

 無料電子ブック

学習

firebase-authentication

Free unaffiliated eBook created from
Stack Overflow contributors.

#firebase-
authenticati

on

.....	1
1: firebase	2
.....	2
Examples.....	2
.....	2
.....	2
Firestore v3	3
.....	4
2: AndroidGoogle Plus	6
Examples.....	6
Google Plus.....	6
.....	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

1: firebaseのい

このセクションでは、firebase-authenticationのと、がそれをいたいについてします。

また、のにあるきなテーマについてもし、するトピックにリンクするがあります。firebaseのためのドキュメンテーションはしいので、それらのトピックのバージョンをするがあるかもしれません。

Examples

インストールまたはセットアップ

シノプシス

Firebase v3 Webのなのデモが[ここにされます](#)。Facebook、Github、Google、Twitter、パスワードベース、アカウントでサインインしてください。Githubでなコードは、みやすく、っており、よくされています。は、にするシステムです。

パスワードベースのユーザーにはリンクがされます。また、メールアドレスとパスワードをすることもできます。これらののイベントは、のセキュリティとしてメールをします。

に、クライアントの、およびFirebaseリアルタイムデータベースセキュリティールでされたサーバーののいについてします。

1. Firebase Webプロジェクト。
 2. IDE。IDEとはですか [Cloud9](#)を試してみてください。
 3. Github、Google、Facebook、Twitterアカウント。
 4. 2つのメールアドレス。

2. IDEの

2. HTML5プロジェクトをします。
3. Firebase Toolsをインストールします。 `npm install -g firebase-tools`
4. Firebase Toolsコマンドラインをして、Firebaseプロジェクトにログインします。
`firebase login --no-localhost`
5. Firebase Toolsコマンドラインをして、のディレクトリにFirebaseプロジェクトをします。
`firebase init`
6. このファイルとフォルダのセットをIDEにコピーします。 `git clone https://github.com/rhroyston/firebase-auth.git`
7. Firebase Toolsコマンドラインをして、IDEプロジェクトをFirebaseプロジェクトにブッシュします。 `firebase deploy`
8. ブラウザでFirebaseプロジェクトをします。れたやエラーはありますかのな。
9. Webホスティングのフォルダにして、すべてのJS、CSS、およびすべてのHTMLファ

イルで、`href`、`src`、および`background: url`をするがあります。

1. をして、`href`と`src`をし、にじてします。
2. ブラウザコンソールにりのなパスのエラーがされます。
3. `script.js`781 `privateLink.href = "../firebase-auth/private" .. privateLink.href =
"../firebase-auth/private" ..`がなようです。
4. すべてのページがFirebase Hostingからしくレンダリングされたらイメージやコンソールエラーがない、します。

3. Firebaseの

1. 6つのすべてののをにします。ソーシャルメディアサイトのののってください。
2. Firebug WebアプリケーションURL + '/ack'をすようにメールアクションハンドラURLをカスタマイズします `https://my-app-1234/ack`。

4. Webアプリケーションにログインする

1. OAuthプロバイダをしてログインします。
2. ブラウザのコマンドラインから、された`demo.update('mynode', 'myKey', 'myValue')`メソッドをして、リアルタイムデータベースになマークアップをします。
 1. のメッセージがブラウザのコンソールにされます。
 2. フォルダにわせて`href`パスをするがあるかもしれません。

```
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>");
```

Firestore v3とデモウォークスルー

1. OAuthプロバイダをしたログイン

1. メールアドレスまたはパスワードオプションのは、[アカウント]ページにはされません。
2. サイドメニューのきしになリンクがあることにしてください。
3. アカウントをしてみてください。がこるのですか

2. パスワードベースのユーザーとしてする

1. メールをけりましたか
2. リンクをクリックするまでプライベートデータをできますか
3. パスワードをできますか

4. メールアドレスをできますか
5. メールのメールのリンクをクリックして、メールのをにすことはできますか

3. ログアウト

1. サイドメニューのきしにはどのようなリンクがありますか
2. なデータにアクセスできますか
3. なデータをすることが出来ますか

パスワードベースのユーザーをする

Firestoreをする、メールアドレスとパスワードをしてFirestoreでをい、アプリケーションのパスワードベースのアカウントをできます。

ここでは、JavaScriptをベースにしたAndroidプロジェクトにこのをしています。

しかし、これをうに、これはにうがあります

1. JavaScriptプロジェクトにFirestoreをします。
2. まだFirestoreプロジェクトにアプリケーションをしていないは、Firestoreコンソールからしてください。
3. Email / PasswordのサインインをにするFirestoreコンソールで、Authセクションをきます。
[サインイン]タブで、メール/パスワードのサインインをにして[]をクリックします。

displayNameをつパスワードベースのユーザー、つまり.createUserWithEmailAndPasswordと、updateProfileをするには、2つのがです。はをネストし、いやしさのためにこれらのメソッドをともするのをしました。

```
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 + '.');
}
```

オンラインでfirebaseのいをもむ <https://riptutorial.com/ja/firebase-authentication/topic/4744/firebaseのい>

2: AndroidでのGoogle Plus

Examples

Google Plus ログイン

プラスログインでユーザーをする

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();
```

onStart

```
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
MActivity)", 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();
            }
        }
    }
}
}

```

プロフィールをする

```

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();
        }
    }
}

```

Firebaseによる、

```

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_SIGNIN) {
    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);
    }
}
}

```

ログアウト

```

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

    switchToLoginFragment();
}

```

オンラインでAndroidでのGoogle Plusをむ <https://riptutorial.com/ja/firebase-authentication/topic/8887/androidでのgoogle-plus>

クレジット

S. No		Contributors
1	firebaseのい	Big Dude , Community , ErstwhileIII , Ron Royston
2	AndroidでのGoogle Plus	Sahaj Rana , ThunderStruct