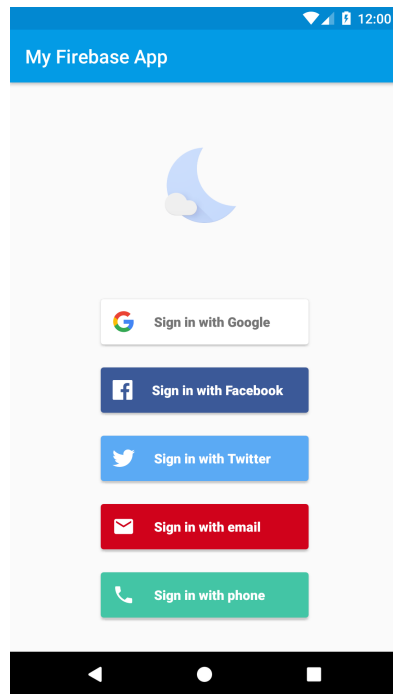


Agrega acceso a tu app para Android fácilmente con FirebaseUI



FirebaseUI (<https://github.com/firebase/firebaseui-android>) es una biblioteca creada a partir del SDK de Firebase Authentication que proporciona flujos directos de IU para usar en la app. FirebaseUI proporciona los siguientes beneficios:

- **Varios proveedores:** Flujos de acceso para correo electrónico y contraseña, vínculos mediante correo electrónico, autenticación telefónica, Acceso con Google y acceso con Facebook, Twitter y GitHub.
- **Administración de cuentas:** Flujos para controlar las tareas de administración de cuentas, como la creación de cuentas y el restablecimiento de contraseñas.
- **Vinculación de cuentas:** Flujos para vincular de forma segura cuentas de usuarios a través de proveedores de identidad.
- **Actualización de usuarios anónimos:** Flujos para actualizar de manera segura los usuarios anónimos.
- **Temas personalizados:** Modifica el aspecto de FirebaseUI para adaptarlo a tu aplicación. Además, FirebaseUI es de código abierto, por lo que puedes hacer tu propia bifurcación del proyecto y personalizarla libremente según tus necesidades.
- **Smart Lock para contraseñas:** Integración automática en [Smart Lock para contraseñas](https://developers.google.com/identity/smartlock-passwords/android/?hl=es-419) (<https://developers.google.com/identity/smartlock-passwords/android/?hl=es-419>), que permite acceder rápidamente en varios dispositivos.

Antes de comenzar

1. Si aún no lo has hecho, agrega Firebase a tu proyecto de Android (<https://firebase.google.com/docs/android/setup?hl=es-419>).
2. En el archivo `build.gradle` de nivel de proyecto, asegúrate de incluir el repositorio Maven de Google en las secciones `buildscript` y `allprojects`.
3. Agrega las dependencias de FirebaseUI al archivo `build.gradle` de nivel de la aplicación. Si deseas admitir el acceso con Facebook o Twitter, incluye también los SDK de Facebook y Twitter:

```
dependencies {
    // ...

    implementation 'com.firebaseui:firebase-ui-auth:4.3.1'

    // Required only if Facebook login support is required
    // Find the latest Facebook SDK releases here: https://goo.gl/Ce5L94
    implementation 'com.facebook.android:facebook-android-sdk:4.x'

    // Required only if Twitter login support is required
    // Find the latest Twitter SDK releases here: https://goo.gl/E5wZvQ
    implementation 'com.twitter.sdk.android:twitter-core:3.x'
}
```

El SDK de FirebaseUI Auth tiene dependencias transitivas del SDK de Firebase y del SDK de Servicios de Google Play.

4. En Firebase console ([//console.firebase.google.com/?hl=es-419](https://console.firebase.google.com/?hl=es-419)), abre la sección **Authentication** y habilita los métodos de acceso que desees admitir. Algunos métodos de acceso requieren información adicional, que se puede encontrar en la consola para programadores del servicio.
5. Si admites el Acceso con Google y aún no especificaste la huella digital SHA-1 de la aplicación, hazlo desde la página Configuración ([//console.firebase.google.com/project/_/settings/general?hl=es-419](https://console.firebase.google.com/project/_/settings/general?hl=es-419)) de Firebase console. Consulta Cómo autenticar tu cliente ([//developers.google.com/android/guides/client-auth?hl=es-419](https://developers.google.com/android/guides/client-auth?hl=es-419)) para ver detalles sobre cómo obtener la huella digital SHA-1 de tu app.
6. Si admites el acceso con Facebook o Twitter, agrega recursos de string a `strings.xml` que especifiquen la información de identificación requerida por cada proveedor:

```
<resources>
    <!-- Facebook application ID and custom URL scheme (app ID prefixed by 'fb'). -->
    <string name="facebook_application_id" translatable="false">YOUR_APP_ID</string>
    <string name="facebook_login_protocol_scheme" translatable="false">fbYOUR_APP_ID</string>
    <!-- Twitter consumer key and secret -->
    <string name="twitter_consumer_key" translatable="false">YOUR_CONSUMER_KEY</string>
    <string name="twitter_consumer_secret" translatable="false">YOUR_CONSUMER_SECRET</string>
</resources>
```

Acceder

Para iniciar el flujo de acceso de FirebaseUI, crea un intent de acceso con los métodos de acceso de tu preferencia:

Java	Kotlin
<pre>// Choose authentication providers val providers = arrayListOf(AuthUI.IdpConfig.EmailBuilder().build(), AuthUI.IdpConfig.PhoneBuilder().build(), AuthUI.IdpConfig.GoogleBuilder().build(), AuthUI.IdpConfig.FacebookBuilder().build(), AuthUI.IdpConfig.TwitterBuilder().build()) // Create and launch sign-in intent startActivityForResult(AuthUI.getInstance() .createSignInIntentBuilder() .setAvailableProviders(providers) .build(), RC_SIGN_IN) </pre>	<pre> </pre>

[FirebaseUIActivity.kt](#) (<https://github.com/firebase/snippets-android/blob/922ad4e7d9586922d0d6636a028471985665a288>)

Cuando se complete el flujo de acceso, recibirás el resultado en `onActivityResult`:

Java	Kotlin
<pre> </pre>	<pre> </pre>

[FirebaseUIActivity.kt](#) (<https://github.com/firebase/snippets-android/blob/922ad4e7d9586922d0d6636a028471985665a288>)

Configura métodos de acceso

Autenticación mediante vínculos de correo electrónico

1. En [Firebase console](https://console.firebase.google.com/?hl=es-419) (<https://console.firebase.google.com/?hl=es-419>), abre la sección **Authentication**. En la pestaña **Método de acceso**, habilita el proveedor de **Correo electrónico/Contraseña**. Ten en cuenta que se debe habilitar el acceso mediante correo electrónico/contraseña para utilizar el acceso mediante un vínculo de correo electrónico.

2. En la misma sección, habilita el método de acceso mediante **Vínculo del correo electrónico (acceso sin contraseña)** y haz clic en **Guardar**.
3. También tendrás que habilitar Firebase Dynamic Links para usar el acceso mediante un vínculo de correo electrónico. En [Firebase console](https://console.firebase.google.com/?hl=es-419) ([//console.firebase.google.com/?hl=es-419](https://console.firebase.google.com/?hl=es-419)), haz clic en **Dynamic Links** en la sección **Crece** de la barra de navegación. Haz clic en **Comenzar** y agrega un dominio. El dominio que elijas aquí se reflejará en los vínculos de correo electrónico que se envíen a tus usuarios.
4. Llama a `enableEmailLinkSignIn` en una instancia `EmailBuilder`, para poder habilitar el acceso mediante vínculos de correo electrónico en FirebaseUI. También tendrás que proveer un objeto `ActionCodeSettings` válido con `setHandleCodeInApp` configurado como `true`. Además, debes incluir en la lista blanca la URL que pasarás a `setUrl`. Puedes hacerlo en [Firebase console](https://console.firebase.google.com/?hl=es-419) ([//console.firebase.google.com/?hl=es-419](https://console.firebase.google.com/?hl=es-419)), en Authentication > Métodos de acceso > Dominios autorizados.

```

ActionCodeSettings actionCodeSettings = ActionCodeSettings.newBuilder()
    .setAndroidPackageName(/* yourPackageName= */ ..., /* installIfNotAvailable= */ true,
    /* minimumVersion= */ null)
    .setHandleCodeInApp(true) // This must be set to true
    .setUrl("https://google.com") // This URL needs to be whitelisted
    .build();

startActivityForResult(
    AuthUI.getInstance()
        .createSignInIntentBuilder()
        .setAvailableProviders(Arrays.asList(
            new AuthUI.IdpConfig.EmailBuilder().enableEmailLinkSignIn()
                .setActionCodeSettings(actionCodeSettings).build())
        ).build(),
    RC_SIGN_IN);

```

5. Si quieres captar el vínculo en una actividad específica, sigue los pasos que se describen [aquí](https://firebase.google.com/docs/auth/android/email-link-auth?hl=es-419) (<https://firebase.google.com/docs/auth/android/email-link-auth?hl=es-419>). De lo contrario, el vínculo redireccionará a la actividad iniciadora.
6. Una vez que captes el vínculo directo, será necesario que verifiques que podemos administrarlo por ti. Si podemos hacerlo, tendrás que pasárnoslo a través de `setEmailLink`.

```

if (AuthUI.canHandleIntent(getIntent())) {
    if (getIntent().getExtras() == null) {
        return;
    }
    String link = getIntent().getExtras().getString(ExtraConstants.EMAIL_LINK_SIGN_IN);
    if (link != null) {
        startActivityForResult(
            AuthUI.getInstance()
                .createSignInIntentBuilder()
                .setEmailLink(link)
                .setAvailableProviders(getAvailableProviders())
                .build(),
            RC_SIGN_IN);
    }
}

```

7. **Opcional** El acceso mediante vínculos de correo electrónico en varios dispositivos es compatible, lo que significa que el vínculo enviado a través de tu app para Android se puede usar a fin de acceder en tus apps web o para iOS. Según la configuración predeterminada, la compatibilidad con varios dispositivos está habilitada. Para inhabilitarla, llama a `setForceSameDevice` en la instancia `EmailBuilder`.

Consulta [FirebaseUI-Web](#)

(<https://github.com/firebase/FirebaseUI-Android/tree/master/auth#configuring-email-link-sign-in>) y [FirebaseUI-iOS](#)

(<https://github.com/firebase/FirebaseUI-iOS/tree/master/Auth#configuring-email-link-sign-in>) para obtener más información.

Salir

FirebaseUI proporciona métodos prácticos para salir de Firebase Authentication y de todos los proveedores de identidad social:

Java	Kotlin
Android	Android
<pre>AuthUI.getInstance() .signOut(this) .addOnCompleteListener { // ... }</pre> <p>FirebaseUIActivity.kt (https://github.com/firebase/snippets-android/blob/922ad4e7d9586922d0d6636a028471985665a286)</p>	

También puedes borrar por completo la cuenta del usuario:

Java	Kotlin
Android	Android
<pre>AuthUI.getInstance() .delete(this) .addOnCompleteListener { // ... }</pre> <p>FirebaseUIActivity.kt (https://github.com/firebase/snippets-android/blob/922ad4e7d9586922d0d6636a028471985665a286)</p>	

Personalización

FirebaseUI usa AppCompatActivity en forma predeterminada para establecer el tema. Por lo tanto, adoptará el esquema de colores de tu app. Si necesitas mayor personalización, puedes pasar un tema y un logotipo al creador del Intent de acceso:

Java	Kotlin
Android	Android
<pre>startActivityForResult(AuthUI.getInstance() .createSignInIntentBuilder()</pre>	

```

        .setAvailableProviders(providers)
        .setLogo(R.drawable.my_great_logo) // Set logo drawable
        .setTheme(R.style.MySuperAppTheme) // Set theme
        .build(),
        RC_SIGN_IN)
    FirebaseAuthUIActivity.kt (https://github.com/firebase/snippets-android/blob/922ad4e7d9586922d0d6636a028471985665a28f

```

También puedes configurar una Política de Privacidad y Condiciones del Servicio personalizadas:

Java	Kotlin
Android	Android

```

startActivityForResult(
    AuthUI.getInstance()
        .createSignInIntentBuilder()
        .setAvailableProviders(providers)
        .setTosAndPrivacyPolicyUrls(
            "https://example.com/terms.html",
            "https://example.com/privacy.html")
        .build(),
    RC_SIGN_IN)
    FirebaseAuthUIActivity.kt (https://github.com/firebase/snippets-android/blob/922ad4e7d9586922d0d6636a028471985665a28f

```

Pasos siguientes

- Para obtener más información sobre el uso y la personalización de FirebaseUI, consulta el archivo [README](#) (<https://github.com/firebase/FirebaseUI-Android/blob/master/auth/README.md>) en GitHub.
- Si encuentras un problema en FirebaseUI y deseas informarlo, usa la [herramienta de seguimiento de problemas de GitHub](#) (<https://github.com/firebase/FirebaseUI-Android/issues>).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies?hl=es-419) (<https://developers.google.com/site-policies?hl=es-419>). Java is a registered trademark of Oracle and/or its affiliates.