

The Graph

Cheat Sheet

Anatomy of a Subgraph

subgraph.yaml: A YAML file that describes the subgraph configuration such as smart contract address, entities, events, etc.

schema.graphql: A GraphQL schema that defines the data you want your GraphQL API to query to serve to your frontend client

AssemblyScript Mappings: AssemblyScript code that translates from the event data emitted from the smart contract to the entities defined in your schema (e.g. **mapping.ts** in this cheatsheet)

* Web3 Development Workshop with The Graph on Friday, 4. November 2022 at 6pm.



Note: The subgraph folder structure includes several other files and folders such as *generated*, *node_modules*, *build*, etc. The above list only delineates the files that are involved in the subgraph definition process.

Up-and-vibing with your first subgraph!

Step 1

Installing the Graph CLI

- Using npm

```
yarn global add @graphprotocol/graph-cli
```

- Using yarn

```
npm install -g @graphprotocol/graph-cli
```

Step 2

Initializing your subgraph

If you already have an existing contract deployed, the best way to get started is to bootstrap your subgraph with autogenerated entities for each event. Simply run the following command and answer the questions:

```
graph init --index-events
```

Currently, the product option **subgraph-studio** is only available for subgraphs on Ethereum Mainnet and Goerli

Step 3

Configure your subgraph

Define the following fields in **subgraph.yaml**

- Datasource:** Smart contract address
- Start block:** Smart contract start block
- Entities:** Data to be defined in **schema.graphql**
- Event Handlers:** Event handlers to be defined in **mapping.ts**

Step 4

Defining the entities

Define the entities in **schema.graphql**

Step 5

Run codegen command

```
graph codegen
```

Step 6

Updating AssemblyScript Mappings

Configure the AssemblyScript mappings in the **src/mapping.ts** file to handle the events defined in **subgraph.yaml**

Step 7

Deploying your subgraph

```
graph deploy --studio <subgraph-slug>
```

Step 8

Authenticating your subgraph

```
graph auth --studio
```

Final Step

Testing your subgraph

Test a sample query in the GraphQL playground in your subgraph dashboard

About Subgraphs

A subgraph is a decentralized GraphQL API layer that sits between your UI layer and blockchain data layer. Subgraphs let you index data from the blockchain and query it over to your app's frontend.

Learn more at thegraph.com/ecosystem/learn/

About The Graph

The Graph is the indexing and query layer of web3. Developers build and publish open APIs, called subgraphs, that applications can easily query using GraphQL.

Cheat Sheet

Querying an existing subgraph

Each subgraph deployed on the decentralized network has a query URL that allows you to query the blockchain data indexed by that subgraph. Let's learn how to use an existing subgraph to query data to your frontend application!

Using the Graph Explorer

Open a subgraph you want to query from the [Graph Explorer](#) and click on "Query" to access the query URL. It looks something like:

```
https://gateway.thegraph.com/api/[api-key]/subgraphs/id/AVZ1dGwmRGKsbDAbwvxNmXzeEkD48voB3LfGqj5w7FUS
```

Note the `[api-key]` placeholder. This needs to be replaced with a real API-key

Creating your first API Key

API Keys are required to query subgraphs. You can receive 1,000 free queries on your first API key. You can create and manage your API keys in the Subgraph Studio¹ from the [API Keys](#) section.

1. Open [Subgraph Studio](#) > [API Keys](#)
2. Click on "Create API Key"
3. Enter the name of your API Key and hit "Create API Key"
4. Enter email to claim 1,000 free queries
5. Click on "Copy" to copy the API Key from the [Overview](#) section. It looks something like:
`106d.....a90b`



Open Subgraph Studio to create you API here:

Adding the API Key

Update the query URL for the subgraph with the newly created API Key from the previous step.



Note: Querying subgraphs using your API keys will generate query fees that will be paid in GRT from the [Billing](#) section.

You can learn more about billing here: <https://thegraph.com/docs/en/billing/>

High Quality Subgraphs to Get Started:



Messari Subgraphs



NFTs



Lens Protocol



Unlock Protocol

Learn More



More about Querying Subgraphs:



Query from the front-end here:



GraphQL Querying Best Practices: