# Vorbereitung

NodeJS installieren

npx create-react-app reactsample

cd reactsample

npm start

<http://localhost:3000/>

# Übung 1

## Timer.js

import React from 'react';

class Timer extends React.Component {

constructor(props) {

super(props);

}

render() {

return (

<div>{new Date().toLocaleTimeString()}</div>

);

}

}

export default Timer;

## App.js

import Timer from './Timer';

render() {

return (

<div className="App">

<header className="App-header">

<img src={logo} className="App-logo" alt="logo" />

</header>

<main>

<Timer></Timer>

</main>

</div>

);

}

## App.css

.App-logo {

height: 20vmin;

…

}

.App-header {

min-height: 40vh;

…

}

# Übung 2

## Timer.js

import React from 'react';

class Timer extends React.Component {

constructor(props) {

super(props);

this.state = { date: new Date()};

setInterval(() => this.tick(), 1000);

}

tick() {

this.setState({ date: new Date()});

}

render() {

return (

<div>{this.state.date.toLocaleTimeString()}</div>

);

}

}

export default Timer;

# Übung 3

## Login.js

import React from 'react';

class Login extends React.Component {

constructor(props) {

super(props);

this.state = { isLoggedIn: false };

this.onClick = this.onClick.bind(this);

}

onClick(e) {

e.preventDefault();

this.setState({ isLoggedIn: !this.state.isLoggedIn });

}

render() {

let welcomeMessage;

if (this.state.isLoggedIn) {

welcomeMessage = <div>{'Welcome ' + this.props.name}</div>;

}

return (

<React.Fragment>

<button onClick={this.onClick}>{this.state.isLoggedIn ? 'Logout' : 'Login'}</button>

{welcomeMessage}

</React.Fragment>

);

}

}

export default Login;

## App.js

import Login from './Login';

render() {

return (

<div className="App">

<header className="App-header">

<img src={logo} className="App-logo" alt="logo" />

</header>

<main>

<Timer></Timer>

<Login name="Markus"></Login>

</main>

</div>

);

}

# Übung 4

## Login.js

import React from 'react';

class Login extends React.Component {

constructor(props) {

super(props);

this.onClick = this.onClick.bind(this);

}

onClick(e) {

e.preventDefault();

this.props.onChange(!this.props.isLoggedIn);

}

render() {

return (

<button onClick={this.onClick}>{this.props.isLoggedIn ? 'Logout' : 'Login'}</button>

);

}

}

export default Login;

## App.js

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

import Timer from './Timer';

import Login from './Login';

class App extends Component {

constructor(props) {

super(props);

this.state = { isLoggedIn: false };

this.onLoginChange = this.onLoginChange.bind(this);

}

onLoginChange(value) {

this.setState({ isLoggedIn: value });

}

render() {

let welcomeMessage;

if (this.state.isLoggedIn) {

welcomeMessage = <div>{'Welcome ' + this.props.name}</div>;

}

return (

<div className="App">

<header className="App-header">

<img src={logo} className="App-logo" alt="logo" />

</header>

<main>

<Timer></Timer>

<Login isLoggedIn={this.state.isLoggedIn} onChange={this.onLoginChange}></Login>

{welcomeMessage}

</main>

</div>

);

}

}

export default App;

## index.js

ReactDOM.render(<App name="Markus" />, document.getElementById('root'));

# Übung 5

## Axios

npm install axios

npm install es6-promise

## index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import \* as serviceWorker from './serviceWorker';

require('es6-promise').polyfill();

ReactDOM.render(<App name="Markus" />, document.getElementById('root'));

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: http://bit.ly/CRA-PWA

serviceWorker.unregister();

## App.js

import PersonSearchPanel from './PersonSearchPanel';

class App extends Component {

render() {

let welcomeMessage;

let personSearchPanel

if (this.state.isLoggedIn) {

welcomeMessage = <div>{'Welcome ' + this.props.name}</div>;

personSearchPanel = <PersonSearchPanel />

}

return (

<div className="App">

<header className="App-header">

<img src={logo} className="App-logo" alt="logo" />

</header>

<main>

<Timer></Timer>

<Login isLoggedIn={this.state.isLoggedIn} onChange={this.onLoginChange}></Login>

{welcomeMessage}

{personSearchPanel}

</main>

</div>

);

}

}

## PersonSearchPanel

import React from 'react';

import PersonSearchForm from './PersonSearchForm';

import PersonSearchResult from './PersonSearchResult';

import axios from 'axios';

class PersonSearchPanel extends React.Component {

constructor(props) {

super(props);

this.state = { persons : [] }

this.onChange = this.onChange.bind(this);

}

onChange(text) {

axios.get('http://localhost:8080/person?name=' + text)

.then(res => {

this.setState({ persons : res.data });

}

);

}

render() {

return (

<div>

<h2>Personensuche</h2>

<PersonSearchForm onChange={this.onChange} />

<PersonSearchResult persons={this.state.persons} />

</div>

);

}

}

export default PersonSearchPanel;

## PersonSearchForm

import React from 'react';

class PersonSearchForm extends React.Component {

constructor(props) {

super(props);

this.onKeyUp = this.onKeyUp.bind(this);

}

onKeyUp(e) {

e.preventDefault();

this.props.onChange(e.target.value);

}

render() {

return (

<div>

<span>Name: </span>

<input type="text" onKeyUp={this.onKeyUp} />

</div>

);

}

}

export default PersonSearchForm;

## PersonSearchResult

import React from 'react';

class PersonSearchResult extends React.Component {

constructor(props) {

super(props);

}

render() {

const rows = [];

this.props.persons.forEach(person => {

rows.push(<PersonRow key={person.name + '\_\_\_' + person.vorname} person={person} />)

});

return (

<table>

<thead>

<PersonHeader />

</thead>

<tbody>

{rows}

</tbody>

</table>

);

}

}

class PersonHeader extends React.Component {

render() {

return (

<tr>

<th>Name</th>

<th>Vorname</th>

</tr>

)

}

}

class PersonRow extends React.Component {

render() {

return (

<tr>

<td>{this.props.person.name}</td>

<td>{this.props.person.vorname}</td>

</tr>

)

}

}

export default PersonSearchResult;

# Übung 6

## ErrorBoundary.js

import React from 'react';

class ErrorBoundary extends React.Component {

constructor(props) {

super(props);

this.state = { error: null, errorInfo: null };

}

componentDidCatch(error, errorInfo) {

this.setState({

error: error,

errorInfo: errorInfo

})

}

render() {

if (this.state.errorInfo) {

return (

<div>

<p><b>Es ist ein Fehler aufgetreten. Sie müssen die Seite neu laden (F5).</b></p>

<p></p>

<p>

{this.state.error && this.state.error.toString()}

</p>

<p></p>

<pre>

{this.state.errorInfo.componentStack}

</pre>

</div>

);

} else {

return this.props.children;

}

}

}

export default ErrorBoundary;

## PersonSearchPanel.js

render() {

return (

<ErrorBoundary>

<div>

<h2>Personensuche</h2>

<PersonSearchForm onChange={this.onChange} />

<PersonSearchResult persons={this.state.persons} />

</div>

</ErrorBoundary>

);

}

## PersonSearchResult.js

class PersonRow extends React.Component {

render() {

if (this.props.person.name === 'XXX1') {

throw new Error('I crashed!');

}

return (

<tr>

<td>{this.props.person.name}</td>

<td>{this.props.person.vorname}</td>

</tr>

)

}

}

# Materialize installieren

npm install react-materialize

## index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<link rel="shortcut icon" href="%PUBLIC\_URL%/favicon.ico">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<meta name="theme-color" content="#000000">

<!--

manifest.json provides metadata used when your web app is added to the

homescreen on Android. See https://developers.google.com/web/fundamentals/engage-and-retain/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json">

<!--

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

-->

<title>React App</title>

<!-- Import Google Icon Font -->

<link href="http://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<!-- Import materialize.css -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.98.0/css/materialize.min.css" rel="stylesheet">

</head>

<body>

<noscript>

You need to enable JavaScript to run this app.

</noscript>

<!-- Import jQuery before materialize.js -->

<script src="https://code.jquery.com/jquery-2.1.1.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.98.0/js/materialize.min.js"></script>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.

To create a production bundle, use `npm run build` or `yarn build`.

-->

</body>

</html>

# Übung 7

## App.js

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

import Timer from './Timer';

import Login from './Login';

import PersonSearchPanel from './PersonSearchPanel';

import { Row, Col } from 'react-materialize';

class App extends Component {

constructor(props) {

super(props);

this.state = { isLoggedIn: false };

this.onLoginChange = this.onLoginChange.bind(this);

}

onLoginChange(value) {

this.setState({ isLoggedIn: value });

}

render() {

let welcomeMessage;

let personSearchPanel

if (this.state.isLoggedIn) {

welcomeMessage = <div>{'Welcome ' + this.props.name}</div>;

personSearchPanel = <PersonSearchPanel />

}

return (

<div className="App container">

<header className="App-header">

<Row>

<Col s={4} m={4} className="left-align">

<Login isLoggedIn={this.state.isLoggedIn} onChange={this.onLoginChange}></Login>

</Col>

<Col s={4} m={4}>

<img src={logo} className="App-logo" alt="logo" />

{welcomeMessage}

</Col>

<Col s={4} m={4}>

</Col>

</Row>

</header>

<main className="grey lighten-3">

{personSearchPanel}

</main>

<footer className="black white-text">

<Row>

<Col s={6} m={6} className="left-align">&copy; Copyright Markus Borer</Col>

<Col s={6} m={6} className="right-align"><Timer /></Col>

</Row>

</footer>

</div>

);

}

}

export default App;

## App.css

.App {

text-align: center;

display: flex;

flex-direction: column;

min-height: 100vh;

}

.App-logo {

animation: App-logo-spin infinite 20s linear;

height: 20vmin;

}

.App-header {

min-height: 30vh;

background-color: #282c34;

font-size: calc(10px + 2vmin);

color: white;

padding: 20px;

}

.App-link {

color: #61dafb;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

main {

flex: 1 0 auto;

padding: 0px 20px 0px 20px;

}

footer {

padding: 20px 20px 0px 20px;

}

## Login.js

import React from 'react';

import { Button, Icon } from 'react-materialize';

class Login extends React.Component {

constructor(props) {

super(props);

this.onClick = this.onClick.bind(this);

}

onClick(e) {

e.preventDefault();

this.props.onChange(!this.props.isLoggedIn);

}

render() {

return (

<Button onClick={this.onClick}>{this.props.isLoggedIn ? 'Logout' : 'Login'}<Icon left>{this.props.isLoggedIn ? 'power\_settings\_new' : 'exit\_to\_app'}</Icon></Button>

);

}

}

export default Login;

## PersonSearchPanel.js

import React from 'react';

import PersonSearchForm from './PersonSearchForm';

import PersonSearchResult from './PersonSearchResult';

import axios from 'axios';

import ErrorBoundary from './ErrorBoundary';

class PersonSearchPanel extends React.Component {

constructor(props) {

super(props);

this.state = { persons : [] }

this.onChange = this.onChange.bind(this);

}

onChange(text) {

axios.get('http://localhost:8080/person?name=' + text)

.then(res => {

this.setState({ persons : res.data });

}

);

}

render() {

return (

<ErrorBoundary>

<div>

<h2 className="left-align">Personensuche</h2>

<PersonSearchForm onChange={this.onChange} />

<PersonSearchResult persons={this.state.persons} />

</div>

</ErrorBoundary>

);

}

}

export default PersonSearchPanel;

## PersonSearchForm

import React from 'react';

import { Row, Input } from 'react-materialize';

class PersonSearchForm extends React.Component {

constructor(props) {

super(props);

this.onKeyUp = this.onKeyUp.bind(this);

}

onKeyUp(e) {

e.preventDefault();

this.props.onChange(e.target.value);

}

render() {

return (

<Row>

<Input label="Name" onKeyUp={this.onKeyUp} />

</Row>

);

}

}

export default PersonSearchForm;

## PersonSearchResult.js

import React from 'react';

import { Table } from 'react-materialize';

class PersonSearchResult extends React.Component {

constructor(props) {

super(props);

}

render() {

const rows = [];

this.props.persons.forEach(person => {

rows.push(<PersonRow key={person.name + '\_\_\_' + person.vorname} person={person} />)

});

return (

<Table hoverable="true">

<thead>

<PersonHeader />

</thead>

<tbody>

{rows}

</tbody>

</Table>

);

}

}

class PersonHeader extends React.Component {

render() {

return (

<tr>

<th>Name</th>

<th>Vorname</th>

</tr>

)

}

}

class PersonRow extends React.Component {

render() {

if (this.props.person.name === 'XXX1') {

throw new Error('I crashed!');

}

return (

<tr>

<td>{this.props.person.name}</td>

<td>{this.props.person.vorname}</td>

</tr>

)

}

}

export default PersonSearchResult;

## ErrorBoundary.js

import React from 'react';

class ErrorBoundary extends React.Component {

constructor(props) {

super(props);

this.state = { error: null, errorInfo: null };

}

componentDidCatch(error, errorInfo) {

this.setState({

error: error,

errorInfo: errorInfo

})

}

render() {

if (this.state.errorInfo) {

return (

<div>

<p className="red-text"><b>Es ist ein Fehler aufgetreten. Sie müssen die Seite neu laden (F5).</b></p>

<p></p>

<p>

{this.state.error && this.state.error.toString()}

</p>

<p></p>

<pre>

{this.state.errorInfo.componentStack}

</pre>

</div>

);

} else {

return this.props.children;

}

}

}

export default ErrorBoundary;

# Übung 8

npm install rxjs

## PersonSearchPanel.js

**import** React **from 'react'**;  
**import** PersonSearchForm **from './PersonSearchForm'**;  
**import** PersonSearchResult **from './PersonSearchResult'**;  
**import** axios **from 'axios'**;  
**import** ErrorBoundary **from './ErrorBoundary'**;  
**import** { *fromEvent*, *of* } **from 'rxjs'**;  
**import** { *switchMap*, *debounceTime* } **from 'rxjs/operators'**;  
**import** { **Modal** } **from 'react-materialize'**;  
  
**class** PersonSearchPanel **extends** React.Component {  
  
 **eventTarget** = **new** EventTarget();  
  
 constructor(props) {  
 **super**(props);  
 **this**.**state** = { **persons** : [] }  
 **this**.onChange = **this**.onChange.bind(**this**);  
 }  
  
 componentDidMount() {  
 *fromEvent*(**this**.**eventTarget**, **"OnChange"**)  
 .pipe(  
 *debounceTime*(200)  
 )  
 .pipe(  
 *switchMap*(  
 event => axios.get(**'http://localhost:8080/person?name='** + event.**detail**)  
 .catch(error => {  
 **console**.log(error);  
 **this**.setState({  
 **error**: **'Fehler bei der Serveranfrage'** })  
 **return** *of*(**undefined**);  
 })  
 )  
 )  
 .subscribe(  
 res => {  
 **if** (res.**data** !== **undefined**) {  
 **console**.log(res.**data**);  
 **this**.setState({  
 **persons**: res.**data**,  
 **error**: **undefined** });  
 }  
 }  
 );  
 }  
  
 onChange(text) {  
 **this**.**eventTarget**.dispatchEvent(  
 **new** CustomEvent(**'OnChange'**, { **detail**: text })  
 );  
 }  
  
 render() {  
 **return** (  
 <**ErrorBoundary**>  
 <**div**>  
 <**h2 className="left-align"**>Personensuche</**h2**>  
 <**PersonSearchForm onChange=**{**this**.onChange} />  
 {**this**.**state**.**error** === **undefined** &&  
 <**PersonSearchResult persons=**{**this**.**state**.**persons**} />  
 }  
 {**this**.**state**.**error** !== **undefined** &&  
 <**Modal header='Error' open='true'**>  
 {**this**.**state**.**error**}  
 </**Modal**>  
 }  
 </**div**>  
 </**ErrorBoundary**>  
 );  
 }  
  
}  
  
**export default** PersonSearchPanel;