Methods

estimateUserOpGas

This method is used to estimate gas for the userOp. It returns estimates for preVerificationGas, verificationGasLimit, and callGasLimit for a given UserOperation. It requires passing a semi-valid/ dummy signature in userOp (e.g. a signature of the correct length and format).

Usage

const userOpGasResponse : UserOpGasResponse =

await bundler . estimateUserOpGas (userOp) ; Parameters

- userOp(UserOperationStruct
- , required): userOperation to calculate gas for.
- stateOverrideSet(StateOverrideSet
-): optional state override set for estimating gas for a userOperation under different blockchain states.

returns

- userOpGasResponse(Promise
-): It returns an object containing the following gas values.
- type
- UserOpGasResponse
- =
- {
- preVerificationGas
- :
- string
- ;
- verificationGasLimit
- :
- string
- ;
- callGasLimit
- :
- string
- ;
- maxPriorityFeePerGas
- :
- string
- ;
- maxFeePerGas
- :
- string
- ;
- }
- ;

sendUserOp

This method is used to execute the userOperation.

Usage

const userOpResponse : UserOpResponse =

await bundler . sendUserOp (userOp) ; Parameters

- userOp(UserOperation
- , required): userOperation to send.
- simulationParam(SimulationType
-): The simulationType enum can be of two types:* validation

-
  - which will only simulate the validation phase, checks if user op is valid but does not check if execution will succeed. By default this flag is set to validation.
-
  - validation_and_execution
-
  - checks if user op is valid and if user op execution will succeed.

returns

- userOpResponse(Promise
-): It returns an object containing the userOpHash and other methods.wait()
- method waits for the receipt until the transaction is mined.waitForTxHash()
- returns transactionHash identifier (not userOpHash) and you can later watch for receipt on your own.
- type
- UserOpResponse
- =
- {
- userOpHash
- :
- string
- ;
- wait
- (
- _confirmations
- ?
- :
- number
-)
- :
- Promise
- <
- UserOpReceipt
-
- ;
- waitForTxHash
- (
-)
- :
- Promise
- <
- UserOpStatus
-
- ;
- }
- ;

getUserOpReceipt

After usingsendUserOp you will receive auserOpResponse which contains a string calleduserOpHash

Using thisuserOpHash you can fetch theuserOpReceipt which verifies that youruserOp was handled on chain as a transaction.

Usage

const userOpReceipt =

await bundler . getUserOpReceipt ("0x....") ; Parameters

- userOpHash(string
- , required): user operation hash.

returns

- userOpReceipt(Promise

-): The full UserOpReceipt object type is shown below:
- type
- UserOpReceipt
- =
- {
- actualGasCost
- :
- Hex
- ;
- actualGasUsed
- :
- Hex
- ;
- entryPoint
- :
- string
- ;
- logs
- :
- any
- [
-]
- ;
- paymaster
- :
- string
- ;
- reason
- :
- string
- ;
- receipt
- :
- UserOperationReceipt
- [
- "receipt"
-]
- ;
- success
- :
- "true"
- |
- "false"
- ;
- userOpHash
- :
- string
- ;
- }
- ;

getUserOpByHash

Using theuserOpHash you can fetch the originaluserOp that was created with this hash.

Usage

const userOp =

await bundler . getUserOpByHash ("0x...") ; Parameters

- userOpHash(string
- , required): user operation hash.

returns

- userOp(Promise

-) : The userOperation will contain the following values:
- type
- BytesLike
- =
- Bytes
- |
- string
- ;
- type
- UserOpByHashResponse
- =
- UserOperationStruct
- &
- {
- transactionHash
- :
- string
- ;
- blockNumber
- :
- number
- ;
- blockHash
- :
- string
- ;
- entryPoint
- :
- string
- ;
- }
- ;
- type
- UserOperationStruct
- =
- {
- callData
- :
- BytesLike
- ;
- callGasLimit
- ?
- :
- number
- |
- bigint
- |
- `
- 0x
- {
- string
- }
- `
- ;
- initCode
- :
- BytesLike
- ;
- maxFeePerGas
- ?
- :
- number
- |
- bigint
- |
- `
- 0x
- {

- string
- }
- `
- ;
- maxPriorityFeePerGas
- ?
- :
- number
- |
- bigint
- |
- `
- 0x
- {
- string
- }
- `
- ;
- nonce
- :
- number
- |
- bigint
- |
- `
- 0x
- {
- string
- }
- `
- ;
- paymasterAndData
- :
- BytesLike
- ;
- preVerificationGas
- ?
- :
- number
- |
- bigint
- |
- `
- 0x
- {
- string
- }
- `
- ;
- sender
- :
- string
- ;
- signature
- :
- BytesLike
- ;
- verificationGasLimit
- ?
- :
- number
- |
- bigint
- |
- `
- 0x
- {
- string

- }
- `
- ;
- }
- ; Edit this page Previous Integration Next APIs