

useSendTransaction

Hook for creating, signing, and sending transactions to networks.

Import

...

```
import{ useSendTransaction }from'@permissionless/wagmi'
```

...

Usage

...

File index.tsx import{ useSendTransaction }from'@permissionless/wagmi' import{ parseEther }from'viem'

```
functionApp() { const{sendTransaction}=useSendTransaction()
```

```
    sendTransaction({ to:'0xd2135CfB216b74109775236E36d4b433F1DF507B', value:parseEther('0.01'), })
    }
```

```
return(      Send transaction    ) }
```

...

sendTransaction from @permissionless/wagmi returns a reference to fetch receipt and not transaction hash. Hence it should not be used to fetch transaction details or shown to the user.

Parameters

...

```
import{typeUseSendTransactionParameters }from'wagmi'
```

...

config

Config | undefined

[Config](#) to use instead of retrieving from the from nearest [WagmiProvider](#) .

index.tsx config.ts ```

File index.tsx import{ useSendTransaction }from'wagmi' import{ config }from'./config'

```
functionApp() { constresult=useSendTransaction({ config, }) }
```

...

mutation

Same mutation object that you can pass to useSendTransaction [hook from wagmi](#) .

Return Type

...

```
import{typeUseSendTransactionReturnType }from'wagmi'
```

...

sendTransaction

(variables: SendTransactionVariables, { onSuccess, onSettled, onError }) => void

The mutation function you can call with variables to trigger the mutation and optionally hooks on additional callback options.

- **variables**

- SendTransactionVariables
- The variables object to pass to the [sendTransaction](#)
- action.

- **onSuccess**

- (data: string, variables: SendTransactionVariables, context: TContext) => void
- This function will fire when the mutation is successful and will be passed the mutation's result.

- **onError**

- (error: SendTransactionErrorType, variables: SendTransactionVariables, context: TContext | undefined) => void
- This function will fire if the mutation encounters an error and will be passed the error.

- **onSettled**

- (data: string | undefined, error: SendTransactionErrorType | null, variables: SendTransactionVariables, context: TContext | undefined) => void
- - This function will fire when the mutation is either successfully fetched or encounters an error and be passed either the data or error
- - If you make multiple requests, onSuccess
- - will fire only after the latest call you've made.

sendTransactionAsync

(variables: SendTransactionVariables, { onSuccess, onSettled, onError }) => Promise

Similar to [sendTransaction](#) but returns a promise which can be awaited.

data

string | undefined

- A transaction reference
- Defaults to undefined
- The last successfully resolved data for the mutation.

error

SendTransactionErrorType | null

The error object for the mutation, if an error was encountered.

failureCount

number

- The failure count for the mutation.
- Incremented every time the mutation fails.
- Reset to 0
- when the mutation succeeds.

failureReason

SendTransactionErrorType | null

- The failure reason for the mutation retry.
- Reset to null
- when the mutation succeeds.

isError / isIdle / isPending / isSuccess

boolean

Boolean variables derived from [status](#) .

isPaused

boolean

- will be true
- if the mutation has been paused
- .
- see [Network Mode](#)
- for more information.

reset

() => void

A function to clean the mutation internal state (e.g. it resets the mutation to its initial state).

status

'idle' | 'pending' | 'error' | 'success'

- 'idle'
- initial status prior to the mutation function executing.
- 'pending'
- if the mutation is currently executing.
- 'error'
- if the last mutation attempt resulted in an error.
- 'success'
- if the last mutation attempt was successful.

submittedAt

number

- The timestamp for when the mutation was submitted.
- Defaults to 0
- .

variables

SendTransactionVariables | undefined

- The variables object passed to [sendTransaction](#)
- .
- Defaults to undefined
- .