NEAR + The Graph: 打造完美的 Web3 用户体验

Web3

基础设施

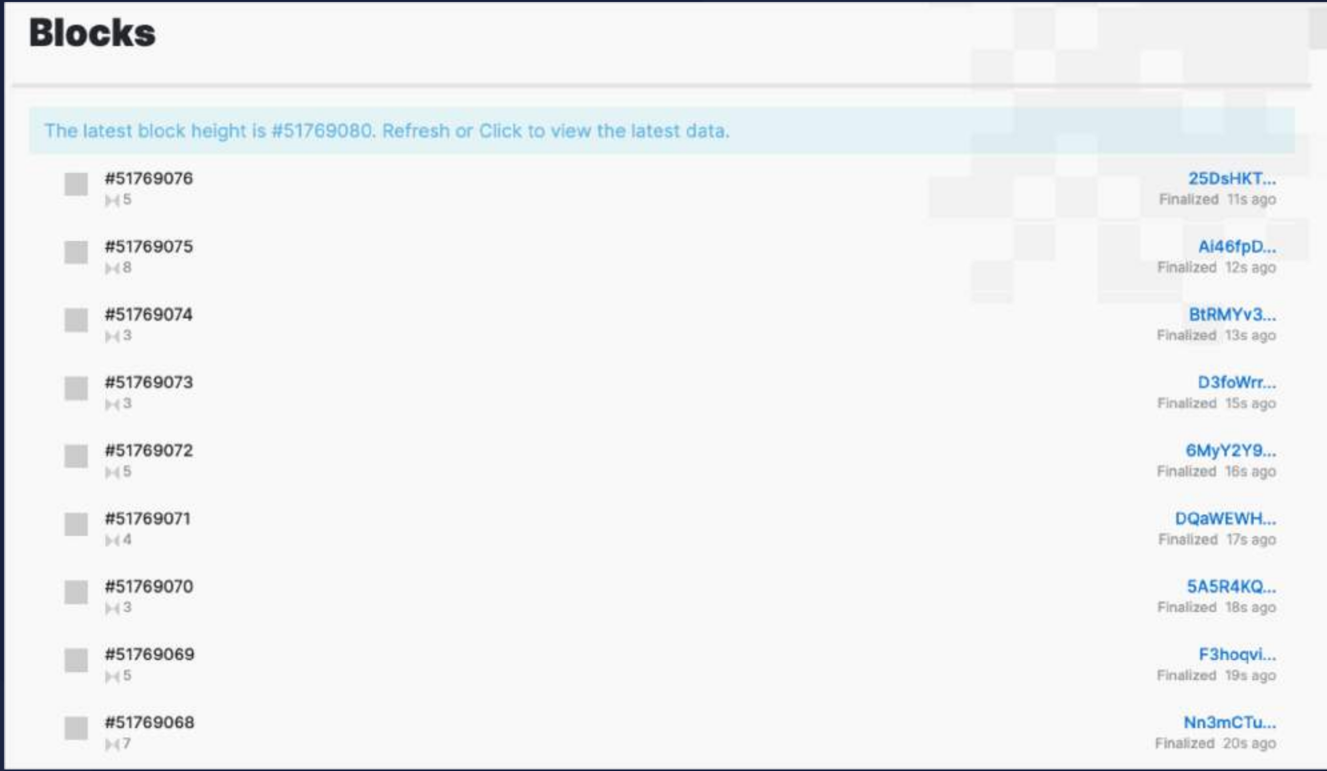
The Graph

Onchain		Offchain	
		Textile	Ceramic
Bridges	L2s	Notifications	IPFS
Near	Other L1s	Storage	Video

Web3

搜索与查询

没有The Graph



Blocks

The latest block height is #51769080. Refresh or Click to view the latest data.

#51769076
5
25DsHKT...
Finalized 11s ago

#51769075
8
Ai46fpD...
Finalized 12s ago

#51769074
3
BtRMYv3...
Finalized 13s ago

#51769073
3
D3foWrr...
Finalized 15s ago

#51769072
5
6MyY2Y9...
Finalized 16s ago

#51769071
4
DQaWEWH...
Finalized 17s ago

#51769070
3
5A5R4KQ...
Finalized 18s ago

#51769069
5
F3hoqvi...
Finalized 19s ago

#51769068
7
Nn3mCTu...
Finalized 20s ago

Web3

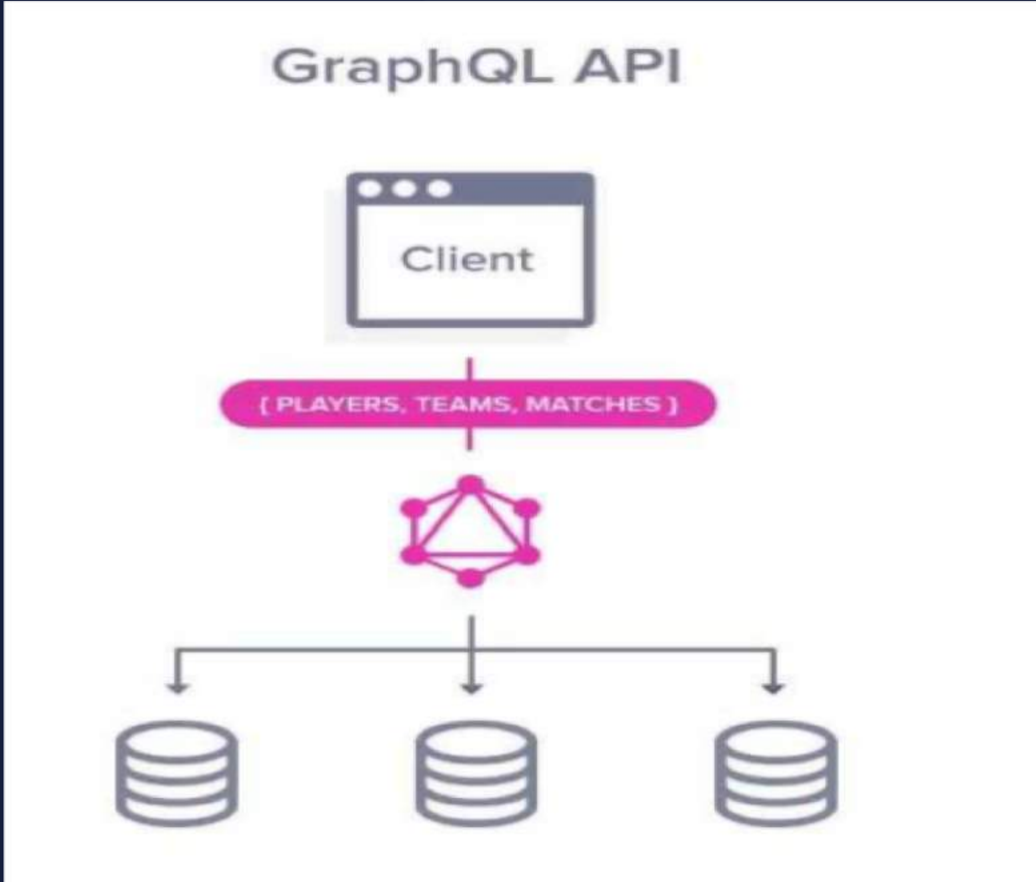
搜索与查询

没有The Graph

Web3

搜索与查询

有了The Graph

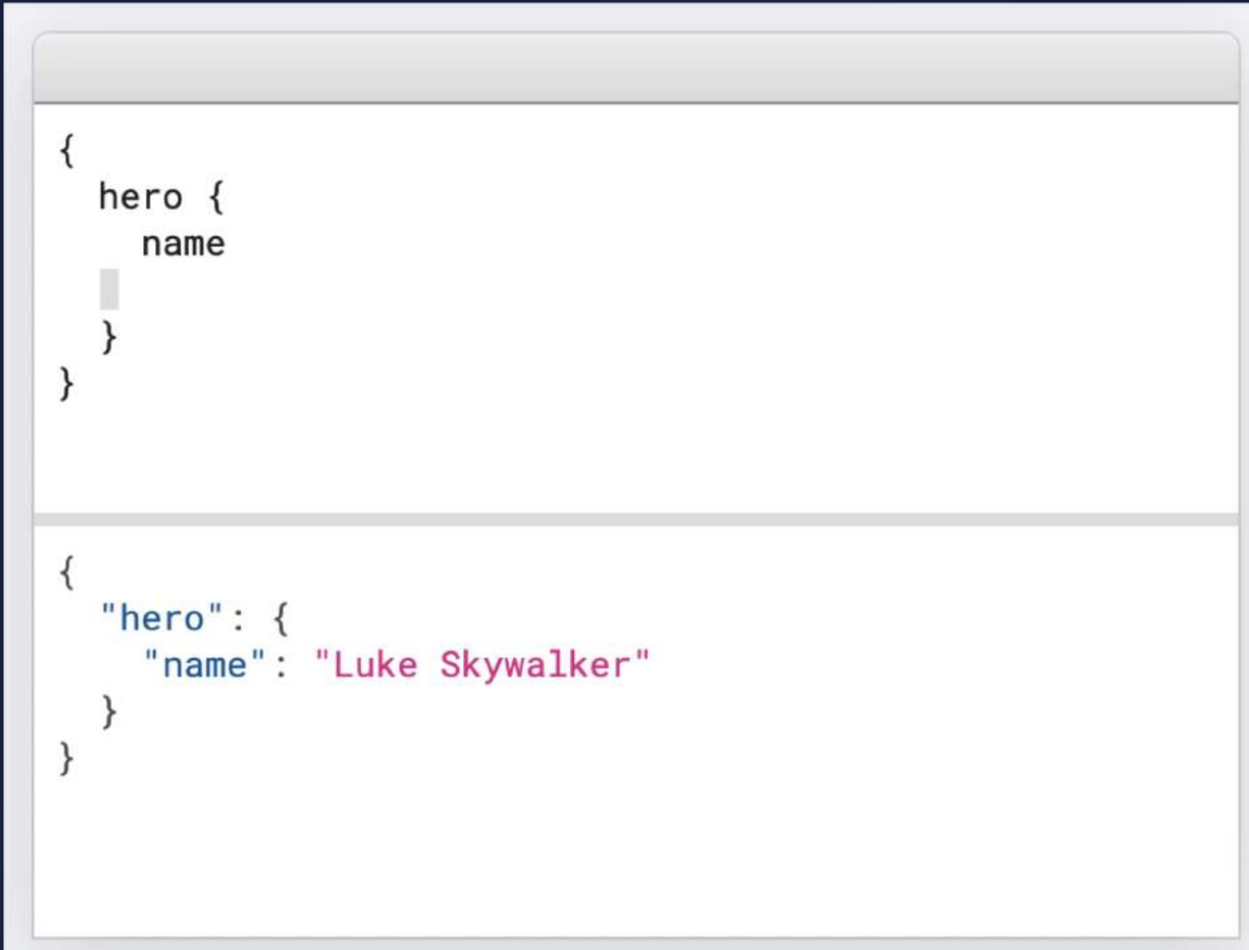


GraphQL API

Web3

搜索与查询

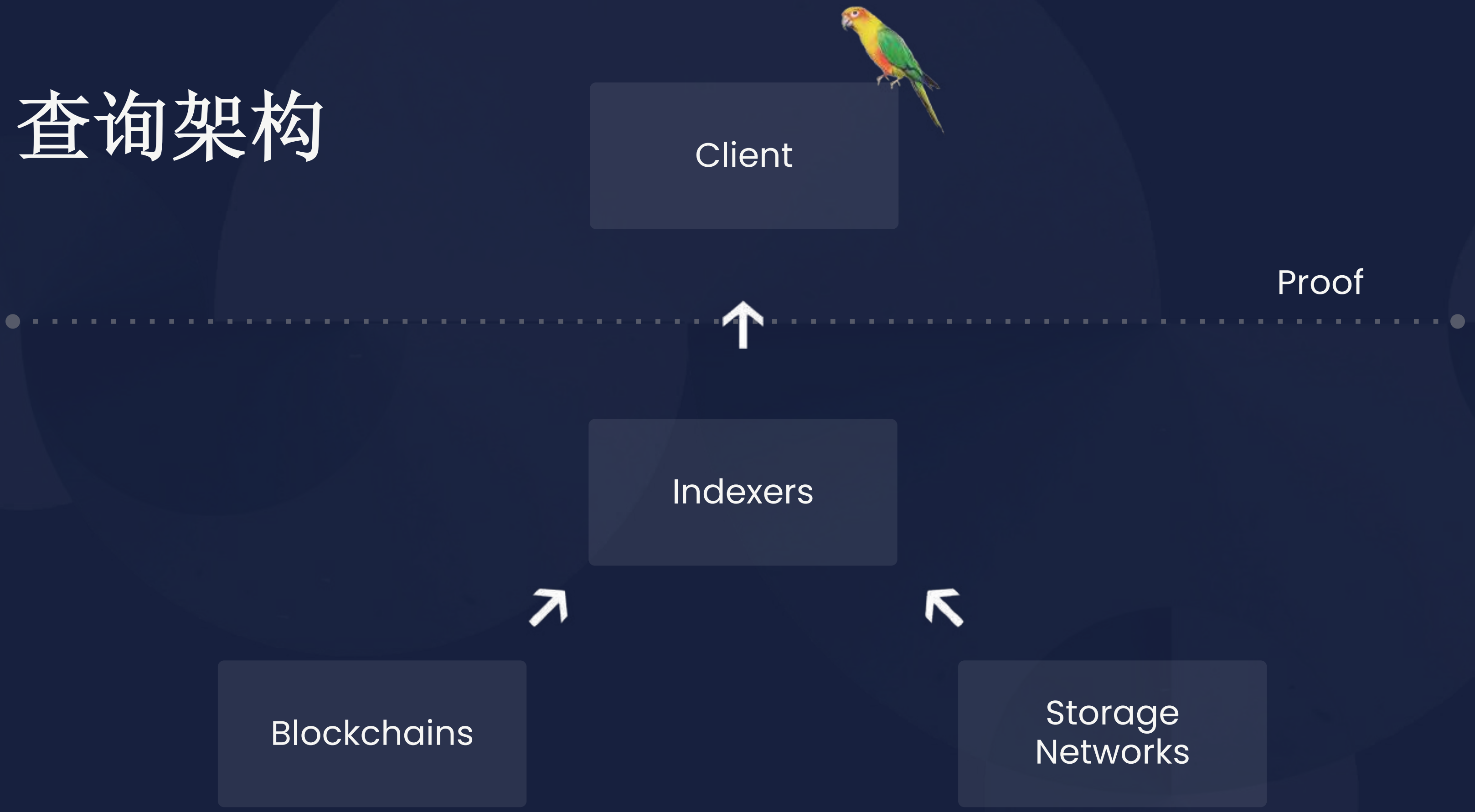
有了The Graph



```
{
  hero {
    name
    
  }
}
```

```
{
  "hero": {
    "name": "Luke Skywalker"
  }
}
```

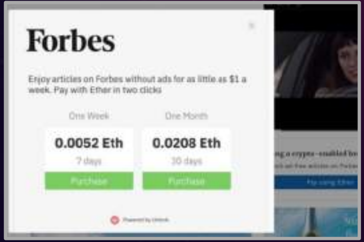
查询架构



支持各种复杂的去中心化应用

交易数据

Forbes 订阅数据



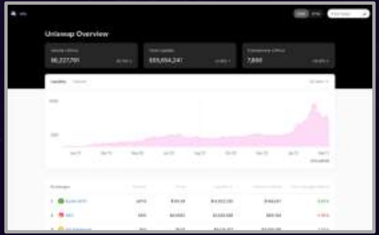
Forbes

Enjoy articles on Forbes without ads for as little as $1 a week. Pay with Ether in two clicks

One Week	One Month
0.0052 Eth	0.0208 Eth
7 days	30 days
Purchase	Purchase

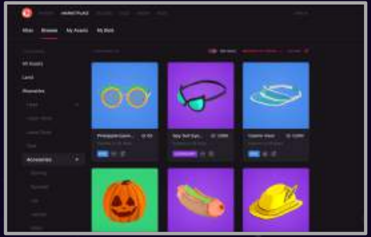
Powered by Unlock

去中心化交易平台

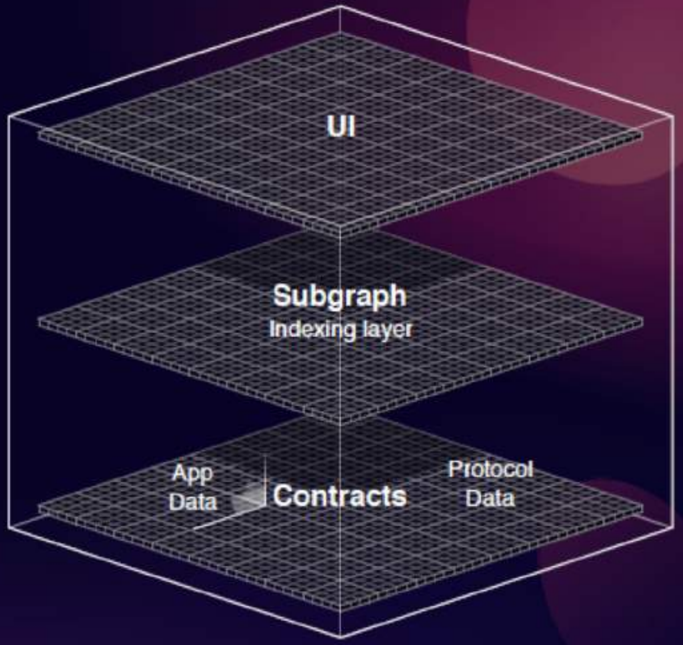


Uniswap Overview

虚拟资产



Web3技术栈不可或缺的基础设施

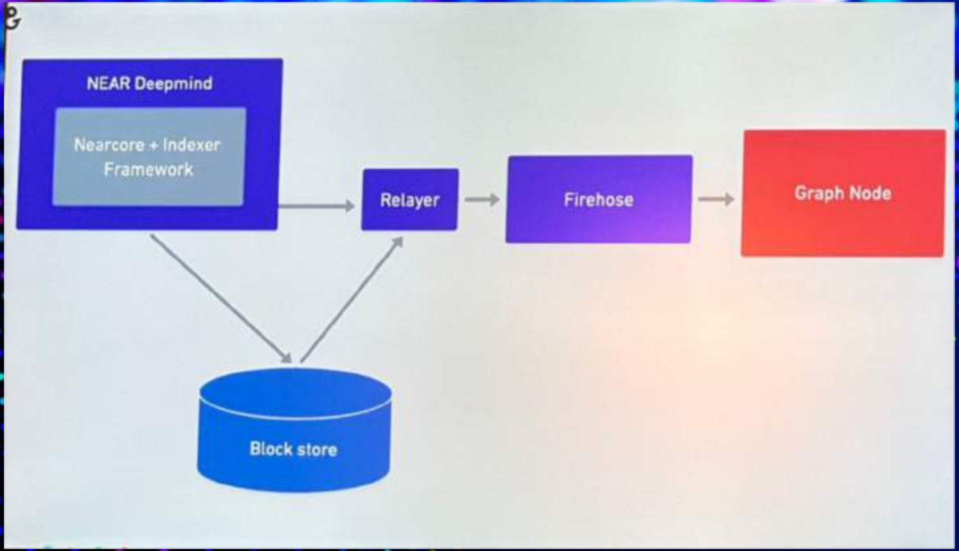


与NEAR强强联手

STREAMING FAST

高速数据架构平台

支持以太坊、EVM 链以及其它多种区块链协议，现为 The Graph 第二核心开发团队



StreamingFast 的 Firehose 架构实现对 Near Graph的集成

https://github.com/streamingfast/sf-near

https://github.com/streamingfast/near-dm-indexer

微信公众号

Twitter
@streamingfastio

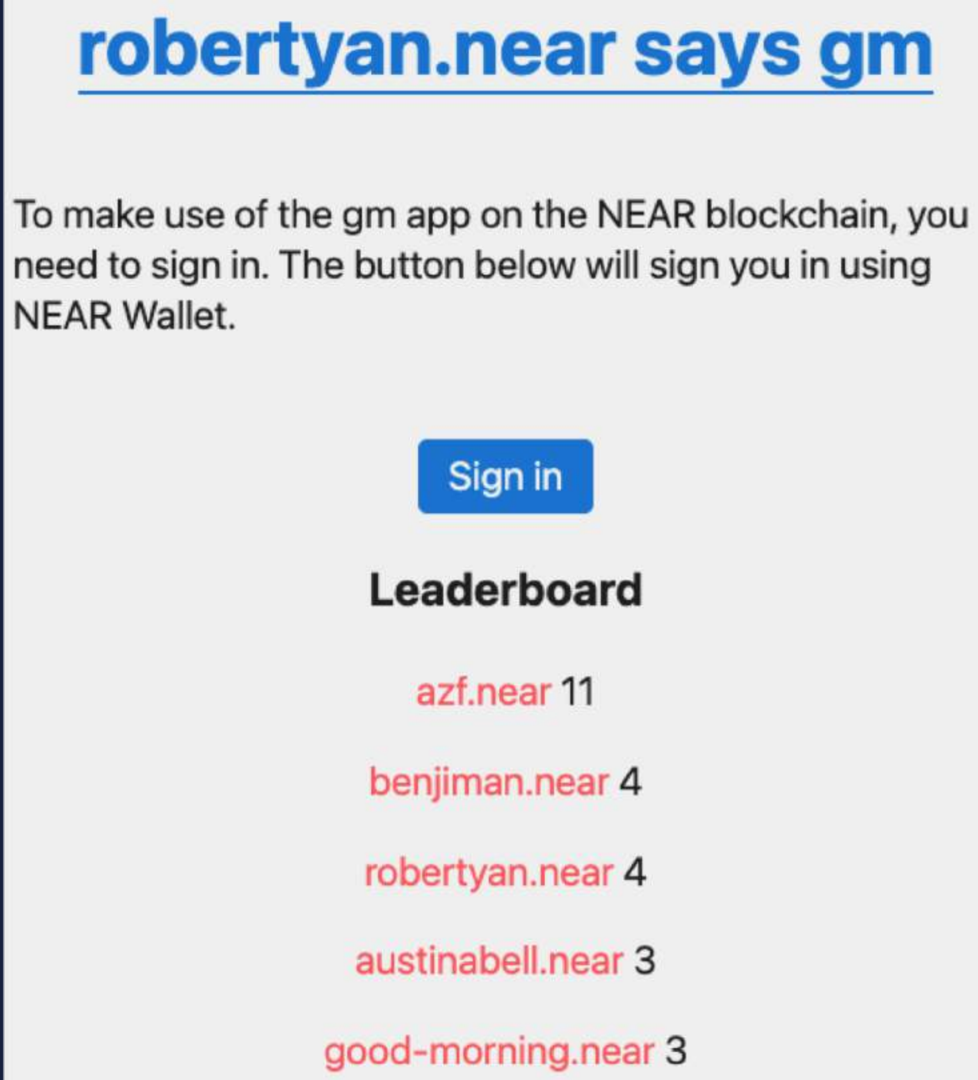
Discord

如何在Near上构建Subgraph?

分布式应用

https://gm-near.surge.sh

https://github.com/azf20/near-app-test/tree/gm-near



robertyan.near says gm

To make use of the gm app on the NEAR blockchain, you need to sign in. The button below will sign you in using NEAR Wallet.

Leaderboard

azf.near 11

benjiman.near 4

robertyan.near 4

austinabell.near 3

good-morning.near 3

如何在Near上构建Subgraph?

工具

@graphprotocol/graph-cli

@graphprotocol/graph-ts

https://thegraph.com/docs/developer/define-subgraph-hosted

如何在Near上构建Subgraph?

代码

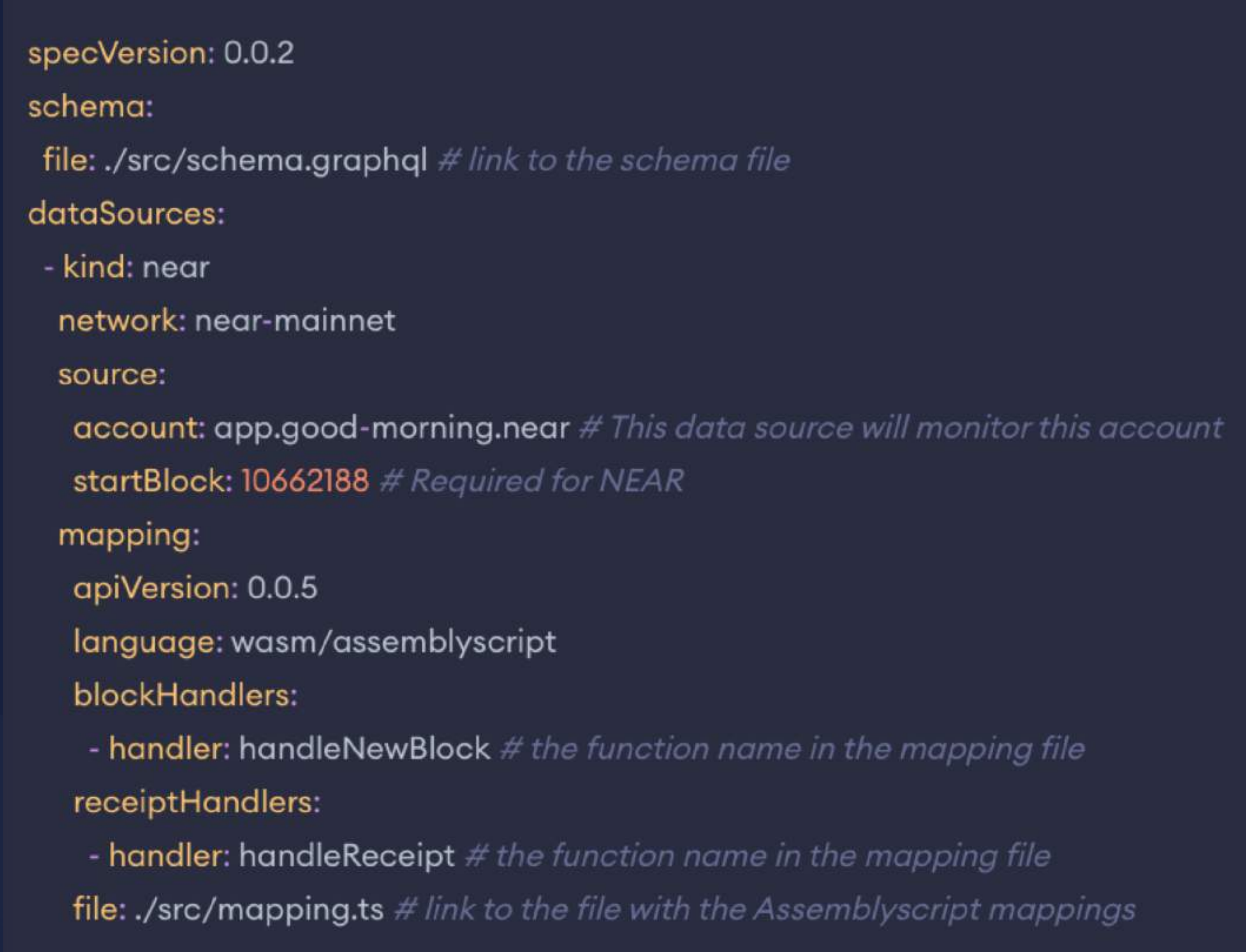
subgraph.yaml

schema.graphql

映射文件

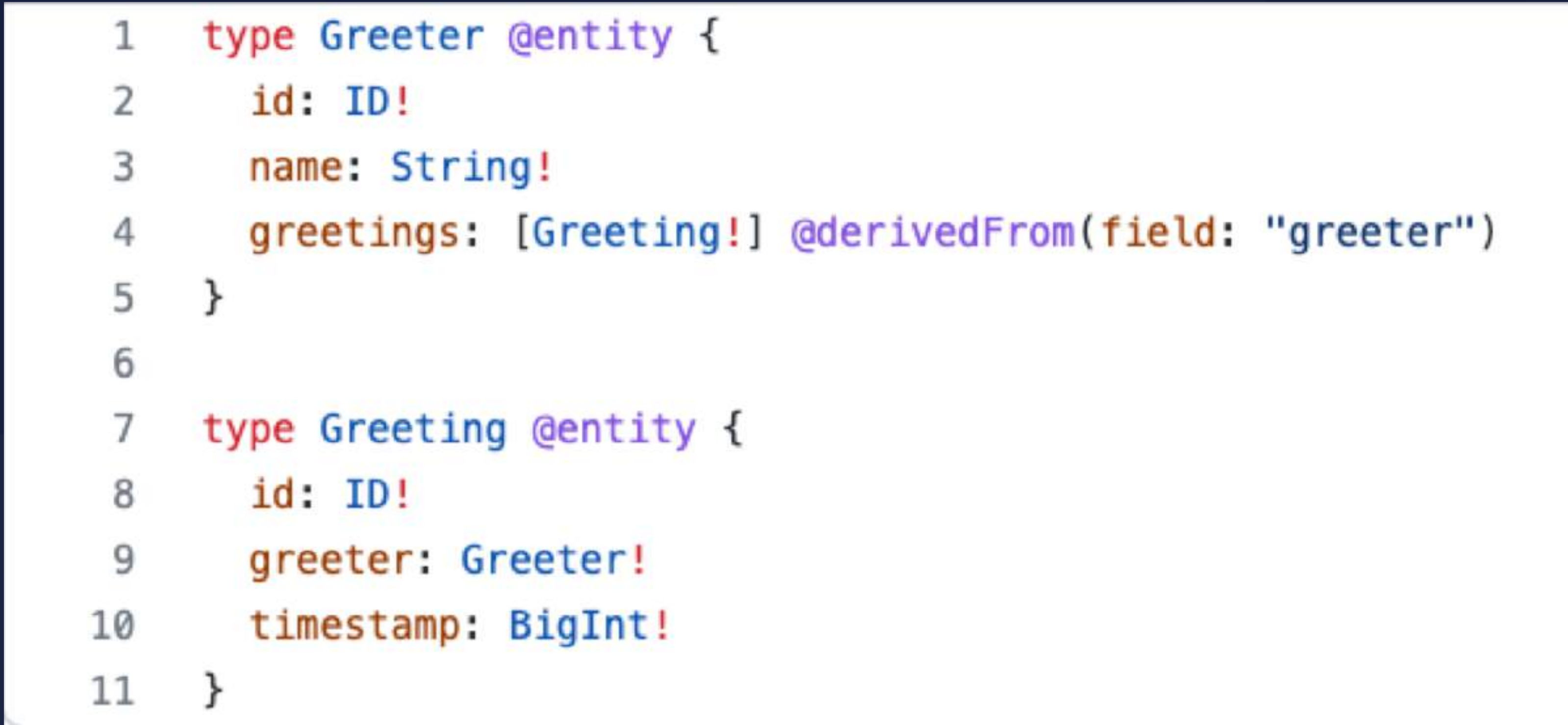
https://thegraph.com/docs/developer/create-subgraph-hosted

subgraph.yaml



```yaml
specVersion: 0.0.2
schema:
  file: ./src/schema.graphql # link to the schema file
dataSources:
  - kind: near
    network: near-mainnet
    source:
      account: app.good-morning.near # This data source will monitor this account
      startBlock: 10662188 # Required for NEAR
    mapping:
      apiVersion: 0.0.5
      language: wasm/assemblyscript
      blockHandlers:
        - handler: handleNewBlock # the function name in the mapping file
      receiptHandlers:
        - handler: handleReceipt # the function name in the mapping file
      file: ./src/mapping.ts # link to the file with the Assemblyscript mappings
```

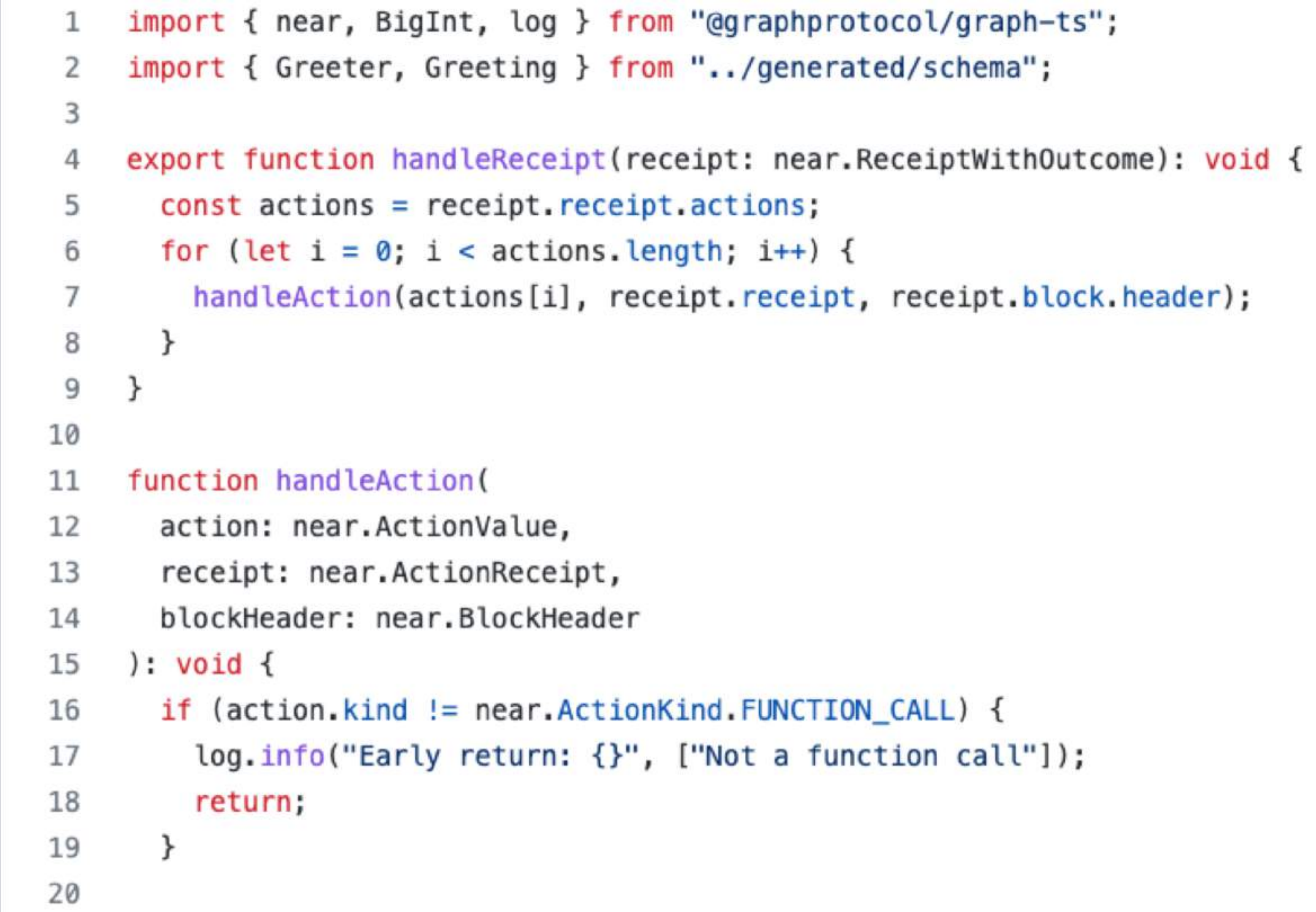
schema.graphql



```graphql
type Greeter @entity {
  id: ID!
  name: String!
  greetings: [Greeting!] @derivedFrom(field: "greeter")
}

type Greeting @entity {
  id: ID!
  greeter: Greeter!
  timestamp: BigInt!
}
```

映射文件



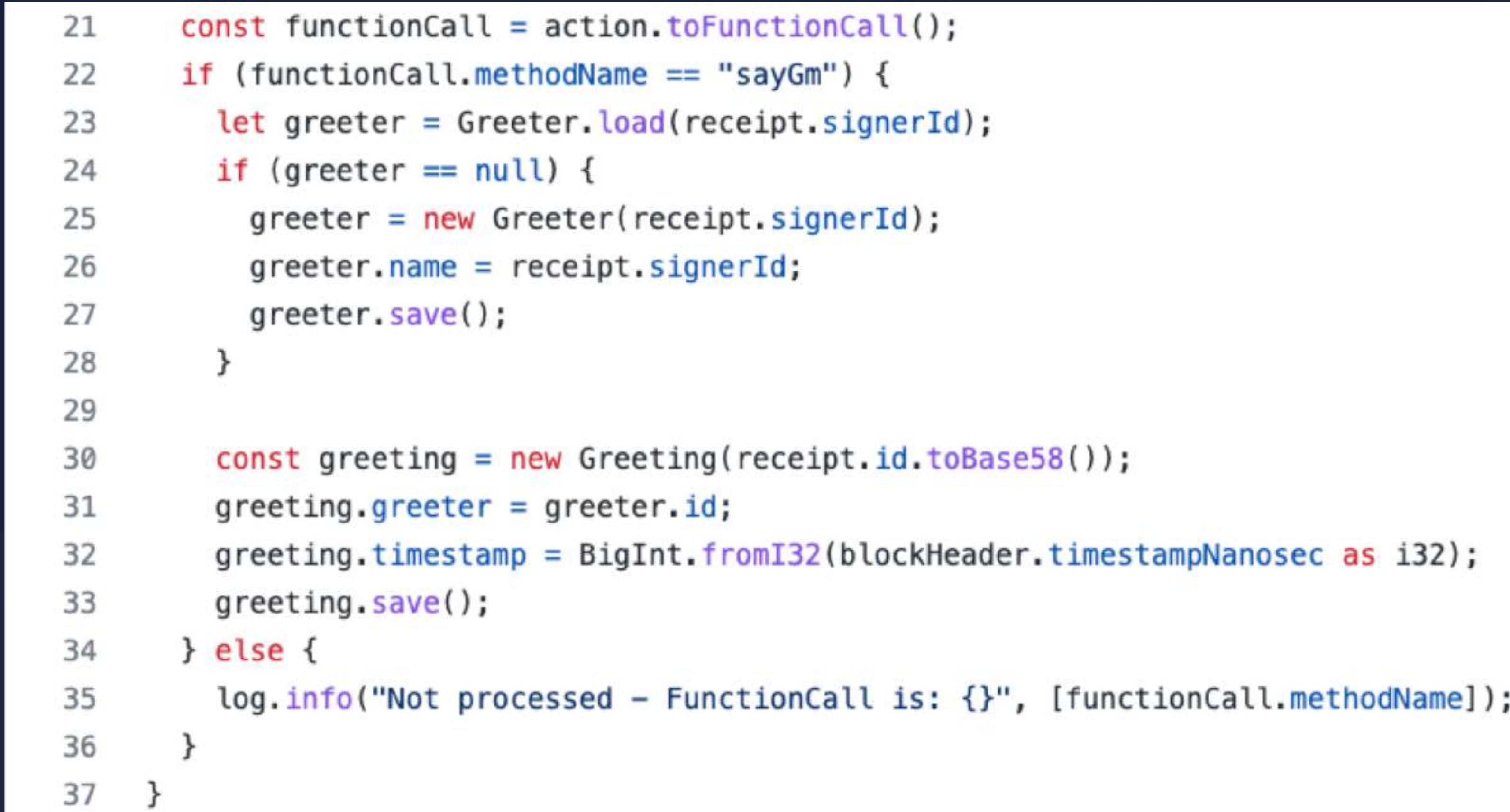
```
import { near, BigInt, log } from "@graphprotocol/graph-ts";
import { Greeter, Greeting } from "../generated/schema";

export function handleReceipt(receipt: near.ReceiptWithOutcome): void {
  const actions = receipt.receipt.actions;
  for (let i = 0; i < actions.length; i++) {
    handleAction(actions[i], receipt.receipt, receipt.block.header);
  }
}

function handleAction(
  action: near.ActionValue,
  receipt: near.ActionReceipt,
  blockHeader: near.BlockHeader
): void {
  if (action.kind != near.ActionKind.FUNCTION_CALL) {
    log.info("Early return: {}", ["Not a function call"]);
    return;
  }

```

映射文件

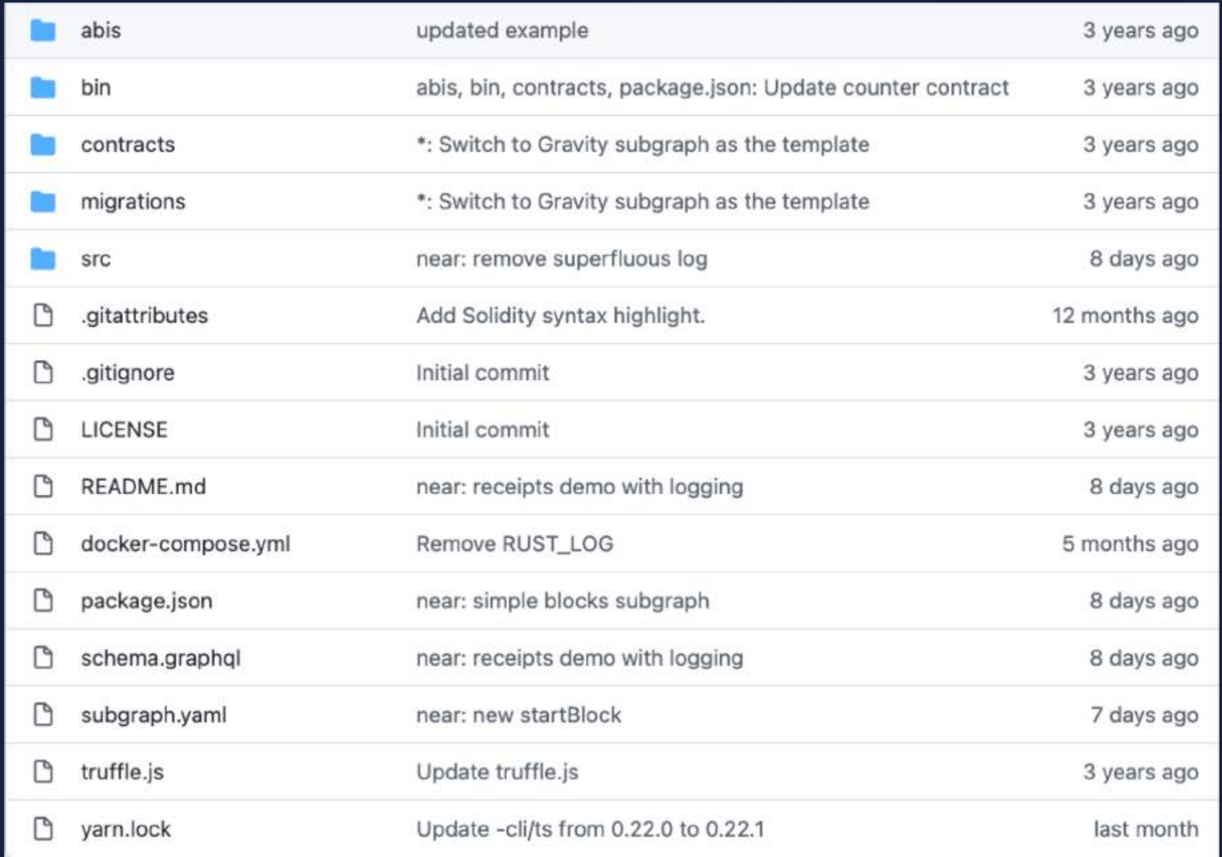


```
  const functionCall = action.toFunctionCall();
  if (functionCall.methodName == "sayGm") {
    let greeter = Greeter.load(receipt.signerId);
    if (greeter == null) {
      greeter = new Greeter(receipt.signerId);
      greeter.name = receipt.signerId;
      greeter.save();
    }

    const greeting = new Greeting(receipt.id.toBase58());
    greeting.greeter = greeter.id;
    greeting.timestamp = BigInt.fromI32(blockHeader.timestampNanosec as i32);
    greeting.save();
  } else {
    log.info("Not processed - FunctionCall is: {}", [functionCall.methodName]);
  }
}
```

完整代码

https://github.com/graphprotocol/example-subgraph/tree/near-receipts-example



abis	updated example	3 years ago
bin	abis, bin, contracts, package.json: Update counter contract	3 years ago
contracts	*: Switch to Gravity subgraph as the template	3 years ago
migrations	*: Switch to Gravity subgraph as the template	3 years ago
src	near: remove superfluous log	8 days ago
.gitattributes	Add Solidity syntax highlight.	12 months ago
.gitignore	Initial commit	3 years ago
LICENSE	Initial commit	3 years ago
README.md	near: receipts demo with logging	8 days ago
docker-compose.yml	Remove RUST_LOG	5 months ago
package.json	near: simple blocks subgraph	8 days ago
schema.graphql	near: receipts demo with logging	8 days ago
subgraph.yaml	near: new startBlock	7 days ago
truffle.js	Update truffle.js	3 years ago
yarn.lock	Update -cli/ts from 0.22.0 to 0.22.1	last month

如何在Near上构建Subgraph?

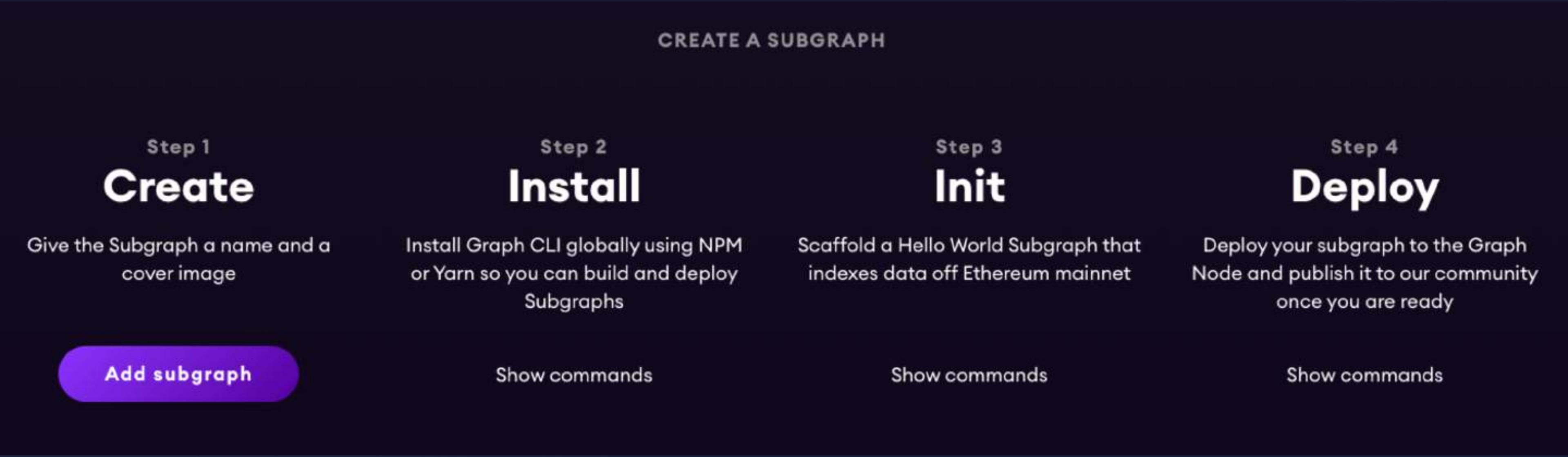
编译

```
$ graph codegen

$ graph build
```

部署

https://thegraph.com/hosted-service/dashboard
https://thegraph.com/docs/hostedservice/deploy-subgraph-hosted



CREATE A SUBGRAPH

Step 1
Create
Give the Subgraph a name and a cover image

Step 2
Install
Install Graph CLI globally using NPM or Yarn so you can build and deploy Subgraphs

Step 3
Init
Scaffold a Hello World Subgraph that indexes data off Ethereum mainnet

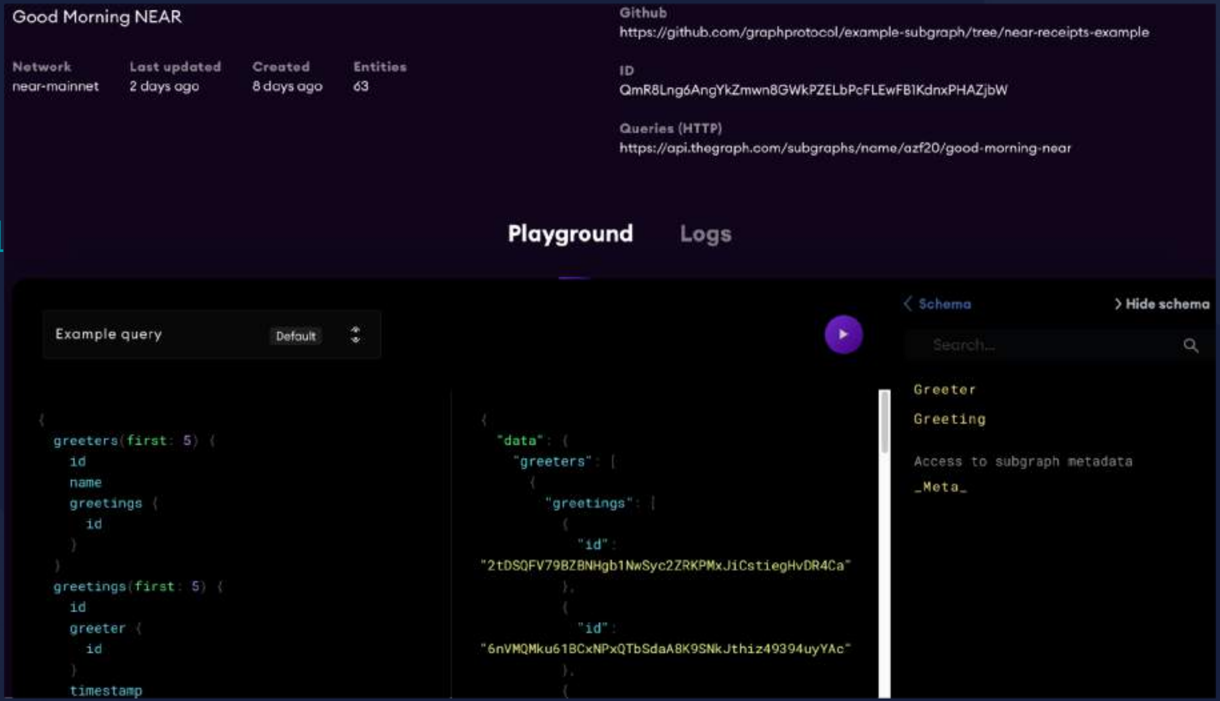
Step 4
Deploy
Deploy your subgraph to the Graph Node and publish it to our community once you are ready

如何查询Subgraph?

查询

https://thegraph.com/hosted-service/subgraph/azf20/good-morning-near?selected=playground

https://thegraph.com/docs/hostedservice/query-hosted-service


Good Morning NEAR

Network
near-mainnet

Last updated
2 days ago

Created
8 days ago

Entities
63

Github
https://github.com/graphprotocol/example-subgraph/tree/near-receipts-example

ID
QmR8Lng6AngYkZmwn8GWkPZELbPcFLEwFB1KdnxPHAZjbW

Queries (HTTP)
https://api.thegraph.com/subgraphs/name/azf20/good-morning-near

```
{
  greeters(first: 5) {
    id
    name
    greetings {
      id
    }
  }
  greetings(first: 5) {
    id
    greeter {
      id
    }
    timestamp
```

```
{
  "data": {
    "greeters": [
      {
        "greetings": [
          {
            "id":
"2tDSQFV79BZBNHgb1NwSyc2ZRKPMxJiCstiegHvDR4Ca"
          },
          {
            "id":
"6nVMQMku61BCxNPxQTbSdaA8K9SNkJthiz49394uyYAc"
          },
          {
```

Greeter
Greeting

Access to subgraph metadata
Meta

更多信息

https://thegraph.com/docs/supported-networks/near

near@thegraph.com

Web3
未来已来

让我们携手共创