Paying for your transactions

Deprecation Notice: Gelato Functions is set to replace Automate, providing enhanced automation capabilities, access to off-chain data, unified payments across networks, and a performance boost.Migrate your tasks to Gelato Functions before Legacy Automate's end of life on June 30th, 2024 for uninterrupted service. When Gelato executes your transaction it needs to pay the transaction fees. The fees can be paid in one of two ways:

1. From Gelato Balance
2. Transaction pays for itself
3.

Gelato currently accepts payment in the native network token or its wrapped version for each network that Automate supports. For example, on Ethereum you can pay either in ETH or WETH and on Polygon either in MATiC or WMATIC - and so on for all the othernetworks we support .

Gelato Balance

This is the easiest option. You simply deposit some tokens into Gelato for each of the network(s) on which you want to execute transactions. Each time an execution occurs, Gelato will deduct the costs from your Gelato Balance to cover the gas costs. In the future we will introduce a small Gelato fee.

If your Gelato Balance is too low to cover the costs of your transaction it will not execute until you top-up your balance. We provide anotification service for you to receive alerts when your balance is running low.

Deposit funds via UI

?

You can deposit funds into your Gelato Balance on the fund's page.

Deposit funds via smart contract

If your smart contract will be the one creating tasks, it would need its own Gelato balance to pay for the task executions.

If you would like to have your smart contract deposit funds at any point of time into it's Gelato balance, inheritAutomateTaskCreator like so. Learn more aboutSmart Contract#automatetaskcreator-functions

```

Copy contractCounterisAutomateTaskCreator{

constructor(address_automate,address_fundsOwner)AutomateTaskCreator(_automate, _fundsOwner) {}

functiondepositForCounter()externalpayable{ _depositFunds(msg.value,ETH); } }

```

AutomateTaskCreator exposeswithdrawFunds function to allowfundsOwner to withdraw from the contract's Gelato balance.

Transaction pays for itself

You can also choose not to pre-deposit funds into your Gelato balance and have your function pay the fee during executions.

This can be done by inheritingAutomateReady .

```

Copy contractCounterWTisAutomateReady{ uint256publiccount; uint256publiclastExecuted;

constructor(address_automate,address_taskCreator) AutomateReady(_automate, _taskCreator) {}

receive()externalpayable{}

functionincreaseCount(uint256amount)externalonlyDedicatedMsgSender{ count+=amount; lastExecuted=block.timestamp;

(uint256fee,addressfeeToken)=_getFeeDetails();

_transfer(fee,feeToken); } }

```

In theincreaseCount function, we use_transfer inherited fromAutomateReady to pay Gelato.

_transfer has two parameters,fee andfeeToken which has to be queried from the Automate contract by usinggetFeeDetails()

Previous Supported Networks Next Execution & Balance Alerts Last updated3 months ago On this page *Gelato Balance * Transaction pays for itself