
Protokoll

Synchronisiered Mobile App

Systemtechnik
5BH1T 2017/18

Zaher Merna

Note:
Betreuer: Borko

Version 0.2
Begonnen am 18. April 2018
Beendet am 18. April 2018

Inhaltsverzeichnis

1	Einführung	1
1.1	Ziele	1
1.2	Voraussetzungen	1
1.3	Aufgabenstellung	1
1.4	Bewertung	1
2	Ergebnisse	3
2.1	Technologien	3
2.2	diff sync	3
2.2.1	couchbase	3
2.2.2	firebase	3
2.3	Firebase	3
2.4	CRUD-Funktionalitäten	5
2.4.1	Realtime Database	5
2.4.2	Offline-Verfügbarkeit	6

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 "Einkaufsliste" 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 Bewertung

- 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 **Erweiterte-Kompetenz überwiegend erfüllt"**
 - CRUD Implementierung
 - Implementierung eines Replikationsansatzes zur Konsistenzwahrung
- Anforderungen **Erweiterte-Kompetenz zur Gänze erfüllt"**
 - Offline-Verfügbarkeit
 - System global erreichbar

2 Ergebnisse

2.1 Technologien

2.2 diff sync

Diff synchronisation bietet die Möglichkeit mit nodejs mit einer Webseite die man zu einer Mobile Web app machen kann eine schnelle und einfache Möglichkeit an.

2.2.1 couchbase

Auch mittels couchbase ist es möglich die Daten zu synchronisieren. Der NoSQL Server ist für die Verwaltung der Daten zuständig. Sync Gateway ist die Schnittstelle zw. Server und Client. Couchbase Lite ist die Client-Datenbank.[1]

Hier gibt es auch einen Vergleich zw. den couchbase und firebase.

<https://db-engines.com/de/system/Couchbase%3BFirebase+Realtime+Database>

2.2.2 firebase

Firebase ist eine vollständig verwaltete Plattform für den Aufbau von iOS-, Android- und Web-Apps, die unter anderem eine automatische Datensynchronisierung, Authentifizierungsdienste, Messaging, Dateispeicher und Analysen bietet.

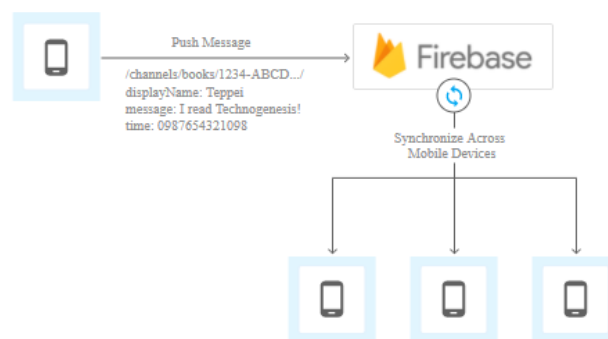


Abbildung 1: firebase

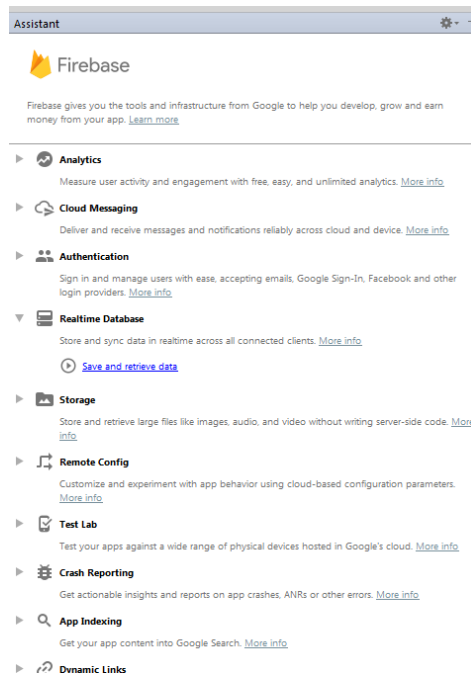
Verwendet wurde das Produkt *Firebase Realtime Database*. Welches die Client-Synchronisierung übernimmt. Leider bietet firebase die Replikation mit dem kostenpflichtigen Abo an.

Der Client sendet die Daten in der firebase Datenbank. Gespeichert wird es in einem JSON auf der Datenbank. Und Firebase synchronisiert die Clients.

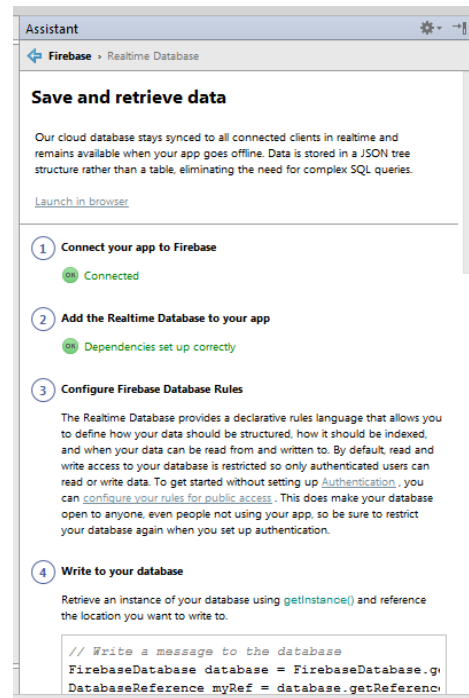
2.3 Firebase

Um Firebase mit Android Studio zu verbinden auf **Tools -> Firebase** den Assistenten starten. Dieser Assistent zeigt die Produkte die Firebase anbietet an. Firebase Realtime Database wurde

ausgewählt, da dies die Synchronisation der Mobile abwickelt.



(a) firebase-assistent



(b) firebase-assistent-step by step

Als erstes wird auf dem Button für das Connecten des Android Projekts mit firebase, dazu wird auf dem eingeloggten Google Account eine Datenbank mit dem Projektnamen aus Android erstellt. Danach werden die Dependencies runtergeladen und dem Android Projekt hinzugefügt. Nach der erfolgreichen Konfiguration von Firebase kann mit Hilfe des Guides Daten gelesen und geschrieben werden.[2][3]

Dabei wird eine Automatische konfigurierte Authentifizierung angeboten, die wird ausgeschaltet, da sie nicht benötigt wird.

Die Gui wurde über Drag and Drop im *res/layout/activity_main.xml* design. Eine *View_List* für das Anzeigen der Elemente der Datenbank wurde gewählt und 3Buttons die die CRUD - Funktionalitäten darstellen. Die GUI-Komponenten wurden mit der *MainActivity* Klasse initialisiert. Ein Array namens *adapter* für die Speicherung der Elemente und Anzeige auf der *View_List* wurde erstellt. Ein Listener wurde für das Selektieren von Elementen gesetzt.

```

1  button_add = (Button) findViewById(R.id.add);
3  button_update = (Button) findViewById(R.id.update);
   button_delete = (Button) findViewById(R.id.delete);
5  eintrag = (EditText) findViewById(R.id.editText2);
   view = (ListView) findViewById(R.id.listView);

7

9  adapter = new ArrayAdapter<String>(this,
   android.R.layout.simple_list_item_single_choice,
   listItems);
11 view.setAdapter(adapter);
   view.setChoiceMode(ListView.CHOICE_MODE_SINGLE);

13

15 view.setOnItemClickListener(
   new AdapterView.OnItemClickListener() {
       public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
17         selectedPosition = position;
       }
   });

```

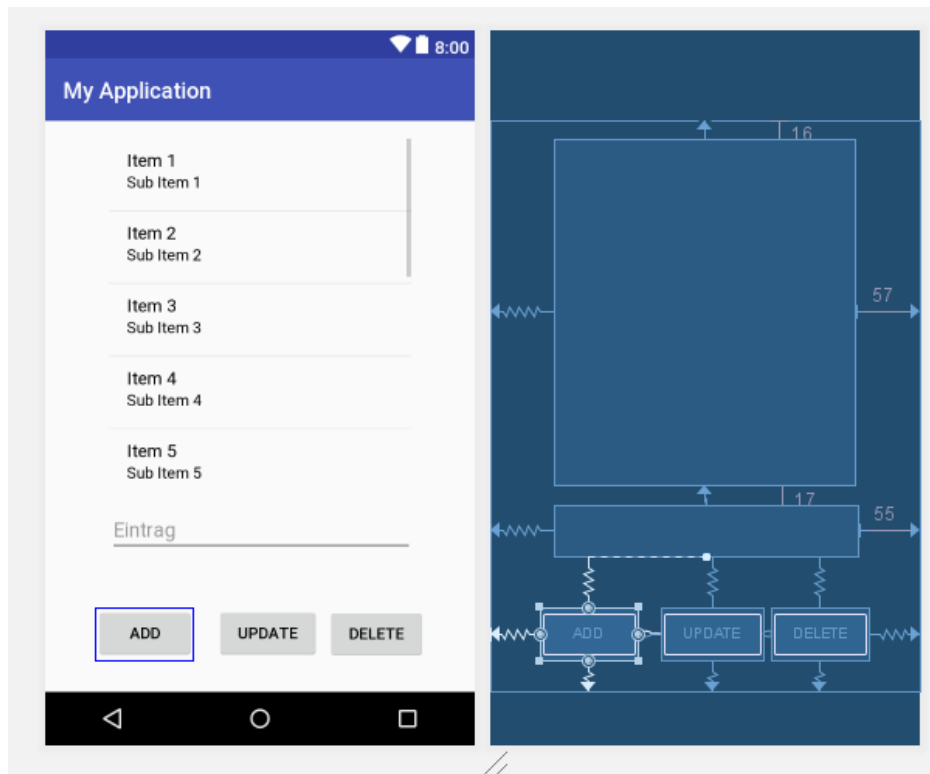


Abbildung 3: User Interface

```

    itemSelected = true;
19    button_delete.setEnabled(true);
    }
21    });

```

Listing 1: Gui, Adapter-Array für die Anzeige der Elemente, Listener für die Selektion

2.4 CRUD-Funktionalitäten

2.4.1 Realtime Database

```

1  FirebaseDatabase database = FirebaseDatabase.getInstance();
   DatabaseReference myRef=database.getReference("list");

```

Listing 2: datenbank Instanz und Referenz

Auf den 3 Button wie vorhin erwähnt sind die CRUD-Funktionalitäten anwendbar.

```

2  myRef.addChildEventListener(childListener);

4  button_add.setOnClickListener(new View.OnClickListener() {
   @Override
6  public void onClick(View v) {
   String key = myRef.push().getKey(); //jedes Element bekommt ein unique key zugewiesen
8  String text = eintrag.getText().toString();

10 myRef.child(key).child("description").setValue(text); //wert aus EditText wird ausgelesen und in die
    datenbak eingefuegt

```

```

12 adapter.notifyDataSetChanged();
    Toast.makeText(MainActivity.this, "Inserting Data", Toast.LENGTH_SHORT).show();
14 }
    });

```

Listing 3: Button Listener fuer den Add_{Button}

Der add_Button mit dem *onClickListener* wird hier veranschaulicht .Es ändert sich nicht viel zu den anderen Button-Funktionalitäten.

```

ChildEventListener childListener = new ChildEventListener() {
2   @Override
   public void onChildAdded(DataSnapshot dataSnapshot, String s) {
4       String d = dataSnapshot.child("description").getValue().toString();
       //aus der Datenbak wird das in den String gespeichert und dem adapter-array hinzugefuegt
6       adapter.add(d);
       Toast.makeText(MainActivity.this, "list", Toast.LENGTH_SHORT).show();
8       listKeys.add(dataSnapshot.getKey());
   }
}

```

Listing 4: childevntListener für das einfuegen in das Adapter-array für die Anzeige

Die dazu gehörige *ChildEventListener* wird hier veranschaulicht.

2.4.2 Offline-Verfügbarkeit

Die Persistence wird eingeschaltet und auf die Referenz das Synchronisieren ebenfalls eingeschalten.[4]

```

1 FirebaseDatabase.getInstance().setPersistenceEnabled(true);
  myRef = database.getReference("list");
3 myRef.keepSynced(true);

```

Listing 5: Offline-Verfügbarkeit

Derzeit ist es möglich das zu testen.Leider erkennt die Applikation nach ca. 2min das sie wieder online ist. Daher könnte man dies noch verbessern.

Hier kann man die firebase-Datenbank sehen und die App. Die grüne Farbe ist laut legende neu Hinzugefügt.Es werden Tomanten eingefügt.Auf der App sieht man das es gerade eingefügt wird.(inserting Data).

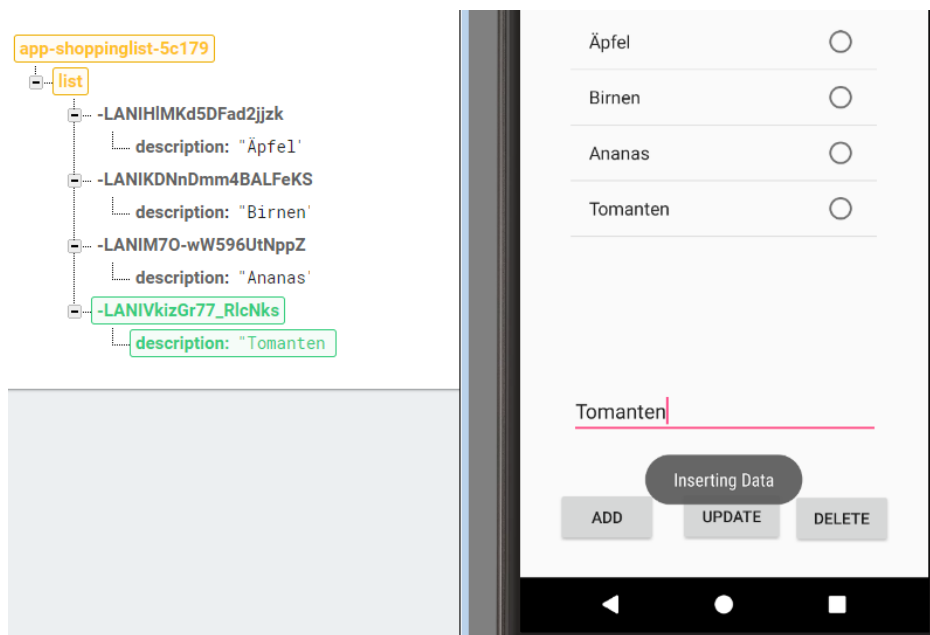


Abbildung 4: fire-Datenbank und Android-App

Literatur

- [1] Getting comfortable with couchbase mobile: Sync gateway via the command line. <https://blog.couchbase.com/getting-comfortable-with-couchbase-mobile-sync-gateway-via-the-command-line/>, February 2017.
- [2] Read and write data on android. <https://firebase.google.com/docs/database/android/read-and-write>, April 2017.
- [3] Working with firebase realtime database lists. http://www.techotopia.com/index.php/Working_with_Firebase_Realtime_Database_Lists, August 2017.
- [4] Enabling offline capabilities on android. <https://firebase.google.com/docs/database/android/offline-capabilities>, April 2018.

Tabellenverzeichnis

Listings

1	Gui, Adapter-Array für die Anzeige der Elemente, Listener für die Selektion	4
2	datenbank Instanz und Referenz	5
3	Button Listener fuer den <i>AddButton</i>	5
4	childeventListener für das einfüegen in das Adapter-array für die Anzeige	6
5	Offline-Verfügbarkeit	6

Abbildungsverzeichnis

1	firebase	3
3	User Interface	5
4	fire-Datenbank und Android-App	7