



Kostenloses eBook

LERNEN

firebase-authentication

Free unaffiliated eBook created from
Stack Overflow contributors.

#firebase-
authenticati

on

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit der Firebase-Authentifizierung	2
Bemerkungen.....	2
Examples.....	2
Installation oder Setup.....	2
Zusammenfassung	2
Demo-Anleitung zur Authentifizierung und Autorisierung von Firebase v3	3
So erstellen Sie einen passwortbasierten Benutzer.....	4
Kapitel 2: Google Plus-Authentifizierung mit Android	6
Examples.....	6
Google Plus-Anmeldungsauthentifizierung.....	6
Credits	10



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [firebase-authentication](#)

It is an unofficial and free firebase-authentication ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official firebase-authentication.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit der Firebase-Authentifizierung

Bemerkungen

Dieser Abschnitt bietet einen Überblick über die Firebase-Authentifizierung und warum ein Entwickler sie verwenden möchte.

Es sollte auch alle großen Themen in der Firebase-Authentifizierung erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für die Firebase-Authentifizierung neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation oder Setup

Zusammenfassung

Eine voll funktionsfähige Demo der Firebase v3-Webauthentifizierung, die [hier](#) angezeigt [werden kann](#). Melden Sie sich mit Facebook, Github, Google, Twitter, kennwortbasierten und anonymen Konten an. Der [auf Github verfügbare](#) Code ist einfach zu lesen und zu befolgen und gut dokumentiert. Der Fokus liegt auf dem voll funktionsfähigen Authentifizierungssystem.

Passwortbasierte Benutzer erhalten einen Validierungslink. Sie können auch ihre E-Mail-Adresse und ihr Kennwort ändern. Bei beiden Ereignissen wird als zusätzliche Sicherheitsmaßnahme eine Bestätigungs-E-Mail gesendet.

Schließlich wird der Unterschied zwischen Authentifizierung, clientseitiger Autorisierung und serverseitiger Autorisierung durch Firebase-Echtzeit-Datenbanksicherheitsregeln gezeigt.

1. Voraussetzungen

1. Ein Firebase-Webprojekt. *KOSTENLOS!*
2. Eine IDE. Was ist eine IDE? Versuchen Sie es mit [Cloud9](#). *KOSTENLOS!*
3. Ein Github-, Google-, Facebook- und Twitter-Konto. *KOSTENLOS!*
4. Zwei E-Mail-Konten. *KOSTENLOS!*

2. Konfigurieren Sie Ihre IDE

2. Erstellen Sie ein HTML5-Projekt.
3. Installieren Sie die Firebase-Tools. `npm install -g firebase-tools`
4. Melden Sie sich über die Befehlszeile der Firebase-Tools bei Ihrem Firebase-Projekt an. `firebase login --no-localhost`
5. Richten Sie über die Befehlszeile der Firebase-Tools ein Firebase-Projekt im aktuellen

Verzeichnis ein. `firebase init`

6. Klonen Sie diese Gruppe von Dateien und Ordnern in Ihre IDE. `git clone https://github.com/rhroyston/firebase-auth.git`
7. Übertragen Sie Ihr IDE-Projekt über die Firebase-Tools-Befehlszeile in Ihr Firebase-Projekt. `firebase deploy`
8. Firebase-Projekt in Ihrem Browser anzeigen. Irgendwelche defekte Bilder oder Fehler? Einfach zu beheben.
9. Sie müssen möglicherweise `href`, `src` und `background: url` in allen JS-, CSS- und allen HTML-Dateien aktualisieren, abhängig von Ihrer Web-Hosting-Ordnerstruktur.
 1. Verwenden Sie die Suchfunktion, um nach `href` und `src` zu suchen und gegebenenfalls zu aktualisieren.
 2. In der Browser-Konsole werden alle verbleibenden fehlerhaften Pfade angezeigt.
 3. Hinweis `script.js line 781 privateLink.href = "../firebase-auth/private" die ..` scheint erforderlich zu sein.
 4. Wenn alle Seiten vom Firebase-Hosting richtig dargestellt werden (keine fehlerhaften Bilder oder Konsolenfehler), fahren Sie fort.

3. Firebase konfigurieren

1. Aktivieren Sie alle 6 Formen der Authentifizierung. Befolgen Sie die Anweisungen zum Konfigurieren der Einstellungen für Social Media-Sites.
2. Passen Sie die E-Mail-Handler-URL so an, dass sie auf die URL der Firebase-Web-App + `/ ack` verweist, z. B. `https://my-app-1234/ack`.

4. Melden Sie sich bei der Web-App an

1. Melden Sie sich mit einem OAuth-Anbieter an.
2. Verwenden Sie von der Browser-Befehlszeile aus die exponierte `demo.update('mynode', 'myKey', 'myValue')`, um Ihrer Echtzeitdatenbank sichere `demo.update('mynode', 'myKey', 'myValue')` hinzuzufügen.
 1. Eine Erfolgsmeldung wird in Ihrer Browserkonsole angezeigt.
 2. Möglicherweise müssen Sie den `href` Pfad entsprechend Ihrer Ordnerstruktur aktualisieren.

```
demo.update("markup", "secureData", "<div class=\"mdl-card__title\"> <h1 class=\"mdl-card__title-text mdl-color-text--white\">Secured Data</h1> </div><div class=\"mdl-card__supporting-text mdl-typography--headline\"> <p>This is a secure card. The HTML markup that renders this card is secured in the Realtime Database. Access is determined server side so no matter what you do with JavaScript on your browser you will not be able to view this card unless you are authorized to.</p><p>Secured data can be markup, JSON, strings, numbers, etc. Your imagination is the limit!</p></div><div class=\"mdl-card__actions mdl-card--border intro-card-actions\"> <a class=\"mdl-button mdl-button--colored mdl-js-button mdl-js-ripple-effect\" href=\"../firebase-auth/\">Home</a></div>");
```

Demo-Anleitung zur Authentifizierung und Autorisierung von Firebase v3

1. Melden Sie sich bei jedem OAuth-Anbieter an

1. Beachten Sie, dass auf Ihrer Kontoseite keine Optionen zum Aktualisieren der E-Mail-Adresse oder des Kennworts vorhanden sind.
2. Beachten Sie zusätzliche Links in der Seitenmenü-Schublade?
3. Versuchen Sie, Ihr Konto zu löschen. Was geschieht?

2. Registrieren Sie sich als Passwort-Benutzer

1. Haben Sie eine Bestätigungs-E-Mail erhalten?
2. Können Sie private Daten anzeigen, bis Sie auf den Bestätigungslink geklickt haben?
3. Kannst du dein Passwort ändern?
4. Können Sie Ihre E-Mail-Adresse ändern?
5. Können Sie die E-Mail-Änderung rückgängig machen, indem Sie auf den E-Mail-Widerrufslink für E-Mail-Benachrichtigungen klicken?

3. Ausloggen

1. Welche Links gibt es in der Seitenmenüleiste?
2. Können Sie auf private Daten zugreifen?
3. Können Sie private Daten anzeigen?

So erstellen Sie einen passwortbasierten Benutzer

Sie können die Firebase-Authentifizierung verwenden, um Ihren Benutzern die Authentifizierung bei Firebase anhand ihrer E-Mail-Adressen und Kennwörter zu ermöglichen und die kennwortbasierten Konten Ihrer App zu verwalten.

In diesem Beispiel verwenden wir diese Schritte, um es für unser Android-Projekt einzurichten, das auf JavaScript basiert.

Aber bevor dies getan wird, muss dies vorher erledigt werden:

1. Fügen Sie Firebase zu Ihrem JavaScript-Projekt hinzu.
2. Wenn Sie Ihre App noch nicht mit Ihrem Firebase-Projekt verbunden haben, tun Sie dies über die Firebase-Konsole.
3. E-Mail- / Passwort-Anmeldung aktivieren: Öffnen Sie in der Firebase-Konsole den Abschnitt Auth. Aktivieren Sie auf der Registerkarte Anmeldemethode die Anmeldemethode E-Mail / Kennwort, und klicken Sie auf Speichern.

Zum Erstellen eines kennwortbasierten Benutzers mit displayName sind zwei Authentifizierungsmethoden erforderlich, nämlich `.createUserWithEmailAndPassword` und `.updateProfile`. Ich habe die letztere verschachtelt und eine einzige Funktion erstellt, die beide Methoden aus Gründen der Benutzerfreundlichkeit auslöst.

```
function registerPasswordUser(email, displayName, password, photoURL) {
  var user = null;
  //NULLIFY EMPTY ARGUMENTS
  for (var i = 0; i < arguments.length; i++) {
    arguments[i] = arguments[i] ? arguments[i] : null;
  }
}
```

```
}
auth.createUserWithEmailAndPassword(email, password)
.then(function () {
  user = auth.currentUser;
  user.sendEmailVerification();
})
.then(function () {
  user.updateProfile({
    displayName: displayName,
    photoURL: photoURL
  });
})
.catch(function(error) {
  console.log(error.message, 7000);
});
console.log('Validation link was sent to ' + email + '.');
}
```

Erste Schritte mit der Firebase-Authentifizierung online lesen: <https://riptutorial.com/de/firebase-authentication/topic/4744/erste-schritte-mit-der-firebase-authentifizierung>

Kapitel 2: Google Plus-Authentifizierung mit Android

Examples

Google Plus-Anmeldungsauthentifizierung

Benutzer mit **Plus-** Login authentifizieren

onCreate

```
GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client_id))
    .requestScopes(new Scope(Scopes.PLUS_LOGIN))

    .requestEmail()
    .build();
mGoogleApiClient = new GoogleApiClient.Builder(this)
    .enableAutoManage(this /* FragmentActivity */, this /* OnConnectionFailedListener */)
    .addConnectionCallbacks(this)
    .addOnConnectionFailedListener(this)
    .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
    .addApi(Plus.API)
    .build();
```

am Start()

```
protected void onStart() {
    super.onStart();
    mGoogleApiClient.connect();
    mAuth.addAuthStateListener(mAuthListener);
}

protected void onStop() {
    super.onStop();
    if (mAuthListener != null) {
        mAuth.removeAuthStateListener(mAuthListener);
    }
    if (mGoogleApiClient.isConnected()) {
        mGoogleApiClient.disconnect();
    }
}

@Override
public void onConnected(Bundle bundle) {
    mSignInClicked = false;
    getProfileInformation();
    //getGoogleOAuthTokenAndLogin();
    Toast.makeText(this, "User is connected! (in onConnected
MainActivity)", Toast.LENGTH_LONG).show();
    // Get user's information
```



```

// Update the UI after signin
// updateUI(true);
}

@Override
public void onConnectionSuspended(int i) {

    mGoogleApiClient.connect();
    //updateUI(false);
}

@Override
public void onConnectionFailed(ConnectionResult connectionResult) {

    if (!connectionResult.hasResolution()) {
        GooglePlayServicesUtil.getErrorDialog(connectionResult.getErrorCode(), this,
            0).show();
    } else if (connectionResult.hasResolution()) {
        mConnectionResult = connectionResult;
        resolveSignInError();
    }
}

private void resolveSignInError() {
    Log.e("pavan", "User RESOLVE SIGN IN ERROR CALLED OUT OF IF");
    if (mSignInClicked) {
        if (mConnectionResult.hasResolution()) {
            try {
                Log.e("pavan", "User RESOLVE SIGN IN ERROR CALLED OUT OF IF TRY");
                mIntentInProgress = true;
                mConnectionResult.startResolutionForResult(this, GOOGLE_SIGNIN);
            } catch (IntentSender.SendIntentException e) {
                Log.e("pavan", "User RESOLVE SIGN IN ERROR CALLED OUT OF IF CATCH");
                mIntentInProgress = false;
                mGoogleApiClient.connect();
            }
        }
    }
}
}

```

Profilinformationen abrufen

```

private void getProfileInformation() {
    try {
        if (Plus.PeopleApi.getCurrentPerson(mGoogleApiClient) != null) {
            Person currentPerson = Plus.PeopleApi
                .getCurrentPerson(mGoogleApiClient);
            String personName = currentPerson.getDisplayName();
            String personPhotoUrl = currentPerson.getImage().getUrl();
            String personGooglePlusProfile = currentPerson.getUrl();
            String email = Plus.AccountApi.getAccountName(mGoogleApiClient);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Authentifizierung mit Firebase,

```
private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
//AuthCredential credential = GoogleAuthProvider.getCredential(tokenCredential, null);
AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(),null);
 mAuth.signInWithCredential(credential)
    .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            Log.d("TAG", "signInWithCredential:onComplete:" + task.isSuccessful());

            // If sign in fails, display a message to the user. If sign in succeeds
            // the auth state listener will be notified and logic to handle the
            // signed in user can be handled in the listener.
            if (!task.isSuccessful()) {
                Log.w("TAG", "signInWithCredential", task.getException());
                Toast.makeText(MainActivity.this, "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
            }else{
                SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
                Log.e("SahajLOG", "Login PREF ISSSSSSSS ONACTIVITYRESULT
"+prefs.getBoolean("AuthByGplus", AuthByGplus));
                prefs.edit().putBoolean("AuthByGplus", true).commit();
                Log.e("SahajLOG", "Login PREF ISSSSSSSS ONACTIVITYRESULT IFTRUE..
"+prefs.getBoolean("AuthByGplus", AuthByGplus));
                Intent intent=new Intent(getApplicationContext(),MainActivity.class);
                startActivity(intent);
                finish();
            }
            // [START_EXCLUDE]
            // hideProgressDialog();
            // [END_EXCLUDE]
        }
    });
}
```

onActivityResult

```
@Override
protected void onActivityResult(int requestCode, int responseCode,
    Intent intent) {

if (requestCode == GOOGLE_SIGIN) {
    if (responseCode != RESULT_OK) {
        mSignInClicked = false;
    }
    mIntentInProgress = false;
    if (!mGoogleApiClient.isConnecting()) {
        mGoogleApiClient.connect();
        if (mGoogleApiClient.isConnected()) {
            // getGoogleOAuthTokenAndLogin();
        }
    }
    GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(intent);
    if (result.isSuccess()) {
        // Google Sign In was successful, authenticate with Firebase
        GoogleSignInAccount account = result.getSignInAccount();
    }
}
```

```

        Log.e("SahajLOG", "account " + account.getIdToken());
        //getGoogleOAuthTokenAndLogin();
        firebaseAuthWithGoogle(account);
    } else {
        // Google Sign In failed, update UI appropriately
        // [START_EXCLUDE]
        Log.e("SahajLOG", "SIGN IN FAILED");
        // [END_EXCLUDE]
    }
}

public void signInWithGplus() {
    if (!mGoogleApiClient.isConnected()) {
        mSignInClicked = true;
        Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
        startActivityForResult(signInIntent, GOOGLE_SIGNIN);
    }
}
}

```

Ausloggen

```

@Override
public void onLogout() {
    mAuth.signOut();
    if (mGoogleApiClient.isConnected()) {
        Plus.AccountApi.clearDefaultAccount(mGoogleApiClient);
        mGoogleApiClient.disconnect();
        mGoogleApiClient.connect();
        SignedInWithGoogle=false;
    }
    // Google sign out

    switchToLoginFragment();
}

```

Google Plus-Authentifizierung mit Android online lesen: <https://riptutorial.com/de/firebase-authentication/topic/8887/google-plus-authentifizierung-mit-android>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit der Firebase-Authentifizierung	Big Dude , Community , ErstwhileIII , Ron Royston
2	Google Plus-Authentifizierung mit Android	Sahaj Rana , ThunderStruct