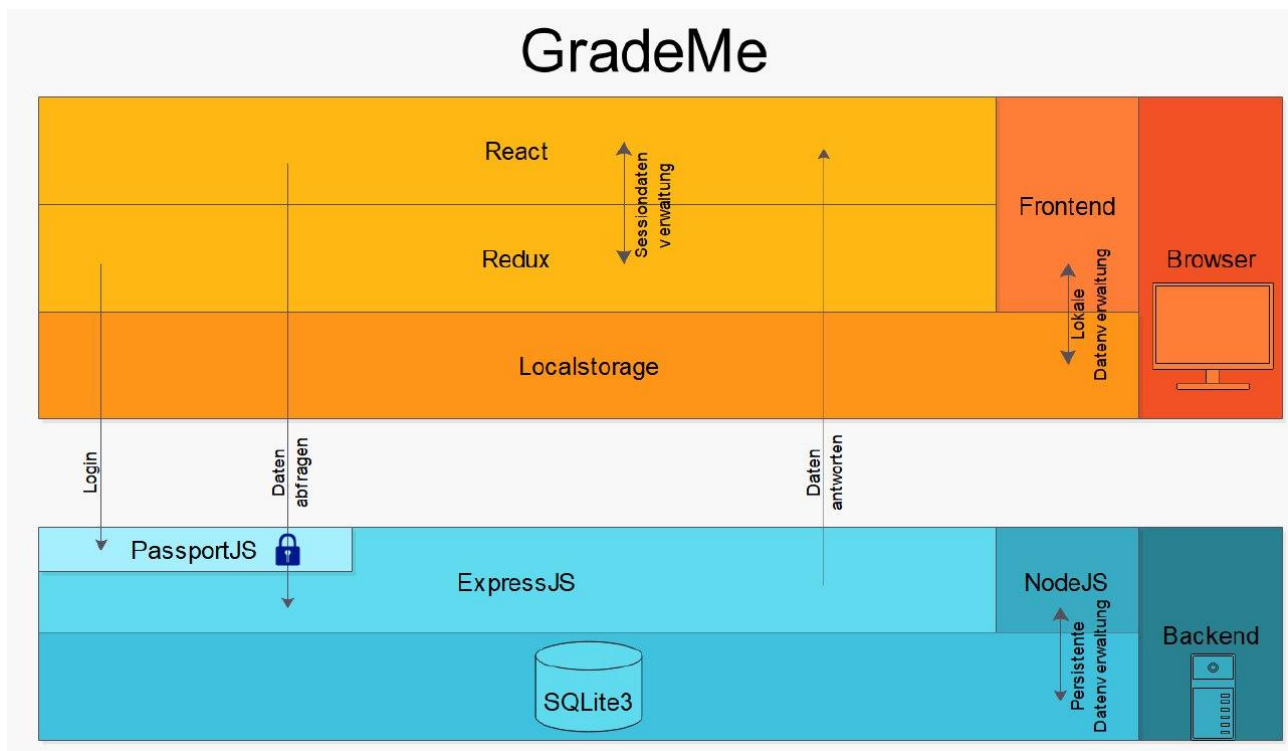


Systemarchitektur IT19A Team 6



Beschreibung

Die Systemarchitektur ist in zwei Hauptkomponenten aufgeteilt.

Zum einen gibt es den Browser, der auf dem Gerät des Endnutzers läuft. In ihm werden im sogenannten "Localstorage" Daten für längere Zeit lokal gespeichert. Zudem führt der Browser das Frontend aus, der Teil der GradeMe Anwendung, der die Schnittstelle mit dem Benutzer darstellt. Das Frontend wiederum wird durch zwei Komponenten realisiert. Für die Nutzereingaben und Datendarstellung dient eine React Instanz, zur Datenverwaltung während einer Session wird Redux verwendet.

Zum anderen gibt es das Backend, welches auf einem Server läuft. Es stellt eine SQLite3 Datenbank und einen NodeJS Server zur Verfügung.

Der NodeJS Server benutzt das ExpressJS Framework, um Webanfragen zu behandeln. In ExpressJS können sogenannte Middleware Komponenten integriert werden. Diese dienen dazu, sich um verschiedene Ebenen einer Abfrage zu kümmern, und können entweder als Bibliothek eingebunden oder vom Programmierer umgesetzt werden.

So könnte die erste Ebene die Anfrage authentifizieren, die zweite Ebene die übermittelten Daten entschlüsseln, die dritte Ebene die Daten auf Richtigkeit prüfen, die vierte Ebene die Daten weiterverarbeiten und die letzte Ebene eine adequate Antwort zurücksenden.

Sollte es in einer der Ebenen ein Problem auftreten, werden alle weiteren Ebenen übersprungen und die Fehler werden als Antwort mit einem entsprechenden Fehlercode übermittelt.

Wie im Diagramm dargestellt wird zur Authentifizierung der Nutzer die Middleware PassportJS eingebunden.

Zwischen den Komponenten der Systemarchitektur werden an unterschiedlichen Stellen Daten ausgetauscht.

Im Frontend speichert React mittels Redux aktuelle Sessiondaten und fragt ebenso Sessiondaten von Redux ab.

Zudem werden Daten, die lokal für eine längere Zeit als die aktuelle Session gespeichert werden sollen, vom Frontend im Localstorage des Browsers gespeichert und bei Bedarf wieder geladen.

Startet ein Nutzer einen Anmeldeversuch, so wird von Redux eine Loginanfrage an PassportJS im Backend gesendet. ExpressJS sendet daraufhin die Antwort an React, ob der Anmeldeversuch erfolgreich war. Falls dies der Fall ist, werden noch zusätzliche Daten übermittelt, darunter auch ein Request-Token, der bei zukünftigen Anfragen an das Backend genutzt werden muss, um sich zu authentisieren.

Stellt React nun also eine Anfrage an das Backend, so geht die Anfrage zunächst durch PassportJS und wird dort darauf geprüft, ob der Nutzer angemeldet und für diese Anfrage berechtigt ist. Ist dies der Fall, wird die Anfrage weiter an ExpressJS gereicht. ExpressJS wertet daraufhin die Anfrage aus und speichert über NodeJS entweder die Daten im Langzeitspeicher des Backends, welcher mit einer SQLite3-Datenbank realisiert ist, oder fragt Daten von diesem ab. Abschließend wird eine Antwort an React gesendet.

Falls der Nutzer nicht angemeldet oder berechtigt ist, gibt es einen Authentifizierungsfehler.