

Subgraph

Introduction

Lido has a Subgraph deployed on [The Graph Decentralized Network](#) which indexes and organises data from the Lido smart contracts events, exposing a GraphQL endpoint for queries. Subgraph data is indexed and served by independent Indexers on the network.

GraphQL Schema

The schema of GraphQL entities available is defined in [schema.graphql](#).

Links

- [Explorer Page](#)
- GraphQL Endpoint: [https://gateway.thegraph.com/api/\[api-key\]/subgraphs/id/HXfMc1jPHfFQoccWd7VMv66km75FoxVHDMvsJj5vG5vf](https://gateway.thegraph.com/api/[api-key]/subgraphs/id/HXfMc1jPHfFQoccWd7VMv66km75FoxVHDMvsJj5vG5vf)
- [Code Repo](#)

Query Examples

Below are some sample queries you can use to gather information from the Lido contracts.

You can build your own queries using [GraphQL Explorer](#) to test it out and query exactly what you need.

Rewards Distribution

Daily staking rewards data with calculated APR and fees distribution.

```
{ totalRewards ( first :  
100 ,  
orderBy :  
block ,  
orderDirection :  
desc )  
{ id totalRewards totalRewardsWithFees insuranceFee treasuryFee totalFee dust nodeOperatorFees  
{ address fee } nodeOperatorsShares  
{ address shares } shares2mint sharesToInsuranceFund sharesToOperators sharesToTreasury totalPooledEtherBefore  
totalPooledEtherAfter totalSharesBefore totalSharesAfter apr aprBeforeFees aprRaw preTotalPooledEther  
postTotalPooledEther timeElapsed block blockTime transactionIndex } }
```

Oracle Reports

Daily completed oracle reports.

```
{ oracleCompleteds ( first :  
500 ,  
orderBy :  
blockTime ,  
orderDirection :  
desc )  
{ epochId beaconBalance beaconValidators block blockTime } }
```

Transfers

stETH transfers between addresses.

```
{ lidoTransfers ( first :  
50 )  
{ from to value block blockTime transactionHash } }
```

Submissions

stETH staking events.

```
{ lidoSubmissions ( first :  
50 )  
{ sender amount block blockTime transactionHash } }
```

Node Operator Keys

Fetch validator keys of a node operator.

```
{ nodeOperatorSigningKeys ( where :  
{  
operatorId :  
0  
} )  
{ pubkey } }
```

Helpful Links

[Creating an API Key Video Tutorial](#)

[Managing your API Key & Setting your indexer preferences](#)

[Querying from an Application](#) [Edit this page](#) [Previous SDKs](#) [Next Specification](#)