JUST DO IT
HÖHERE ABTEILUNG FÜR
INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Protokoll

Synchronisation bei mobilen Diensten

Systemtechnik
5BHIT 2017/18

Ottomaier

Version 0.1
Begonnen am 27.03.2018
Beendet am 17.04.2018

Note:
Betreuer: Pro.Borko

Inhaltsverzeichnis

1 Einführung

Diese Übung soll die möglichen Synchronisationsmechanismen bei mobilen Applikationen aufzeigen.

1.1 Ziele

Das Ziel dieser Übung ist eine Anbindung einer mobilen Applikation an ein Webservices zur gleichzeitigen Bearbeitung von bereitgestellten Informationen.

1.2 Voraussetzungen

- Grundlagen einer höheren Programmiersprache
- Grundlagen über Synchronisation und Replikation
- Grundlegendes Verständnis über Entwicklungs- und Simulationsumgebungen
- Verständnis von Webservices

1.3 Aufgabenstellung

Es ist eine mobile Anwendung zu implementieren, die einen Informationsabgleich von verschiedenen Clients ermöglicht. Dabei ist ein synchronisierter Zugriff zu realisieren. Als Beispielimplementierung soll eine Ëinkaufsliste"gewählt werden. Dabei soll sichergestellt werden, dass die Information auch im Offline-Modus abgerufen werden kann, zum Beispiel durch eine lokale Client-Datenbank.

Es ist freigestellt, welche mobile Implementierungsumgebung dafür gewählt wird. Wichtig ist dabei die Dokumentation der Vorgehensweise und des Designs. Es empfiehlt sich, die im Unterricht vorgestellten Methoden sowie Argumente (pros/cons) für das Design zu dokumentieren.

1.4 Aufgabenstellung

- Gruppengröße: 1 Person
- Anforderungen "Grundkompetenz überwiegend erfüllt"
  - Beschreibung des Synchronisationsansatzes und Design der gewählten Architektur (Interaktion, Datenhaltung)
  - Recherche möglicher Systeme bzw. Frameworks zur Synchronisation und Replikation der Daten
  - Dokumentation der gewählten Schnittstellen
- Anforderungen "Grundkompetenz zur Gänze erfüllt"

  - Implementierung der gewählten Umgebung auf lokalem System
  - Überprüfung der funktionalen Anforderungen zur Erstellung und Synchronisation der Datensätze
- Anforderungen Ërweiterte-Kompetenz überwiegend erfüllt"
  - CRUD Implementierung
  - Implementierung eines Replikationsansatzes zur Konsistenzwahrung
- Anforderungen Ërweiterte-Kompetenz zur Gänze erfüllt"
  - Offline-Verfügbarkeit
  - System global erreichbar

2 Recherche

Nach der Recherche kamen folgende Systeme in die ängere Auswahl **Couchbase Mobile Firebase Real-Time Database** und **DiffSync**.

2.1 Couchbase Mobile

Ist eine Platform um Mobile Apps zu erstellen. Die Daten werden als JSON in einer NoSQL Datenbank gespeichert.[1]
Vorteile:[1]

- Real-time synchronisation
- Open-source
- Daten sind offline verfügbar
- keine Einschränkungen bei der Datenmenge

Nachteile:[1]

- Schwererer Anfang
- Dokumentation

2.2 DiffSync

Bei DiffSync werden die als JSON Objekte mittels Websockets synchronisiert.Für die Synchronisation wird der Differential Synchronization Algorithmus verwendet.[2]

Vorteile:[1]

- einfaches Einrichten des Systemes
- keine Datenbank erforderlich(speicherung über zum Beispiel Arrays)

Nachteile:[1]

- Server notwendig
- keine Offline Verfügbarkeit

2.3 Firebase Real-Time Database

Ist eine NoSQL Datenbank. Firebase hilft dem Nutzer bei den einstieg sehr, denn sie haben eine gute Dokumentation und eine leicht zu verstehendes beziehungsweise zu benutzendes UI. Die Daten in der Datenbank werden in Form von JSON Objekten gespeichert diese sind dann wie ein Baum aufgebaut.[1]

Vorteile:[1]

- Real-time synchronisation
- Kostenlos wenn man nicht zuviel Storage braucht
- offline Arbeit ist möglich

Nachteile:

- Kosten fallen an wenn man zu viel Storage benötigt
- Man kann Firebase nicht selber hosten und man kann es nur durch die Cloud laufen lassen(d.h. nicht lokal)

2.4 Entschluss

Schlussendlich wurde dann der Entschluss gefasst mit Firebase zu arbeiten. Da es sich hierbei sowieso um kein sehr großes Projekt handelt ist einer der größten Nachteile von Firebase weg. Ein weiterer Grund für die Entscheidung für Firebase war das es wesentlich einfacher ist damit zu arbeiten. Die Dokumentation ist auch die meiste Zeit gut und es gibt eine Menge an tutorials. Auch das einfach zu benutzende UI war ein Grund warum die Entscheidung dann zu gunsten von Firebase ausfiehl.

3 Vorgangsweise

Als ersten Schritt musste man sich bei Firebase erstmal einen Account erstellen wenn dies gemacht wurde kann man auch schon ein Projekt erstellen. Nun kann man auch schon loslegen und auf den Menüpunkt Database klicken dann kann man sich zwischen den beiden Datenbanken entscheiden hierbei wurde Realtime Database gewählt. Danach kann man noch ein paar Einstellungen treffen wie den Namen der Datenbank.

Als nächsten Schritt muss man wieder auf den Menüpunkt Projektoverview gehen und dort auf Firebase zu meiner Web-App hinzufügen klicken das folgende Script muss man dann koppieren und in html file hineinkoppieren.

Nachdem man damit fertig ist kann man auf den Menüpunkt hosten gehen, dort wird einem dann genau erklärt was man machen muss und welche Befehle man in seine cmd reinschreiben muss.

Nach diesem Schritt ist dann alles miteinander verbunden und man sollte in der Lage sein seine Seite über das Firebase hosting aufzurufen.

Als letzten Schritt wurde noch eingefügt das man auf der Webseite seine gewünschten Sachen der Einkaufsliste hinzufügen kann diese werden dann in der Datenbank sowie auch auf der Webseite ausgegeben, weiters kann man auch die Datensätze synchron auf mehreren Browsern aufrufen.[3]

4 Synchronisationansatz Firebase Real-Time Database

In der Firebase Real-Time Database ist die Synchronisation wie folgt aufgebaut:
Wenn die Datenbank geupdated wird dann wird dies in der Cloud gespeichert und leitet dies an alle User die „interessiert“ sind weiter.
Wenn jemand die etwas in der Datenbank ändert ohne eine Verbindung werden diese in einen lokale Cache auf dem Gerät gespeichert sobald der User dann wieder eine Verbindung hat werden die Daten dann automatisch wieder synchronisiert.[4]
Da die Benutzeroberfläche Webbasiert ist ist man weiters noch plattformunabhängig.

4.1 Schnittstellendokumentation

Für jede Datenbank wird von Firebase für Webanwendungen ein Codesnippet für die richtige Referenzierung auf die Datenbank zur Verfügung gestellt.

```
<script src="https://www.gstatic.com/firebasejs/4.12.1/firebase.js"></script>
<script language = "javascript" type="text/javascript">
      // Initialize Firebase
      var config = {
        apiKey: "AIzaSyCOzBAP0fMC7LT8S8H2M9QQqqYx83DURwI",
        authDomain: "einkaufsliste-e6aa8.firebaseapp.com",
        databaseURL: "https://einkaufsliste-e6aa8.firebaseio.com",
        projectId: "einkaufsliste-e6aa8",
        storageBucket: "einkaufsliste-e6aa8.appspot.com",
        messagingSenderId: "613130881509"
      };
      firebase.initializeApp(config);
</script>
```

Listing 1: Referenzierungscodesnippet

Dadurch ist der Zugriff auf die Datenbank möglich.

Literatur

[1] Firebase und Couchbase. `https://dzone.com/articles/firebase-vs-couchbase-for-server-side-differences`, 2018. [Online; accessed 17-April 2018].

[2] DiffSync. `https://github.com/janmonschke/diffsync`, 2018. [Online; accessed 17-April 2018].

[3] Dokumentation. `https://firebase.google.com/docs/database/web/start`, 2018. [Online; accessed 17-April 2018].

[4] Synchronisieren. `https://www.youtube.com/watch?v=U5aeM5dvUpA`, 2018. [Online; accessed 17-April 2018].

Tabellenverzeichnis

Listings

Abbildungsverzeichnis