



Subsquid cheatsheet for Substrate networks

v0.0

EVM cheatsheet



Decoded 2023 | June 28-29 | Copenhagen

Get a template squid up and running:

```
npm install --global @subsquid/cli
sqd init my-new-squid -t substrate
cd my-new-squid
npm ci
sqd up
sqd process
```

Data ingestion is done, start an API with

sqd serve

Quickstart
guide

Requesting data (src/processor.ts):

```
const processor =
  new SubstrateBatchProcessor()
.setDataSource({
  archive: lookupArchive('kusama'),
  chain: KUSAMA_NODE_RPC
})
.addEvent('Balances.Transfer', {
  data: {
    event: {
      args: true,
      extrinsic: {
        fee: true
      }
    }
  } as const
})
.addCall('Balances.transfer_all', {
  data: {
    call: {
      success: true,
      args: true,
      parent: true
    }
  } as const
})
```

Protocol/VM-specific settings available for EVM, WASM/ink!, Gear, Acala.

Processor
config docs

Processing and saving data (src/processor.ts):

```
processor.run(new TypeormDatabase(), async context => {
  // THIS function gets called
  // for each batch of 1-1000 blocks
  // DEFINE some buffers to hold the processed data
  for (let block of context.blocks) {
    for (let item of block.items) {
      if (item.kind === 'event' &&
          item.name === 'Balances.Transfer') {
        // DO whatever you want with the event data!
        // QUERY chain state, use IPFS, HTTP APIs
        // SDK provides tools for decoding and
        // handling historical runtimes
      }
    }
  }
  // SAVE the results for the whole batch
  // Postgres or CSV/Parquet/JSON files supported
})
```

Batch context
interfaceState
queries

Data decoding is done using auto-generated TypeScript classes:

- Set the network and list the items at typegen.json;
- To generate decoders e.g. for all events set events: true;
- Run sqd typegen to generate classes at ./src/types

Classes are aware of runtime versions. A config like this

```
{ "outDir": "src/types",
  "specVersions": "https://kusama.archive.subsquid.io/graphql",
  "events": [ "Balances.Transfer" ],
  "calls": true }
```

enables batch handler code like this

```
import {BalancesTransferEvent}
from './types/events'
if (item.kind === 'event' &&
  item.name === 'Balances.Transfer') {
  let e = new BalancesTransferEvent(
    ctx,
    item.event
  )
  if (e.isV1020) {
    let [from, to, amount] = e.asV1020
  } // decode all runtime
  // versions similarly
}
```

Typegen

Live squid
examplesKodaDot
NFT squidPolkassembly
squidGiant
Squid

Postgres & GraphQL share a data definition. Config at schema.graphql

```
type Account @entity {  
    id: ID!  
    transfersTo: [Transfer!] @derivedFrom(field: "to")  
}  
type Transfer @entity {  
    id: ID!  
    to: Account!  
    hash: String! @index  
}
```

creates a foreign key column `to_id` at the “transfer” table on the database side and allows queries like this:

```
query {  
    transfers {  
        to {  
            transfersTo } }  
    accounts {  
        transfersTo } }
```

The hash column will be `@index`-ed for fast queries.

The GraphQL server can be extended with arbitrary queries.

Data definition



API extensions



It is possible to write to file-based datasets:

Formats:

- CSV
- Parquet

Destinations:

- Local
- S3

Persisting to files



The ‘Aquarium’ - an optional managed service for squids. Free tier forever.

How it works:

- Register at app.subsquid.io
 - Authorize your Squid CLI
 - Run `sqd deploy`. <- Mind the dot!
- ...and that's it!



Aquarium

Deployment docs

No time to code? Contract indexers can be autogenerated from ABIs!

WASM/ink! (e.g. Astar):

```
sqd init my-new-squid -t ink-abi  
cd my-new-squid && npm ci  
# save JSON ABI as abi/myAbi.json  
sqd generate --name contractName \  
--address <contractAddress> \  
--archive <networkArchiveAlias> \  
--abi ./abi/myAbi.json \  
--event '*' # or a comma-separated list
```

EVM (Moonbeam, Moonriver, etc):

```
sqd init my-new-squid -t abi  
cd my-new-squid && npm ci  
# save JSON ABI as abi/myAbi.json  
sqd generate --name contractName \  
--address <contractAddress> \  
--archive <networkArchiveAlias> \  
--abi ./abi/myAbi.json \  
--function '*' \  
--event '*' # or a comma-separated list
```

Squid-gen quickstart



Squid-gen reference



SUBSQUID



Web



Docs



SDK overview



Dev chat



Discord