

Envio HyperIndex

[Envio](#) HyperIndex is a feature-rich indexing solution that provides developers with a seamless and efficient way to index and aggregate real-time or historical blockchain data for any EVM. The indexed data is easily accessible through custom GraphQL queries, providing developers with the flexibility and power to retrieve specific information.

Envio offers native support for Arbitrum One (testnet & mainnet), Arbitrum Nova (testnet & mainnet), and Arbitrum Orbit chains, and has been designed to support high-throughput blockchain applications that rely on real-time data for their business requirements.

Designed to optimize the user experience, Envio offers automatic code generation, flexible language support, quickstart templates, and a reliable cost-effective [hosted service](#).

Indexers on Envio can be written in JavaScript, TypeScript, or ReScript.

Envio HyperSync

Envio supports [HyperSync](#) on Arbitrum.

HyperSync is an accelerated data query layer for the Arbitrum networks, providing APIs that bypass JSON-RPC for 20-100x faster syncing of historical data. HyperSync is used by default in Envio HyperIndex, with the use of RPC being optional. Using HyperSync, application developers do not need to worry about RPC URLs, rate-limiting, or managing infrastructure and can easily sync large datasets in a few minutes, something that would usually take hours or days via RPC.

HyperSync is also available as a standalone API for data analytic use cases. Data analysts can interact with the HyperSync API using JavaScript, Python, or Rust clients and extract data in JSON, Arrow, or Parquet formats. For more information, visit the HyperSync documentation [here](#).

HyperIndex key features

- Contract Import: Autogenerate the key boilerplate for an entire Indexer project off a single or multiple smart contracts. Deploy within minutes.
- Multi-chain Support: Aggregate data across multiple networks into a single database. Query all your data with a unified GraphQL API.
- Asynchronous Mode: Fetch data from off-chain storage such as IPFS, or contract state (e.g. smart contract view functions).
- Quickstart Templates: Use pre-defined indexing logic for popular OpenZeppelin contracts (e.g. ERC-20).

Getting started

Users can choose whether they want to start from a quickstart template, perform a subgraph migration, or use the contract import feature to get started with Envio HyperIndex.

The following files are required to run the Envio indexer:

- Configuration (defaults to config.yaml)
-)
- GraphQL Schema (defaults to schema.graphql)
-)
- Event Handlers (defaults to src/EventHandlers.*)
- depending on the language chosen)

These files are auto-generated according to the template and language chosen by running the `envio init` command.

Contract import tutorial

This walkthrough explains how to initialize an indexer using a single or multiple contracts that are already deployed on Arbitrum. This process allows a user to quickly and easily start up a basic indexer and a queryable GraphQL API for their application in less than 3 minutes.

Initialize your indexer

`cd` into the folder of your choice and run

`envio init Name your indexer`

? Name your indexer: Choose the directory where you would like to setup your project (default is the current directory)

? Set the directory: (.)

. SelectContract Import as the initialization option.

? Choose an initialization option Template

ContractImport SubgraphMigration [↑↓ to move, enter to select, type to filter] ? Would you like to import from a block explorer or a local abi?

Block Explorer Local ABI [↑↓ to move, enter to select, type to filter] Block Explorer option only requires user to input the contracts address and chain of the contract. If the contract is verified and deployed on one of the supported chains, this is the quickest setup as it will retrieve all needed contract information from a block explorer.

Local ABI option will allow you to point to a JSON file containing the smart contract ABI. The Contract Import process will then populate the required files from the ABI.

Select the blockchain that the contract is deployed on

? Which blockchain would you like to import a contract from? ethereum-mainnet goerli

arbitrum-one arbitrum-nova bsc gnosis v polygon [↑↓ to move, enter to select, type to filter]

Enter the address of the contract to import

? What is the address of the contract? [Use the proxy address if your abi is a proxy implementation] Note if you are using a proxy contract with an implementation, the address should be for the proxy contract.

Choose which events to include in theconfig.yaml

file

? Which events would you like to index?

[x] ClaimRewards (address indexed from, address indexed reward, uint256 amount) [x] Deposit (address indexed from, uint256 indexed tokenId, uint256 amount) [x] NotifyReward (address indexed from, address indexed reward, uint256 indexed epoch, uint256 amount) [x] Withdraw (address indexed from, uint256 indexed tokenId, uint256 amount) [↑↓ to move, space to select one, → to all, ← to none, type to filter]

Select the continuation option

? Would you like to add another contract?

I'm finished Add a new address for same contract on same network Add a new network for same contract Add a new contract (with a different ABI) [Current contract: BribeVotingReward, on network: arbitrum-one] TheContract Import process will prompt the user whether they would like to finish the import process or continue adding more addresses for same contract on same network, addresses for same contract on different network or a different contract.

For more information on contract import feature, visit the documentation[here](#) .

Envio examples

Click[here](#) for Envio HyperIndex examples. Click[here](#) for Envio HyperSync examples.

Getting help

Indexing can be a rollercoaster, especially for more complex use cases. Our engineers are available to help you with your data availability needs.

Join our growing community of elite builders, and find peace of mind with Envio.

- [Discord](#)
- Email:hello@envio.dev [Edit this page](#) Last updated on Mar 19, 2024 [Previous](#) [Crossmint](#) [Next](#) [Flair](#)