



EBook Gratis

APRENDIZAJE

firebase-authentication

Free unaffiliated eBook created from
Stack Overflow contributors.

#firebase-

authenticati

on

Tabla de contenido

Acerca de.....	1
Capítulo 1: Comenzando con la autenticación firebase.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Sinopsis.....	2
Autenticación de Firebase v3 y tutorial de demostración.....	3
Cómo crear un usuario basado en contraseña.....	4
Capítulo 2: Autenticación de Google Plus con Android.....	6
Examples.....	6
Autenticación de inicio de sesión de Google Plus.....	6
Creditos.....	10

Acerca de

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

Capítulo 1: Comenzando con la autenticación firebase

Observaciones

Esta sección proporciona una descripción general de qué es la autenticación Firebase y por qué un desarrollador puede querer usarla.

También debe mencionar cualquier tema grande dentro de la autenticación de base de fuego, y vincular a los temas relacionados. Dado que la Documentación para la autenticación con base de fuego es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Sinopsis

Una demostración completamente funcional de Firebase v3 Autenticación web se puede ver [aquí](#) . Inicie sesión con Facebook, Github, Google, Twitter, cuentas basadas en contraseñas y cuentas anónimas. El código, [disponible en Github](#) , es fácil de leer y seguir y está bien documentado. El foco está en el sistema de autenticación completamente funcional.

Los usuarios basados en contraseña reciben un enlace de validación. También pueden cambiar su dirección de correo electrónico y contraseña: ambos eventos envían un correo electrónico de verificación como una medida de seguridad adicional.

Por último, se demuestra la diferencia entre la autenticación, la autorización del lado del cliente y la autorización del lado del servidor aseguradas a través de las Reglas de seguridad de la base de datos en tiempo real de Firebase.

1. Prerrequisitos

1. Un proyecto web de Firebase. ¡*GRATIS!*
2. Un IDE. ¿Qué es un IDE? Prueba [Cloud9](#) . ¡*GRATIS!*
3. Una cuenta de Github, Google, Facebook y Twitter. ¡*GRATIS!*
4. Dos cuentas de correo electrónico. ¡*GRATIS!*

2. Configure su IDE

2. Crea un proyecto HTML5.
3. Instale las herramientas de Firebase. `npm install -g firebase-tools`
4. Usando la línea de comandos de las herramientas de Firebase, inicie sesión en su proyecto de Firebase. `firebase login --no-localhost`

5. Usando la línea de comandos de las herramientas de Firebase, configure un proyecto de Firebase en el directorio actual. `firebase init`
6. Clone este conjunto de archivos y carpetas a su IDE. `git clone https://github.com/rhroyston/firebase-auth.git`
7. Usando la línea de comandos de las herramientas de Firebase, empuje su proyecto IDE a su proyecto de Firebase. `firebase deploy`
8. Ver el proyecto Firebase en su navegador. ¿Imágenes rotas o errores? Fácil solución a continuación.
9. Es posible que deba actualizar `href`, `src` y `background: url` en todos los archivos JS, CSS y HTML, según la estructura de la carpeta de alojamiento web.
 1. Use la función Buscar para buscar `href` y `src` y actualice según sea necesario.
 2. La consola del navegador mostrará los errores restantes de las rutas incorrectas.
 3. Note `script.js` line 781 `privateLink.href = "../firebase-auth/private"` el `..` parece ser requerido.
 4. Una vez que todas las páginas se representen correctamente desde Firebase Hosting (sin imágenes rotas o errores de consola), continúe.

3. Configurar Firebase

1. Habilitar las 6 formas de autenticación. Siga las instrucciones para configurar los ajustes del sitio de redes sociales.
2. Personalice la URL del controlador de acciones de correo electrónico para que apunte a la URL de la aplicación web de Firebase + `' / ack'`, por ejemplo, `https://my-app-1234/ack`.

4. Iniciar sesión en la aplicación web

1. Inicie sesión con un proveedor de OAuth.
2. Desde la línea de comandos del navegador, use el `demo.update('mynode', 'myKey', 'myValue')` para agregar un marcado seguro a su base de datos en tiempo real.
 1. Un mensaje de éxito aparecerá en la consola de su navegador.
 2. Es posible que deba actualizar la ruta `href` para que coincida con la estructura de su carpeta.

```
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>");
```

Autenticación de Firebase v3 y tutorial de demostración

1. Inicie sesión utilizando cada proveedor OAuth

1. Tenga en cuenta que la actualización de las opciones de dirección de correo electrónico o contraseña no están presentes en su página de Cuenta.
2. ¿Observa algún enlace extra en el cajón del menú lateral?
3. Intenta borrar tu cuenta. ¿Lo que pasa?

2. Registrarse como usuario basado en contraseña

1. ¿Recibiste un correo electrónico de verificación?
2. ¿Puede ver datos privados hasta que haga clic en el enlace de verificación?
3. ¿Puedes cambiar tu contraseña?
4. ¿Puedes cambiar tu dirección de correo electrónico?
5. ¿Puede deshacer el cambio de correo electrónico haciendo clic en el enlace de revocación de correo electrónico de notificación de cambio de correo electrónico?

3. Cerrar sesión

1. ¿Qué enlaces están presentes en el cajón del menú lateral?
2. ¿Se puede acceder a los datos privados?
3. ¿Puedes ver datos privados?

Cómo crear un usuario basado en contraseña

Puede usar la autenticación Firebase para permitir que sus usuarios se autenticuen con Firebase usando sus direcciones de correo electrónico y contraseñas, y para administrar las cuentas basadas en contraseña de su aplicación.

En este ejemplo, utilizamos estos pasos para configurarlo para nuestro proyecto de Android que se basa en JavaScript.

Pero antes de hacerlo, esto es lo que hay que hacer antes:

1. Agrega Firebase a tu proyecto de JavaScript.
2. Si aún no ha conectado su aplicación a su proyecto Firebase, hágalo desde la consola Firebase.
3. Habilitar el inicio de sesión de correo electrónico / contraseña: en la consola de Firebase, abra la sección Autenticación. En la pestaña Método de inicio de sesión, active el método de inicio de sesión de correo electrónico / contraseña y haga clic en Guardar.

Se requieren 2 métodos de autenticación para crear un usuario basado en contraseña con el nombre de usuario, a saber, `.createUserWithEmailAndPassword` y `.updateProfile`. He anidado este último y he creado una función única que activa estos dos métodos para facilitar su uso.

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

Lea Comenzando con la autenticación firebase en línea: <https://riptutorial.com/es/firebase-authentication/topic/4744/comenzando-con-la-autenticacion-firebase>

Capítulo 2: Autenticación de Google Plus con Android

Examples

Autenticación de inicio de sesión de Google Plus

Autenticar usuario con inicio de sesión **Plus**

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
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();
            }
        }
    }
}
}

```

Obtener información de perfil

```

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();
        }
    }
}

```

Autenticando con 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);
    }
}
}

```

Cerrar sesión

```

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

    switchToLoginFragment();
}

```

Lea Autenticación de Google Plus con Android en línea: <https://riptutorial.com/es/firebase-authentication/topic/8887/autenticacion-de-google-plus-con-android>

Creditos

S. No	Capítulos	Contributors
1	Comenzando con la autenticación firebase	Big Dude , Community , ErstwhileIII , Ron Royston
2	Autenticación de Google Plus con Android	Sahaj Rana , ThunderStruct