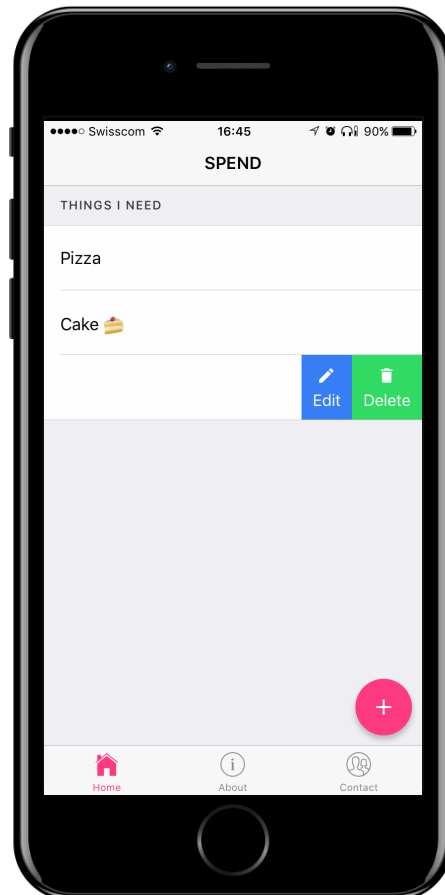


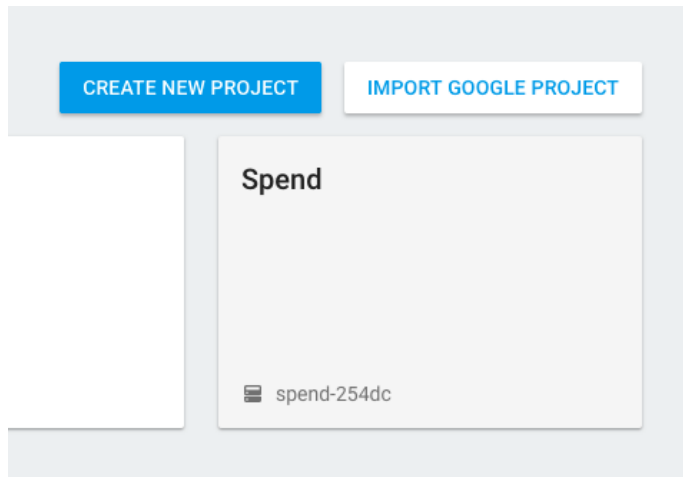
Aufgabe 1

Wir erstellen eine simple App, welche eine Liste deiner Wünsche speichert und dies in einer „Realtime Database“ speichert. Du wirst auch die CRUD Operationen implementieren.

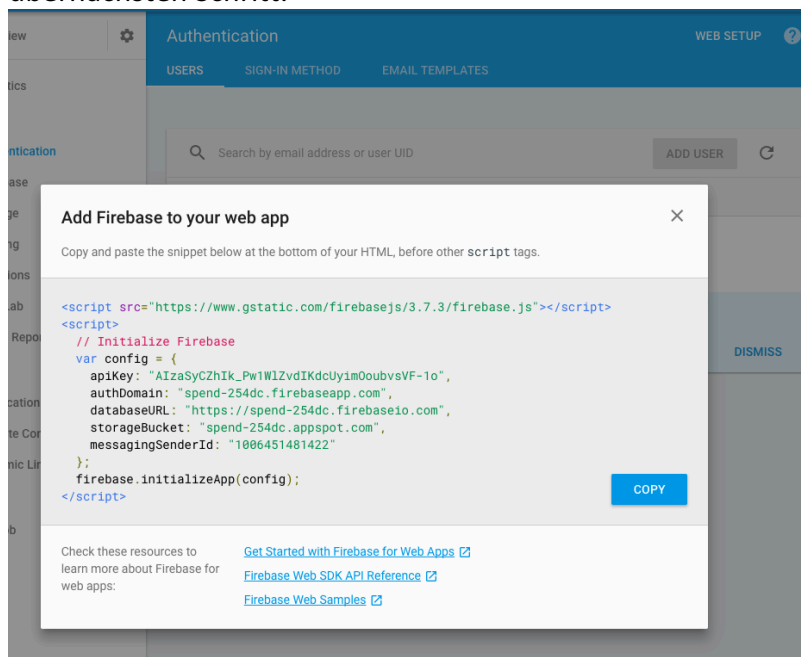
Was wir bauen



1. Öffne die Konsole und installiere Ionic Cordova und TypeScript damit sie auf der aktuellsten Version sind. Führe dazu folgenden Befehl aus. (Auf Mac evt. mit sudo vor dem Befehl).
 - a. `npm install --g ionic cordova typescript`
2. Erstelle ein neues Ionic Projekt
 - a. `sudo ionic start --v2 Spend tabs`
 - b. navigiere in das Projekt mit `cd Spend`
3. Aktualisiere app-scripts damit wir Webpack benutzen können und somit Firebase Support aus der Box haben.
 - a. `npm install @ionic/app-scripts@latest --save-dev`
4. Jetzt installieren wir noch Friebase in unser Projekt mit folgendem Befehl (führe diesen Befehl im Root-Ordner aus, also im Spend Ordner wo du dich seit Schritt 2b. befinden solltest).
 - a. `npm install angularfire2 firebase --save`
5. Weg von der Konsole in den Browser! Erstelle deinen Firebase Account
 - a. <https://firebase.google.com/>
 - b. Erstelle ein neues Projekt, nenne es Spend



- c.
- d. aktiviere die Authentifizierungsmodus damit anonymer Zugriff erlaubt ist indem du im Sidemenu auf Authentication gehst und dann auf SIGN-IN METHOD klickst. Klicke auf Anonymous in der Liste und setze es den Status auf **Enable**.
- e. Wir brauchen die Config Infos zu unserer neu erstellten Firebase DB damit wir aus unserer Daten lesen und schreiben können. Die Config Infos findest du auf dieser Authentication Seite oben rechts unter **WEB SETUP**. Kopiere den Code ab apiKey bis und mit messagingSenderId. Wir brauchen Code im übernächsten Schritt.



- f. Nun müssen wir noch die Regeln für Datenbankzugriffe ändern. Damit wir einfach los legen können definieren wir keine Regeln. Gehe dazu auf **Database** im Sidemenu und dann auf den Tab **RULES**. Aktualisiere den Eintrag mit folgendem Code

```
{
  "rules": {
    ".read": "auth == null",
    ".write": "auth == null"
  }
}
```

6. Zurück in den Code Editor! Wir importieren und initialisieren nun Firebase. Öffne die Datei src/app/app.module.ts und ergänze den Code. **Bitte verwende deinen Firebase Settings damit du auch deine Daten in der Firebase DB siehst!**

```
import { NgModule, ErrorHandler } from '@angular/core';
import { IonicApp, IonicModule, IonicErrorHandler } from 'ionic-angular';
import { MyApp } from './app.component';
import { AboutPage } from '../pages/about/about';
import { ContactPage } from '../pages/contact/contact';
import { HomePage } from '../pages/home/home';
import { TabsPage } from '../pages/tabs/tabs';

import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

// Import AngularFire Module
import { AngularFireModule } from 'angularfire2';

// AngularFire Settings aus DEINER der Firebase Konsole (WEB SETUP siehe 5e)
export const firebaseConfig = {
  apiKey: "AIzaSyCZhIk_Pw1WlZvdIKdcUyimOoubvsVF-1o",
  authDomain: "spend-254dc.firebaseio.com",
  databaseURL: "https://spend-254dc.firebaseio.com",
  storageBucket: "spend-254dc.appspot.com",
  messagingSenderId: "1006451481422"
};

@NgModule({
  declarations: [
    MyApp,
    AboutPage,
    ContactPage,
    HomePage,
    TabsPage
  ],
  imports: [
    IonicModule.forRoot(MyApp),
    AngularFireModule.initializeApp(firebaseConfig)
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    AboutPage,
    ContactPage,
    HomePage,
    TabsPage
  ],
})
```

7. Wir passen jetzt die View bzw. unsere Page an damit wir eine Liste anzeigen können und Einträge hinzufügen können. Öffne home.html und passe den Code an.

```
<ion-header>
  <ion-navbar>
    <ion-title>SPEND</ion-title>
  </ion-navbar>
</ion-header>

<ion-content class="outer-content">
  <ion-fab right bottom>
    <button ion-fab class="fab fab-md fab-md-accent" (click)="addItem()"><ion-
icon name="md-add"></ion-icon></button>
  </ion-fab>

  <ion-list>
    <ion-list-header>
      Things I need
    </ion-list-header>
    <ion-item-sliding *ngFor="let item of items | async">
      <ion-item (tap)="viewItem(item.$key, item.name)">
        {{item.name}}
      </ion-item>
      <ion-item-options side="right">
        <button ion-button color="primary" (click)="updateItem(item.$key,
item.name)">
          <ion-icon name="md-create"></ion-icon>
          Edit
        </button>
        <button ion-button color="secondary" (click)="removeItem(item.$key)">
          <ion-icon name="md-trash"></ion-icon>
          Delete
        </button>
      </ion-item-options>
    </ion-item-sliding>
  </ion-list>
</ion-content>
```

8. Nun müssen wir noch die Daten von Firebase aufrufen und die Funktionen addItem, updateItem und removeItem definieren. Öffne dazu home.ts und passe den Code an.

```
import { Component } from '@angular/core';

import { NavController, AlertController } from 'ionic-angular';

import { AngularFire, FirebaseListObservable } from 'angularfire2';

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
```

```

})
export class HomePage {

  // Member variable
  items: FirebaseListObservable<any>;

  constructor(public navCtrl: NavController, af: AngularFire, public alertCtrl:
AlertController) {
    this.items = af.database.list('/items');
  }

  // add an item to the list with help of a prompt
  // let is block scope definition
  addItem(){
    let prompt = this.alertCtrl.create({
      title: 'Item Name',
      message: "What do you need to spend money on?",
      inputs: [
        {
          name: 'name',
          placeholder: 'Name',
        },
      ],
      buttons: [
        {
          text: 'Cancel',
          handler: data => {
            console.log('Cancel clicked');
          }
        },
        {
          text: 'Save',
          handler: data => {
            if(data.name != ""){
              this.items.push({
                name: data.name
              });
            }
          }
        }
      ]
    });
    prompt.present();
  }

  removeItem(itemId: string){
    this.items.remove(itemId);
  }
}

```

```
    }

    updateItem(itemId, itemTitle){
      let prompt = this.alertCtrl.create({
        title: 'Item Name',
        message: "Edit the name for this item",
        inputs: [
          {
            name: 'name',
            placeholder: 'Name',
            value: itemTitle
          },
        ],
        buttons: [
          {
            text: 'Cancel',
            handler: data => {
              console.log('Cancel clicked');
            }
          },
          {
            text: 'Save',
            handler: data => {
              this.items.update(itemId, {
                name: data.name
              });
            }
          }
        ]
      });
      prompt.present();
    }
  }
}
```

9. DONE! Du kannst jetzt sofort jede Änderung im Browser oder auf der App sehen.
Kein Refresh nötig.
10. Bilde deine App damit du sie auf deinem Phone auch testen kannst.
 - a. Für iOS -> sudo ionic build ios
 - i. Der Build ist unter Spend/platforms/ios/Spend.xcodeproj
 - ii. Öffne das Projekt und schliesse dein iPhone an.
 - iii. Wenn nötig wähle dein Team Signing Profil
 - iv. Wähle dein iPhone aus der Liste der Devices oben links gleich neben dem Stop Button aus
 - v. Drücke Play
 - b. Für Android-> sudo ionic run android
 - i. Verbinde dazu dein Android Phone und stell sicher dass der Developer Mode aktiviert ist und USB debugging aktiviert ist.