Gelato's Fee Oracle

How to get a quote for your relay fee before sending the request After reading this page:

- You'll understand how to query the fee oracle using either the SDK or the API directly and which methods/endpoints are available.
- You'll learn the difference between different fee modalities and trade offs: for example, allowing your user to sign off on a maximum fee they are willing to pay helping account for gas volatility, or allowing them to sign off on the exact fee but with a higher risk of non-execution. * You can query our fee oracle before the relay request to get an overall estimated fee (gas costs + Gelato fee) for your relay request.

Querying via the SDK

SDK method: isOracleActive

```
Copy constisOracleActive=async(chainId:bigint):Promise
```

Arguments:

- chainId
- : the chain ID of the network on which to check if the fee oracle is active.
-

Return Object:

- true/false
- : depending on the status of the fee oracle on the requested network.
-

SDK method: getPaymentTokens

```
Copy constgetPaymentTokens=async(chainId:bigint):Promise
```

Arguments:

- chainId
- : the chain ID of the network where you want to check if the fee oracle is active there.
-

Return Object:

- An array of strings listing all accepted payment tokens on the requested network.
-

SDK method: getEstimatedFee

```
Copy getEstimatedFee=( chainId:bigint, paymentToken:string, gasLimit:bigint, isHighPriority:boolean, gasLimitL1?:bigint
):Promise
```

Arguments:

- chainId
- : the chain ID of the network where you want to check if the fee oracle is active there.
- paymentToken
- : the address of the token you would like to pay in.

- gasLimit
- : a custom gas limit for your transaction, please remember to add an overhead for Gelato Relay's contract calls and security checks, seehere
- for more info.
- isHighPriority
- :EIP-1559 flag
- for increasing your priority gas fee. If true, you will incur higher costs but have a higher certainty of block inclusion.
- gasLimitL1
- : gas limit for theL1 data fee
- which is required to properly estimate fees on OP Stack chains, e.g. Optimism, Base, Zora. Can be omitted on all other chains.
-

Return Object

- The value of your estimated fee, including gas costs + gelato fee on top.
-

NOTE: please be aware that if your relayed transaction incurs gas refunds, for example, for clearing out storage slots, this is not known beforehand. These refunds can only be known after the fact, from the transaction receipts. This means that the fee oracle will give you a price which does not include the gas refunds, so it may be higher than you think. This is due to how the EVM works, and the initial gas allocated for execution should still be the total amount before refunds, otherwise you will get an 'Out of gas' error. Seehere for more info.

Querying via the API

Please see the available endpoints on theAPI page.

Previous Relay Context Contracts Next Optional Parameters Last updated2 months ago On this page *Querying via the SDK * SDK method: isOracleActive * SDK method: getPaymentTokens * SDK method: getEstimatedFee * Querying via the API