

# Bridging

## How to implement Bridging

### Summary

#### 1 Fetch the Price

This optional step is quicker than fetching an executable quote but invaluable when crafting a snappy user experience. In this step we'll fetch the price to display a quote preview to the user. This can be skipped if the bridge is on the server without user interaction. 2 Fetch the Quote

In this step we'll fetch a full quote that expires after 30s. It will be a complete quote with all the details required to execute it. 3 Execute the Quote

Once we have the quote all that's left to do is execute the quote and display a progress state to the user.

#### Fetch the Price

The Price is a lightweight quote, excluding steps and calldata but it includes a preview of what the full quote might contain. Prices are subject to change once you get the quote so you should handle price changes accordingly. Once you determine where to bridge from and to, the amount and the currency then you can pass everything into the [getPrice](#) or the [usePrice](#) hook depending on if you're using the sdk or the ui hooks:

```
sdk hooks Copy import { getClient } from '@reservoir0x/relay-sdk' import { useAccount } from 'wagmi' const { address } = useAccount ( ) const price = await getClient ( ) ?. actions. getPrice ( { user: address originChainId: 1 , // The chain id to bridge from destinationChainId: 7777777 , // The chain id to bridge to amount: '10000000000000000' , // Amount in wei to bridge originCurrency: '0x00000000000000000000000000000000' , // ERC20 Address destinationCurrency: '0x00000000000000000000000000000000' , // ERC20 Address tradeType: 'EXACT_INPUT' } )
```

#### Fetch the Quote

If you followed the previous step, you can pass that same data into the [getQuote](#) action or [useQuote](#) hook.

```
sdk hooks Copy import { getClient } from '@reservoir0x/relay-sdk' import { useWalletClient } from 'wagmi' const { data: wallet } = useWalletClient ( ) const quote = await getClient ( ) ?. actions. getQuote ( { wallet, chainId: 1 , // The chain id to bridge from toChainId: 7777777 , // The chain id to bridge to amount: '10000000000000000' , // Amount in wei to bridge currency: '0x00000000000000000000000000000000' , // ERC20 Address toCurrency: '0x00000000000000000000000000000000' , // ERC20 Address tradeType: 'EXACT_INPUT' } ) In the above example, we fetch the quote to bridge 0.1 ETH from Ethereum Mainnet to Zora. In the quote object we'll get data on the fees required to do this bridge. You can learn more about fees here .
```

Note - Quotes can go stale after 30 seconds. Clients should regularly fetch fresh quotes so that users are always submitting valid transactions. [Learn more](#) about Relay quotes.

#### Execute the Quote

After getting the quote all that's left is to execute the quote and render the progress to the user. Follow the code snippets below to execute the quote:

```
sdk hooks Copy import { getClient } from '@reservoir0x/relay-sdk' import { useWalletClient } from 'wagmi' const { data: wallet } = useWalletClient ( ) const quote = await getClient ( ) ?. actions. getQuote ( { ... options } ) await getClient ( ) ?. actions. execute ( { quote, wallet, onProgress : ( steps, currentStep, currentStepItem, fees, details, txHashes ) => { console . log ( steps, currentStep, currentStepItem, fees, details, txHashes ) } } ) While the quote is being executed if you're rendering the quote in an interface you may want to display progress to the user. There's a provided progress callback to get the updated state of execution. Learn more about the progress state here .
```

#### Preflight Checklist

☐ Verify that the user has enough balance on the origin chain to cover the full bridge amount + fees. Check out our [fees](#) doc for more details on calculating the fees. ☐ Have the Relay SDK set up and the Relay hooks package installed if applicable ☐ Ensure the user is on the correct chain (it should match the chainId you get the quote from) ☐ Handle any errors that may come about ☐ Render the progress of the bridge to provide a best in class user experience [Cross-Chain Execution](#)  
[Swapping twitter Powered by Mintlify](#)

On this page \* [Summary](#) \* [Fetch the Price](#) \* [Fetch the Quote](#) \* [Execute the Quote](#) \* [Preflight Checklist](#)