introduction)

Optimizing quote response timing

Optimizing /quote and /routes response timing using timeout parameter

Introduction

The timing options are designed to give partners flexibility in managing how long they are willing to wait for results from the API

These options are crucial for partners who want tooptimize their integrations based on their specific needs , whether that means prioritizing speed, maximizing the number of results, or maintaining a balance between the two

By exposing these parameters, partners can tailor the timing strategies to their operational requirements. This way, they can ensure they get the most relevant results without unnecessary delays

Why These Options Are Exposed

Partners may encounter various use cases when interacting with theLI.FI API.

For instance, some might prefer the first result available , while others may be willing towait longer to receive more results . By providing these timing strategies, we empower partners to optimize their user experience according to their unique needs

The offering

We provide two types of timing strategies to give partners control over how they manage waiting times for responses in their requests:swapStepTimingStrategies androuteTimingStrategies

These options allow partners to customize how their requests handle timing, thereby enhancing their control over the responsiveness and results of their applications

Timing Strategies Overview

1.swapStepTimingStrategies

- Purpose
- : This strategy is applied at the swap step level. It determines how long the system will wait for results from exchanges before proceeding
- Use Case
- : Partners can prioritize quick responses for initial swap results while allowing for some flexibility in waiting for additional results

2.routeTimingStrategies

- Purpose
- : This strategy is applied at the route level after obtaining the best results from the swap step. It manages the waiting time for the complete route results, which may involve multiple swaps and bridges
- Use Case
- : Useful for partners who want to ensure they receive the best possible route by allowing for a longer wait time to gather comprehensive results

How Timing Strategies Work

The two-level approach to timing strategies allows for a more structured evaluation of results:

1. For Each Swap Step
2. : When requesting exchanges, theswapStepTimingStrategies
3. are applied to evaluate the results from the exchanges against each other. This helps in selecting the best swap option based on the predefined timing parameters
4. For the Overall Route
5. : After determining the best swap, therouteTimingStrategies
6. are applied. This manages the waiting time for the complete route results, ensuring that if there are multiple components (like swaps and bridges), the overall waiting period is managed consistently

This structure allows for flexibility and efficiency in obtaining results, making it adaptable to various routing types such as swap-only, bridge-only, or combined routes

Parameters Overview

Both timing options share 4 parameters:

- strategy: for now onlyminWaitTime
- is available
- minWaitTimeMs: min amount of time external tools have to provide a result
- startingExpectedResults: amount of resultsLI.FI
- API need to return withinminWaitTime
- . If at leaststartingExpectedResults
- is not returned byminWaitTime
- , the amount of expected results reduces everyreduceEveryMs
- reduceEveryMs: amount of time afterminWaitTime
- after which the amount of expected results reduces. For example, everyminWaitTime
- +reduceEveryMs
- ,startingExpectedResults
- =-1

Example Parameters Explained

For instance, using theswapStepTimingStrategies parameters as follows:

```

Copy { "strategy":"minWaitTime", "minWaitTimeMs":600, "startingExpectedResults":4, "reduceEveryMs":300 }

```

This setup indicates that the system will wait for 600ms for results, expecting at least 4 results, and if that is not met, it will reduce the expected results every 300ms

Common Use Cases

1. Immediate Results
2. : A partner may want the first available result without delay, even if it means sacrificing completeness. Seexample strategy 3
3. for the example.
4. Balanced Approach
5. : A partner may prefer to wait up to one second to receive as many results as possible but will not wait longer than that, even if no results are returned. Seexample strategy 4

6. for example.
7. Maximize Results
8. : Timing is not a concern for a partner who prefers to wait until all potential results are gathered for the best options. Se[ample strategy 1
9. for example.

By allowing partners to specify these parameters, we provide them with the flexibility to control the balance between speed and comprehensiveness according to their business needs

Additional Considerations

- Currently, there are no plans to implement extra rules for BridgeStep alone, as the existing structure covers routes comprehensively
- Applying timing strategies at the route level allows for consistent handling across various routing types, enhancing performance and predictability

Examples and visuals

Examples forswapStepTimingStrategies

1./routes request UsingswapStepTimingStrategies

This request retrieves routes (set of routes, single step AND multi step) information based on specified parameters, focusing on the timing strategy for swap steps

```

Copy { "fromChainId":1, "fromAmount":"1000000000000000000", "fromTokenAddress":"0x0000000000000000000000000000000000000000", "toChainId":1, "toTokenAddress":"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48", "options": { "timing":{ "swapStepTimingStrategies":[ { "strategy":"minWaitTime", "minWaitTimeMs":600, "startingExpectedResults":4, "reduceEveryMs":300 } ] } } }

```

Explanation of Parameters

- swapStepTimingStrategies
- :
- strategy
- : The timing strategy to use for swap steps, which isminWaitTime
- minWaitTimeMs
- : Minimum wait time set to600 milliseconds
- startingExpectedResults
- : The initial number of results expected from the swap set to4
- reduceEveryMs
- : After every300 milliseconds
- , the expected results may be reduced

2./quote request UsingswapStepTimingStrategies

This request retrieves a quote (best, single step route) with a focus on the timing strategy for swap steps

/quote request

```

Copy {{API_URL}}/v1/quote?fromChain=1&toChain=1&fromToken=ETH&toToken=USDC&fromAddress=0x29DaCdF7cCaDf4eE67c923b4C22255A4B2494eD7&toAddress=0x29DaCdF7cCaDf4eE67c923b4C22255A4B2494eD7&fr600-4-300

```

Explanation of Parameters

- swapStepTimingStrategies
- : Specifies the timing strategy for swap steps.
- Format
- :minWaitTime-{minWaitTimeMs}-{startingExpectedResults}-{reduceEveryMs}
- Example:minWaitTime-600-4-300
- means:
- Minimum wait of600 ms
- Expecting at least4 results
- Results may reduce every300 ms
- if not met

Examples forrouteTimingStrategies

1./routes Request UsingrouteTimingStrategies

```

Copy { "fromChainId":1, "fromAmount":"1000000000000000000", "fromTokenAddress":"0x0000000000000000000000000000000000000000", "toChainId":1, "toTokenAddress":"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48", "options": { "timing":{ "routeTimingStrategies":[{ "strategy":"minWaitTime", "minWaitTimeMs":1500, "startingExpectedResults":6, "reduceEveryMs":500 } ] } } }

```

Explanation of Parameters

- routeTimingStrategies
- :
- strategy
- : The timing strategy to use for route evaluations, which isminWaitTime
- minWaitTimeMs
-

    - : Minimum wait time set to1500 milliseconds
-
    - startingExpectedResults
-
    - : The initial number of results expected from the route set to6
-
    - reduceEveryMs
-
    - : After every500 milliseconds
-
    - , the expected results may be reduced

2./quote Request UsingrouteTimingStrategies

```

Copy {{API_URL}}/v1/quote?
fromChain=1&toChain=1&fromToken=ETH&toToken=USDC&fromAddress=0x29DaCdF7cCaDf4eE67c923b4C22255A4B2494eD7&toAddress=0x29DaCdF7cCaDf4eE67c923b4C22255A4B2494eD7&fr
1500-6-500

```

Explanation of Parameters

- routeTimingStrategies
- : Specifies the timing strategy for route evaluations.
-
    - Format
-
    - :minWaitTime-{minWaitTimeMs}-{startingExpectedResults}-{reduceEveryMs}
-
    - Example:minWaitTime-1500-6-500
-
    - means:
-
        - Minimum wait of1500 ms
-
        - Expecting at least6 results
-
        - Results may reduce every500 ms
-
        - if not met

Visual strategies examples

The examples below apply to both swapStepTimingStrategies and routeTimingStrategies

Sample strategy 1

The above diagram demonstrates a timing strategy where the system determines when to return results based on several conditions

- Strategy
- :minWaitTime
- minWaitTimeMs
- : 900 milliseconds
- startingExpectedResults
- : 5
- reduceEveryMs
- : 300 milliseconds

Process Timeline: Initially, the system waits for all results (up tominWaitTimeMs), which is 900ms in this case. After this, it evaluates whether the number of received results meets the expected number (startingExpectedResults), which is 5

Dynamic Result Evaluation

- If at least 5 results are received by 900ms, the system returns those results immediately
- If fewer than 5 results are received by 900ms, the system reduces the expected number of results by 1 every 300ms (reduceEveryMs
-)
- This means the system will lower its expectation to 4 results at 1200ms, 3 results at 1500ms, and so on
- The strategy continues reducing the expected results until it either returns a result set that meets the reduced expectation or it hits the fallback point, which is returning the first result received after 2100ms

Fallback Point:

- Return first result after 2100ms:
- If the system has not received the minimum required results even after multiple reductions (e.g., from 5 to 4 to 3, etc.), it eventually reaches a point where it will simply return the first result it receives after 2100ms. This ensures that the system doesn't wait indefinitely and that some result is returned even if the expected conditions aren't fully met.

Sample strategy 2

The above diagram demonstrates a timing strategy where the system determines when to return results based on a minimum wait time and a dynamic reduction strategy

- Strategy
- :minWaitTime
- minWaitTimeMs
- : 900 milliseconds
- startingExpectedResults
- : 1
- reduceEveryMs
- : 300 milliseconds

Process Timeline: Initially, the system waits for a minimum time (minWaitTimeMs) of 900ms before evaluating the results. This ensures that the system does not return any results before this minimum period has elapsed

Dynamic Result Evaluation

- Return Condition
- : After 900ms, if at least one result is received (startingExpectedResults
- = 1), the system immediately returns that result
- Reducing Wait Expectation
- : If fewer than the expected results are received by 900ms, the system continues to wait and evaluates again every 300ms (reduceEveryMs
-)
- This means the system will continuously check every 300ms to see if it has received at least one result and will return the first result as soon as it is available after the initial 900ms wait

Fallback Point

- If no results are received after the 900ms initial wait, the system does not impose additional waiting time or expectations beyond the minimum threshold and returns the first result as soon as it is received

Sample strategy 3

The above diagram demonstrates a timing strategy where the system returns results as soon as the first result is received, without a mandatory minimum wait time

- Strategy
- :minWaitTime
- minWaitTimeMs
- : 0 milliseconds
- startingExpectedResults
- : 1
- reduceEveryMs
- : 300 milliseconds

Process Timeline: The system is configured with no initial wait time (minWaitTimeMs = 0ms), meaning it can immediately evaluate and potentially return the first result received

Dynamic Result Evaluation

- Return Condition
- : The system is set to return as soon as the first result is received, with no enforced wait time or need to accumulate a specific number of results
- Reducing Wait Expectation
- : Since the initial expected results are set to 1 (startingExpectedResults
- = 1), and the system does not impose any delay, it is ready to return the first available result instantly
- The strategy essentially allows the system to deliver results as quickly as possible, prioritizing the speed of response over the number of results collected

Fallback Point

- As there is no mandatory wait time, the system is optimized for scenarios where the speed of receiving the first result is critical, regardless of the quantity of results

Sample strategy 4

The above diagram demonstrates a timing strategy where the system returns results under 900 ms, without a mandatory minimum amount of expected results (meaning none can be returned)

- Strategy
- :minWaitTime
- minWaitTimeMs
- : 900 milliseconds
- startingExpectedResults
- : 0
- reduceEveryMs
- : 0 milliseconds

Process Timeline: The system waits for a minimum time (minWaitTimeMs) of 900ms before evaluating the results. This ensures that the system does not return any results before this minimum period has elapsed

Dynamic Result Evaluation

- Return Condition
- : The system is set to return any result received within 900ms, with no enforced minimum number of results
- Reducing Wait Expectation
- : Since the initial expected results are set to 0 (startingExpectedResults
- = 0), and the system does not impose any delay, it is ready to return any result within 900ms.
- The strategy essentially allows the system to deliver results within imposed time limit, prioritizing the time over the number of results collected

Fallback Point

- As there is no mandatory minimum results, the system is optimized for scenarios where the speed of receiving the first result is critical within a specified time frame, regardless of the quantity of results Last updated2 months ago