

# useUserOpWait

[@biconomy/use-aa](#) • Hook

[@biconomy/use-aa](#) / useUserOpWait

## Description

This function will wait for the userOp to be mined and return the receipt

## Parameters

type

UserOpResponse

=

{ userOpHash :

string ; wait ( \_confirmations ? :

number ) :

Promise < UserOpReceipt

; waitForTxHash ( ) :

Promise < UserOpStatus

; } ;

## Returns

[userOpReceipt](#)

type

UserOpReceipt

=

{ userOpHash :

string ; entryPoint :

string ; paymaster :

string ; actualGasCost : Hex ; actualGasUsed : Hex ; success :

"true"

|

"false" ; reason :

string ; logs :

Array < any

; receipt :

any ; } ;

## Example

import

{ useSendTransaction , useUserOpWait , useSmartAccount , }

```

from
"@biconomy/useAA" ; import
{ polygonAmoy }
from
"viem/chains" ; import
{ encodeFunctionData , parseAbi }
from
"wagmi" ;
export
const
SendTx
=
( )
=>
{ const
{ smartAccountAddress }
=
useSmartAccount ( ) ;
const
{ mutate , data : userOpResponse , error , isPending , }
=
useSendTransaction ( ) ;
const
{ isLoading : waitIsLoading , isSuccess : waitIsSuccess , error : waitError , data : waitData , }
=
useUserOpWait ( userOpResponse ) ;
useEffect ( ( )
=>
{ if
( waitIsSuccess && waitData ?. success ===
"true" )
{ console . log ( "Successful mint: "
+ { polygonAmoy . blockExplorers . default . url } /tx/ { waitData ?. receipt ?. transactionHash } ) ; } } ,
[ waitIsSuccess ] ) ;
const
mintNftTx
=

```

```
( )
=> mutate ( { transactions :
{ to :
"0x1758f42Af7026fBbB559Dc60EcE0De3ef81f665e" , data :
encodeFunctionData ( { abi :
parseAbi ( [ "function safeMint(address _to)" ] ) , functionName :
"safeMint" , args :
[ smartAccountAddress ] , } ) , } , } ) ;
return
( < ErrorGuard
```

## errors

```
{ [ error , waitError ] }
    < Button title = " Mint NFT " onClickFunc = { mintNftTx } isLoading = { isPending || waitIsLoading } /> </
    ErrorGuard
  ) ; } ;
```

## Source

[hooks/useUserOpWait.ts:76](#) [Previous](#) [useTokenFees](#) [Next](#) [Paymaster](#)