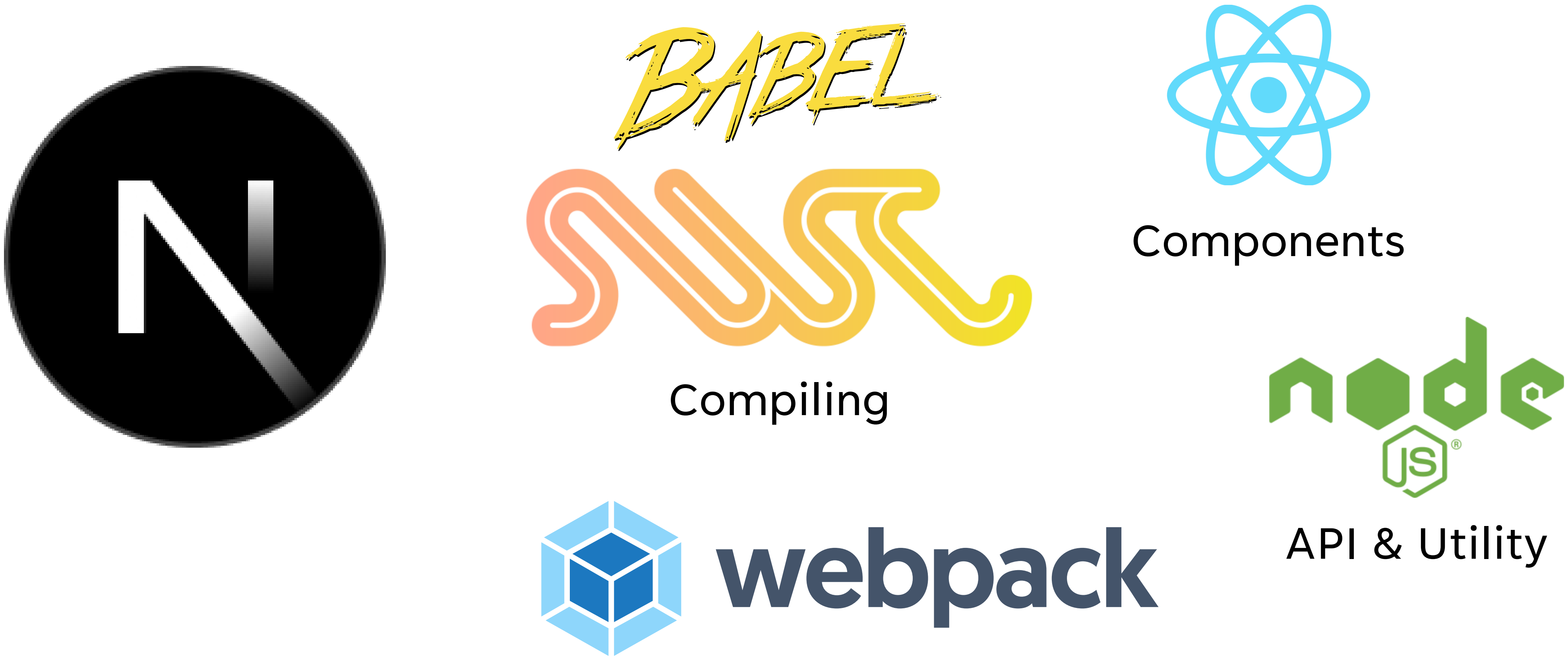
Solution:

Introduction to Next.js

What is Next.js?

- Fully-featured React framework
- Created by Vercel in 2016
- GitHub: vercel/next.js (96.3 k stars)

Introduction to Next.js



Compiling

Components

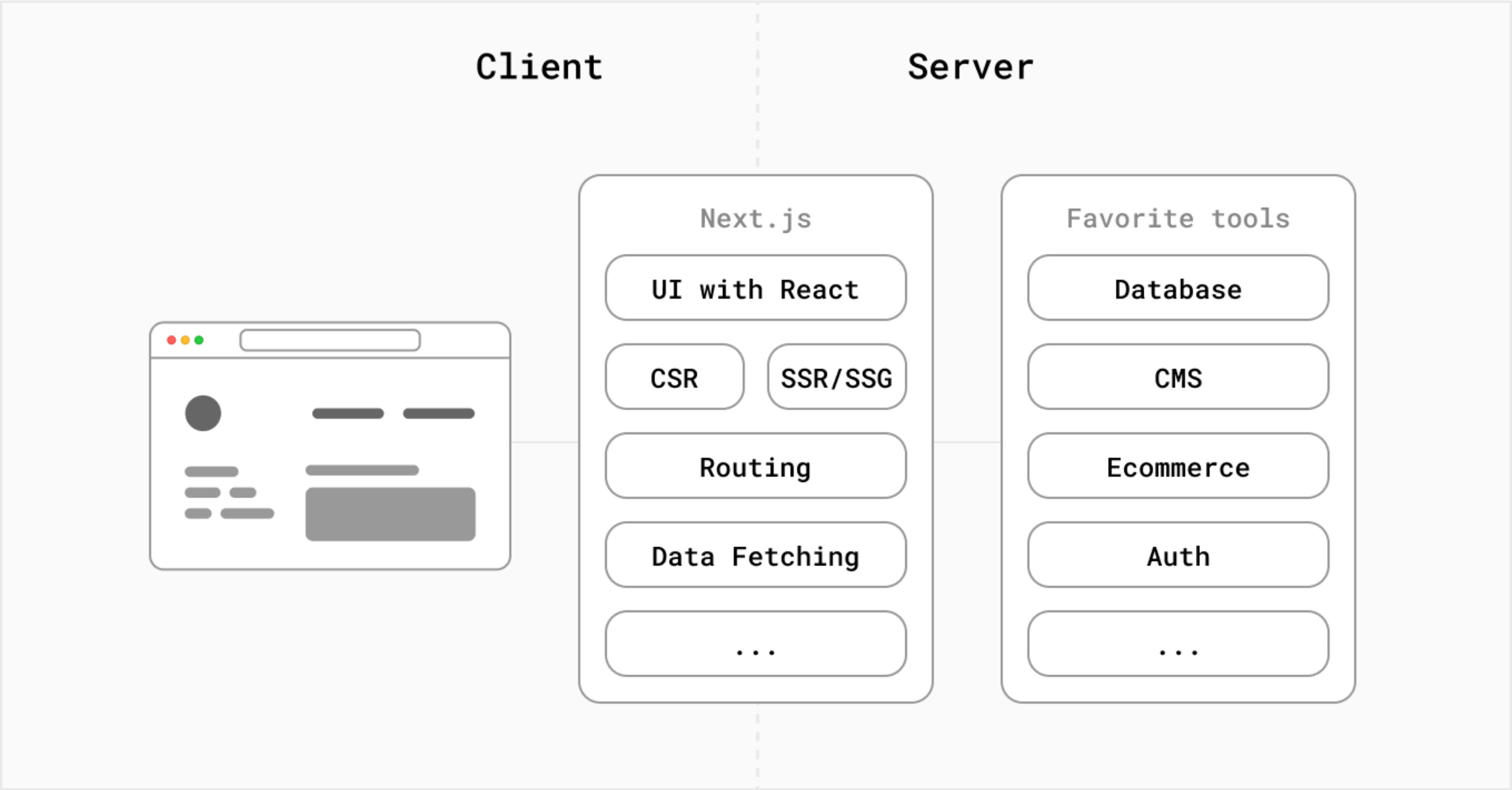
API & Utility

Bundling

Core features of Next.js

- Routing
- Rendering approaches
- Optimization
- Build & Deployment

Feature overview



Routing

File-based routing

- “pages” directory
  - Exported React component
  - “.js”, “.jsx”, “.ts”, “.tsx”
- Types of routes
  - Index routes – index.js
  - Nested routes – folders
  - Dynamic routes – [param].js
    - Nested – [...all].js

```
next-js-example/
├─ pages/
│  ├─ api/
│  │  ├─ order/
│  │  │  ├─ index.js
│  │  │  ├─ [orderId].js
│  │  ├─ post/
│  │  │  ├─ [...param].js
│  ├─ team/
│  │  ├─ index.js
│  │  ├─ [member].js
│  ├─ _app.js
│  ├─ index.js
│  ├─ about.js
├─ ....
```

Routing

API Routes

- Build API with Next.js
- “api” folder
  - Files as API endpoints
  - Server-side only bundles

```
next-js-example/
├─ pages/
│  ├─ api/
│  │  ├─ order/
│  │  │  ├─ index.js
│  │  │  ├─ [orderId].js
│  │  ├─ post/
│  │  │  ├─ [...param].js
│  ├─ team/
│  │  ├─ index.js
│  │  ├─ [member].js
│  ├─ _app.js
│  ├─ index.js
│  ├─ about.js
├─ ....
```

Routing

Examples

- /
- /about
- /team
- /team/kevin
- /api/order
- /api/order/1234
- /api/post/1/2

```
next-js-example/
├─ pages/
│  ├─ api/
│  │  ├─ order/
│  │  │  ├─ index.js
│  │  │  ├─ [orderId].js
│  │  ├─ post/
│  │  │  ├─ [...param].js
│  ├─ team/
│  │  ├─ index.js
│  │  ├─ [member].js
│  ├─ _app.js
│  ├─ index.js
│  ├─ about.js
├─ ....
```

Rendering approaches

Concept of Pre-Rendering

- Generating HTML for the page in advance
- Different approach based on the generation time
- Approach can be chosen per page

Static-site generation (SSG)

Server-side rendering (SSR)

Rendering approaches

Static-site generation (SSG)

- Generating HTML for the page at **build** time
  - Page **default**
  - Possible to inject **external content data**
  - Can be used for **dynamic** pages
  - Is later used for **each** request
- Good for e.g. blog posts, marketing pages, etc.
- Variant: Incremental Static Regeneration (ISR)
  - Developer sets interval for **regeneration**

Rendering approaches

Server-side rendering (SSR)

- Generating HTML for the page on **each request**
    - Possible to inject **external content data**
    - **Less performant**
- Good for e.g. personalized content, etc.

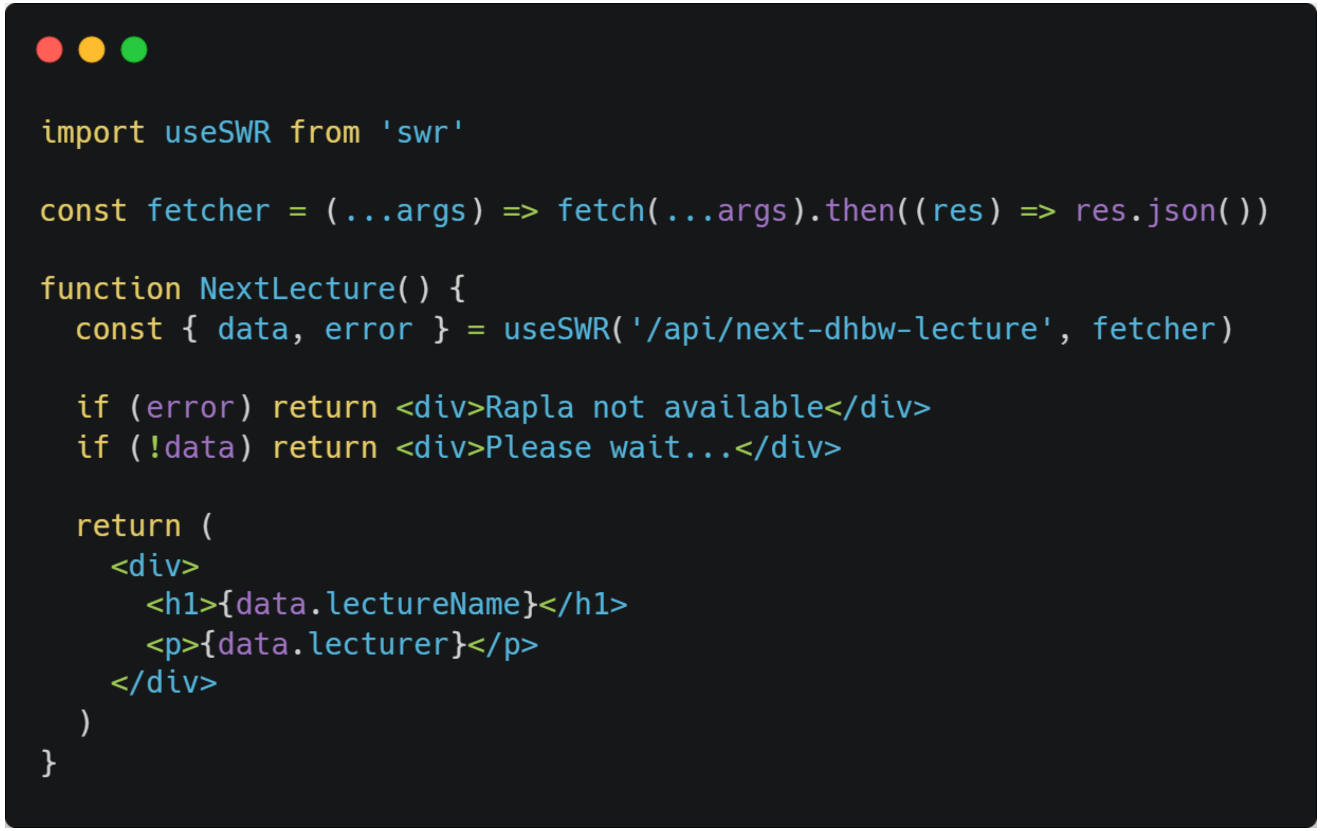
Rendering approaches

Client-side rendering (CSR)

- Fetching data at **runtime** or **component mount time**
  - Performance depends on **client**
  - Impacts **SEO ratings**
  - Preferred in Next.js: **SWR** (stale-while-revalidate)
- Good for e.g. frequent page updates, etc.

Rendering approaches

Client-side rendering (CSR)



```
import useSWR from 'swr'

const fetcher = (...args) => fetch(...args).then((res) => res.json())

function NextLecture() {
  const { data, error } = useSWR('/api/next-dhbw-lecture', fetcher)

  if (error) return <div>Rapla not available</div>
  if (!data) return <div>Please wait...</div>

  return (
    <div>
      <h1>{data.lectureName}</h1>
      <p>{data.lecturer}</p>
    </div>
  )
}
```

Optimization

Use additional Next.js tools to improve performance

- Image Component (next/image)
  - Automatic size optimization
  - Prevention of Cumulative Layout Shifts
- Optimizing Fonts (@next/font)
  - Self-hosted Google Fonts
  - Downloaded at build time

Build & Deployment

`next build`

- Command for the optimized server build
- Can be hosted on any service with Node.js runtime
  - Vercel
- Application can be dockerized and scaled with Kubernetes

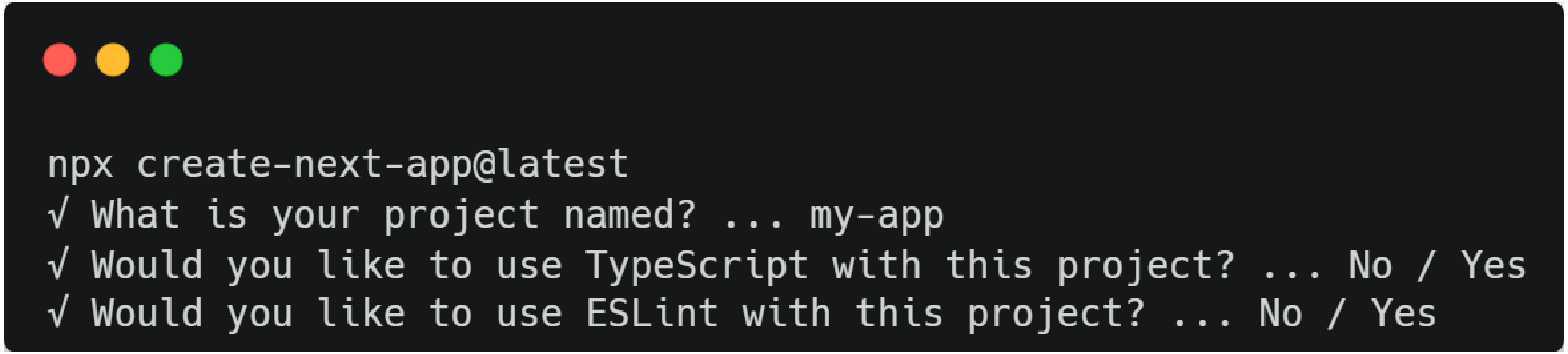
`next export`

- Static HTML export of application
- Can be used without a Node.js server
- Limited features

Case study

How to create Next.js apps

Create a Next.js application



```
npx create-next-app@latest
√ What is your project named? ... my-app
√ Would you like to use TypeScript with this project? ... No / Yes
√ Would you like to use ESLint with this project? ... No / Yes
```

Next.js setup for case study

SQLite

Database

Prisma

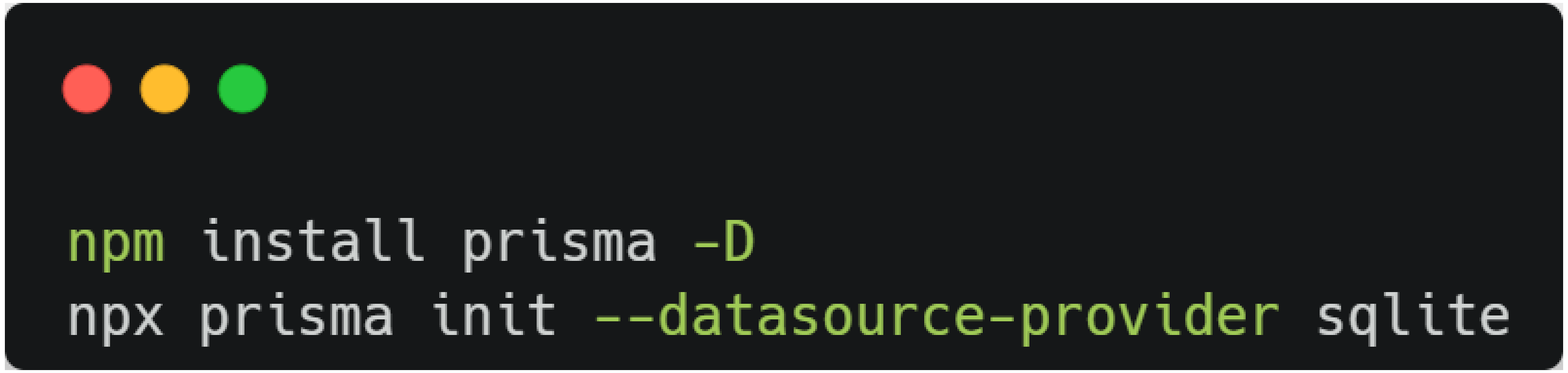
ORM

Next.js application

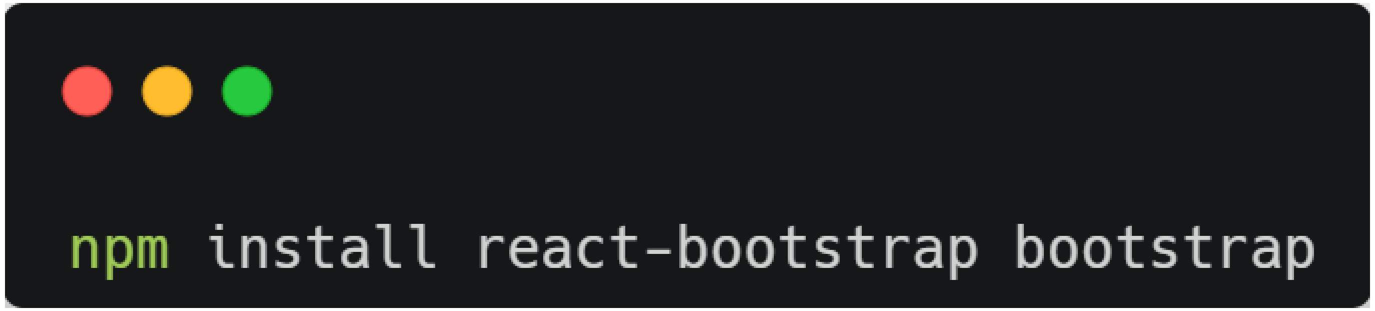
Bootstrap

Finnhub Stock API

Setup dependencies



```
npm install prisma -D
npx prisma init --datasource-provider sqlite
```



```
npm install react-bootstrap bootstrap
```

Live demonstration

Summary

What are the key takeaways?

Key aspects of Next.js

1

React framework

Fully-featured React framework for server-side web applications

2

Routing

File-based routing

API routes

3

Rendering

Static-site generation

Incremental static regeneration

Server-side rendering

Client-side data fetching

4

Additional features

Image optimization

Font optimization

Server build

Static HTML export

5

Integrability

Database integration

Style integration

API integration

Sources

- https://app.pluralsight.com/library/courses/nextjs-12-big-picture (Cory House, Access: 14.11.2022)
- https://app.pluralsight.com/library/courses/react-big-picture (Cory House, Access: 14.11.2022)
- Getting Started | Next.js (nextjs.org) (Vercel, Access: 14.11.2022)
- React – A JavaScript library for building user interfaces (reactjs.org) (Meta Platforms, Access: 14.11.2022)
- Create a New React App – React (reactjs.org) (Meta Platforms, Access: 14.11.2022)
- vercel/next.js: The React Framework (github.com) (Vercel, Access: 14.11.2022)
- ASCII Tree Generator (ascii-tree-generator.com) (ASCII Tree Generator 2022, Access: 14.11.2022)
- Quickstart with TypeScript & SQLite (prisma.io) (Prisma Data, Access: 14.11.2022)
- Next.js Database with Prisma | Next-Generation ORM for SQL Databases (Prisma Data, Access: 14.11.2022)
- API Documentation | Finnhub - Free APIs for realtime stock, forex, and cryptocurrency. Company fundamentals, economic data, and alternative data. (Finnhub.io, Access: 14.11.2022)
- https://getbootstrap.com/docs/5.2 (Bootstrap, Access: 14.11.2022)

Image Sources

- https://upload.wikimedia.org/wikipedia/commons/a/a7/React-icon.svg (React logo, Access: 14.11.2022)
- https://carbon.now.sh/ (Terminal pictures, Access: 14.11.2022)
- Icon_dark_background.png (180×180) (vercel.com) (Next.js logo, Access: 14.11.2022)
- https://raw.githubusercontent.com/babel/logo/master/babel.png (Babel, Access: 14.11.2022)
- https://raw.githubusercontent.com/swc-project/logo/master/swc.png (SWC Project, Access: 14.11.2022)
- https://webpack.js.org/site-logo.1fcab817090e78435061.svg (Webpack, Acess: 14.11.2022)
- https://nodejs.org/static/images/logo.svg (OpenJS Foundation, Access: 14.11.2022)
- https://nextjs.org/static/images/learn/foundations/user-interface.png (Vercel, Access: 14.11.2022)
- https://www.sqlite.org/images/sqlite370_banner.gif (SQLite Consortium, Access: 14.11.2022)
- https://website-v9.vercel.app/logo-dark.svg (Prisma Data, Access: 14.11.2022)
- https://getbootstrap.com/docs/5.2/assets/brand/bootstrap-logo-shadow.png (Bootstrap, Access: 14.11.2022)
- https://static.finnhub.io/img/finnhub_2020-05-09_20_51/logo/logo-gradient2.png (Finnhub.io, Access: 14.11.2022)

Thank you

Kevin Wiesner

4KevR - GitHub