

Use the MetaMask onboarding library

Sending users away from your dapp to install MetaMask presents challenges. You must inform the user to return to your dapp and refresh their browser after the installation. Your dapp detects the user's newly installed MetaMask extension only after that refresh.

You can use MetaMask's [onboarding library](#) to improve and simplify the onboarding experience. The library exposes an API to initiate the onboarding process.

During the onboarding process, the library registers your dapp as the origin of the onboarding request. MetaMask checks for this origin after the user completes the onboarding flow. If it finds an origin, the final confirmation button of the MetaMask onboarding flow indicates that the user will be redirected back to your dapp.

tip [MetaMask SDK](#) incorporates the functionality of the MetaMask onboarding library. You don't need to set up the onboarding library if you use the SDK.

Steps

```
1. Install @metamask/onboarding
2. .
3. Import the library or include it in your page:
4. // As an ES6 module
5. import
6. MetaMaskOnboarding
7. from
8. "@metamask/onboarding"
9. ;
10. // Or as an ES5 module
11. const
12. MetaMaskOnboarding
13. =
14. require
15. (
16. "@metamask/onboarding"
17. )
18. ;
19. Alternatively, you can include the prebuilt ES5 bundle that ships with the library:
20. <
21. script
22. src
23. =
24. "
25. ./metamask-onboarding.bundle.js
26. "
27.
28. <!--
29. script
30. -->
31. Create a new instance of the onboarding library:
32. const
33. onboarding
34. =
35. new
36. MetaMaskOnboarding
37. (
38. )
39. ;
40. Start the onboarding process in response to a user event (for example, a button click):
41. onboarding
42. .
43. startOnboarding
44. (
45. )
46. ;
```

Example

The following are example ways to use the onboarding library in various frameworks:

- React
- TypeScript
- Vanilla JavaScript and HTML

```
import
MetaMaskOnboarding
from
"@metamask/onboarding" ; import
React
from
"react" ;
const
ONBOARD_TEXT
=
"Click here to install MetaMask!" ; const
CONNECT_TEXT
=
"Connect" ; const
CONNECTED_TEXT
=
"Connected" ;
export
function
OnboardingButton ( )
{ const
[ buttonText , setButtonText ]
=
React . useState ( ONBOARD_TEXT ) ; const
[ isDisabled , setDisabled ]
=
React . useState ( false ) ; const
[ accounts , setAccounts ]
=
React . useState ( [ ] ) ; const onboarding =
React . useRef ( ) ;
React . useEffect ( ( )
=>
{ if
```

```

( ! onboarding . current )
{ onboarding . current
=
new
MetaMaskOnboarding ( ) ; } } ,
[ ] ) ;
React . useEffect ( ( )
=>
{ if
( MetaMaskOnboarding . isMetaMaskInstalled ( ) )
{ if
( accounts . length
0 )
{ setButtonText ( CONNECTED_TEXT ) ; setDisabled ( true ) ; onboarding . current . stopOnboarding ( ) ; }
else
{ setButtonText ( CONNECT_TEXT ) ; setDisabled ( false ) ; } } } ,
[ accounts ] ) ;
React . useEffect ( ( )
=>
{ function
handleNewAccounts ( newAccounts )
{ setAccounts ( newAccounts ) ; } if
( MetaMaskOnboarding . isMetaMaskInstalled ( ) )
{ provider // Or window.ethereum if you don't support EIP-6963. . request ( {
method :
"eth_requestAccounts"
} ) . then ( handleNewAccounts ) ; provider // Or window.ethereum if you don't support EIP-6963. . on ( "accountsChanged" ,
handleNewAccounts ) ; return
( )
=>
{ provider // Or window.ethereum if you don't support EIP-6963. . removeListener ( "accountsChanged" ,
handleNewAccounts ) ; } } } ,
[ ] ) ;
const
onClick
=
( )
=>

```

```

{ if
  ( MetaMaskOnboarding . isMetaMaskInstalled ( ) )
{ provider // Or window.ethereum if you don't support EIP-6963. . request ( {
method :
"eth_requestAccounts"
} ) . then ( ( newAccounts )
=>
setAccounts ( newAccounts ) ) ; }
else
{ onboarding . current . startOnboarding ( ) ; } } ; return
( < button

```

disabled

```
{ isDisabled }
```

onClick

```
{ onClick }
```

```
{ buttonText } </ button
```

); } The onboarding library ships with MetaMask's TypeScript types. Modify the React example as follows to get type safety:

- const onboarding =

React . useRef () ; + const onboarding =

React . useRef < MetaMaskOnboarding

(); This gives you editor auto-completion for the methods exposed by the library, and helpful documentation:

```
< html
```

lang

```
" en-CA "
```

```
< head
```

```
< title
```

```
MetaMask Onboarding Example </ title
```

```
< meta
```

charset

```
" UTF-8 "
```

```
/> </ head
```

```
< body
```

```
< h1
```

```
Sample Dapp </ h1
```

```
< button
```

id

```
" onboard "
```

```
Loading... </ button
```

```
< script
```

src

```
" ./metamask-onboarding.bundle.js "
```

```
</ script
```

```
< script
```

```
window . addEventListener ( "DOMContentLoaded" ,
```

```
( )
```

```
=>
```

```
{ const onboarding =
```

```
new
```

```
MetaMaskOnboarding ( ) ; const onboardButton =
```

```
document . getElementById ( "onboard" ) ; let accounts ;
```

```
const
```

```
updateButton
```

```
=
```

```
( )
```

```
=>
```

```
{ if
```

```
( ! MetaMaskOnboarding . isMetaMaskInstalled ( ) )
```

```
{ onboardButton . innerText
```

```
= "Click here to install MetaMask!" ; onboardButton . onclick
```

```
=
```

```
( )
```

```
=>
```

```
{ onboardButton . innerText
```

```
=
```

```
"Onboarding in progress" ; onboardButton . disabled
```

```
=
```

```
true ; onboarding . startOnboarding ( ) ; } ; }
```

```
else
```

```
if
```

```

( accounts && accounts . length
0 )
{ onboardButton . innerText
=
"Connected" ; onboardButton . disabled
=
true ; onboarding . stopOnboarding ( ) ; }
else
{ onboardButton . innerText
=
"Connect" ; onboardButton . onclick
=
async
( )
=>
{ await provider // Or window.ethereum if you don't support EIP-6963. . request ( { method :
"eth_requestAccounts" , } ) ; } ; } ;
updateButton ( ) ; if
( MetaMaskOnboarding . isMetaMaskInstalled ( ) )
{ provider // Or window.ethereum if you don't support EIP-6963. . on ( "accountsChanged" ,
( newAccounts )
=>
{ accounts = newAccounts ; updateButton ( ) ; } ) ; } } ; </ script
</ body
</ html

```

[Edit this page](#)