

React – Report

Ennemoser Patrick ga92fay@mytum.de
Menz Gregor ga92xib@mytum.de
Rinner Fabian ga48gov@mytum.de

Ursprung:

- 2013 von Facebook ins Leben gerufen
- Entwickelt um Cross-Site-Scripting-Angriffe zu verhindern
- React ist eine reine Oberflächenbibliothek
- Es unterstützt sowohl server- als auch clientseitiges Rendering

Projekte:

- Facebook
- Netflix
- AirBnB
- Imagehoster Imgur
- Yahoo! Mail Client
- WhatsApp Web
- Instagram

Vorteile der GUI Technologie:

- Ständig wachsende Community
- wiederverwendbare Komponenten
- das virtuelle DOM
- React Developer Tools
- JSX

Probleme:

- langsame Lernkurve für Programmier-Anfänger
- nur eine Library für User Interfaces
- umständlich in ein MVC Model umzuwandeln

Architektur:

- React besitzt keinen Paketmanager
- Empfohlen wird Node.js
- Benötigte Dateien: index.html, index.js und optional index.css
- JSX:
 - visuelle Hilfe beim Arbeiten mit JavaScript zum Erstellen der Benutzeroberfläche
 - Syntaxerweiterung für JavaScript
 - Mischung aus HTML und JavaScript
 - Ermöglicht zusätzliche Warn- und Fehlermeldungen
 - Verhindert XSS Angriffe

- Komponenten:
 - teilen die Benutzerfläche in unabhängige wiederverwendbare Teile
 - Funktionale Komponenten:
 - aufgebaut wie eine Funktion
 - erhalten props und returnen ein React Element
 - Klassen Komponenten:
 - vergleichbar mit Klassen in Java
 - müssen React.Component extenden
 - das Einzige nicht optionale ist die render()-Methode
 - Properties/props:
 - props sind read only
 - werden beim Aufruf der Komponente mitgegeben
 - States
 - privat und vollständig von der Komponente kontrolliert
 - mit dem Konstruktor festgelegt
 - nur veränderbar mit der Methode setState()
 - bei Veränderung, re-rendern der betroffenen Abschnitte
- Lifecycle Hooks
 - von React vorgegebene Methoden
 - zu 4 verschiedenen Zeitpunkten automatisch aufgerufen
 - folgende lifecycle hooks in der aktuellen React Version:
 - constructor()
 - render()
 - componentDidMount
 - componentDidUpdate
 - componentWillUnmount
- Virtuelle DOM
 - abstrakte Kopie des tatsächlichen DOMs
 - deutlich kleiner und nur das Nötigste an Informationen
 - wird verglichen mit dem richtigen DOM
 - nur Änderungen werden auf dem realen DOM angewandt
 - bietet eine minimale Aktualisierungszeit und höhere Leistung

Zusammenarbeit mit anderen Technologien:

- Arbeit mit vielen Backend-Technologien gut zusammen
- Node.js wird aber empfohlen
- Weitere oft verwendete Technologie -> Redux:
 - sogenannter predictable state container für JavaScript
 - konsistente Anwendungen, leicht zu testen, Live-Code Bearbeitung

Community:

- Auf der Internetplattform „Stack Overflow“ gibt es zum Tag react.js über 90.000 Fragen
- 40.000 Repositories auf Github
- kein Mangel an Antworten und Beispielen