

# Chainlink Functions Billing

## Request costs

To send Chainlink Functions requests, you must maintain a sufficient amount of LINK in your subscription balance. Because Chainlink Functions follows the [Request and Receive Data](#) model, billing is done in two steps:

1. During the request step, the cost to fulfill a Chainlink Functions request is estimated and blocked on the subscription balance by adding it to the subscription reservation.
2. During the receive step, the exact fulfillment cost is calculated, then billed to the subscription account, and the subscription reservation is removed.

You can break down total costs into the following components:

- Gas cost: This cost is paid back to the transmitter oracle in LINK for fulfilling the request.
- Premium fees: These fees are paid in LINK to compensate nodes for their work and for the maintenance of the FunctionsRouter contract.

Gas cost calculation includes the following variables:

- Gas price: The current gas price fluctuates depending on network conditions.
- Overestimated gas price: refers to a deliberately higher gas price estimation to account for potential price fluctuations between the time a request is made and the time Chainlink Functions fulfills it. By setting the gas price higher than the current one, Chainlink Functions increases the likelihood that the request will be fulfilled, even if gas prices rise unexpectedly in the short term. The overestimation is calculated as a percentage increase over the current gas price.
- Callback gas: The amount of gas used for the callback request. See the [Cost Calculation](#) section.
- Callback gas limit: The maximum amount of gas that can be used to call the `handleOracleFulfillmentcallback` function of the consumer contract in order to provide the response. See the [Cost Simulation](#) section.
- Gas overhead: The amount of gas used by the FunctionsRouter and FunctionsCoordinator contracts. It is an estimate of the total gas cost of fulfilling a request.
- Native to LINK translation: Your subscription balance can be billed only in LINK tokens.\* Translate the network's native gas tokens to LINK: The total gas cost in native units is translated using the correct [Price Feed](#). For example, on Sepolia, the translation uses the ETH/LINK Price Feed, and on Polygon Mumbai, the translation uses the MATIC/LINK Price Feed.
- Fallback Wei to LINK ratio: In the unlikely event that the Native to LINK price data feed is unavailable, the fallback translation is used. You can find this ratio by running the `getConfig` function in the FunctionsCoordinator contract. See the [Supported Networks](#) page to find the contract addresses for each network.

## Cost simulation (reservation)

During the request step:

1. The total cost in LINK is estimated using the following formula:

total estimated gas cost in LINK = (Overestimated gas price \* (Gas overhead + Callback gas limit)) / Native to LINK translation

total estimated cost in LINK = total estimated gas cost + premium fees

2. The total estimated cost is then added to the subscription reservation.

## Cost calculation (fulfillment)

When a DON's oracle reports the response, subscription accounts are charged based on the gas amount used in the callback:

1. The total cost in LINK is calculated using the following formula:

total gas cost in LINK = (Gas price \* (Gas overhead + Callback gas)) / Native to LINK translation

total cost in LINK = total gas cost + premium fees

2. The FunctionsRouter contract performs several accounting movements.

## Cost calculation example

This is an example of cost estimation for a request and then cost calculation at fulfillment.

## Cost Simulation (Reservation)

Parameter	Value
Overestimated gas price	9 gwei
Callback gas limit	300000
Gas overhead	185000
Premium fee	0.2 LINK

1.

Calculate the total estimated gas cost in LINK, using an overestimated gas price, the gas overhead, and the full callback gas limit:

Gas cost calculation  
Total estimated gas cost  
Overestimated gas price x (Gas overhead + Callback gas limit)  
9 gwei x (300000 + 185000) = 4365000 gwei (0.004365 ETH)  
2. Convert the gas cost to LINK using the [LINK/ETH feed](#). For this example, assume the feed returns a conversion value of  $\approx 0.007$  ETH per 1 LINK.

ETH to LINK cost conversion  
Total gas cost (LINK) = 0.004365 ETH / 0.007 ETH/LINK = 0.62 LINK  
3. Add the premium fee to get the estimated cost for a subscription reservation:

Adding LINK premium  
Maximum request cost (LINK) = Total gas cost (LINK) + Premium fee  
0.62 LINK + 0.2 LINK = 0.82 LINK

For this example request, 0.82 LINK is reserved from your subscription balance, but not yet deducted. When the request is fulfilled, the exact request cost is deducted. The estimated cost of a request is overestimated to allow for 99% of gas price fluctuation increases in between request and response time.

### Cost calculation (fulfillment)

Parameter | Value  
Gas price | 1.5 gwei  
Callback gas | 200000  
Gas overhead | 185000  
Premium fee | 0.2 LINK  
1. Calculate the total gas cost:

Gas cost calculation  
Total gas cost  
Gas price x (Gas overhead + Callback gas)  
1.5 gwei x (200000 + 185000) = 577500 gwei (0.0005775 ETH)  
2. Convert the gas cost to LINK using the [LINK/ETH feed](#). For this example, assume the feed returns a conversion value of  $\approx 0.007$  ETH per 1 LINK.

ETH to LINK cost conversion  
Total gas cost (LINK) = 0.0005775 ETH / 0.007 ETH/LINK = 0.0825 LINK  
3. Add the premium fee to get the total cost of a request:

Adding premium fee  
Total request cost (LINK) = Total gas cost (LINK) + premium fee  
0.0825 LINK + 0.2 LINK = 0.2825 LINK

This example request would cost 0.2825 LINK when it is fulfilled. The subscription reservation for the 0.82 LINK is released, and the actual cost of 0.2825 LINK is deducted from your subscription balance.

## Minimum balance for uploading encrypted secrets

If you choose to store the encrypted secrets with the DON (learn more on the [Secrets Management page](#)), then one of your subscriptions must maintain a balance greater than the minimum required for uploading encrypted secrets. This balance is blockchain-specific. You can find the specific values for each blockchain on the [Supported Networks](#) page. Note: Uploading encrypted secrets is free of charge. However, to prevent misuse of Chainlink Functions, you are required to maintain a minimum balance in one of the subscriptions owned by your externally-owned account (EOA) before you can upload encrypted secrets to a DON.

## Withdrawing funds

To withdraw your LINK balance, you must first cancel your subscription. To prevent misuse of Chainlink Functions, any subscription with fulfilled requests below the request threshold will incur a cancellation fee. Both the request threshold and the cancellation fee are blockchain-specific. You can find their values for each blockchain on the [Supported Networks](#) page.

Example 1:

- Request Threshold: Two requests
- Cancellation Fee: 0.5 LINK
- Your Balance: 0.4 LINK
- Number of Fulfilled Requests: One
- Outcome: Canceling your subscription results in a non-refundable balance

Example 2:

- Request Threshold: Two requests
- Cancellation Fee: 0.5 LINK
- Your Balance: 1 LINK
- Number of Fulfilled Requests: One
- Outcome: You will receive 0.5 LINK if you cancel your subscription

Example 3:

- Request Threshold: Two requests
- Cancellation Fee: 0.5 LINK
- Your Balance: 1 LINK

- Number of Fulfilled Requests: Two or more
- Outcome: You will receive 1 LINK if you cancel your subscription. Your subscription will not incur the cancellation fee