

getUserOperationGasPrice

Return a User Operation based on a User Operation hash.

Usage

example.ts client.ts ```

File example.ts import{ pimlicoBundlerClient }from"./client"

```
const smartAccountClient=createSmartAccountClient({ account: safeAccount, entryPoint:ENTRYPOINT_ADDRESS_V06, chain: sepolia, bundlerTransport:http("https://api.pimlico.io/v1/sepolia/rpc?apikey=API_KEY"), middleware: { gasPrice:async()=>(await pimlicoBundlerClient.getUserOperationGasPrice()).fast,// use pimlico bundler to get gas prices, if using pimlico sponsorUserOperation: paymasterClient.sponsorUserOperation,// optional }, })
```

```
/ other gas alternatives/ const gasPriceResult=await pimlicoBundlerClient.getUserOperationGasPrice() /* * { "slow": { "maxFeePerGas": 3267781404n, "maxPriorityFeePerGas": 3267781404n }, "standard": { "maxFeePerGas": 3439769899n, "maxPriorityFeePerGas": 3439769899n }, "fast": { "maxFeePerGas": 3620810421n, "maxPriorityFeePerGas": 3620810421n } } /
```

```

## Returns

- Type:
- GetUserOperationGasPriceReturnType

The maxFeePerGas and maxPriorityFeePerGas values for the different gas price tiers. Choosing a faster gas price will mean faster inclusion by the Pimlico Bundler.

## JSON-RPC Method

[pimlico\\_getUserOperationGasPrice](#)