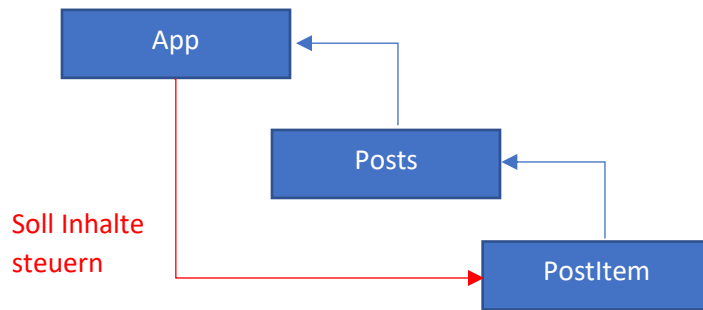


Webtechnologien Übungen React 3

Aufgabe 1:

Was ist der Weg, wenn Inhalte einer Subkomponente (bspw. Texte, die angezeigt werden sollen) aus einer höheren Komponenten-Ebene gesteuert werden soll?

Beispiel:



Aufgabe 2:

Für diese Aufgabe soll die Applikation aus der Übung „APIs“ in React neu zusammengebaut werden. Dafür verwenden wir die UI-Bibliothek „material ui“ für React (ist im vorgegebenen Projekt enthalten). Als fertige Komponenten können wir hierbei die Komponenten Card, CardMedia, CardContent, CardActions, Button, Typography (alle für die Darstellung einer Karte), TextField (für das Suchfeld), und Container (als Oberkomponente das alle Unterelemente beinhaltet).

Für diese Aufgabe ist ein Projekt im Ordner „react3_1“ vorgegeben, welches weiterentwickelt werden soll. Hier sind die Komponenten „App.js“, „Posts.js“ sowie „PostItem.js“ enthalten.

App.js stellt den EntryPoint dar und beinhaltet einen Container, welcher die Posts.js Komponente beinhaltet.

Posts.js rendert alle PostItems der jeweiligen Anfrage. Sie übernimmt außerdem den Suchknopf, sowie die Bearbeitung des Suchfeldes und den damit verbundenen Suchvorgang. Die Daten für jeden Post sollen in einem Array (z.B. „items“) im Zustand der Komponente gespeichert werden.

PostItem.js stellt einen Post in Form einer material-ui Card-Komponente dar. Sie soll folgende Props erhalten:

Prop	Beschreibung
<i>img</i>	Pfad zu einer Bilddatei
<i>titel</i>	Titel der Karte
<i>description</i>	Beschreibung der Karte
<i>url</i>	Pfadziel für den Knopf

Tabelle 1: Props für die Komponente PostItem

Arbeitsschritte:

1. Importiere zunächst einmal das Projekt. Führe dann im Projektordner den Befehl „npm install“ aus, um alle Dependencies zu installieren. Starte dann mit „npm start“ das Projekt und überprüfe, ob der Import funktioniert hat.
2. Schreibe einen Konstruktor für die Klasse Posts, der einen Zustand mit einem leeren Array initialisiert.
3. Die Applikation soll beim Laden zunächst standardmäßig alle Posts anzeigen. Schreibe eine Funktion componentDidMount (vgl. React.Component „componentDidMount“), die von der jsonplaceholder api (<https://jsonplaceholder.typicode.com/posts>) alle Posts fetched und in den Zustand speichert.
4. Überarbeite die render Methode, damit diese für jedes Element des Arrays im Zustand ein PostItem Element generiert und anzeigt. Übergebe dieser hierzu alle erforderlichen Props (vgl. Tabelle 1 oder PostItem.js). *Hinweis: Da es sich hierbei intern um eine Liste handelt, benötigt jedes PostItem ein key prop. Das key Attribut soll der id des jeweiligen Posts entsprechen (vgl. Vorlesung UUID)*
5. Ergänze die Komponente um eine Klassenvariable (z.B. searchString), die den Suchbegriff zwischenspeichert. Ergänze dann das Textfeld, sodass bei Veränderungen (Stichwort „onChange“) der Suchbegriff aktualisiert wird
6. Ergänze den Button, sodass bei Knopfdruck eine Suche nach bestimmten Posts ausgeführt wird. Dazu muss der Zustand aktualisiert werden.