useSendWithdrawals

@biconomy/use-aa •Hook

@biconomy/use-aa / useSendWithdrawals

Description

Withdraws funds from Smart Account to a recipient (defaults to EOA)

Parameters

type

WithdrawalRequest

=

{ / **The address of the asset */ address : Hex ;** /The amount to withdraw. Expects unformatted amount. Will use max amount if unset _/ amount ? : bigint ; /The destination address of the funds. The second argument from the withdraw(...) function will be used as the default if left unset._ / recipient ? : Hex ; } ;

type

UseSendWithdrawalsProps

=

{ / **The BuildUserOpOptions options. See https://bcnmy.github.io/biconomy-client-sdk/types/BuildUserOpOptions.html for further detail */ options ? : BuildUserOpOptions ;** / Withdrawal requests */ withdrawalRequests ? : WithdrawalRequest [ ]

|

null ; } ;

Returns

userOpReceipt

type

UserOpResponse

=

{ userOpHash :

string ; wait (_confirmations ? :

number) :

Promise < UserOpReceipt

; waitForTxHash ( ) :

Promise < UserOpStatus

; } ;

Example

import

{ useSendWithdrawals , useUserOpWait , useSmartAccount , }

from

"@biconomy/useAA" ; import

```
{ polygonAmoy }
from
"viem/chains" ; import
{
NATIVE_TOKEN_ALIAS
}
from
"@biconomy/account" ;
export
const
Withdraw
=
( )
=>
{ const
{ smartAccountAddress }
=
useSmartAccount ( ) ;
const
{ mutate , data : userOpResponse , error , isPending , }
=
useSendWithdrawals ( ) ;
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
withdrawalHandler
=
```

( )

=> mutate ({ withdrawalRequests :

[{ token :

"0x747A4168DB14F57871fa8cda8B5455D8C2a8e90a"

} ,

// omit the amount to withdraw the full balance { address :

NATIVE_TOKEN_ALIAS , amount :

BigInt (1)

} ,] , options :

Options . Sponsored , }) ;

return

(< ErrorGuard

errors

{ [error , waitError] }

> < Button title = " Withdraw " onClickFunc = { withdrawalHandler } isLoading = { isPending || waitIsLoading } /> </ ErrorGuard
>
>) ; } ;

Source

hooks/useSendWithdrawals.ts:89 Previous useSendTransaction Next useSession