

Using Core Kit SFA iOS 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()

[^](#)

getKey()

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

Variable Type Description Mandatory loginParams object Login Parameters Yes

Returns^{[^](#)}

public func getKey (loginParams :

LoginParams)

async throws -

TorusKey

LoginParams

[^](#)

getKey(loginParams: LoginParams)

- Table
- Type

Parameter Type Description Mandatory verifier string Details of the verifier (verifier type, ie.torus ,metamask ,openlogin etc.)
Yes verifierId string Verifier ID's value,sub oremail value present in the idToken. Yes idToken string A newly createdJWT
Token that has not already been sent to Web3Auth or aDuplicate Token error will be thrown. Yes subVerifierInfoArray?
TorusSubVerifierInfo[] Sub verifier info No public

class

LoginParams

{ public

let verifier :

String public

let verifierId :

String public

let idToken :

String public

let subVerifierInfoArray :

[TorusSubVerifierInfo] ?

```

public
init ( verifier :
String , verifierId :
String , idToken :
String )
{ self . verifier = verifier self . verifierId = verifierId self . idToken = idToken self . subVerifierInfoArray =
nil }
init ( verifier :
String , verifierId :
String , idToken :
String , subVerifierInfoArray :
[ TorusSubVerifierInfo ] )
{ self . verifier = verifier self . verifierId = verifierId self . idToken = idToken self . subVerifierInfoArray = subVerifierInfoArray }
} Usage let loginParams =
LoginParams ( verifier :
"YOUR_VERIFIER_NAME" , verifierId :
"YOUR_VERIFIER_ID" , idToken :
"YOUR_ID_TOKEN" ) let torusKey =
try
await singleFactorAuth . getKey ( loginParams : loginParams ) NOTE Web3Auth SFA iOS SDK only works for users who
havenot enabled MFA . MFA enabled users For MFA enabled users, you'll see an Error message.

```

Example

```

import
SingleFactorAuth import
JWTDecode
/ . . ./
let jwt =
try
decode ( jwt : id_token ) let result =
try
await
SingleFactorAuth ( singleFactorAuthArgs :
. init ( network :
. CYAN ) ) . getKey ( loginParams :
. init ( verifier :
"web3auth-firebase-examples" , verifierId : jwt . subject ??
"YOUR_VERIFIER_ID" , idToken : id_token ) )
await

```

```
MainActor . run ( body :
```

```
{ privateKey = result . getPrivateKey ( ) publicKey = result . getPublicKey ( ) } )
```

Session Management[^](#)

We have also included Session Management in this SDK, so calling the initialize function to get the TorusKey value without re-logging in the user if a user has an active session will return the TorusKey struct; otherwise, it will return nil.

```
if
```

```
let savedKey =
```

```
try
```

```
await singleFactorAuth . initialize ( )
```

```
{ print ( savedKey . getPrivateKey ( ) ) print ( savedKey . getPublicKey ( ) ) }
```

[Edit this page](#) [Previous Authentication](#) [Next Web3Auth Core Kit Single Factor Auth React Native SDK](#)