

Using Core Kit SFA Android SDK

After successfully installing and initializing SingleFactorAuth, you can use it to authenticate your users and obtain their private and public keys.

The SingleFactorAuth instance natively provides the following functions:

- getKey()
- - Returns the torus key using the verifier
- ,verifierId
- &idToken
- .

getKey()

[â](#)

getKey()

To obtain a user's Torus key using the Web3Auth SFA Android SDK, you can call the `getKey()` function.

Variable Description loginParams Parameters for the login. It takes LoginParams as the value. context? Android activity context sessionTime sessionTime parameter is used for session management. The default value is 1 day ~ 86400, and max value is 7 days. It takes Long as a value.

Returns [â](#)

```
public CompletableFuture < TorusKey
```

```
getKey ( loginParams : LoginParams , context : Context ?
```

```
=
```

```
null , sessionTime : Long =
```

```
86400 ) ` On successful call, the getKey() function returns a CompletableFuture instance.
```

LoginParams

[â](#)

```
getKey(loginParams: LoginParams, context: Context? = null, sessionTime: Long = 86400)
```

- Table
- Type

Parameter Description verifier The verifier obtained from the Web3Auth Dashboard. It's a mandatory field and takes String as a value. verifierId The verifierId used for the verification. It takes String as a value. idToken The idToken of the user obtained from login provider. It takes String as a value. subVerifierInfoArray? Sub verifier info. Usually used during aggregate verifier. It takes Array as a value. class LoginParams { val verifier : String val verifierId : String val idToken : String var subVerifierInfoArray : Array < TorusSubVerifierInfo

```
?
```

```
=
```

```
null
```

```
constructor ( verifier : String , verifierId : String , idToken : String )
```

```
{ this . verifier = verifier this . verifierId = verifierId this . idToken = idToken }
```

```
constructor ( verifier : String , verifierId : String , idToken : String , subVerifierInfoArray : Array < TorusSubVerifierInfo  
)
```

```
{ this . verifier = verifier this . verifierId = verifierId this . idToken = idToken this . subVerifierInfoArray = subVerifierInfoArray }  
} Usage loginParams =
```

```
LoginParams ( "YOUR_VERIFIER_NAME" ,
```

```
"YOUR_VERIFIER_ID" ,
```

```
"YOUR_ID_TOKEN" ) ; torusKey = singleFactorAuth . getKey ( loginParams ) . get ( ) ; Note Web3Auth SFA Android SDK only works for users who havenot enabled MFA . MFA enabled users For MFA enabled users, you'll see an Error message.
```

Example

```
import android . widget . Button import android . os . Bundle import com . github . web3auth . singlefactorauth . SingleFactorAuth import com . github . web3auth . singlefactorauth . types . SingleFactorAuthArgs import com . github . web3auth . singlefactorauth . types . LoginParams import com . github . web3auth . singlefactorauth . types . TorusKey import org . torusresearch . fetchnodedetails . types . TorusNetwork import com . auth0 . android . jwt . JWT
```

```
class MainActivity :
```

```
AppCompatActivity ( )
```

```
{ // ... private
```

```
lateinit
```

```
var singleFactorAuth : SingleFactorAuth private
```

```
lateinit
```

```
var singleFactorAuthArgs : SingleFactorAuthArgs private
```

```
lateinit
```

```
var loginParams : LoginParams private
```

```
var torusKey : TorusKey ?
```

```
=
```

```
null
```

```
override
```

```
fun
```

```
onCreate ( savedInstanceState : Bundle ? )
```

```
{ super . onCreate ( savedInstanceState ) setContentView ( R . layout . activity_main )
```

singleFactorAuthArgs

```
SingleFactorAuthArgs ( TorusNetwork . TESTNET ) singleFactorAuth =
```

```
SingleFactorAuth ( singleFactorAuthArgs )
```

```
// Setup UI and event handlers val signInButton = findViewById < Button
```

```
( R . id . signIn ) signInButton . setOnClickListener
```

```
{
```

```
signIn ( )
```

```
}}}
```

```
private
```

```
fun
```

```
signIn ( )
```

```
{ val jwt =
```

```
JWT ( idToken )
```

```

// idToken is the JWT token obtained from auth provider or your custom JWT provider val sub = jwt . getClaim ( "sub" ) .
asString ( )

// get sub claims loginParams =

LoginParams ( "web3auth-firebase-examples" ,

"

sub " ,

"

idToken " ) try

{ torusKey = singleFactorAuth . getKey ( loginParams ) . get ( ) }

catch

( e : ExecutionException )

{ e . printStackTrace ( ) }

catch

( e : InterruptedException )

{ e . printStackTrace ( ) } println ( ""Private Key: { torusKey !! . privateKey . toString ( 16 ) } "" . trimIndent ( ) ) println (

""Public Address: { torusKey !! . publicAddress } "" . trimIndent ( ) ) } //... } Example You can have a look at thethis example

made on top of this SDK and try it out yourself. Edit this page Previous Authentication Next Overview

```