useDeploySmartAccount

@biconomy/use-aa •Hook

@biconomy/use-aa / useDeploySmartAccount

Description

Deploys a users smartAccount contract. It is useful for deploying in a moment when you know that gas prices are low, and you want to deploy the account before sending the first user operation. This step can otherwise be skipped, as the deployment will alternatively be bundled with the first user operation.

Parameters

type

UseDeploySmartAccountProps

=

{ /* *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

{ useDeploySmartAccount , useUserOpWait ,

Options

}

from

"@biconomy/useAA" ; import

{ polygonAmoy }

from

"viem/chains" ; import

{ encodeFunctionData , parseAbi }

```
from
"wagmi" ;
const
DeploySmartAccount
=
( )
=>
{ const
{ mutate , data : userOpResponse , error , isPending , }
=
useDeploySmartAccount ( ) ;
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
deployTx
=
( )
=> mutate ( { options :
Options . Sponsored , } ) ;
return
( < ErrorGuard
```

errors

```
{ [ error , waitError ] }
    < Button title = " Deploy Smart Account " onClickFunc = { deployTx } isLoading = { isPending || waitIsLoading } />
    </ ErrorGuard
    ) ; } ;
```

Source

hooks/useDeploySmartAccount.ts:69 Previous useCreateSession Next useGasEstimate