5CFE9D;}

.css-kun0x7{fill:transparent;opacity:0.5;margin:0 0.2rem;}.css-kun0x7:hover{fill:#FAF40A;}

.css-1ix0nx7{fill:transparent;opacity:0.5;}.css-1ix0nx7:hover{fill:#F14544;} On this page

Background

Before integrating with Uniswap, it may be helpful for newcomers to review the following background information on some important developer web3 concepts, the structure of our examples, and SDK concepts.

info

Already familiar with web3 development and/or the basics of our SDK and want to get right to the code? Start with our first guide,Getting a Quote !

Providers

Communication with the blockchain is typically done through a provider and local models of smart contracts and theirABIs .

To achieve this, our examples use theethers.js library. To instantiate a provider you will need a data source. Our examples offer two options:

- JSON RPC URL
- : If you are working directly with the Ethereum mainnet or a local fork, products such asInfura
- offer JSON RPC URLs for a wide variety of chains and testnets. For our examples, we'll only be using the Ethereum mainnet.
- Wallet Extension
- : If you are connecting to a wallet browser extension, these wallets embed a source directly into the Javascript window object aswindow.ethereum
- . This object surfaces information about the user's wallets and provides the ability to communicate with the connected chain. Importantly for our examples, it can be used withethers.js
- to construct a provider.

Uniswap's Runnable Examples

Each guide is accompanied and driven byrunnable examples using React to provide a basic UI for interacting with the example. Each examples provides relevant options such as running against a local blockchain or connecting to the Ethereum mainnet directly. You also have the option of using a wallet extension which can be connected to either environment.

Inputs and environment settings are configured in each example'sconfig.ts and allows for simple setup and configuration.

Developing and Testing

To test your code, we recommend utilizing a local fork of the Ethereum mainnet. To help facilitate easy testing, each example includes a quickstart for running the local chain with a test wallet. To further test, we also recommend using a wallet extension and connecting to the local chain. Finally, each example can be run against the Ethereum mainnet if desired. Full development instructs can be found in theREADME.md of each code example.

Utility Libraries

Each example is concentrated into a single file within thelibs/ folder of the example, with the entry points noted in each guide and README.

To allow the guides to focus on the SDK's core functionality, additional basic building blocks can be found in each example'slibs folder. The exported functionality from these files is intended to be the minimum needed for each example and not a complete library for production usage. These also include storing core constants such as definitions for tokens, ABI's, and blockchain addresses that can distract from the core concepts. Below are summaries of the helping libraries you will encounter.

Provider Utilities

provider.ts wraps the basics ofethers.js and connecting to wallet extensions into an abstracted view of a provider, a wallet address, and the ability to send transactions. It also helps abstract the configured environment you wish to run against in

your example without making code changes outside of your configuration.

Wallet Utilities

wallet.ts offers the ability to query a wallet (whether connected via an extension or defined in code/config) for its balances and other essential information.

Pool Information

pool.ts contains the basic querying of pool information when not essential / core to the relevant guide

Display Utilities

conversion.ts provides display and light math wrappers to help show human readable prices when dealing with currency amounts (typically stored as raw numbers and the decimal placement separate for precision reasons) in the form of two functions:fromReadableAmount andtoReadableAmount

Notable SDK Structures and Concepts

When working with the SDK it can be helpful to understand some of the design choices and why they are needed. Below you can find a few important concepts.

ABI's

To allow others to interact with a smart contract, each contract exposes an ABI (Application Binary Interface). As these are defined on the blockchain, we must ensure the correct definitions are provided to our Javascript functions. ABI's are provided from various SDK's and imported in as needed. Some examples will define an ABI directly as needed.

CurrencyAmount and JSBI

Cryptocurrency applications often work with very small fractions of tokens. As a result, high precision is very important. To ensure precision is upheld, theCurrencyAmount class helps store exact values as fractions and utilizesJSBI for compatibility across the web. To display these amounts nicely to users, additional work is sometimes required.

Currency

TheCurrency class can represent both native currency (ETH) and an ERC20Token . Currencies vary in their relative value, so theToken class allows your application to define the number of decimals needed for each currency along with the currency's address, symbol, and name. Edit this page .css-1tclyyl{margin-top:1.5rem;} .css-1c3fvx8{display:-webkit-box;display:-webkit-flex;display:-ms-flexbox;display:flex;-webkit-flex-direction:row;-ms-flex-direction:row;flex-direction:row;-webkit-align-items:center;-webkit-box-align:center;-ms-flex-align:center;align-items:center;-webkit-box-pack:center;-ms-flex-pack:center;-webkit-justify-content:center;justify-content:center;} .css-1wsnqg4{font-size:1rem;padding-right:0.5rem;} Helpful? .css-y2jwfw{fill:transparent;opacity:0.5;}.css-y2jwfw:hover{fill:#5CFE9D;}

.css-kun0x7{fill:transparent;opacity:0.5;margin:0 0.2rem;}.css-kun0x7:hover{fill:#FAF40A;}

.css-1ix0nx7{fill:transparent;opacity:0.5;}.css-1ix0nx7:hover{fill:#F14544;} Previous Overview Next Local Development * Providers * Uniswap's Runnable Examples * * Developing and Testing * * Utility Libraries * Notable SDK Structures and Concepts * * ABI's * * CurrencyAmount and JSBI * * Currency