

lab73



REACT

INHALTSVERZEICHNIS

Was sind die Vor- und Nachteile?	3
Wie setzt man das ganze auf?	5
Wie funktionieren die REST-Calls?	9
Auf welche Probleme seid ihr gestoßen?	10
Was würdet ihr euren Mitschülern empfehlen?	10
Gibt es besonders coole Features?	10
Was ist die Geschichte des Frameworks?	11
Wie sieht es mit der Community aus?	11
Berühmte Applikationen die React verwenden?	12
Wie sieht das Tooling hinter dem Framework aus?	12
Geschwindigkeit des Frameworks?	13
Lernkurve des Frameworks	13
Wie findet ihr das Framework persönlich?	14
Würdet Ihr den Einsatz empfehlen?	14

Was sind die Vor- und Nachteile?

Die Vorteile von ReactJS

1. Virtuelles DOM in ReactJS macht die Benutzerfreundlichkeit besser und die Arbeit der Entwickler schneller

DOM (document object model) ist eine logische Struktur von Dokumenten in den Formaten HTML, XHTML oder XML. Laienhaft beschrieben, handelt es sich um eine Anzeigevereinbarung über Daten ein- und -ausgänge, die eine Baumform hat. Web-Browser verwenden Layout-Engines, um die Darstellung der HTML-Syntax in ein Dokumenten-Objektmodell zu transformieren oder zu „parsen“, das wir in Browsern sehen können.

2. Die Erlaubnis zur Wiederverwendung von React-Komponenten spart erheblich Zeit

Ein weiterer Vorteil, den Facebook mit React eingeführt hat, ist die Möglichkeit, Code-Komponenten einer anderen Ebene jederzeit wiederzuverwenden, ein weiterer bedeutender Zeitgewinn.

3. Datenfluss in einer Richtung in ReactJS bietet einen stabilen Code

ReactJS ermöglicht die direkte Arbeit mit Komponenten und nutzt die Datenbindung nach unten, um sicherzustellen, dass Änderungen der Kinderstrukturen sich nicht auf die Eltern auswirken. Das macht den Code stabil.

4. Eine Open-Source-Bibliothek von Facebook: ständig weiterentwickelt und offen für die Gemeinschaft

ReactJS war eines der ersten Projekte mit JavaScript-Anbindung, das von Facebook als Open Source veröffentlicht wurde. Das bedeutet, dass ReactJS alle Vorteile des freien Zugangs nutzt - viele nützliche Anwendungen und zusätzliche Tools von firmenfremden Entwicklern.

Nachteile von ReactJS

1. Hohes Entwicklungstempo

Dieser Nachteil wird von den Entwicklern Michael Jackson und Ryan Florence auf Modern Web treffend beschrieben: "Falls Sie nicht bemerkt haben, dass wir hier ein Auto mit zwei platten Reifen fahren, die Motorhaube ist gerade vor die Windschutzscheibe geflogen, und wir haben keine Ahnung mehr, was hier los ist! Das Umfeld ändert sich ständig, und die Entwickler müssen sich regelmäßig mit den neuen Vorgehensweisen vertraut machen. Alles entwickelt sich weiter, und einige Entwickler fühlen sich nicht wohl dabei, mit einem solchen Tempo Schritt zu halten."

2. Schlechte Dokumentation

Das Problem mit der Dokumentation geht auf die ständige Veröffentlichung neuer Werkzeuge zurück. Verschiedene und neue Bibliotheken wie Redux und Reflux versprechen, die Arbeit einer Bibliothek zu beschleunigen oder das gesamte React-Ökosystem zu verbessern. Am Ende kämpfen die Entwickler mit der Integration dieser Tools in ReactJS.

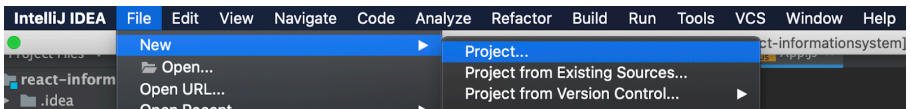
3. 'HTML in meinem JavaScript' - JSX als Hindernis

ReactJS verwendet JSX. Es ist eine Syntaxerweiterung, die es erlaubt, HTML mit JavaScript zu mischen. JSX hat seine eigenen Vorteile (z.B. den Schutz von Code vor Injektionen), aber einige Mitglieder der Entwicklungsgemeinschaft halten JSX für einen ernsthaften Nachteil. Entwickler und Designer beklagen sich über die Komplexität von JSX und die daraus resultierende steile Lernkurve.

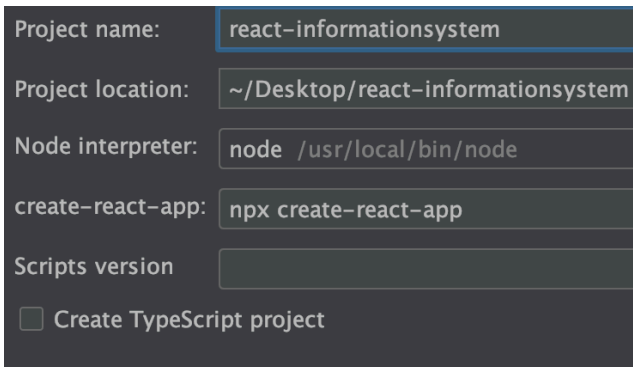
Wie setzt man das ganze auf?

Create Project

Klicke auf New Project and wähle React > Next.



Wähle einen Projektnamen and deine Speicherlocation > Finish.

A screenshot of the 'Create Project' dialog in IntelliJ IDEA. The dialog has several fields: 'Project name:' with the value 'react-informationsystem', 'Project location:' with the value '~/Desktop/react-informationsystem', 'Node interpreter:' with the value 'node /usr/local/bin/node', 'create-react-app:' with the value 'npx create-react-app', and 'Scripts version' which is empty. There is also a checkbox labeled 'Create TypeScript project' which is unchecked.

Hole dir am besten nun einen Kaffee während das Projekt „Indext“, dies wird eine Zeit dauern.

Nun müssen wir noch in das generierte File „App.js“ gehen und dort den p tag in Hello World umändern.

Davor:

```
import ...

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

Dannach:

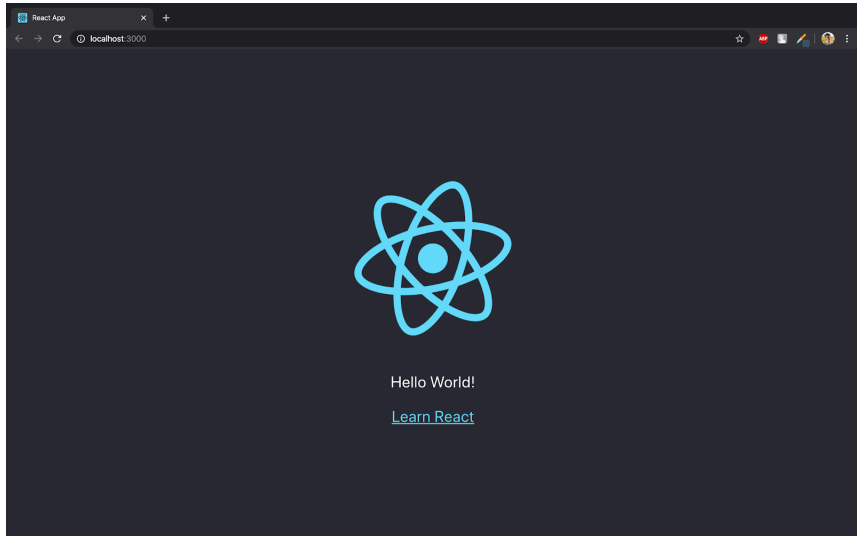
```
import ...

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Hello World!
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

Zu guter letzt musst du den Code noch ausführen und überwältigt sein wie schnell das alles gegangen ist.

Das finale Ergebnis sollte nun nämlich so aussehen:



Wie funktionieren die REST-Calls?

Sie können jede beliebige AJAX-Bibliothek mit React verwenden. Einige beliebte sind Axios, jQuery AJAX und die im Browser eingebaute `window.fetch`.

Wir haben uns dabei für Axios entschieden, da wir zuvor auch schon einmal damit gearbeitet haben und es auch am meisten verbreitet ist. Der Code dafür ist eigentlich ganz einfach gehalten. Hier ein Beispiel:

```
function getAllServices() {  
  const Http = new XMLHttpRequest();  
  const url = "http://localhost:8080/api/services";  
  Http.open("GET", url);  
  Http.send();  
  return new Promise(resolve => {  
    Http.onreadystatechange = (e) => {  
      if (Http.readyState === 4) {  
        let response = Http.responseText;  
        let array = JSON.parse(response);  
        resolve(array);  
      }  
    }  
  })  
}
```

Auf welche Probleme seit ihr gestoßen?

Am meisten Schwierigkeiten hatten wir mit dem aktualisieren der Felder. ... Entweder es aktualisierte sich zu wenig, oder es aktualisierte einfach alles. Wir haben dies ganz gut mit den States von React gelöst. Man musste nur wissen wann, wo und vor allem wie wir diese genau einsetzen mussten. Nach einigen try und error hat es schließlich auch geklappt.

Sonst sind wir auf eher weniger Probleme gestoßen, da das gesamte Team im Vorhinein schon mit React vertraut war.

Was würdet ihr euren Mitschülern empfehlen?

Functional Components zu verwenden. Und das konstant! In React kann man seine einzelnen Components (also Stücke einer Seite) entweder als „Functional Component“ oder als „Class Component“ anlegen. Viele StackOverflow Antworten verwenden zwar noch häufig Class Components, was aber ein veraltetes Modell ist und so zu Leistungseinbuße führen kann.

Gibt es besonders coole Features?

Eigentlich nicht - es ist nur immer wieder erstaunlich wie schnell man in React zu einer schönen und effektiven Seite gerät.

Was ist die Geschichte des Frameworks?

React ist eine sehr neue Technologie im Vergleich zu den anderen Technologien auf dem Markt. Es entstand 2011, als Jordan Walke, ein Software-Ingenieur bei Facebook, das Framework ins Leben rief. React wird von XHP beeinflusst, einem einfachen HTML-Komponenten-Framework für PHP. Der erste Anwendungsfall von React war der Newsfeed von Facebook im Jahr 2011.

Später wurde es auch von Instagram verwendet. Mit der Zeit wuchs React, und Facebook beschloss, es im Mai 2013 bei JSConf US zu Open Source zu machen.

2015 wurde React Native auf der JSConf angekündigt, was eine einfache Entwicklung mit Android und iOS ermöglichte. Im selben Jahr wurde React Native als Open Source veröffentlicht, um sein Wachstum zu unterstützen.

Wie sieht es mit der Community aus?

Die Community hinter React ist sehr stark - laut StackOverflow Statistik die Stärkste nach Angular. Angefangen von den zahlreichen StackOverflow Antworten, bis hin zu den Millionen von Websites, welche ebenfalls Probleme von React aus der Welt schaffen, findet man meist eine gute Antwort.

Berühmte Applikationen die React verwenden?

- Facebook.
- Dropbox.
- Mattermark.
- Tesla.
- Atlassian.
- Airbnb.
- Netflix.
- Reddit.

Wie sieht das Tooling hinter dem Framework aus?

React funktioniert mit fast allen Web Development IDEs - wir persönlich haben IntelliJ dafür verwendet und waren auch sehr zufrieden.

Um die Entwicklung zu erleichtern bietet React auch noch eine Chrome Extension mit der man auch in der geränderten Website eine Übersicht über seine Components hat.

Geschwindigkeit des Frameworks?

Persönlich können wir kaum eine Aussage über die Geschwindigkeit treffen, da unsere App dafür nicht komplex genug war und wir es auch nicht aktiv getestet haben.

Laut Internet Recherchen aber ist React eines der schnelleren Frameworks. Schneller als Angular und auch am leistungsstärksten. Der Dank gebührt dabei dem Virtual DOM (Document Object Model). Anstatt eines echten DOM verwendet React lieber ein virtuelles, was zu massiven Leistungssteigerungen führt.

Lernkurve des Frameworks

Eigentlich sehr „steil“. Nach unserem ersten Projekt in React war das meiste klar und wir konnten sicher doppelt so schnell eine zweite Seite erstellen. Bei diesem Projekt jetzt war der Lerneffekt an sich nicht sehr hoch, da wir mit dem meisten schon vertraut waren. Trotzdem aber sind wir wieder etwas schneller geworden und konnten einiges verbessern.

Wie findet ihr das Framework persönlich?

Wir persönlich sind sehr überzeugt von diesem Framework. In den meisten unserer Applikationen verwenden wir es sogar. Öfters gibt es bei uns noch Debatten über einen Wechsel zu Angular, aber meist wird dann doch wieder React verwendet.

Würdet Ihr den Einsatz empfehlen?

Absolut. React ist unserer Meinung nach eines der besten Frameworks, weswegen wir es auch jedem weiterempfehlen würden. Die Lernkurve ist nicht all zu flach und die Community ist besonders stark. Auch praktisch an React ist die Vielzahl an externen Librarys, welche eine noch schnellere und angenehmere Entwicklung ermöglichen.