

useSendSponsoredTransaction

[@biconomy/use-aa](#) • Hook

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

Description

Sends a transaction, using a paymaster for sponsorship.

Parameters

type

Transaction

=

{ to :

string ; value : BigNumberish |

string ; data :

string ; } ;

type

UseSendSponsoredTransactionProps

=

{ / The transaction or transactions to send */ transactions : Transaction | Transaction [] ; //The BuildUserOpOptions options. See <https://bcnmy.github.io/biconomy-client-sdk/types/BuildUserOpOptions.html> for further detail */ options ? : BuildUserOpOptions ; } ;

Returns

[userOpReceipt](#)

type

UserOpResponse

=

{ userOpHash :

string ; wait (_confirmations ? :

number) :

Promise < UserOpReceipt

 ; waitForTxHash () :

Promise < UserOpStatus

 ; } ;

Example

import

{ useSendSponsoredTransaction , useUserOpWait , useSmartAccount , }

from

"@biconomy/useAA" ; import

```

{ polygonAmoy }
from
"viem/chains" ; import
{ encodeFunctionData , parseAbi }
from
"wagmi" ;
const
SendSponsoredTx
=
( )
=>
{ const
{ smartAccountAddress }
=
useSmartAccount ( ) ;
const
{ mutate , data : userOpResponse , error , isPending , }
=
useSendSponsoredTransaction ( ) ;
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/useSendSponsoredTransaction.ts:77](#) [Previous useSendGasTokenTransaction](#) [Next useSendTransaction](#)