

RBBID e The Graph com Blockchain Local

requisitos

Para este mini tutorial, temos os seguintes pré requisitos

- criar uma blockchain local com o contrato da rbb-identificacao

Obs.: olhar o tutorial Instalacao_RBBID_fora BNDES_bclocal.doc

0.instalando o Graph CLI

Para o uso do the Graph é preciso ter o Graph CLI instalado na sua máquina :

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

1.Inicie um nó de gráfico local

Para iniciar um nó do the graph, é preciso baixar o projeto pré configurado

```
git clone https://github.com/graphprotocol/graph-node/
```

Após clonar o projeto, acesse a pasta /graph-node/docker. Lá você vai encontrar o docker-compose.yml, dentro desse arquivo é preciso mudar a linha ethereum: 'mainnet:<http://host.docker.internal:8545>'. Essa linha, especifica a rede blockchain ('<http://host.docker.internal:8545>') que o the graph precisa acessá e o nome da rede (mainnet).

Obs: o nome da rede é escolhido de forma arbitrária, para o nosso exemplo, vamos chamá-la de rbb. Já a rede, vai depender do ip/porta, da sua rede blockchain.

Ethereum: 'rbb:http://ip:port'

dado que tudo acima já esteja configurado, basta levantar o contêiner:

```
docker-compose up
```

o docker-compose up irá criar 3 containers

1. graphprotocol/graph-node
2. postgres
3. ipfs/go-ipfs

you pode verificar a criação dos containers com o
docker ps

2.Criando o projeto Graph cli

Escolha um diretório e rode o seguinte comando

graph init

rodando o graph init no terminal, ele vai fazer uma série de perguntas.

O tipo do projeto: hosted-service

o nome do subgraph: deveríamos colocar o nome de um github aqui, mas para o nosso exemplo isso não será necessário, basta colocar rbb/id

nome do diretório que será criado:

Ethereum network: aqui pode ser escolhido qualquer um, pois mudaremos esse valor mais a frente

Contract address:

caminho para o seu ABI:

após o passo anterior você vai ter um projeto pré configurado.
Substitua os arquivos do projeto pelo seguintes arquivos do github de mesmo nome

theGraph/id/schema.graphql
theGraph/id/subgraph.yaml
theGraph/id/src/mapping.ts

2.1 Configurando Graph cli

Esse passo só é necessário caso os container não estejam na mesma máquina que você instalou o Graph CLI

Agora é preciso acessar a pasta id, onde pode ser encontrado o arquivo package.json. Nele vamos editar as seguintes linhas.

```
"create-local": "graph create --node http://localhost:8020/ rbb/id",
```

```
"remove-local": "graph remove --node http://localhost:8020/ rbb/id",
```

```
"deploy-local": "graph deploy --node http://localhost:8020/ --ipfs  
http://localhost:5001 rbb/id"
```

Mude o localhost:8020 para o ip do graph-node e o localhost:5001 para o ip do ipfs

2.2 configurando Graph cli

para finalizar, acesse o arquivo subgraph.yaml, e adicione o contract address em address e adicione em network o nome colocado no docker-compose.yml no passo 1, no nosso caso vai ficar

```
network: rbb
```

3.EXECUTAR O SISTEMA

Agora basta rodar os seguintes comandos

```
yarn codegen
```

gera alguns código CLI do Graph:

```
yarn create-local
```

esse comando coloca o nome do subGraph no graph-node

```
yarn deploy-local
```

para fazer o deploy

no yarn deploy-local, você vai encontra a Queries (HTTP), basta colocá-la no browser, e começar a fazer as consultas

consulta de exemplo:

```
{  
  accounts(first:100) {  
    id  
    addr  
    RBBId  
    CNPJ  
    responsible  
    reason  
    hashProof  
    dateTimeExpiration  
  }  
}
```