

SDK

About the SDK

The Across SDK is written and maintained by the engineering team at Risk Labs. It is written in typescript and available on NPM at [@across-protocol/sdk-v2](https://www.npmjs.com/package/@across-protocol/sdk-v2) . It's compatible with both Node JS environments and in the browser.

How can I use the SDK?

The SDK can be used currently to query suggested deposit fees and limits, and get liquidity pool statistics. It is imported and used in the [API's](#) implementation.

If I want to integrate Across into my dApp, should I use the SDK or the API?

We recommend using the API, which wraps SDK functions and has an easier interface. However, if speed is a concern then we recommend reviewing the [API implementation of the SDK](#) to understand best how to use the SDK.

Installation

To add the SDK to your project, use npm or yarn to npm install @across-protocol/sdk-v2 or yarn add @across-protocol/sdk-v2 . This can be used either in a frontend application or a node js project.

Basic Usage

You can read about the different SDK modules on the [Github README](#) page. For convenience, the available modules are: * lpFeeCalculator: Get * [liquidity provider fee](#) * that will be charged on deposit for its * quoteTimestamp * relayFeeCalculator: Get suggested * relayerFeePct * for a deposit, which accounts for * [opportunity cost of capital and gas costs](#) * . If the depositor opts to set this fee lower than the suggested fee, then there is a chance that the deposit goes unfilled for a long time. * pool: Get HubPool statistics, such as available * liquidReserves * that be used to refund relayers and * estimatedApy * for * [liquidity providers](#) * . [Reference -Previous API Next- Reference Contracts](#) Last modified 22d ago On this page About the SDK How can I use the SDK? If I want to integrate Across into my dApp, should I use the SDK or the API? Installation Basic Usage