React Hello Component Blog

Einfachste React Component

Mit dem Blog React Hello World HTML Blog haben wir erste Schritte mit React aufgezeigt.Jetzt bauen wir die erste React Component und beginnen mit der einfachsten aller Komponenten. Wir verwenden hierzu wiederum HTML und den Babel Compiler, also JSX.Als Grundlage beginnen wir mit dem folgenden HTML JSX Gerüst:<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8" />
 <title>Hello Comonent</title>
 <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
 <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
 <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
 </head>
 <body>
 <div id="root"></div>
 </body>
</html>Im <head> Element definieren wir die erste einfache React Component wie folgt:<script type="text/babel">
class HelloComponent extends React.Component {
 render() {
 return <h1>Hello {this.props.name}</h1>;
 }
}
</script>React Components sind aus React.Component abgeleitet und überschreiben minimal die render() Methode.Der Name einer React Komponente kann als Element direkt referenziert werden wie das folgende Script platziert im <body>-Element zeigt:<script type="text/babel">
 ReactDOM.render(<HelloComponent name="James" />, document.getElementById('root'));
</script>Das name-Attribut wird der Komponente als Property via props zugeführt und in der Komponente direkt referenziert (this.props.name).Damit ist unsere erste React Komponente fertig programmiert und einsatzbereit.

Lösung Einfachste React Component

Das HTML Script der einfachsten React Component:<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8" />
 <title>Hello Component</title>
 <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
 <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
 <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
 <script type="text/babel">
 class HelloComponent extends React.Component {
 render() {
 return <h1>Hello {this.props.name}</h1>;
 }
 }
 </script>
 </head>
 <body>
 <div id="root"></div>
 <script type="text/babel">
 ReactDOM.render(<HelloComponent name="James" />, document.getElementById('root'));
 </script>
 </body>
</html>Die laufende Lösung zu diesem Blog und Einfachste React Component finden Sie hier

React Component Constructor

React Components werden über einen Konstruktor initialisiert. Man spricht hier von der Initalisierung von State. Dieser State wird in jeder React Component im "state" Object geführt.In React werden Konstruktoren normalerweise nur für zwei Zwecke verwendet:Initialisieren des lokalen Status durch Zuweisen eines Objekts zu this.state.Binden von Ereignishandlermethoden an eine Instanz.Der Konstruktor für eine React-Komponente wird für jede Instanz aufgerufen. Wenn Sie den Konstruktor für eine React.Component-Unterklasse implementieren, sollten Sie vor jeder anderen Anweisung super (props) aufrufen. Andernfalls ist this.props im Konstruktor undefiniert, was zu

Fehlern führen kann.Wir ergänzen unsere HelloComponent mit einem constructor welcher die Background und Foreground Color im state setzt:constructor(props) {
  super(props);
 this.state = {color: "red", background: "black"};
}Den CSS Style setzen wir in der render-Methode wie folgt:render() {
  const style = {};
  style.background = this.state.background;
  style.color = this.state.color;
  return <h1 style={style}>Hello {this.props.name}</h1>;
}Unsere Komponente arbeitet mit state und props. In einem weiteren Blog werden wir dies thematisieren.

Lösung Constructor React Component

Das HTML Script der Constructor React Component:<!DOCTYPE html><!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello Component</title>
    <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
    <script type="text/babel">
      class HelloComponent extends React.Component {
        constructor(props) {
          super(props);
 this.state = {color: "red", background: "black"};
        }
        render() {
          const style = {};
          style.background = this.state.background;
 style.color = this.state.color;
          return <h1 style={style}>Hello {this.props.name}</h1>;
        }
      }
    </script>
  </head>
  <body>
    <div id="root"></div>
    <script type="text/babel">
      ReactDOM.render(<HelloComponent name="James"/>, document.getElementById('root'));
    </script>
  </body>
</html>Die laufende Lösung zu diesem Blog und Constructor React Component finden Sie hier

Feedback

War dieser Blog für Sie wertvoll. Wir danken für jede Anregung und Feedback

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

https://www.simtech-ag.ch/thematisieren