Methods

estimateUserOperationGas

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 estimateUserOperationGasResponse : EstimateUserOperationGas =

await gasEstimator . estimateUserOperationGas ({ userOperation , supportsEthCallStateOverride , supportsEthCallByteCodeOverride , stateOverrideSet baseFeePerGas }) ; Parameters

- userOperation(UserOperation
- , required): userOperation to calculate gas estimates for.
- stateOverrideSet(StateOverrideSet
-): optional state override set for estimating gas for a userOperation under different blockchain states.
- supportsEthCallStateOverride (boolean
-): optional param, default set to true, set to false if eth_call does not support state overrides
- supportsEthCallByteCodeOverride (boolean
-): optional param, default set to true, set to false if eth_call does not give the correct response to bytecode overrides
- baseFeePerGas (bigint
-): optional param, but required for Optimism based networks

returns

- estimateUserOperationGasResponse(Promise
-): It returns an object containing the following gas limits.
- type
- EstimateUserOperationGas
- =
- {
- callGasLimit
- :
- bigint
- ;
- verificationGasLimit
- :
- bigint
- ;
- preVerificationGas
- :
- bigint
- ;
- validAfter
- :
- number
- ;
- validUntil
- :
- number
- ;
- }
- ;

estimateVerificationGasLimit

This method is used to estimate the verificationGasLimit for a given userOperation.

Usage

const verificationGasLimitResponse : EstimateVerificationGasLimit = await gasEstimator . estimateVerificationGasLimit ({

userOperation , supportsEthCallStateOverride , supportsEthCallByteCodeOverride , stateOverrideSet , }) ; Parameters

- userOperation(UserOperation
- , required): userOperation to calculate gas estimates for.
- stateOverrideSet(StateOverrideSet
-): optional state override set for estimating gas for a userOperation under different blockchain states.
- supportsEthCallStateOverride (boolean
-): optional param, default set to true, set to false if eth_call does not support state overrides
- supportsEthCallByteCodeOverride (boolean
-): optional param, default set to true, set to false if eth_call does not give the correct response to bytecode overrides

returns

- verificationGasLimitResponse(Promise
-): It returns an object containing the verificationGasLimit, validUntil, and validAfter values
- type
- EstimateVerificationGasLimit
- =
- {
- verificationGasLimit
- :
- bigint
- ;
- validAfter
- :
- number
- ;
- validUntil
- :
- number
- ;
- }
- ;

estimateCallGasLimit

This method is used to estimate the callGasLimit for a given userOperation.

Usage

const callGasLimitResponse =

await gasEstimator . estimateCallGasLimit ({ userOperation , supportsEthCallStateOverride , supportsEthCallByteCodeOverride , stateOverrideSet , }) ; Parameters

- userOperation(UserOperation
- , required): userOperation to calculate gas estimates for.
- stateOverrideSet(StateOverrideSet
-): optional state override set for estimating gas for a userOperation under different blockchain states.
- supportsEthCallStateOverride (boolean
-): optional param, default set to true, set to false if eth_call does not support state overrides
- supportsEthCallByteCodeOverride (boolean
-): optional param, default set to true, set to false if eth_call does not give the correct response to bytecode overrides

returns

- callGasLimitResponse(Promise
-): It returns an object containing the callGasLimit value
- type
- EstimateCallGasLimit
- =
- {
- callGasLimit
- :
- bigint
- ;
- }
- ;

calculatePreVerificationGas

This method is used to estimate the preVerificationGas for a given userOperation. The exact implementation of this method is network-dependent hence make sure to use network-specific gas estimator clients

Usage

const preVerficationGasResponse = await gasEstimator . calculatePreVerificationGas ({ userOperation , baseFeePerGas , }) ; Parameters

- userOperation(UserOperation
- , required): userOperation to calculate gas estimates for.
- baseFeePerGas (bigint
-): optional param, but required for Optimism based networks

returns

- preVerificationGasResponse(Promise
-) : It returns an object containing the preVerficationGas value
- type
- CalculatePreVerificationGas
- =
- {
- preVerificationGas
- :
- bigint
- ;
- }
- ;

setEntryPointAddress

This method is used to set the entryPointAddress that is being used in the Gas Estimator instance

Usage

await gasEstimator . setEntryPointAddress ('") ; Parameters

- entryPointAddress(string
- , required): entry point address that one might need to change to.

returns

- void Edit this page Previous Integration Next Modules