

## UseExecutePosition

## Hook

Used to run shortcuts based on positions. It will find a route for the specified position, an input token and an amount.

## Example

```
const { data: positions } =  
usePositions ({ chain :  
1 , token :  
"0x6b175474e89094c44da98b954eedeac495271d0f" , })  
  
const { executionDetails } =  
useExecutePosition ({ position : positions ? positions[ 0 ] :  
undefined , tokenIn :  
"0xeaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa", amountIn :  
parseUnits ( 0.1 ,  
18 ) // from viem })  
  
// ... < button
```

## onClick

```
{ executionDetails .executeRoute}> Execute </ button>
// ...
```

## Options

**options.chainId**

number

ChainID to run the shortcut on.

Example: { chainId: 1 }

## position

Position (See [Position](#) )

Position to find a route for. Positions are returned from the [usePositions](#) endpoint.

Example: { position: positionFromUsePositionsHook }

**tokenIn**

string | string[]

If tokenIn is specified as an array, amountIn must also be an Array.

Tokens to use as input tokens for the shortcut. Can be one or multiple.

Use 0x for the chains native token.

Example: { tokenIn: "0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48" }

**amountIn**

string | string[]

If amountIn is specified as an array, tokenIn must also be an Array.

Amount(s) to use for the input tokens to run the shortcut. Can be one or multiple.

Convert amount(s) to string before passing to this function.

Specified amounts must be in the tokens smallest units. For example, for ETH, 1 ETH = 1e18. You can use [view sparseUnits function for this \(opens in a new tab\)](#)

Example: { amountIn: '1000000000000000000' }

## options

{}

Additional options with sensible defaults.

### options.slippage

number

Allowed slippage for this shortcut execution, in base units (1/10000). Default is 300 = 3%.

Example: { options: { slippage: 100 } }

### options.pricImpact

boolean

Calculate and return the price impact of executing this shortcut in return value.

Example: { options: { priceImpact: true } }

### options.transferMethod

'APPROVE\_TRANSFERFROM' | 'TRANSFER' | 'PERMIT2' | 'NONE'

Method to use for transferring funds to the [Smart Wallet](#).

There is multiple different ways to transfer funds to the Smart Wallet, namely:

- APPROVE\_TRANSFERFROM
  - Use a combination of [EOA](#)
  - approve transactions, and a transferFrom
  - executed inside the shortcut.
- TRANSFER
  - Use [EOA](#)
  - transfers, before execution of the shortcut.
- PERMIT2
  - Use [Permit2 \(opens in a new tab\)](#)
  - signature for running the shortcut, with a batchTransferFrom
  - executed inside the shortcut.
- NONE
  - No transfer or approvals are done, all funds need to be inside the [Enso Smart Wallet](#)
  - before execution.

Example: { options: { transferMethod: 'APPROVE\_TRANSFERFROM' } }

## Response

The useExecutePosition returns multiple data objects to inspect and execute a route that was found.

The following properties are available:

- executionDetails
- - Available after a route was found\* executionDetails.route
-

- - Route details (See [Route](#))
- 
- )
- 
- executionDetails.route.execute
- 
- - Execution of the route, creates a transaction via Wagmi
- 
- executionDetails.approvals
- 
- - Array of approvals needed to be executed before the Route can be executed.
- 
- executionDetails.approvals[].execute
- 
- - Execution function for an approval function.
- 
- executionDetails.transfers
- 
- - Array of transfers needed to be executed before the Route can be executed.
- 
- executionDetails.transfers[].execute
- 
- - Execution function for a transfer function.
- executeRoute
- - Execute the route directly, alias of `executionDetails.route.execute`
- executeApprovalsOrTransfers
- - Executes all transfers or approval functions in order at once.
- errorMessage
- - Details about errors occurred during route finding.
- hasRoute
- - Check if a route was found.
- hasApprovalsOrTransfers
- - Check if the route has any approvals or transfers.
- status
- -success
- |loading
- |error
- |idle

[UsePositions](#) [UseCrossChainRoute](#)