JavaScript SDK options

TheJavaScript version of MetaMask SDK takes the following options.

checkInstallationImmediately

- Syntax
- Example

checkInstallationImmediately :

< boolean

    checkInstallationImmediately :

true Enables or disables immediately checking if MetaMask is installed on the user's browser. Iftrue , the SDK checks for installation upon page load and sends a connection request, prompting the user to install MetaMask if it's not already installed. Iffalse , the SDK waits for the connect method to be called to check for installation.

The default isfalse .

checkInstallationOnAllCalls

- Syntax
- Example

checkInstallationOnAllCalls :

< boolean

    checkInstallationOnAllCalls :

true Enables or disables checking if MetaMask is installed on the user's browser before each RPC request. The default isfalse .

communicationLayerPreference

- Syntax
- Example

communicationLayerPreference :

< type

    communicationLayerPreference :

SOCKET The preferred communication layer to use. The default and only option isSOCKET forSocket.IO .

communicationServerUrl

- Syntax
- Example

communicationServerUrl :

< string

    communicationServerUrl :

"https://metamask-sdk-socket.metafi.codefi.network/" The URL of the communication server to use. This option is mainly used for debugging and testing the SDK.

dappMetadata

- Syntax
- Example

dappMetadata :

{ name :

< string

, url :

< string

, base64Icon :

< string

, } dappMetadata :

{ name :

"My Dapp" , url :

"https://mydapp.com" , base64Icon :

"data:image/png;base64,..." , } Metadata about the dapp using the SDK. The metadata options are:

- name
-
    - Name of the dapp
- url
-
    - URL of the dapp
- base64Icon
-
    - A base64-encoded icon

important SettingdappMetaData creates a clear and trustworthy user experience when connecting your dapp to MetaMask Mobile. MetaMask Mobile displays this metadata in the connection modal to help users identify and verify the connection request.

defaultReadOnlyChainId

- Syntax
- Example

defaultReadOnlyChainId :

< number or hexadecimal string

defaultReadOnlyChainId :

"0x1" Enables sendingread-only RPC requests to this chain ID before the user connects to MetaMask. The value is automatically updated to the chain ID used in MetaMask once connected.

enableDebug

- Syntax
- Example

enableDebug :

< boolean

enableDebug :

false Enables or disables sending anonymous analytics to MetaMask to help improve the SDK. The default istrue .

extensionOnly

- Syntax
- Example

extensionOnly :

< boolean

extensionOnly :

true Enables or disables automatically using the MetaMask browser extension if it's detected. The default istrue .

infuraAPIKey

- Syntax
- Example

infuraAPIKey :

< string

infuraAPIKey : process . env . INFURA_API_KEY TheInfura API key to use for RPC requests. Configure this option tomake read-only RPC requests from your dapp.

important UseInfura allowlists to protect against other people submitting requests to your API key. You can restrict interactions to specific addresses, origins, user agents, and request methods. We recommend using all allowlist options to maximize the security of your API key and dapp.

modals

- Syntax
- Example

modals :

< object

modals :

{ onPendingModalDisconnect :

( )

=>

{ // Custom logic for pending modal disconnect. } , install :

(params)

=>

{ // Custom install modal logic. const

{ link , debug , installer , terminate , connectWithExtension }

= params ; return

{ mount :

(link)

=>

{

/ *Custom mount logic* /

} , unmount :

(shouldTerminate)

=>

{

/ *Custom unmount logic* /

} , } ; } , otp :

({ debug , onDisconnect })

=>

{ // Custom OTP modal logic. return

{ mount :

( )

=>

{

/ *Custom mount logic* /

} , updateOTPValue :

(otpValue)

=>

{

/ *Custom OTP value update logic* /

} , unmount :

( )

=>

{

/ *Custom unmount logic* /

} , } ; } , } An object that allows you to customize the logic and UI of the displayed modals. This is useful if your dapp requires a custom way to handle connection and reconnection scenarios.

openDeeplink

- Syntax
- Example

openDeeplink :

< function

openDeeplink :

(link : string)

=>

{ if

(canOpenLink)

{ Linking . openURL (link) ; } } A function that is called to open a deeplink to the MetaMask Mobile app.

preferDesktop

- Syntax
- Example

preferDesktop :

< boolean

preferDesktop :

true When set totrue , the MetaMask browser extension is displayed as the default option in the SDK connection modal. When set tofalse , MetaMask Mobile is displayed as the default option. The default isfalse .

readonlyRPCMap

- Syntax
- Example

readonlyRPCMap :

< map

readonlyRPCMap :

{ "0x539" :

"http://localhost:8545" , } A map of RPC URLs to use foread-only RPC requests .

shouldShimWeb3

- Syntax
- Example

shouldShimWeb3 :

< boolean

shouldShimWeb3 :

false Enables or disables shimming thewindow.web3 object with the Ethereum provider returned by the SDK (useful for compatibility with older browsers). The default istrue .

timer

- Example

timer :

BackgroundTimer Used by React Native dapps to keep the dapp alive while usingreact-native-background-timer .

transports

- Syntax
- Example

transports :

< array of strings

transports :

["websocket" ,

"polling"] Sets the preference onSocket.IO transports.

ui

- Syntax

ui :

{ installer :

< function

, confirm :

< function

, } Options for customizing the SDK UI.

useDeeplink

- Syntax
- Example

useDeeplink :

< boolean

useDeeplink :

true Enables or disables using deeplinks to connect with MetaMask Mobile. Iffalse , the SDK uses universal links instead. The default isfalse .

wakeLockType

- Syntax
- Example

wakeLockType :

< type

wakeLockType :

Temporary The type of wake lock to use when the SDK is running in the background. Options areDisabled ,Temporary , andUntilResponse .

Edit this page