LI.FI Terminology AtLI.FI , we strive to simplify the complexities of any-to-any cross-chain transfers. To describe these transfers and bridge funds from chain to chain, we've introduced some LI.FI-specific terminology.

Route

A "Route" object represents an entire transfer, whether it occurs within one single chain, such as a token swap, or across chains. It summarizes transfer details, featuring properties prefixed withfrom for source information, including the originating chain, transferred funds, and sender address. Conversely,to -prefixed properties denote destination specifics, such as the target chain, received funds, and recipient address. These attributes collectively define transfer parameters, facilitating seamless execution within applications or platforms. Each Route object includes one or moreStep Objects that represent the individual transactions of that Route.

```

Copy interfaceRoute{ id:string insurance:Insurance fromChainId:number fromAmountUSD:string fromAmount:string fromToken:Token fromAddress?:string

toChainId:number toAmountUSD:string toAmount:string toAmountMin:string toToken:Token toAddress?:string

gasCostUSD?:string// Aggregation of underlying gas costs in usd

containsSwitchChain?:boolean// Features required for route execution

steps:Step[]

tags?:Order[] }

```

Step

Step Objects represent the individual transactions a wallet must sign to execute the entire transfer. These actions move funds by interacting with contracts. It's crucial that Steps are signed and executed in the exact order listed since subsequent steps rely on the output funds of preceding ones.

Note that for every Step the user has to sign either one or two transactions: the main transaction described in the step and possibly one precedingapproval transaction .

What is an approval transaction?

Contracts that a user interacts with cannot just access that user's funds. So, before any contract can utilize a user's funds to perform a transaction, the user has to give permission first. These records of approvals are separate transactions that happen before the main contract interaction. Approval transactions set an upper limit on how much of a cryptocurrency the contract can use. If the approved amount for a contract is zero, then that contract cannot access a user's funds. Example: Why do we need multiple Steps?

Consider a transfer bridging USDC on Arbitrum to some obscure token on Base. While there's no direct bridge between these assets, multiple transactions chained together can achieve the desired outcome. A Route for this transfer might, for example, involve two Steps. The first Step swaps USDC on Arbitrum to USDC on Base using a cross-chain bridge, returning the swapped tokens to the executing wallet. Upon receiving the funds, the first Step can be marked as completed. The second Step involves a same-chain token swap from USDC to the obscure token, with the resulting funds again returned to the wallet. If the destination wallet differs from the source, the last Step directs the funds accordingly.

Anatomy of a Step

```

Copy interfaceStep{ id:string type:StepType tool:StepTool toolDetails:StepToolDetails integrator?:string referrer?:string action:Action estimate?:Estimate transactionRequest?:TransactionRequest }

```

Please note that the interface depicted above is a simplified version of the Step object. We have broken down all possible variants and condensed them into one for easier understanding.

Each step is identified by a uniqueid and characterized by itstype , indicating the nature of the step (e.g., swap, cross-chain transfer). Thetool andtoolDetails properties specifies the tool / protocol used to fulfill the action described. Furthermore, in order to accurately describe its transaction each step is broken down into two more objects:action andestimate .

Action

```

Copy interfaceAction{ fromChainId:number fromAmount:string fromToken:Token fromAddress?:string

toChainId:number toToken:Token toAddress?:string

slippage:number }

```

Describes the action taken in the Step. In this object you can find information about thefrom andto token, source and destination chain, the tool that will be used as well as the amount that will be sent to that tool to be bridged or swapped.

Estimate

```

Copy interfaceEstimate{ tool:string fromAmount:string fromAmountUSD?:string toAmount:string toAmountMin:string toAmountUSD?:string approvalAddress:string feeCosts?:FeeCost[] gasCosts?:GasCost[] executionDuration:number }

```

This object contains all estimated values for that step. Here you can find estimated gas prices and fees a user has to pay in order to fulfill the transaction. This object also contains the estimated return value in the currency a user will receive when successfully executing the transaction. Please note that the values in this object are only estimates, they may be less accurate for extremely volatile markets.

Execution

This object is not included in Steps when they are returned from the server. These object are appended to each Step when executed via theLI.FI SDK . It contains all information necessary to track a Steps execution progress like monitoring the state of approval transactions and of the main transaction. It plays a crucial part in the SDKs ability order to resume a step if an error occurs.

TheLiFiStep variant

This is a special variant of theStep object. In order to minimize the amount of transactions a user has to sign and to improve the UX,LI.FI has deployedSmart Contracts that can condense multiple steps into one by directly executing transactions on the blockchain. The steps can be recognized via the step type which is set totype: lifi . These Steps have the additionalincludedSteps property. This property contains a list of Steps that are executed via theLI.FI Smart Contracts .

Quotes

"Quotes" are essentially standalone Steps with additional transaction data that can directly be executed by a wallet. They come without an encompassing Route object as they contain all necessary information to execute and display the transfer within one single step.

Last updated19 days ago On this page * Route * Step * Anatomy of a Step * The LiFiStep variant * Quotes

Was this helpful? Export as PDF