

FIREBASE



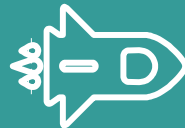
Hello

Kamil Richert

Junior Software Engineer at Spartez

Gdańsk, 13-14 lipca 2019 roku

AGENDA



1. TEORIA
2. FIREBASE W PRAKTYCE (auth, database, storage)
3. COUNTER
4. COUNTER LIVE
5. PROFILE COMPONENT I LOGOWANIE
6. TODO LIST
7. FIREBASE CHAT

FIREBASE

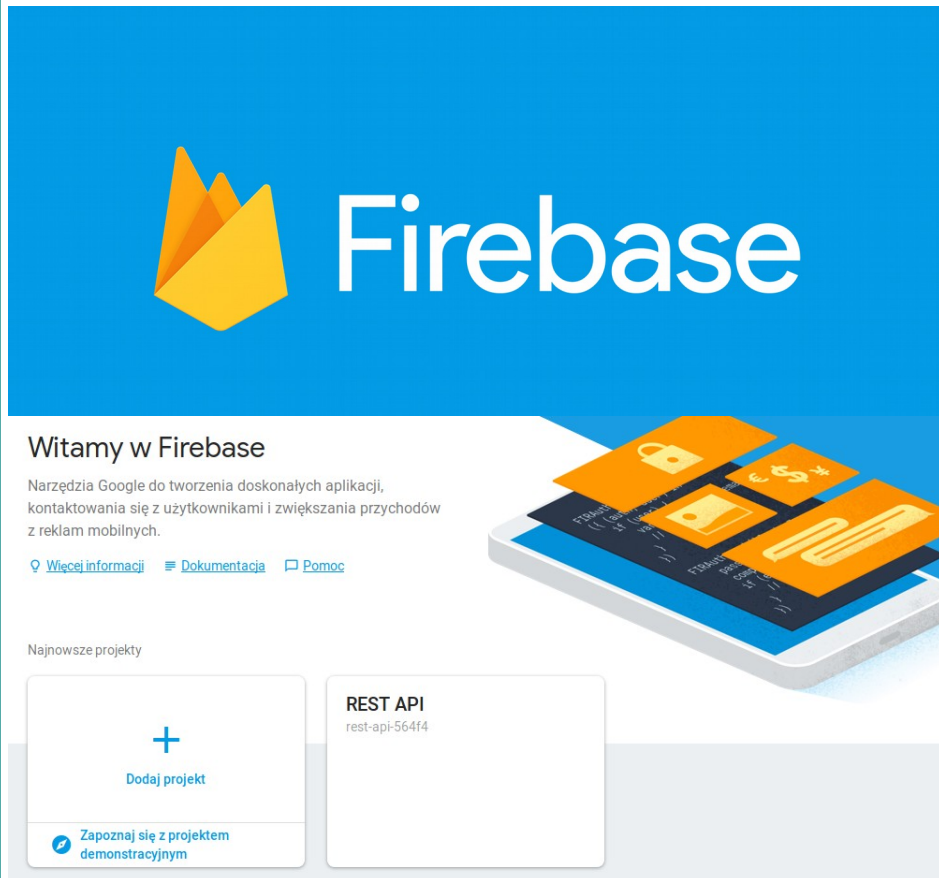


Firestore to darmowe narzędzie od Google, które w swoim panelu zawiera kilkanaście funkcjonalności pomagającym developerom i marketerom w zarządzaniu aplikacją.

Firestore

By korzystać z Firestore
potrzebujemy jedynie konta w
Google.

**Stwórzmy sobie naszą bazę
danych w Firestore. Zaczniemy
od stworzenia licznika
(countera).**



Web setup

Z Firebase możemy korzystać nie tylko w aplikacjach opartych na React, Angular czy View.

<https://firebase.google.com/docs/web/setup>

Stwórz własną bazę na potrzeby zajęć. Póki co ustaw uprawnienia na testowe. Dodaj w niej obiekt people z jednym elementem o kluczu 1 posiadającym kluczem name, surname i age.

WEB

NODE.JS

Click **Copy**, then paste the code snippet into your application HTML. The code snippet should look like this:

```
<script src="https://www.gstatic.com/firebasejs/5.9.3/firebase.js"></script>
<script>
  // Initialize Firebase
  // TODO: Replace with your project's customized code snippet
  var config = {
    apiKey: "<API_KEY>",
    authDomain: "<PROJECT_ID>.firebaseapp.com",
    databaseURL: "https://<DATABASE_NAME>.firebaseio.com",
    projectId: "<PROJECT_ID>",
    storageBucket: "<BUCKET>.appspot.com",
    messagingSenderId: "<SENDER_ID>",
  };
  firebase.initializeApp(config);
</script>
```

Firestore auth

Korzystając z firestore możemy łatwo autoryzować użytkownika poprzez **firebase.auth()** oraz ustawiać uprawnienia w konsoli na stronie.

- createUserWithEmailAndPassowrd
- signInWithEmailAndPassowrd
- onAuthStateChanged
- signOut

<https://firebase.google.com/docs/auth/>

<https://firebase.google.com/docs/database/security/quickstart>

ZADANIE



Zaimplementuj odpowiednie przyciski, które będą wywoływały wcześniej napisane funkcje w `examples/index.html`. Pamiętaj by strona ciągle odświeżała dane po każdej akcji.

Składowe: **formularz logowania / rejestracji, przycik wyloguj, powitalny nagłówek.**

Firestore database

Korzystając z firestore możemy dostać się do bazy danych poprzez referencje do niej: **firebase.database().ref()** , która udostępnia szereg metod:

- set
- update
- remove
- once, on
- child

<https://firebase.google.com/docs/database/web/read-and-write>

ZADANIE



Zaimplementuj odpowiednie przyciski, które będą wywoływały wcześniej napisane funkcje w `examples/index.html`. Pamiętaj by strona ciągle odświeżała dane po każdej akcji.

Składowe: **przycisk usuwania, edycji wiersza, formularz dodawania nowego wiersza.**

Firestore storage

Korzystając z firestore możemy dostać się do bazy danych poprzez referencje do niej: **firebase.storage().ref()** , która udostępnia szereg metod:

- child
- put
- pause, resume, cancel
- getDownloadUrl
- delete

<https://firebase.google.com/docs/storage/web/upload-files>

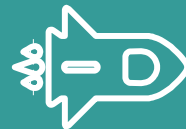
ZADANIE



Zaimplementuj odpowiednie przyciski, które będą wywoływały wcześniej napisane funkcje w `examples/index.html`. Pamiętaj by strona ciągle odświeżała dane po każdej akcji.

Składowe: **przycisk usuwania zdjęcia, zmiana zdjęcia po uploadzie.**

REACT



Wróćmy do React'a! Uffff... nareszcie, prawda?

Counter

Zaimplementuj prostego countera z użyciem metod firebase w firebase-app.



Counter Live

Websockets pozwalają nam na live update danych. Niestety ich użycie jest ograniczone.



Profile page

Zaimplementuj odpowiednie metody, które pozwolą na zmianę zdjęcia profilowego lub jego usunięcie.

Profile



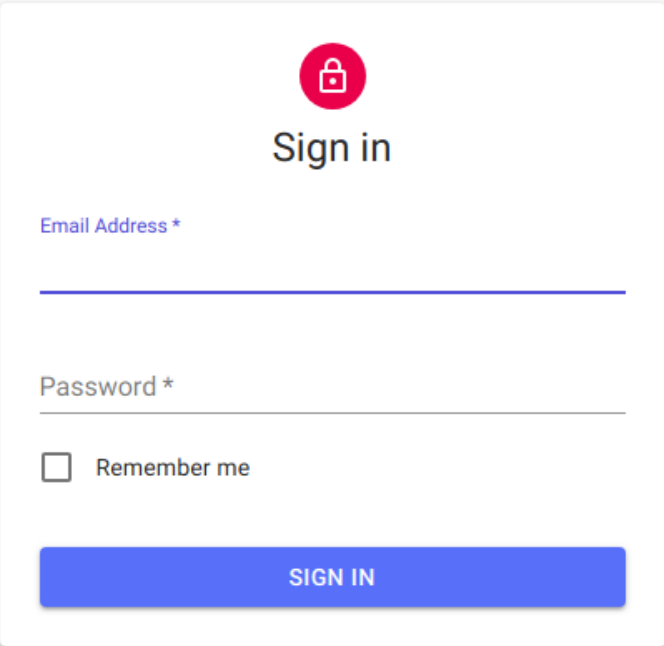
Email


User Id



Logowanie

Zaimplementuj odpowiednie metody, które pozwolą na rejestrację, logowanie i wylogowanie się. Skorzystaj z gotowych komponentów w folderze `auth`. Spraw by profile page nie był dostępny dla nie zalogowanego użytkownika.





Sign in

Email Address *

Password *

☐ Remember me

SIGN IN

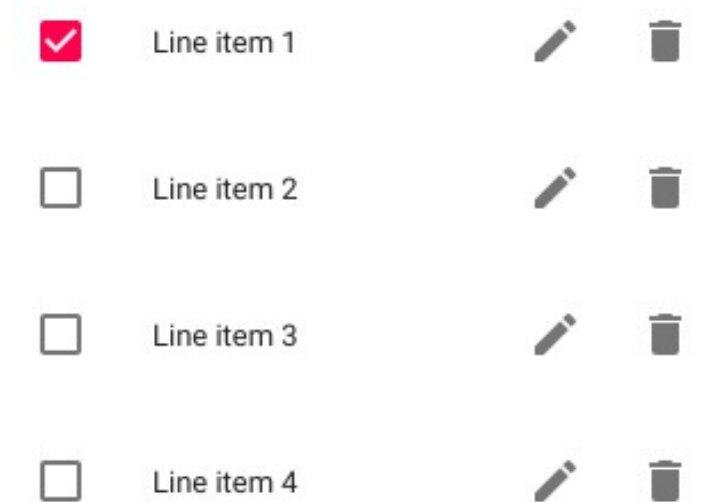
Built with ❤️ by the Material-UI team.

ToDo lista

Pora na coś bardziej złożonego:

Stworzymy listę ToDo z opcją dodawania, usuwania, edycji oraz zaznaczania wykonania. Do wszystkich poleceń zastosujemy firebase.

Od czego zaczniemy?



ToDo lista składowe

Dodajemy nowy komponent do aplikacji krok po kroku:

- tworzymy pusta class, która renderuje `<h1>ToDo</h1>`
- dodajemy nowy ścieżkę do routingu - `/todo`
- dodajemy nowy przycisk do nawigacji z odpowiednim linkiem
- upewniamy się, że nasza aplikacja poprawnie zmienia widoki
- dodajemy metodę na pobranie danych i wyświetlamy dane w renderze
- dodajemy formularz i metodą do dodawania danych
- dodajemy przyciski usuwania i zaznaczania wykonania zadania
- dodajemy przycisk edycji, który edytuje dany wiersz

CHAT



Sprawdźmy jak działają websockets w praktyce – zrobmy własny chat!

Dodaj nowy komponent z listą wiadomości, formularzem dodawania nowej. Zmień config firebase na ten z mastera. Dane są dostępne pod `/messages`'.



Dzięki

Znajdziecie mnie:

<https://www.linkedin.com/in/kamil-richert/>

<https://github.com/krichert>