useWaitForTransactionReceipt

Hook that waits for the transaction to be included on a block, and then returns the transaction receipt. If the transaction reverts, then the action will throw an error. Replacement detection (e.g. sped up transactions) is also supported.

Import

```

import{ useWaitForTransactionReceipt }from'@permissionless/wagmi'

```

Usage

```

File index.tsx import{ useSendTransaction, useWaitForTransactionReceipt }from"@permissionless/wagmi"

functionApp() { const{ sendTransaction, data:transactionReference, isPending }=useSendTransaction()

const{data:receipt, }= useWaitForTransactionReceipt({ id: transactionReference })

return(

## Send test transaction

{isPending&&

Sending transaction...
} {transactionReference&&isReceiptPending&&(

Awaiting confirmation of transaction.

)} {receipt&&

{receipt.status}
} sendTransaction({ to:'0xd2135CfB216b74109775236E36d4b433F1DF507B', value:parseEther('0.01'), }) }type="button">
Send Transaction ) }

```

Parameters

```

import{typeUseWaitForTransactionReceiptParameters }from'@permissionless/wagmi'

```

chainId

config['chains'][number]['id'] | undefined

ID of chain to use when fetching data.

```

File index.ts import{ useWaitForTransactionReceipt }from'wagmi' import{ mainnet }from'wagmi/chains'

functionApp() { constresult=useWaitForTransactionReceipt({ chainId: mainnet.id,
hash:'0x4ca7ee652d57678f26e887c149ab0735f41de37bcad58c9f6d3ed5824f15b74d', }) }

```

config

Config | undefined

Config to use instead of retrieving from the from nearestWagmiProvider .

::: code-group

```

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

functionApp() { constresult=useWaitForTransactionReceipt({
hash:'0x4ca7ee652d57678f26e887c149ab0735f41de37bcad58c9f6d3ed5824f15b74d', config, }) }

```

```

File config.ts import{ http, createConfig }from'wagmi' import{ mainnet, sepolia }from'wagmi/chains'

exportconstconfig=createConfig({ chains: [mainnet, sepolia], transports: {

}, })

```

:::

confirmations

number | undefined

The number of confirmations (blocks that have passed) to wait before resolving.

```

File index.ts import{ useWaitForTransactionReceipt }from'wagmi'

functionApp() { constresult=useWaitForTransactionReceipt({ confirmations:2,
hash:'0x4ca7ee652d57678f26e887c149ab0735f41de37bcad58c9f6d3ed5824f15b74d', }) }

```

onReplaced

(({ reason: 'replaced' | 'repriced' | 'cancelled'; replacedTransaction: Transaction; transaction: Transaction;
transactionReceipt: TransactionReceipt }) => void) | undefined

Optional callback to emit if the transaction has been replaced.

```

File index.ts import{ useWaitForTransactionReceipt }from'wagmi'

functionApp() { constresult=useWaitForTransactionReceipt({
hash:'0x4ca7ee652d57678f26e887c149ab0735f41de37bcad58c9f6d3ed5824f15b74d',
onReplaced:replacement=>console.log(replacement), }) }

```

pollingInterval

number | undefined

- Polling frequency (in milliseconds).
- Defaults to theConfig'spollingInterval config
- .

```

File index.ts import{ useWaitForTransactionReceipt }from'wagmi'

functionApp() { constresult=useWaitForTransactionReceipt({
hash:'0x4ca7ee652d57678f26e887c149ab0735f41de37bcad58c9f6d3ed5824f15b74d', pollingInterval:1_000, }) }
```

```

### id

string | undefined

The transaction reference to wait for.enabled set tofalse ifhash isundefined .

```

File index.ts import{ useWaitForTransactionReceipt }from'wagmi'

functionApp() { constresult=useWaitForTransactionReceipt({
id:'4ca7ee652d57678f26e887c149ab0735f41de37bcad58c9f6d3ed5824f15b74d', }) }

```

### query

Same query object that you can pass touseWaitForTransactionReceipt hook from wagmi .

## Return Type

```

import{typeUseWaitForTransactionReceiptReturnType }from'wagmi'

```

Same return type asuseWaitForTransactionReceipt fromwagmi .
```