

# FRUSTIERENDE WEB-ENTWICKLUNG

mit Fallbeispielen in React :)

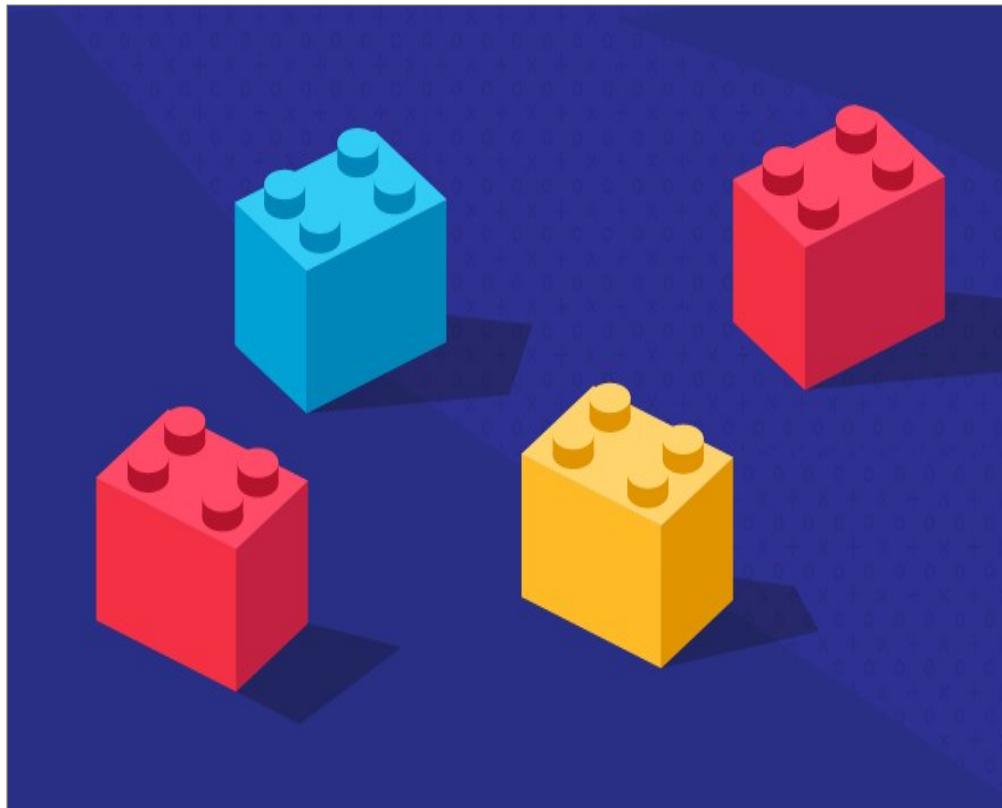
# ZU MIR

- Artur Zimmermann
- Master in Informatik (Juli 2019)
- Vollzeit bei Ambient (Mai 2019)
- Diverse Werkstudentenstellen (Galaria Kaufhof, Questback, SWYP GmbH, etc.)

# MOTIVATION FÜR VORTRAG

- Die selbe Melody, nur anderer Text
- "Wir wollen von React weg"
- Der Fehler ist nicht React exklusiv
- Alle Frameworks, Library, etc. sind Tools
- Anhand von React als Beispiel

# MIT REACT ENTWICKELT MAN WEB-KOMPONENTEN



<https://www.sitepen.com/blog/web-components-in-2018/>

und das alles schon seit 2013

# WEB-COMPONENTS

## NATIVE WEB-KOMPONENTEN BESTEHEN AUS:

- HTML Templates
- Custom Elements
- Shadow DOM

# HTML TEMPLATES

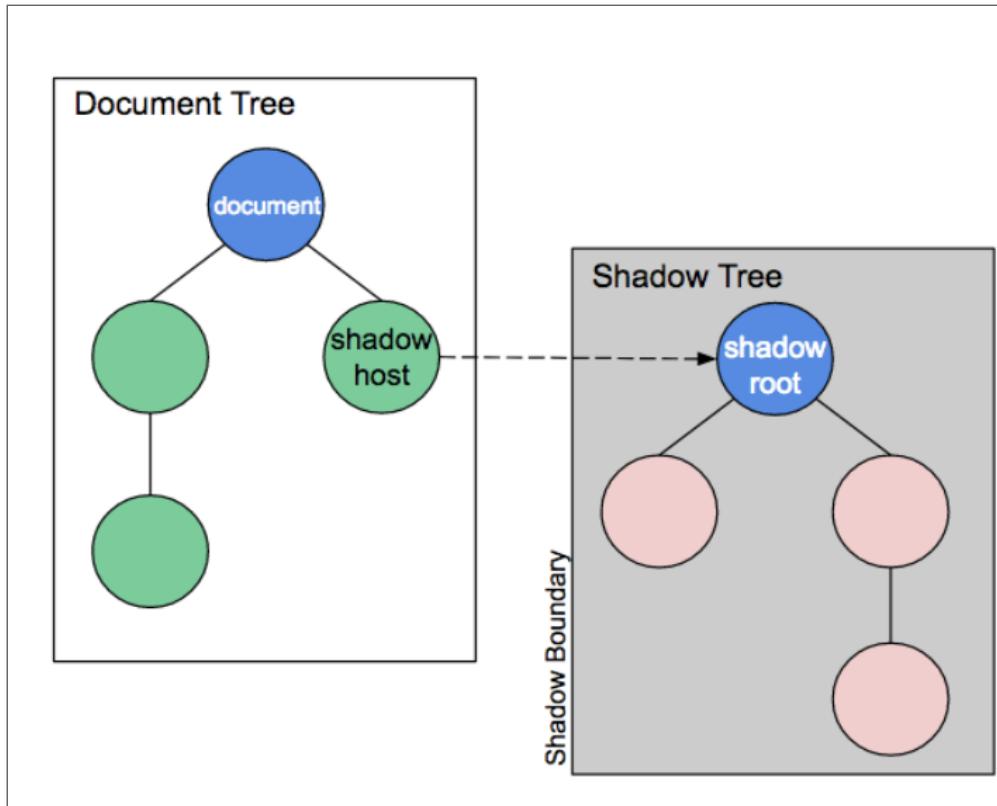
```
<template id="my-paragraph">
    <p>My paragraph</p>
</template>
```

# CUSTOM ELEMENTS

```
customElements.define('my-paragraph',
class extends HTMLElement {
  constructor() {
    super();
    let template = document.getElementById('my-paragraph');
    let templateContent = template.content;

    const shadowRoot = this.attachShadow({mode: 'open'})
      .appendChild(templateContent.cloneNode(true));
  }
})
```

# SHADOW DOM

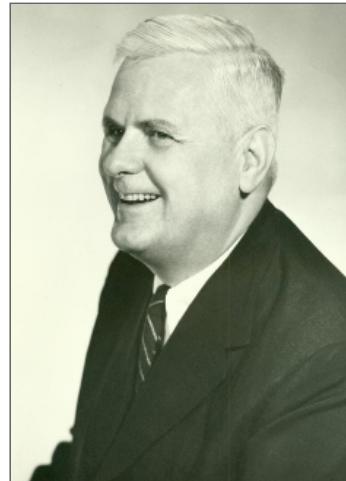
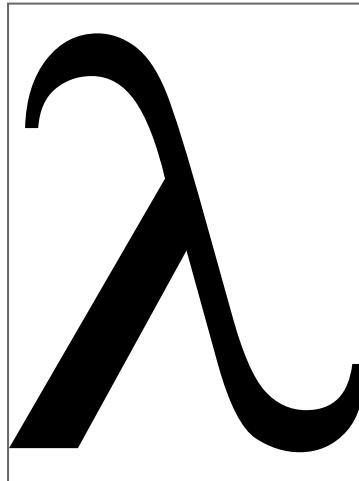


<https://medium.com/javascript-in-plain-english/understanding-the-shadow-dom-8495860b39bf>

Zur Kapselung von Styles und Funktionalität

# FUNKTIONALE PROGRAMMIERUNG

# DAS LAMBDA KALKÜL



<https://de.wikipedia.org/wiki/Lambda-Kalk%C3%BCl> [https://en.wikipedia.org/wiki/Alonzo\\_Church](https://en.wikipedia.org/wiki/Alonzo_Church)

# KEIN NÄHERES EINGEHEN AUF DETAILS

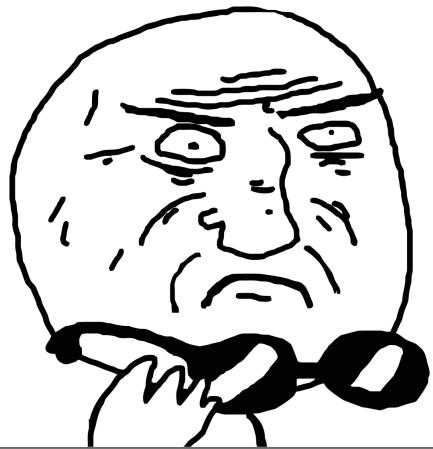
$$\text{true} = \lambda x. \lambda y. x$$

$$\text{false} = \lambda x. \lambda y. y$$

$$\begin{aligned} & (\lambda f. \lambda g. \lambda h. fg(h h))(\lambda x. \lambda y. x)h(\lambda x. xx) \\ \xrightarrow{\beta} & (\lambda g. \lambda h. (\lambda x. \lambda y. x)g(h h))h(\lambda x. xx) \quad (1) \\ \xrightarrow{\alpha} & (\lambda g. \lambda k. (\lambda x. \lambda y. x)g(k k))h(\lambda x. xx) \quad (2) \\ \xrightarrow{\beta} & (\lambda k. (\lambda x. \lambda y. x)h(k k))(\lambda x. xx) \quad (3) \\ \xrightarrow{\beta} & (\lambda x. \lambda y. x)h((\lambda x. xx)(\lambda x. xx)) \quad (4) \\ \xrightarrow{\beta} & (\lambda y. h)((\lambda x. xx)(\lambda x. xx)) \quad (5) \\ \xrightarrow{\beta} & h \quad (6) \end{aligned}$$

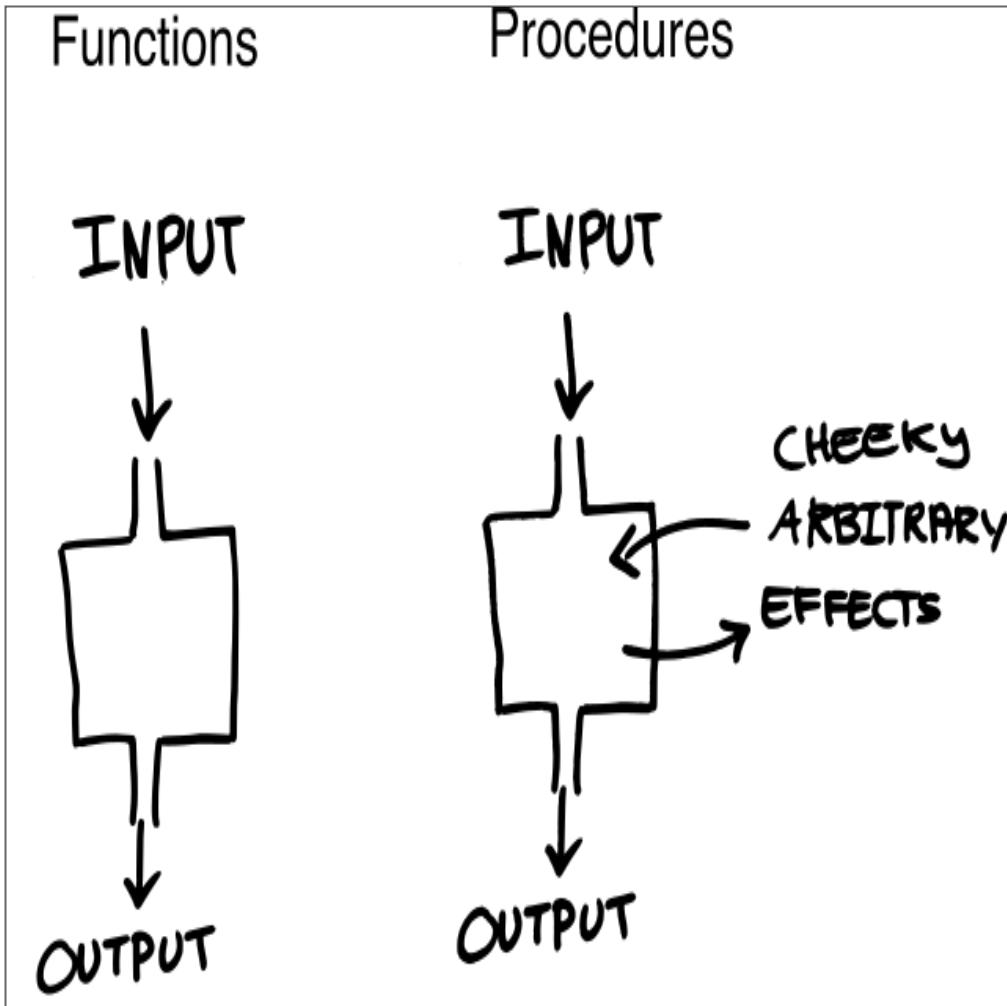
<https://www.slideshare.net/MahsaSeifikar/lambda-calculus-57240548>

MOTHER OF GOD...



# EINE "PURE" FUNKTION

$$f(x) = x + 3$$



```
const fac = (n) => {
    if (n === 0) return 1;
    if (n >= 1) return n * fac(n-1);
}
```

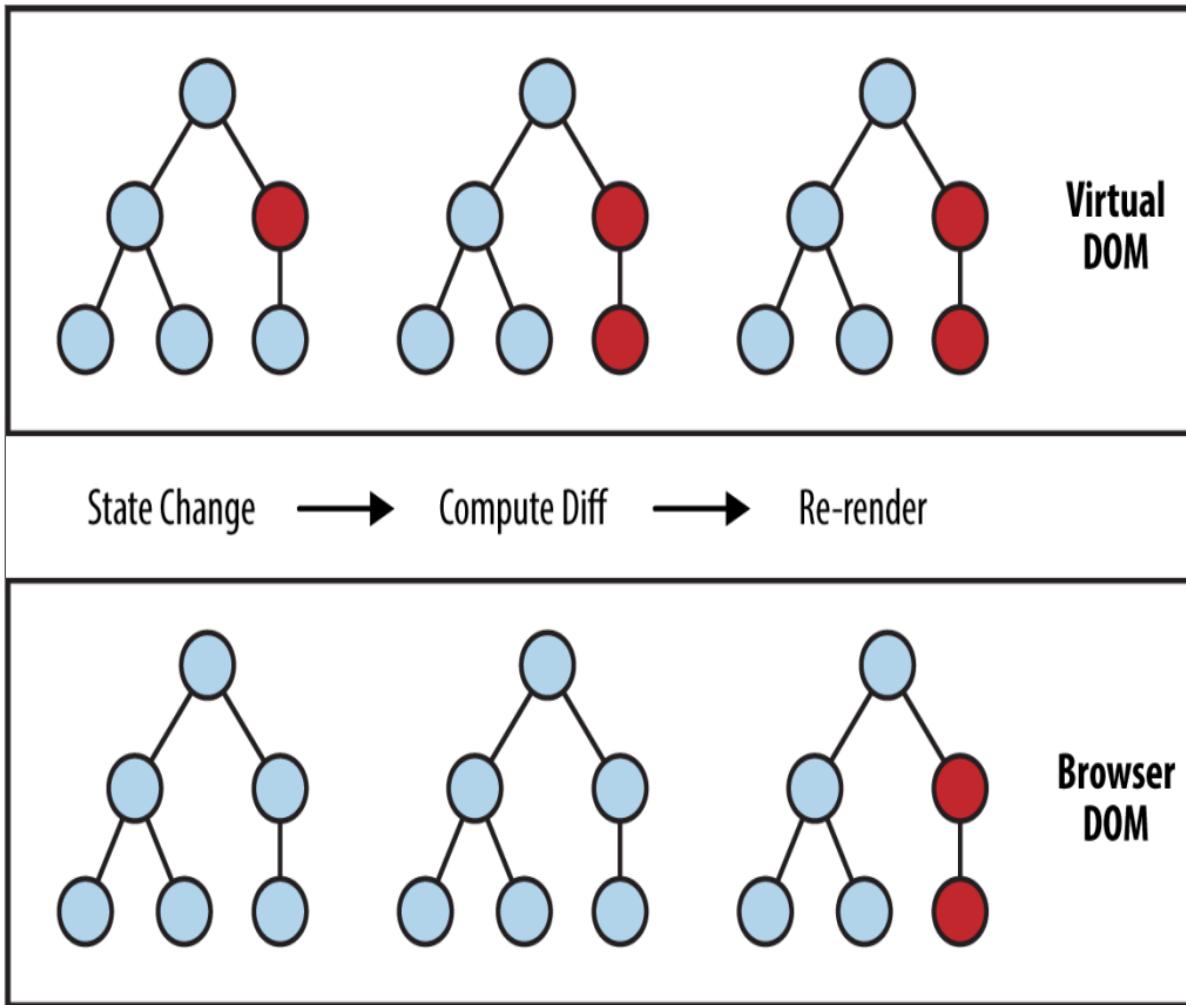
# REACT

# REACT IST EIN FRAMEWORK



The image shows the official React.js website homepage. The background is dark with a large, stylized, light-grayReact logo watermark. The word "React" is written in a bright blue, sans-serif font at the top center. Below it, the text "A JavaScript library for building user interfaces" is written in a smaller, gray, sans-serif font. At the bottom, there are two buttons: a blue "Get Started" button on the left and a smaller "Take the Tutorial >" link next to it.

# VIRTUAL DOM



# ERSTELLEN UND EINBINDEN VON REACT COMPONENTS

```
1 import React from "react";
2 import ReactDOM from "react-dom";
3 import "./styles.css";
4 const App = () => {
5   return (
6     <div className="App">
7       <h1>Hello CodeSandbox</h1>
8       <h2>Start editing to see some magic happen!</h2>
9     </div>
10   );
11 }
12 const rootElement = document.getElementById("root");
13 ReactDOM.render(<App />, rootElement);
```

## JSX: JAVASCRIPT MIT XML

```
<?xml version="1.0"?>
- <job>
  - <production>
    <ApprovalType>WebCenter</ApprovalType>
    <Substrate>carton 150 gr</Substrate>
    <SheetSize>220-140</SheetSize>
    <press>SuperFlat2</press>
    <finishing>standard</finishing>
    <urgency>normal</urgency>
  </production>
  - <customer>
    <name>FruitCo</name>
    <number>2712</number>
    <currency>USD</currency>
  </customer>
</job>
```

# ERGEBNISS IM ECHTEM DOM

```
▼ <Game>
  ▼ <div className="game">
    ▼ <div className="game-board">
      ▼ <Board>
        ▼ <div>
          <div className="status">Next player: X</div>
          ▼ <div className="board-row">
            ► <Square>...</Square>
            ► <Square>...</Square>
            ► <Square>...</Square>
          </div>
          ▼ <div className="board-row">
            ► <Square>...</Square>
            ► <Square>...</Square>
            ► <Square>...</Square>
          </div>
          ▼ <div className="board-row">
            ► <Square>...</Square>
            ► <Square>...</Square>
            </div>
          ▼ <div className="board-row">
            ► <Square>...</Square>
            ► <Square>...</Square>
            ► <Square>...</Square>
          </div>
        </div>
      </Board>
    </div>
    ▼ <div className="game-info">
      <div/>
      <ol/>
    </div>
  </div>
</Game>
```

# PROPS, ODER AUCH DER INPUT FÜR DIE FUNKTION

```
<Counter count=0 />
```

# DAS MÄCHTIGE PROP-OBJECT

```
import React, { useState } from 'react';

export const Counter = ( props ) => {

  return (
    <div>
      <p>{this.props.count}</p>
    </div>
  );
}
```

Alle datentypen in JavaScript können als Prop verwendet werden

# CHILDREN, DAS BESONDERE PROP-OBJECT

```
import React from 'react';

export const Counter = ( props ) => {

  return (
    <div>
      <div>{this.props.children}</div>
    </div>
  );
}
```

# VERWENDUNG VON "CHILDREN"

```
<Counter>
  <ChildComponent>...</ChildComponent>
  <ChildComponent>...</ChildComponent>
  <ChildComponent>...</ChildComponent>
</Counter>
```

# STATES UND CONDITIONAL RENDERING

- Props sind Immutable
- Man möchte Daten anpassen innerhalb einer Komponente
- Häufig um Styles anzupassen

## CODE BEISPIEL:

```
import React, { useState } from 'react';

export const Counter = (props) => {
  const [count, setCount] = useState(this.props.count);

  return (
    <div>
      {count >= 1 ? <p>You clicked {count} times</p> : ""}
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

# WEITERE HOOKS

## useEffect()

```
import React, { useEffect } from 'react';

...
useEffect(() => {
  // Update the document title using the browser API
  document.title = `You clicked ${count} times`;
});
...
```

## useMemo()

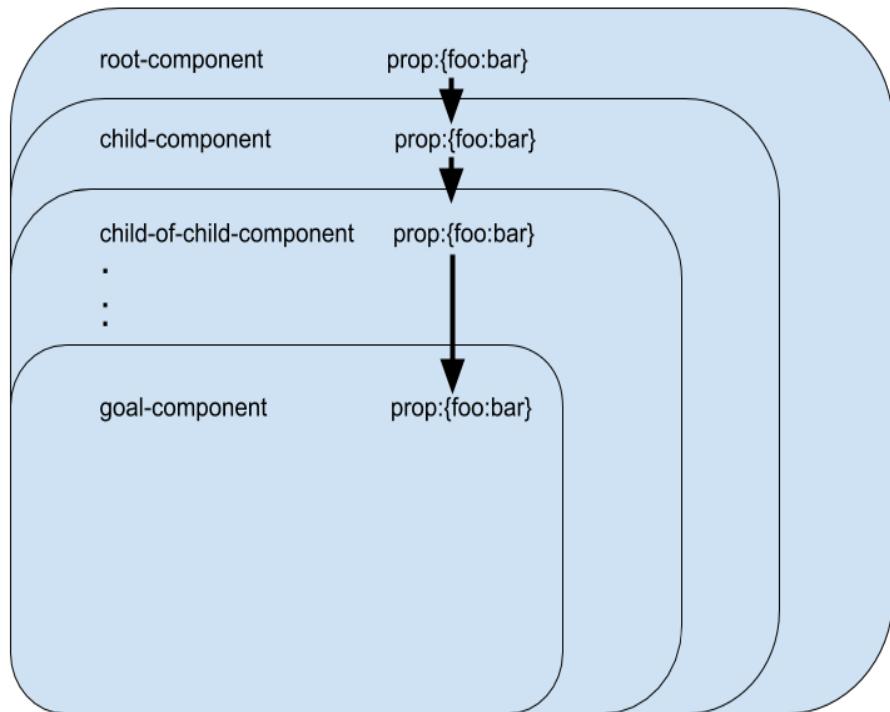
```
import React, { useMemo } from 'react';

...
const myMemo = useMemo(() => computeExpensiveValue(a, b), [a, b]);
...
```

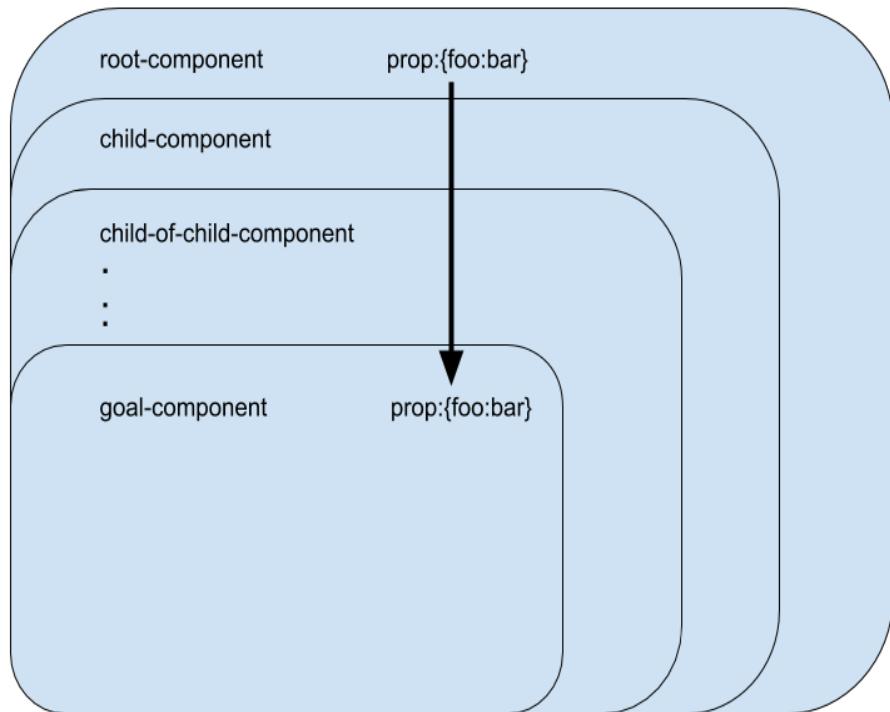
# CONTEXT API

## DIE RETTUNG VON PROP-DRILLING

# WAS IST PROP-DRILLING?



# LÖSUNG: GEZIELT PROPS AN KOMPONENTE ÜBERGEBEN



## CODE BEISPIELE:

```
const themes = {
  light: {
    foreground: "#000000",
    background: "#eeeeee"
  },
  dark: {
    foreground: "#ffffff",
    background: "#222222"
  }
};

const ThemeContext = React.createContext(themes.light);

function App() {
  return (
    <ThemeContext.Provider value={themes.dark}>
      <Toolbar />
    </ThemeContext.Provider>
  );
}

function Toolbar(props) {
  return (
    <div>
      <ThemedButton />
    </div>
  );
}

function ThemedButton() {
  const theme = useContext(ThemeContext);

  return (
    <button style={{ background: theme.background, color: theme.foreground }}>
      I am styled by theme context!
    </button>
  );
}
```

# WIE STYLED MAN EINE KOMPONENTE

# STYLE OBJEKT IN DER COMPONENTE

```
...
  const mystyle = {
    color: "white",
    backgroundColor: "DodgerBlue",
    padding: "10px",
    fontFamily: "Arial"
  };
  return (
    <div>
      <h1 style={mystyle}>Hello Style!</h1>
      <p>Add a little style!</p>
    </div>
  );
...
}
```

# STYLE ÜBER EINE CSS KLASSE

Definiere ein CSS Klasse

Importiere die CSS-Datei

Verwende className statt class

```
...
import "./styles.css";
const App = () => {
  return (
    <div className="App">
      <h1>Hello CodeSandbox</h1>
      <h2>Start editing to see some magic happen!</h2>
    </div>
  );
...
}
```

## ZUSAMMENGEFASST KANN MAN SAGEN:

- React ist eine Erweiterung des Web-Browsers und seiner API
- Es ist reaktive bei Änderung
- Deswegen verhält es sich eigentlich wie ein Framework
- Ist aber eine Library

# DER FRUSTIERENDE TEIL

# WORUM SOLL ES GEHEN?

- Die Komponente Mensch
- Overengineering, Overrefactoring, Overeverything
- Mythen werden zu Wahrheiten
- ...es wird kein holen auf einzelne

## SIE WERDEN NICHT ALLEINE ARBEITEN

- Entweder in einem Team
- Im Kontakt mit Kunden/Stakeholdern

## HÄUFIG FÄLLEN KUNDEN ENTSCHEIDUNGEN, WEIL...

- ...Sie denken sie müsse die Entscheidung treffen
- ...Sie mal gehört haben dass das (Technologie) total im trend wäre
- ...Sie auf diejenigen hören die es besser verkaufen können

## IHRE KOLLEGEN WERDEN...

- ...ihre eigene Karriere haben
- ...eher nach deren Meinung gefragt werden
- ...nicht auf schlüssige Argumente hören

# **WENN SIE NACH EINER STELLE SUCHEN, WORAUF ACHTEN SIE?**

## AUF XING FINDET MAN DANN HÄUFIG FOLGENDES:

- Erfahrung in [Liste von Frameworks]
- Leidenschaft für Design
- Oder, Sie sind Experte in [beliebges Framework]

*“Wir suchen jemanden mit Erfahrung in, HTML, CSS, JavaScript, Less, Sass, Angular, React, Vue.js, Next.js, Node, XML, JSON, JWT, Drupal, CMS, WordPress, Fetch und Cypress”*

## "EXPERTE"

- Anekdot: Der Koch der nur Kartoffeln kann



- Quereinsteiger die nur React o.ä. gelernt haben

## TECHNOLOGIE STACKS WERDEN GRÖSSER

*“because frontend developers identify personally with the frameworks they use, you can’t just impose a single framework - like React or Vue - across an entire enterprise. They’ll be sharpening their pitchforks and chasing you out of town. That’s where Web Components come in. Because they can work with any framework, a central UX or design team can standardize on a single set of fully-custom, fully-functional UI components, and share them throughout their entire org, without having to persuade any teams to give up their religion.”*

HTML, CSS, JavaScript

Node

React

TypeScript

Emotion (Styled Component Library)

UI-Kit Antd

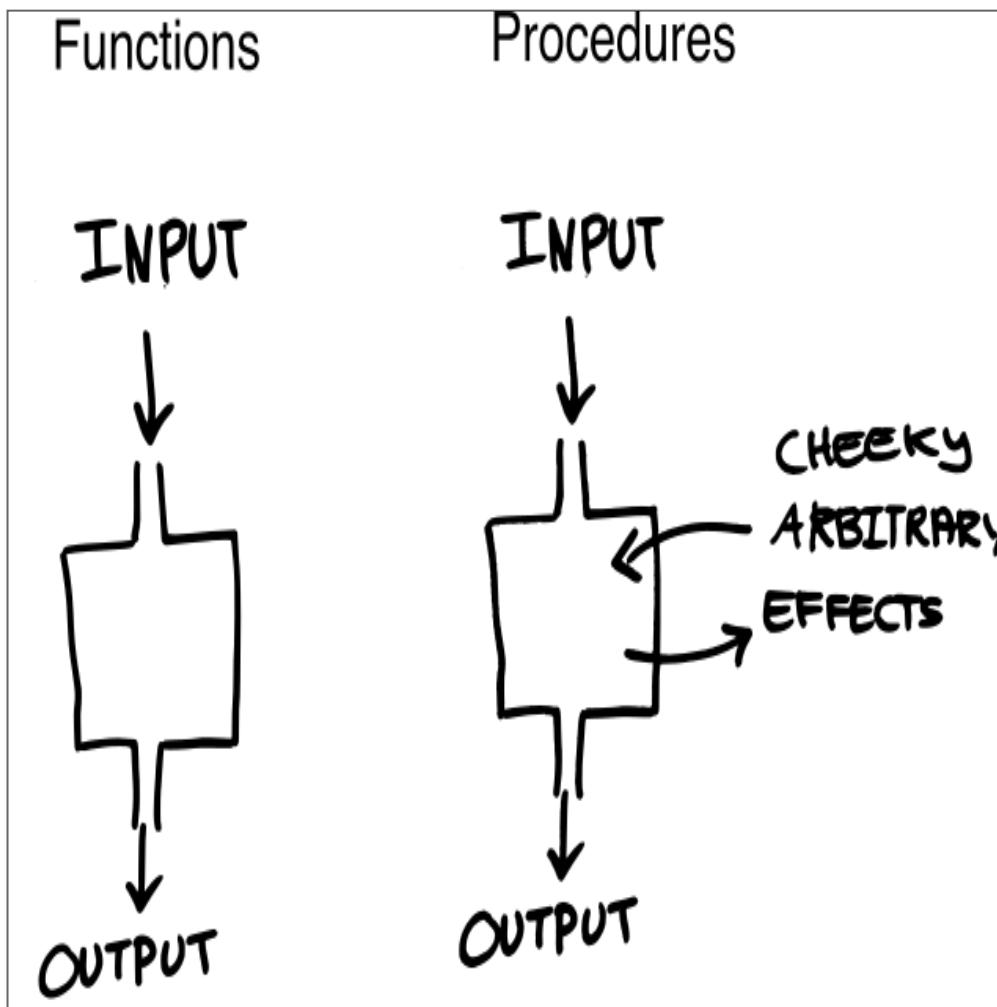
Gatsby

# PROBLEME WEGEN DER KOMBIATION

# MAN BAUT FIESSE HACKS

```
const ResponseButton = styled(({ children, ...props }) => (
  <Button {...props}>{children}</Button>
))(props => ({
  margin: "0.7em",
  width: "10em",
  backgroundColor: props.buttoncolor,
  color: colors.white
}));
```

# ES WIRD OOP VERWENDET OBWOHL MAN FUNKTIONAL ENTWICKELN SOLLTE



```
const fac = (n) => {
    if (typeof n == string) {
        if (n < 0) {
            if (n === 0) return 1;
            if (n >= 1) return n * fac(n-1);
        } else {
            return "Input should be not smaller then 0"
        }
    } else {
        return "Input should be not a string"
    }
}
```

```
const fac = (n) => {
    if (n === 0) return 1;
    if (n >= 1) return n * fac(n-1);
}
```

Home > Conferences > ICFP > Proceedings > ERLANG '03 > Evaluating distributed functional languages for telecommunications software

ARTICLE

# Evaluating distributed functional languages for telecommunications software



**Authors:** J. H. Nyström, P. W. Trinder, D. J. King [Authors Info & Affiliations](#)

**Publication:** ERLANG '03: Proceedings of the 2003 ACM SIGPLAN workshop on Erlang • August 2003 • Pages 1–7 • <https://doi.org/10.1145/940880.940881>

7 439



Get Access

ERLANG '03:  
Proceedings of the...

Evaluating distributed  
functional languages...

Pages 1–7

← Previous

Next →

ABSTRACT

References

Comments

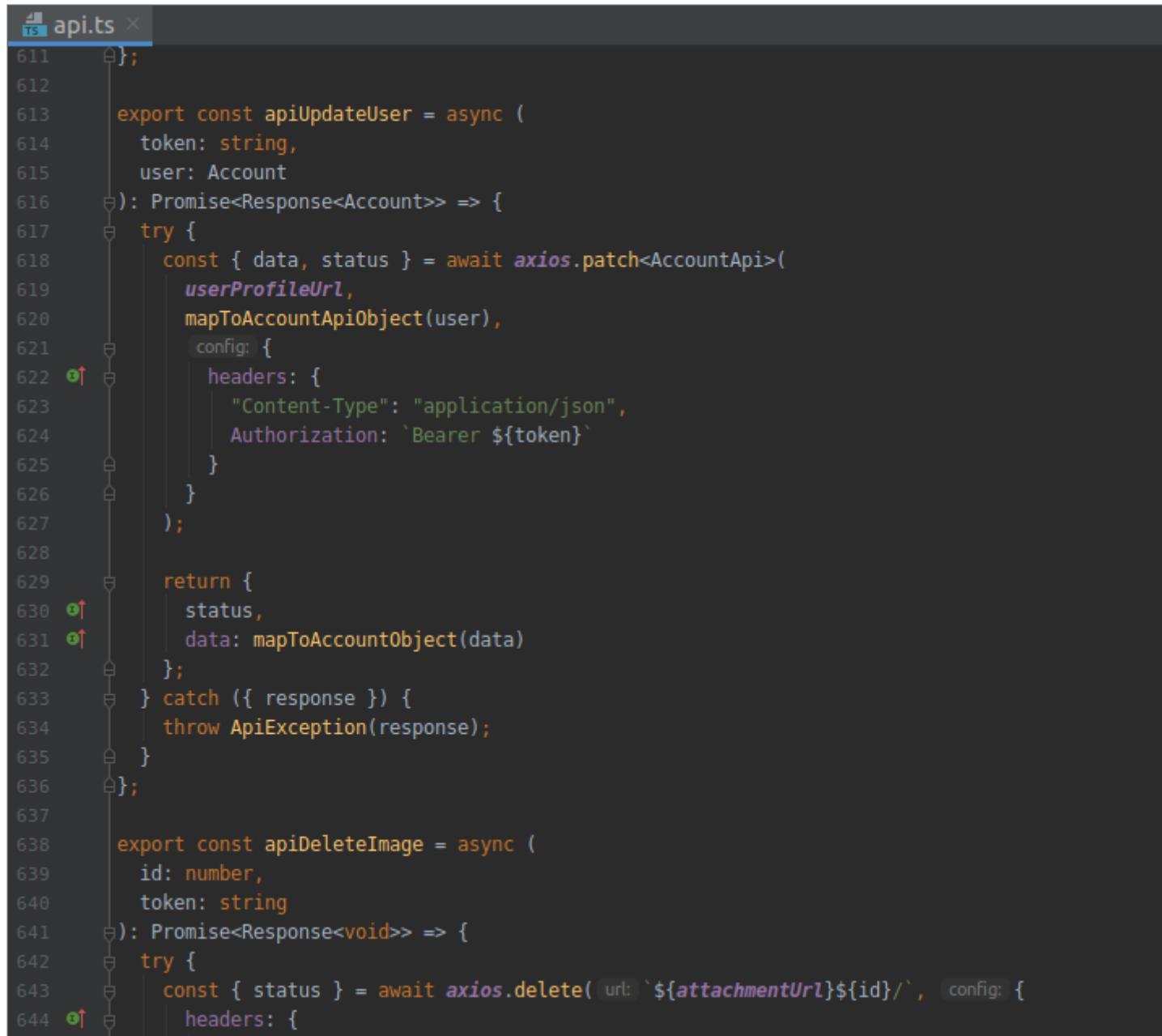


## ABSTRACT

The distributed telecommunications sector not only requires minimal time to market, but also software that is reliable, available, maintainable and scalable. High level programming languages have the potential to reduce development time and improve maintainability due to their compact code size. Moreover reliability is improved by safe type systems and relatively easy verification. This paper outlines plans and initial results from a joint project between Motorola and Heriot-Watt University that aims to evaluate the suitability of distributed functional languages for constructing telecommunications software. The evaluation will use the ERLANG and Glasgow distributed Haskell(GdH) languages, and be based on the construction of several typical applications. The evaluation will focus on reliability issues like ~~area of verification, availability issues like fault tolerance or resilience, as well as whether the~~

# **WEITERE EIGENARTEN VON OOP**

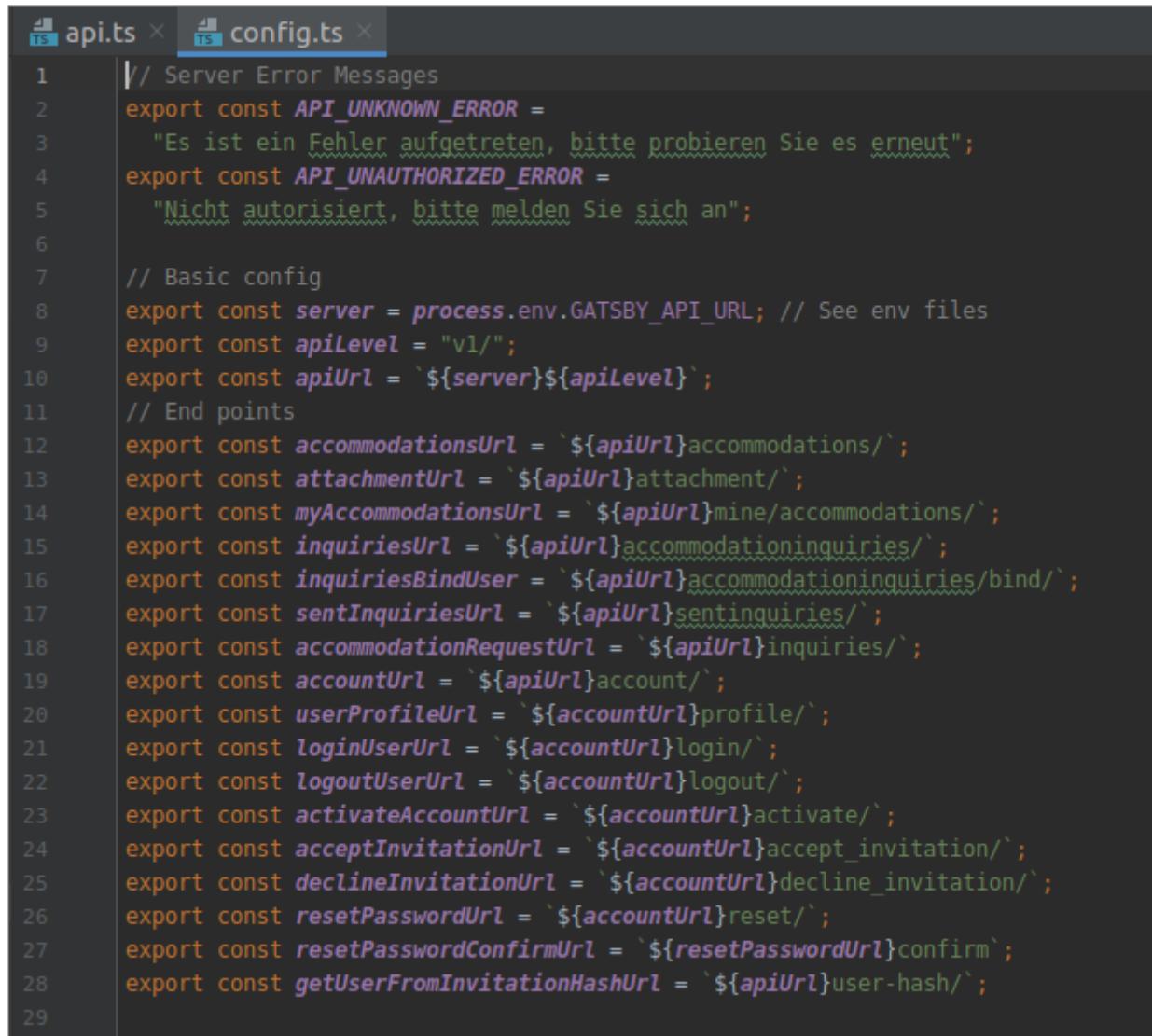
## **ZUM BEISPIEL: MODULARIESIERUNG VON CODE**



The screenshot shows a code editor window with a dark theme. The file is named `api.ts`. The code defines two asynchronous functions: `apiUpdateUser` and `apiDeleteImage`.

```
api.ts
611     };
612
613     export const apiUpdateUser = async (
614         token: string,
615         user: Account
616     ): Promise<Response<Account>> => {
617         try {
618             const { data, status } = await axios.patch<AccountApi>(
619                 userProfileUrl,
620                 mapToAccountApiObject(user),
621                 config: {
622                     headers: {
623                         "Content-Type": "application/json",
624                         Authorization: `Bearer ${token}`
625                     }
626                 }
627             );
628
629             return {
630                 status,
631                 data: mapToAccountObject(data)
632             };
633         } catch ({ response }) {
634             throw ApiException(response);
635         }
636     };
637
638     export const apiDeleteImage = async (
639         id: number,
640         token: string
641     ): Promise<Response<void>> => {
642         try {
643             const { status } = await axios.delete(`url: ${attachmentUrl}${id}/`, config: {
644                 headers: {
```

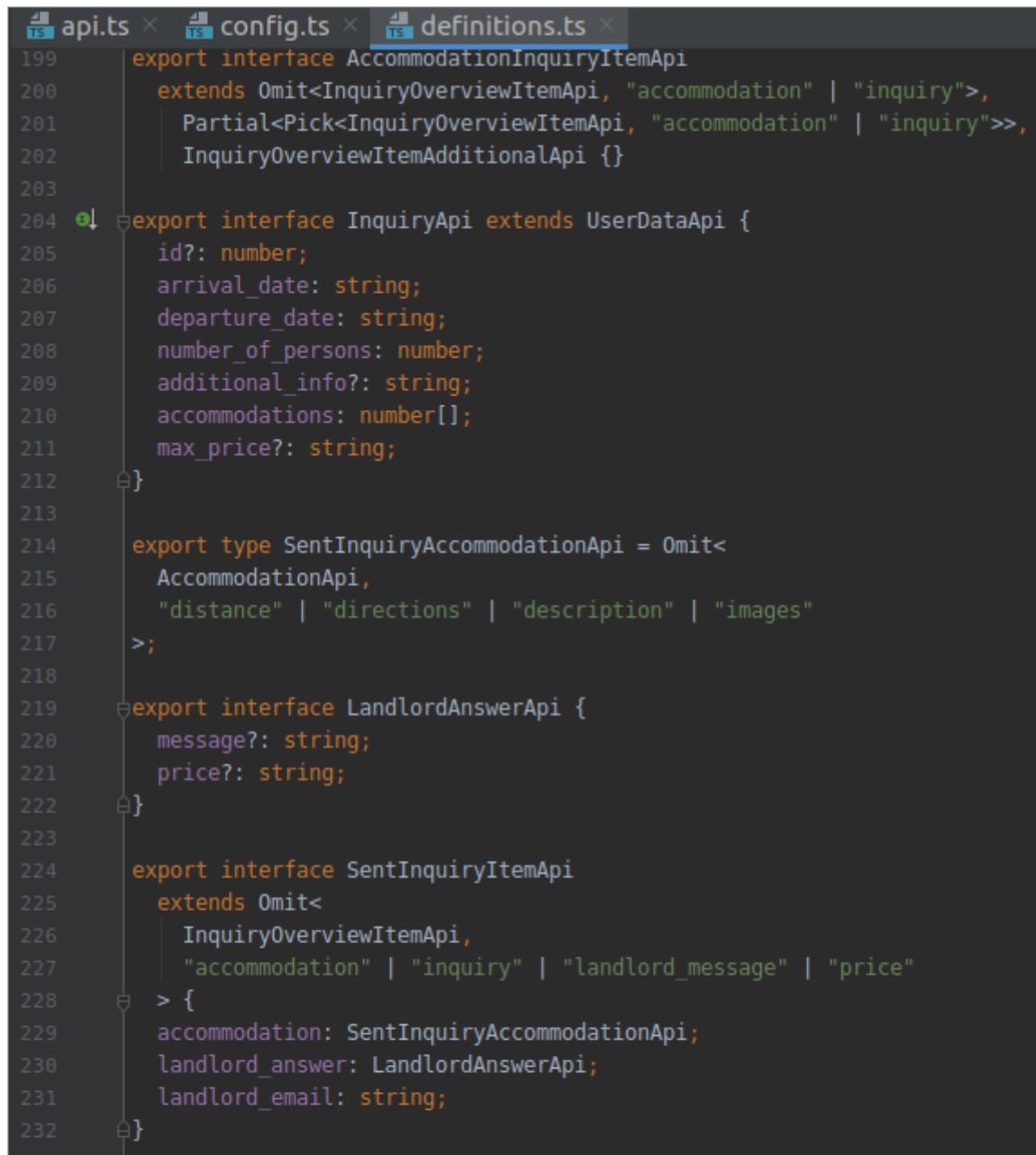
```
645         "Content-Type": "application/json",
646         Authorization: `Bearer ${token}`
647     }
648 );
649
650     return {
651     ↑   status,
652     ↑   data: undefined
653     };
654 } catch ({ response }) {
655     throw ApiException(response);
656 }
657 };
658
```



The screenshot shows a code editor with two tabs open: 'api.ts' and 'config.ts'. The 'api.ts' tab is active, displaying the following TypeScript code:

```
// Server Error Messages
export const API_UNKNOWN_ERROR =
  "Es ist ein Fehler aufgetreten, bitte probieren Sie es erneut";
export const API_UNAUTHORIZED_ERROR =
  "Nicht autorisiert, bitte melden Sie sich an";

// Basic config
export const server = process.env.GATSBY_API_URL; // See env files
export const apiLevel = "v1/";
export const apiUrl = `${server}${apiLevel}`;
// End points
export const accommodationsUrl = `${apiUrl}accommodations/`;
export const attachmentUrl = `${apiUrl}attachment/`;
export const myAccommodationsUrl = `${apiUrl}mine/accommodations/`;
export const inquiriesUrl = `${apiUrl}accommodationinquiries/`;
export const inquiriesBindUser = `${apiUrl}accommodationinquiries/bind/`;
export const sentInquiriesUrl = `${apiUrl}sentinquiries/`;
export const accommodationRequestUrl = `${apiUrl}inquiries/`;
export const accountUrl = `${apiUrl}account/`;
export const userProfileUrl = `${accountUrl}profile/`;
export const loginUserUrl = `${accountUrl}login/`;
export const logoutUserUrl = `${accountUrl}logout/`;
export const activateAccountUrl = `${accountUrl}activate/`;
export const acceptInvitationUrl = `${accountUrl}accept_invitation/`;
export const declineInvitationUrl = `${accountUrl}decline_invitation/`;
export const resetPasswordUrl = `${accountUrl}reset/`;
export const resetPasswordConfirmUrl = `${resetPasswordUrl}confirm`;
export const getUserFromInvitationHashUrl = `${apiUrl}user-hash/`;
```



The screenshot shows a code editor with three tabs open: `api.ts`, `config.ts`, and `definitions.ts`. The `definitions.ts` tab is active, displaying the following TypeScript code:

```
199 export interface AccommodationInquiryItemApi
200   extends Omit<InquiryOverviewItemApi, "accommodation" | "inquiry">,
201   Partial<Pick<InquiryOverviewItemApi, "accommodation" | "inquiry">>,
202   InquiryOverviewItemAdditionalApi {}
203
204 export interface InquiryApi extends UserDataApi {
205   id?: number;
206   arrival_date: string;
207   departure_date: string;
208   number_of_persons: number;
209   additional_info?: string;
210   accommodations: number[];
211   max_price?: string;
212 }
213
214 export type SentInquiryAccommodationApi = Omit<
215   AccommodationApi,
216   "distance" | "directions" | "description" | "images"
217 >;
218
219 export interface LandlordAnswerApi {
220   message?: string;
221   price?: string;
222 }
223
224 export interface SentInquiryItemApi
225   extends Omit<
226     InquiryOverviewItemApi,
227     "accommodation" | "inquiry" | "landlord_message" | "price"
228   > {
229   accommodation: SentInquiryAccommodationApi;
230   landlord_answer: LandlordAnswerApi;
231   landlord_email: string;
232 }
```

```
233
234     export interface SentInquiryMainApi extends InquiryApi {
235         created_at: string;
236     }
237
238     export interface SentInquiryApi {
239         main: SentInquiryMainApi;
240         sub: SentInquiryItemApi[];
241     }
242 }
```

```
api.ts x config.ts x definitions.ts x utils.ts x
500     landlord_email : string ,
501     accommodation : SentInquiryAccommodationApi ,
502     status : string ,
503     ...rest : Pick<SentInquiryItemApi, "id" | "created_at" | "number_of_persons">
504   ): SentInquiryItemApi): SentInquiryItem => ({
505     ...
506     ↑ accommodation: maptoSentAccommodationObject(accommodation),
507     ↑ landlordAnswer: maptoLandlordAnswerObject(landlord_answer),
508     ↑ landlordEmail: landlord_email,
509     ↑ status: status as InquiryStatus
510   });
511
512   export const maptoSentInquiryItemObject = ({
513     landlordAnswer: LandlordAnswer ,
514     landlordEmail: string ,
515     accommodation: SentInquiryAccommodation ,
516     ...rest : Pick<SentInquiryItem, "id" | "status" | "createdAt" | "numberOfPersons">
517   ): SentInquiryItem): SentInquiryItemApi => ({
518     ...
519     ↑ accommodation: maptoSentAccommodationApiObject(accommodation),
520     ↑ landlord_answer: maptoLandlordAnswerApiObject(landlordAnswer),
521     ↑ landlord_email: landlordEmail
522   });
523
524   export const maptoSentInquiryObject = ({
525     main: SentInquiryMainApi ,
526     sub: SentInquiryItemApi[],
527     ...
528   ): SentInquiryApi): SentInquiry => ({
529     ...
530     ↑ main: maptoSentInquiryMainObject(main),
531     ↑ sub: sub.map(maptoSentInquiryItemObject)
532   });
533
534   export const maptoSentInquiryApiObject = ({

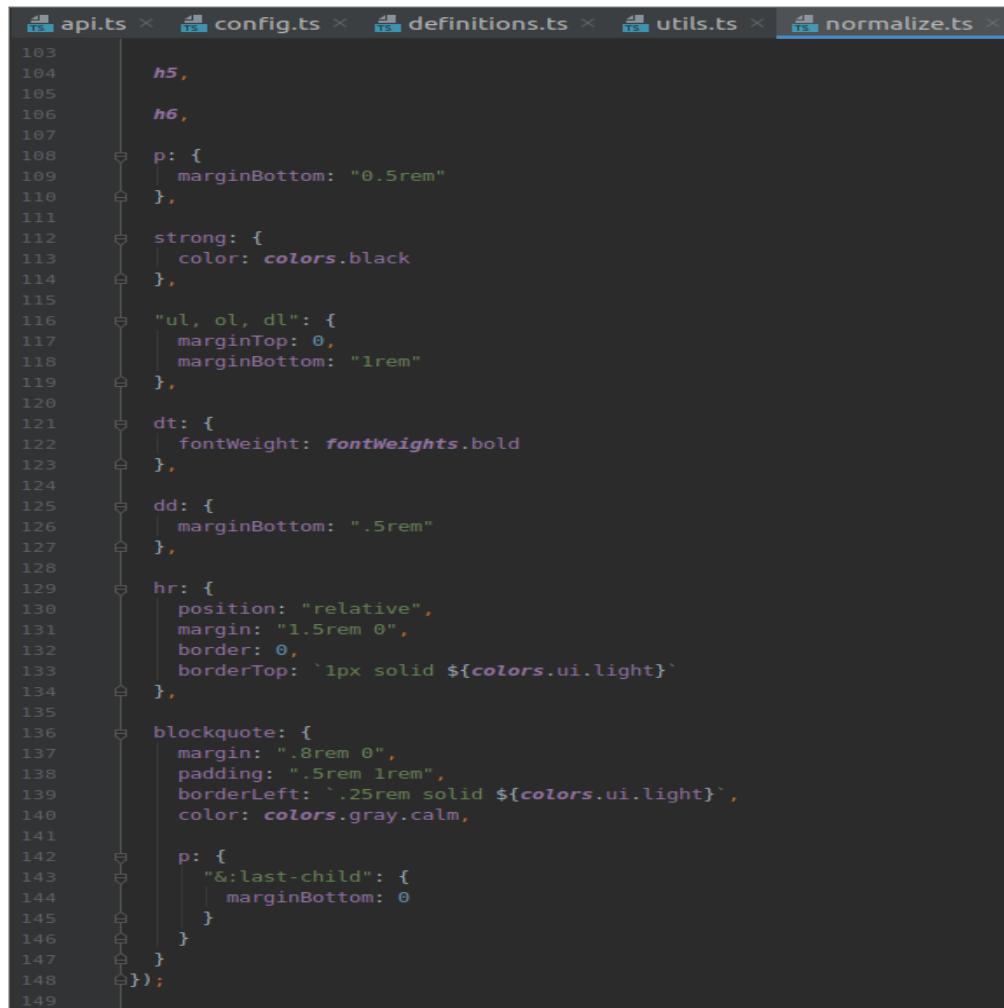

```

```
534  export const mapPresentInquiryApiObject = ({  
535    main : SentInquiryMain ,  
536    sub : SentInquiryItem[] ,  
537    ...rest  
538  }: SentInquiry): SentInquiryApi => ({  
539    ...rest,  
540    main: mapToSentInquiryMainApiObject(main),  
541    sub: sub.map(mapToSentInquiryItemApiObject)  
542  });
543
```

## UND JETZT, WAS GANZ BESONDERES

- Emotion für Styled Components
- Verwendet die Objekt-Notation damit man Typscript verwenden kann
- Erkennt das man in vielen Kompoenten die gleichen Styled Components verwendet

# WAS HAT MAN ALSO GEMACHT?



```
api.ts × config.ts × definitions.ts × utils.ts × normalize.ts ×
103
104     h5,
105
106     h6,
107
108     p: {
109         marginBottom: "0.5rem"
110     },
111
112     strong: {
113         color: colors.black
114     },
115
116     "ul, ol, dl": {
117         marginTop: 0,
118         marginBottom: "1rem"
119     },
120
121     dt: {
122         fontWeight: $fontWeights.bold
123     },
124
125     dd: {
126         marginBottom: ".5rem"
127     },
128
129     hr: {
130         position: "relative",
131         margin: "1.5rem 0",
132         border: 0,
133         borderTop: `1px solid ${colors.ui.light}`
134     },
135
136     blockquote: {
137         margin: ".8rem 0",
138         padding: ".5rem 1rem",
139         borderLeft: ".25rem solid ${colors.ui.light}",
140         color: colors.gray.calm,
141
142         p: {
143             "&:last-child": {
144                 marginBottom: 0
145             }
146         }
147     }
148 });
149
```



# **DER TREND GEHT ZU WENIGER FRAMEWORK MEHR VANILLA**

[Star the repo!](#) 245



The **Frameworkless Movement** is a group of developers interested in developing applications without frameworks. **We don't hate frameworks**, nor we will ever create campaigns against frameworks, but we perceive the **misuse of frameworks as a lack of knowledge regarding technical debt** and the availability of alternatives given by the vanilla language or by dedicated libraries.

## Why the Movement

Every time a team uses a framework, it also takes a **risk**. The risk is that after some time has passed the team ends up with a tool that does not provide any kind of value anymore and that most of the times represents a major roadblock to change. Most importantly a framework could "die" way before the software that uses it, leaving the developers with a heavy burden.

This risk is amplified without the presence of a strong bond between technical decision making, business goals, and user experience. Instead, the non-functional requirements like deadline, lifespan, budget, usability, future business scenarios and domain-specific constraints should be the primary decision drivers for the architectural choices as well as for the implementation roadmap.

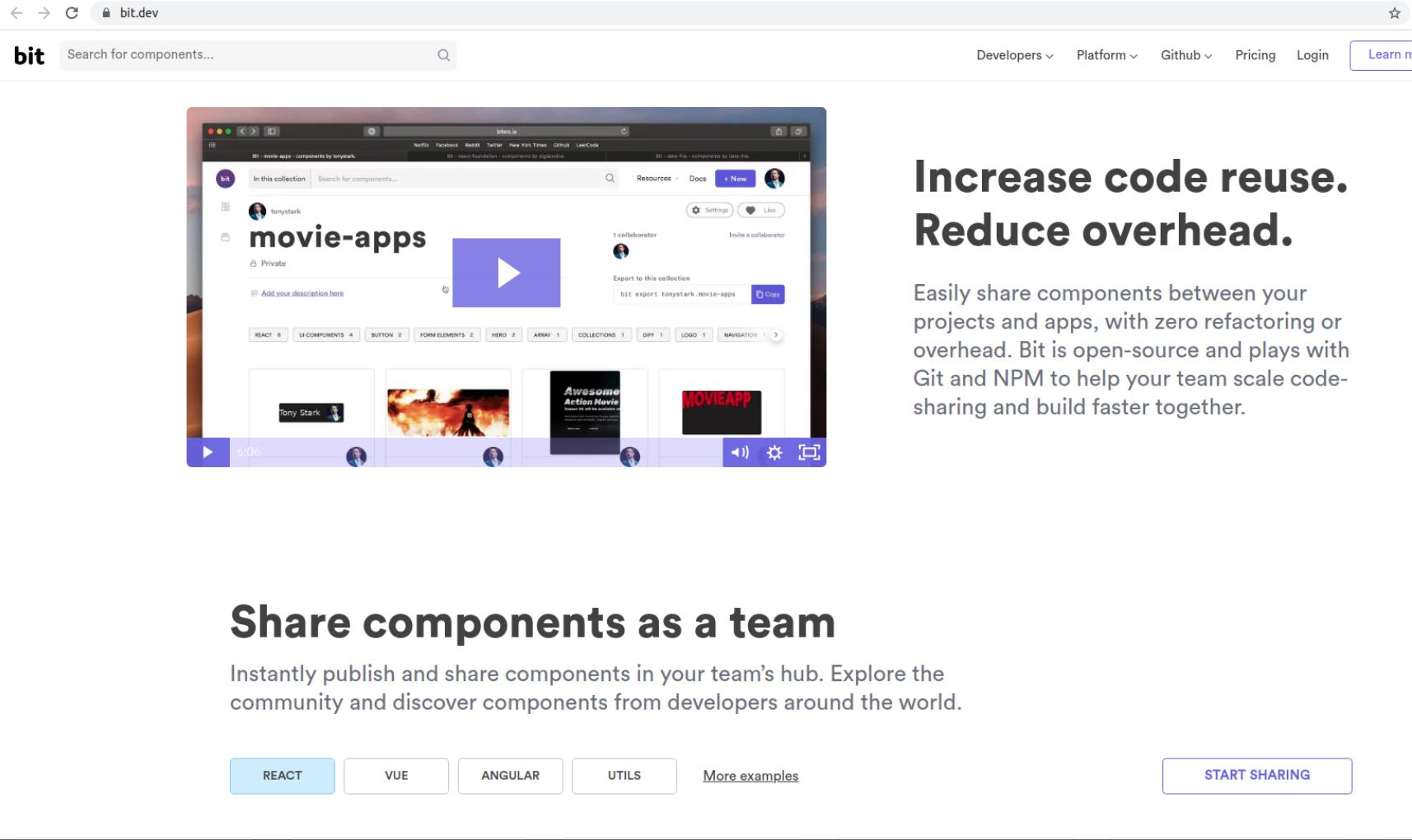
## What Frameworkless Movement is

Like we stated above, **we don't think that frameworks are evil**. They are very powerful tools, written in plain vanilla language and in a very generic way, a fact that makes them great learning tools which can help anyone learn how to code **without them**. In order to know how and when you should use a particular framework you have to understand how frameworks work. Understanding the strengths and the constraints of each framework will enable you to make the right choice of a framework. Even more, it will be very clear when you don't need one at all. Grasping the concepts of how frameworks work behind the scenes will deepen your knowledge about the language itself. Knowing the framework's rationale will enable you to write better code for each specific problem you encounter.

Nonetheless, we believe that a framework, when chosen, should be used responsibly throughout the life of the project. Continuously remind yourself that every tool has some kind of tradeoff.

Frameworkless means that you must give the right importance to the technological decision making and realize that the choice to develop a project or a single feature with no





The screenshot shows the bit.dev website interface. At the top, there's a navigation bar with links for Developers, Platform, Github, Pricing, and Login. A search bar is also present. Below the header, a video player window displays a component library titled "movie-apps". The video player has a play button, a progress bar showing 6:06, and a controls bar at the bottom. The main content area shows a collection of components, including a "Tony Stark" card, a "Action Movie" card, and a "MOVIEAPP" card. Below the video player, there's a section titled "Share components as a team" with a sub-section about publishing components. At the bottom, there are buttons for REACT, VUE, ANGULAR, UTILS, and More examples, along with a large "START SHARING" button.

## Increase code reuse. Reduce overhead.

Easily share components between your projects and apps, with zero refactoring or overhead. Bit is open-source and plays with Git and NPM to help your team scale code-sharing and build faster together.

## Share components as a team

Instantly publish and share components in your team's hub. Explore the community and discover components from developers around the world.

REACT   VUE   ANGULAR   UTILS   [More examples](#)

[START SHARING](#)

The screenshot shows the homepage of [webcomponents.dev](https://webcomponents.dev). The main heading is "All-in-one Web Component Platform". Below it, there's a list of features:

- ✓ Code your components in isolation
- ✓ Create Collections and Design Systems
- ✓ Use them everywhere

Two buttons are present: "Try now" and "Learn more". A sidebar on the left says "Follow us to get the latest news!" with icons for GitHub, LinkedIn, and Twitter.

A large screenshot on the right shows a code editor with a component named "atomic-clock" and a preview window displaying a 3D wireframe clock.

Below the main section, the text "Try now with the library of your choice" is displayed.

**NATIVE CUSTOM ELEMENT**

Icons for "JavaScript" and "+ LighterHTML".

**LIBRARIES AND COMPILERS**

Icons for Riot, Omi, Stencil, LitElement, LitElement TS, LWC SFC, LWC, Atomico, Haunted, Heresy, Hybrids, HyperHTML, Litedom, Ottavino, SkateJS, and SlimJS.

**MAJOR FRAMEWORKS WRAPPED INTO CUSTOM-ELEMENTS**

Icons for Angular, Preact X, React, Svelte 3, and Vue.js.

blog.bitsrc.io/11-must-know-frontend-trends-for-2020-cea8a629b08

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#). [X](#)

Sign in [Get started](#)

 Bits and Pieces WRITE BIT BLOG JAVASCRIPT WEBDEV REACT ANGULAR VUE | BUILD A COMPONENT LIBRARY →

Enjoy!

**Bits and Pieces**  
Scale frontend ·  
Build and deploy  
components ·  
Compose UIs  
together ·  
<https://bit.dev> ↗

[Follow](#)

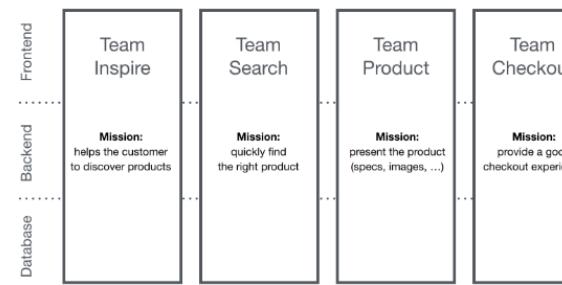
1.5K

1. Micro frontends

Micro Frontends are the buzziest frontend topic for lunch conversations.

Ironically, while frontend development enjoys the modular advantages of components, it is still largely more monolithic than backend microservices.

### End-to-End Teams with Micro Frontends



Frontend

Backend

Database

Team Inspire  
Mission: helps the customer to discover products

Team Search  
Mission: quickly find the right product

Team Product  
Mission: present the product (specs, images, ...)

Team Checkout  
Mission: provide a good checkout experience

Micro frontends bring the promise of splitting your frontend architecture into different frontends for different teams working on different parts of your app. Each team can gain autonomy over the end-to-end lifecycle of their micro frontend which can be developed, versioned, tested, built,

[jaxenter.de/micro-frontends-angular-elements-web-components-steyer-89889](https://jaxenter.de/micro-frontends-angular-elements-web-components-steyer-89889)

**JAXenter** NEWS ARTIKEL VIDEOS JAVA MAGAZIN E.ACADEMY JAX 2020 SUCHE

Java DevOps Microservices Serverless Kubernetes Blockchain Eclipse Android Agile Machine Learning

Quickvote: [Womit befassen Sie sich 2020?](#)

 DevOpsCon 2020 | 23. bis 26. März, Singapur  Kiosk Special Vol. 1: Technologien, die Sie kennen sollten!  JAX 2020 | 11. bis 15. Mai, Mainz

Manfred Steyers Session auf der W-JAX 2019

## Micro Frontends mit Angular Elements und Web Components – eine perfekte Kombination?

#Angular #W-JAX 2019 #Web Components

5. Dezember 2019 Redaktion JAXenter

Die Idee von Micro Frontends ist sehr verlockend: Anstatt eines großen monolithischen Clients erstellt man entsprechend der Microservices-Philosophie mehrere kleine und gut wartbare UIs. Doch wie lassen sich diese einzelnen Inseln mit einem integrierten UI präsentieren? Framework-unabhängige Web Components, die sich dynamisch laden lassen, ermöglichen hier gleich mehrere attraktive Lösungsansätze. Wieso das so ist, erklärt Manfred Steyer (SOFTWAREarchitekt.at) in seiner Session auf der W-JAX 2019.

In seiner Session auf der W-JAX 2019 erklärt Manfred Steyer, wie Sie diese Idee mit Angular Elements umsetzen können und wie Sie durch den Einsatz des neuen Ivy-Compilers praxistaugliche Bundle-Größen für Ihre Web Components erreichen. Darauf aufbauend besprechen wir mehrere Umsetzungsstrategien und diskutieren die Vor- und Nachteile. Am Ende sind Sie in der Lage, die einzelnen Optionen für Ihre Vorhaben zu bewerten.



**POLLS**

**Womit befassen Sie sich 2020?**

- Update auf eine neue Java-Version
- Enterprise Java mit Jakarta EE
- Java im Container (Kubernetes / Docker / etc...)
- Cloud Computing mit Java
- Java als UI-Technologie: JavaFX
- Entwicklung im Mobile-Bereich
- Big-Data-Technologien
- Neuerungen im Tooling-Bereich (Eclipse, NetBeans, Git, Hudson/Jenkins, Maven, etc.)
- Neuerungen bei anderen etablierten JVM-Sprachen wie Groovy, Scala, Clojure, JRuby

## VIELE FIRMEN SIND SCHON AUF DEM TREND

- Karstadt Galeria Kaufhof
- Rewe Digital
- Douglas

# WIESO WERDEN NOCH FEHLENTSCHEIDUNGEN GETROFFEN?

Die meisten Online Materialien sind zu Oberflächlich

Die meisten Verbreiten Mythen und gefährliche Behauptungen

The image shows a grid of five video thumbnail cards for JavaScript tutorials, each with a title, a brief description, and a thumbnail image.

- Learn JavaScript - Full Course for Beginners**  
freeCodeCamp.org 2,5 Mio. Aufrufe • vor 12 Monaten  
This complete 134-part JavaScript tutorial for beginners will teach you everything you need to know to get started with the ...  
Thumbnail: A dark blue background with a yellow 'JS' logo and a flame icon, followed by the text "JavaScript in 3 hours NO ADS" and a duration of 3:26:43.
- JavaScript Tutorial for Beginners: Learn JavaScript in 1 Hour [2019]**  
Programming with Mosh 1,6 Mio. Aufrufe • vor 1 Jahr  
Watch this JavaScript tutorial for beginners to learn JavaScript basics in one hour. Get my Complete JavaScript Course to master ...  
Thumbnail: A yellow background with the text "JAVASCRIPT IN 1 HOUR" and a photo of a smiling man with glasses. A timer icon and a subtitle "Untertitel" are also visible. The duration is 48:17.
- Learn JavaScript in 12 Minutes**  
Jake Wright • 2,1 Mio. Aufrufe • vor 5 Jahren  
Learn the fundamental features of JavaScript - the language used to add dynamic, interactive content to websites. I teach you how ...  
Thumbnail: A screenshot of a computer monitor showing a terminal window with code and a file explorer window. The duration is 12:01.
- JavaScript Tutorial for Beginners - Full Course in 8 Hours [2020]**  
Clever Programmer 197.641 Aufrufe • vor 2 Monaten  
Here is the best free javascript programming course on the planet. Made with lots of ❤️. Take your web development skills to ...  
Thumbnail: A yellow background with the text "JAVASCRIPT 8 HOUR COURSE" and a photo of a man with glasses looking at a computer screen. The duration is 7:57:28.
- JavaScript Tutorial for Beginners**  
Telusko  
Thumbnail: A yellow background with a large 'JS' logo and the word "JavaScript" below it, with a duration of 60.

The screenshot shows a reveal.js presentation slide. On the left, there's a large thumbnail image for a video titled "JAVASCRIPT IN 7 MINUTES". The thumbnail features the text "INTRODUCTION" at the top and "JAVASCRIPT IN 7 MINUTES" in the center. A progress bar at the bottom of the thumbnail indicates the video is 7:36 minutes long. To the right of the thumbnail, the slide content is displayed. At the top, there are two video links: "#0 JavaScript Tutorial | Introduction • 2:44" and "#1 JavaScript Tutorial | Why you should Learn JavaScript Today • 6:48". Below these links is a button labeled "KOMPLETTE PLAYLIST ANSEHEN". The main content area starts with the title "Learn JavaScript in 7 minutes | Create Interactive Websites | Code in 5". Underneath this, it says "blondiebytes • 310.772 Aufrufe • vor 3 Jahren". The final part of the slide contains a descriptive text: "Learn how to make your websites interactive with the JavaScript in 7 minutes! We'll review some basic HTML/CSS concepts as ...".

#0 JavaScript Tutorial | Introduction • 2:44

#1 JavaScript Tutorial | Why you should Learn JavaScript Today • 6:48

KOMPLETTE PLAYLIST ANSEHEN

Learn JavaScript in 7 minutes | Create Interactive Websites | Code in 5

blondiebytes • 310.772 Aufrufe • vor 3 Jahren

Learn how to make your websites interactive with the JavaScript in 7 minutes! We'll review some basic HTML/CSS concepts as ...



## Computer Science Degree vs Coding Bootcamps vs Self Taught (2018 - 2019)

ENGINEERED TRUTH 36.906 Aufrufe • vor 1 Jahr

When it comes between you getting a 4-year computer science degree, specializing in a high demand skill by attending a coding ...



## Coding Bootcamp vs Computer Science Degree | Career Health

Senegoddess 89.034 Aufrufe • vor 2 Monaten

Hi friend! This is a question I get asked a LOT: doing a Coding Bootcamp versus going to college. I hope this can clear up some ...

⋮



## Coding Bootcamp vs Self-Taught vs Computer Science Degree

Coder Foundry 6163 Aufrufe • vor 6 Monaten

There are three main learning paths to becoming a coding career - bootcamps, college, and online courses. Most articles pit one ...



## Coding Bootcamp vs Computer Science degree WHAT SHOULD YOU PICK?

Program With Erik 1443 Aufrufe • vor 1 Jahr

Should you get a full stack certificate or pursue a coding bootcamp or get a computer science degree? How do you get into IT ...



## CS Degree vs Coding Bootcamp (2019) - by Rubén Harris, CEO of #CareerKarma



Career Karma • 3344 Aufrufe • vor 4 Monaten

If you're here, your brains are about to explode because you're not sure about a CS Degree vs. Coding Bootcamp. Well, if you ...



## CODING BOOTCAMPS: What is it and should you go to one?!

mayuko 157.596 Aufrufe • vor 1 Jahr

Hellooo everybody! Signup for your FREE trial to The Great Courses Plus here: <http://ow.ly/TIIA30IEoPL>  
Today I'm talking ...

Untertitel

```
private class IntegerArgumentMarshaler extends ArgumentMarshaler {
    private int intValue = 0;
    private Map<Character, ArgumentMarshaler> intArgs = new HashMap<Character, ArgumentMarshaler>();

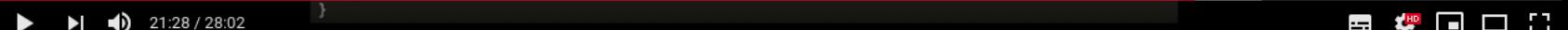
    public void set(String s) throws ArgsException {
        try {
            intValue = Integer.parseInt(s);
        }
        catch (NumberFormatException e) {
            throw new ArgsException();
        }
    }

    public Object get() {
        return intValue;
    }

    private void parseIntegerSchemaElement(char elementId) {
        intArgs.put(elementId, new IntegerArgumentMarshaler());
    }

    public int getInt(char arg) {
        Args.ArgumentMarshaler am = intArgs.get(arg);
        return am == null ? 0 : am.getInteger();
    }

    private void setIntArg(char argChar) throws ArgsException {
        currentArgument++;
        String parameter = null;
        try {
            parameter = args[currentArgument];
            intArgs.get(argChar).setInteger(Integer.parseInt(parameter));
        } catch (ArrayIndexOutOfBoundsException e) {
            valid = false;
            errorArgumentId = argChar;
            errorCode = ErrorCode.MISSING_INTEGER;
            throw new ArgsException();
        } catch (NumberFormatException e) {
            valid = false;
            errorArgumentId = argChar;
            errorParameter = parameter;
            errorCode = ErrorCode.INVALID_INTEGER;
            throw new ArgsException();
        }
    }
}
```



## Object-Oriented Programming is Embarrassing: 4 Short Examples

462.566 Aufrufe • 04.03.2016

1 like 6810 1 dislike 4916 TEILEN SPEICHERN ...



Brian Will  
50.600 Abonnenten

ABONNIEREN

The screenshot shows a browser window with developer tools open. On the left is a code editor with a file named 'script.js' containing JavaScript code. On the right is a DevTools console tab labeled 'Console'. The console output shows two log statements:

```
Making Request to Facebook          script.js:3
We can only talk to Google         script.js:25
```

Below the browser window, a YouTube video player interface is visible, showing the video title 'JavaScript Async Await' and other video details.

```
script.js
19 makeRequest('Facebook').then(response => {
20   console.log('Response Received')
21   return processRequest(response)
22 }).then(processedResponse => {
23   console.log(processedResponse)
24 }).catch(err => {
25   console.log(err)
26 })
27
28 async function doWork() {
29   const response = await makeRequest('Google')
30   console.log('Response Received')
31   const processedResponse = await processRequest(response)
32 }
```

DevTools - 127.0.0.1:5500/

Elements Console Sources >

top

Making Request to Facebook script.js:3

We can only talk to Google script.js:25

Console

#JavaScript #AsyncAwait #Promises

JavaScript Async Await

55.205 Aufrufe • 09.02.2019

2280 28 TEILEN SPEICHERN ...

Web Dev Simplified 112.000 Abonnenten

ABONNIEREN

# **KOOPERATION IM BEZUG AUF PRAXIS-PROJEKTE ODER BACHELOR-ARBEITEN**

Mögliche Themen Praxis-Projekt:

- Modularisierung von Web-Componenten
- Erstellung eines Vertriebstool

## MÖGLICHE THEMEN BACHELOR-ARBEIT:

- Entwicklung und Evaluierung eines Konzepts für Agenturen für Micro-Frontends
- Konzeptzonierung von Micro-Services mit gRPC

Kontaktieren Sie mich: [artur.zimmermann@ambient-innovation.com](mailto:artur.zimmermann@ambient-innovation.com)

# PRAKTISCHE ÜBUNG: SUPERHELDEN LISTE

# ANFORDERUNGEN

- Eine Startseite mit Bild und Login (Username, Password, Login-Button)
- Username soll der Name eines Superhelden sein
- Password soll mindesten 6 Zeichen lang sein und ein Sonderzeichen haben.
- Bei erfolgreichem Login kommt man auf eine neue Seite
- Auf der nächsten Seite; Die Helden als Liste darstellen (Name, Beschreibung, Bild)

Codesandbox link: <https://codesandbox.io/s/hbrs-webengineering2020-6pd52>