



blockscout 安装过程

[环境变量](#)

[启动脚本：](#)

[server 信息](#)

[安装各种第三方包](#)

[安装erlang和elixir和nodejs](#)

[安装 rust](#)

[\(本地, 可选\) 安装postgres与redis](#)

[\(远程, 可选\) postgres数据库](#)

[moonbeam rpc \(需要续费\)](#)

[部署位置](#)

[参考资料 手动部署方式](#)

[clone代码](#)

[使用的moonbeam节点：](#)

[设置变量](#)

[运行elixir命令, 编译和安装代码](#)

[保证postgres 已经启动运行](#)

[创建和运行数据库迁移：mix do ecto.create, ecto.migrate](#)

[开始安装nodejs的各种依赖：](#)

[生成签名。](#)

[修改 /etc/hosts](#)

[修改浏览器：](#)

[启动:](#)

[踩过的坑](#)

[务必使用香港服务器。](#)

[官方文档过期了。](#)

[配置文件, 一定要注意不多不少。](#)

[对于moonbeam, 这样的网络, 需要使用配置项: ganache](#)

[DATADOG_PORT 的设置, 这里有问题。](#)

[类型转换](#)

[left value not match ...](#)

[访问入口](#)

<https://github.com/blockscout/blockscout>

配置需求参考 <https://docs.blockscout.com/for-developers/information-and-settings/requirements>

手动安装参考：<https://docs.blockscout.com/for-developers/manual-deployment>

环境变量

把这个文件保存到本地，记得每次启动前要运行：

```
export NETWORK=Moonbeam
export SUBNETWORK=Coreth
export LOGO=/images/blockscout_logo.svg
export LOGO_FOOTER=/images/blockscout_logo.svg

# 对于moonbeam这样的网络, 只能选择ganache
export ETHEREUM_JSONRPC_VARIANT=ganache
export ETHEREUM_JSONRPC_HTTP_URL=https://moonbeam.api.onfinality.io/rpc?apikey=c78dd454-f16f-4f67-8a03-67b316c4aaee
export ETHEREUM_JSONRPC_TRACE_URL=https://moonbeam.api.onfinality.io/rpc?apikey=c78dd454-f16f-4f67-8a03-67b316c4aaee
export ETHEREUM_JSONRPC_WS_URL=wss://moonbeam.api.onfinality.io/ws?apikey=c78dd454-f16f-4f67-8a03-67b316c4aaee
```

```

# 这里必须是http 否则会走ipc 协议。
export ETHEREUM_JSONRPC_TRANSPORT=http
export SECRET_KEY_BASE=VTIB3uHDnbvY0+60ZWgUoUBKdn9ppLR8MI4CpRz4/qLyEFs54ktJfaNT6Z221No
export DATABASE_URL=postgres://postgres:88888888@localhost:5432/blockscout

export ETHEREUM_JSONRPC_DISABLE_ARCHIVE_BALANCES=false
#export IPC_PATH=
export NETWORK_PATH=/
export API_PATH=/
export SOCKET_ROOT=/

# 这里用localhost 就好, 记得前端用caddy
export BLOCKSCOUT_HOST=localhost
export BLOCKSCOUT_PROTOCOL=http
export CHECK_ORIGIN=false
export PORT=4000
export COIN=GLMR
export COIN_NAME=GLMR
export REWARDS_CONTRACT=0xeca443e8e1ab29971a45a9c57a6a9875701698a5
export TOKEN_BRIDGE_CONTRACT=0x7301CFA0e1756B71869E93d4e4Dca5c7d0eb0AA6
export EMISSION_FORMAT=DEFAULT
export POOL_SIZE=40
export POOL_SIZE_API=10
export ECTO_USE_SSL=false
export DATADOG_HOST=
export DATADOG_PORT=
export SPANDEX_BATCH_SIZE=10
export SPANDEX_SYNC_THRESHOLD=100
export HEART_BEAT_TIMEOUT=30
#export HEART_COMMAND=
#export BLOCKSCOUT_VERSION=
#export RELEASE_LINK=
#export ELIXIR_VERSION=

export BLOCK_TRANSFORMER=base
export LINK_TO_OTHER_EXPLORERS=false
export OTHER_EXPLORERS={}
export SUPPORTED_CHAINS={}
export CACHE_BLOCK_COUNT_PERIOD=7200
export CACHE_TXS_COUNT_PERIOD=7200
export CACHE_ADDRESS_COUNT_PERIOD=7200
export CACHE_ADDRESS_SUM_PERIOD=3600
export CACHE_TOTAL_GAS_USAGE_PERIOD=3600
export CACHE_ADDRESS_TRANSACTIONS_GAS_USAGE_COUNTER_PERIOD=1800
export CACHE_TOKEN_HOLDERS_COUNTER_PERIOD=3600
export CACHE_TOKEN_TRANSFERS_COUNTER_PERIOD=3600
export CACHE_ADDRESS_WITH_BALANCES_UPDATE_INTERVAL=1800
export CACHE_AVERAGE_BLOCK_PERIOD=1800
export CACHE_MARKET_HISTORY_PERIOD=21600
export CACHE_ADDRESS_TRANSACTIONS_COUNTER_PERIOD=1800
export CACHE_ADDRESS_TOKENS_USD_SUM_PERIOD=1800
export CACHE_ADDRESS_TOKEN_TRANSFERS_COUNTER_PERIOD=1800
export CACHE_BRIDGE_MARKET_CAP_UPDATE_INTERVAL=1800
export CACHE_TOKEN_EXCHANGE_RATE_PERIOD=1800
export TOKEN_METADATA_UPDATE_INTERVAL=172800
export ALLOWED_EVM_VERSIONS=homestead,tangerineWhistle,spuriousDragon,byzantium,constantinople,petersburg,istanbul,berlin,london,default
export UNCLER_IN_AVERAGE_BLOCK_TIME=false
export DISABLE_WEBAPP=false
export DISABLE_READ_API=false
export DISABLE_WRITE_API=false
export DISABLE_INDEXER=false
export DISABLE_REALTIME_INDEXER=false
export DISABLE_TOKEN_INSTANCE_FETCHER=false
export INDEXER_DISABLE_PENDING_TRANSACTIONS_FETCHER=false
export INDEXER_DISABLE_INTERNAL_TRANSACTIONS_FETCHER=false
export INDEXER_CATCHUP_BLOCKS_BATCH_SIZE=10
export INDEXER_CATCHUP_BLOCKS_CONCURRENCY=10
export INDEXER_INTERNAL_TRANSACTIONS_BATCH_SIZE=10
export INDEXER_INTERNAL_TRANSACTIONS_CONCURRENCY=4
export INDEXER_COIN_BALANCES_BATCH_SIZE=500
export INDEXER_COIN_BALANCES_CONCURRENCY=4
export INDEXER_RECEIPTS_BATCH_SIZE=250
export INDEXER_RECEIPTS_CONCURRENCY=10
export TOKEN_ID_MIGRATION_FIRST_BLOCK=0
export TOKEN_ID_MIGRATION_CONCURRENCY=1
export TOKEN_ID_MIGRATION_BATCH_SIZE=500
#export WEBAPP_URL=
#export API_URL=
export WOBSERVER_ENABLED=false
export SHOW_ADDRESS_MARKETCAP_PERCENTAGE=true
export CHECKSUM_ADDRESS_HASHES=true

```

```
export CHECKSUM_FUNCTION=eth
export DISABLE_EXCHANGE_RATES=true
export DISABLE_KNOWN_TOKENS=false
export ENABLE_TXS_STATS=true
export SHOW_PRICE_CHART=false
export SHOW_TXS_CHART=true
export HISTORY_FETCH_INTERVAL=60
export TXS_HISTORIAN_INIT_LAG=0
export TXS_STATS_DAYS_TO_COMPILE_AT_INIT=365
export COIN_BALANCE_HISTORY_DAYS=90
export APPS_MENU=false
export EXTERNAL_APPS=[]

export BRIDGE_MARKET_CAP_UPDATE_INTERVAL=1800

export SHOW_MAINTENANCE_ALERT=false
export CUSTOM_CONTRACT_ADDRESSES_TEST_TOKEN=
export ENABLE_SOURCIFY_INTEGRATION=false

export MAX_SIZE_UNLESS_HIDE_ARRAY=50
export HIDE_BLOCK_MINER=false
export DISPLAY_TOKEN_ICONS=false
export SHOW_TENDERLY_LINK=false

export MAX_STRING_LENGTH_WITHOUT_TRIMMING=2040

export API_RATE_LIMIT=50
export API_RATE_LIMIT_BY_KEY=50
export API_RATE_LIMIT_BY_IP=50

export FETCH_REWARDS_WAY=trace_block
export ENABLE_RUST_VERIFICATION_SERVICE=true
export RUST_VERIFICATION_SERVICE_URL=http://localhost:8043/
export VISUALIZE_SOL2UML_ENABLED=true
export VISUALIZE_SOL2UML_SERVICE_URL=http://localhost:8050/
# 这两个千万不要留空。
#export DATABASE_READ_ONLY_API_URL=
#export ACCOUNT_DATABASE_URL=
export ACCOUNT_POOL_SIZE=100
export ACCOUNT_ENABLED=false
export ACCOUNT_REDIS_URL=redis://localhost:6379
```

启动脚本：

```
source env_source
iex -S mix phx.server
```

server 信息

ssh ubuntu@blockscout.enterdao.im

域名：blockscout.raindrop.link

安装各种第三方包

```
sudo apt update
sudo apt install libapr1-dev libxml2-dev libxslt1-dev libmysqlclient-dev git curl build-essential libssl-dev libreadline-dev build-essential
sudo apt install automake libtool gcc make g++ libgmp-dev
```

安装erlang和elixir和nodejs

使用asdf 来安装。

具体过程在这里: <http://siwei.me/blog/posts/erlang-asdf-erlang-elixir-nodejs-ruby>

```
elixir 1.14.2-otp-25
erlang 25.1.2
nodejs 16.16.0
```

安装 rust

```
curl https://sh.rustup.rs -sSf | sh
```

(本地, 可选) 安装 postgres 与 redis

由于 postgres 比较麻烦, 所以我选择使用 docker 来安装。

```
/opt/app$ cat docker-compose.yml
version: '3'
services:
  postgres14:
    container_name: "postgres14"
    image: postgres:14
    volumes:
      - /opt/app/docker_data/postgres14:/var/lib/postgresql/data
      #command: 'tail -F /dev/null'
    command: 'postgres'
    ports:
      - "5432:5432"
    stdin_open: true
    tty: true
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=88888888

  redis:
    image: redis
    ports:
      - "6379:6379"
    volumes:
      - /opt/app/docker_folder/redis:/data
```

运行 postgres , redis:

```
cd /opt/app
sudo docker-compose up -d
```

(远程, 可选) postgres 数据库

见: <https://www.notion.so/pns-team/blockscout-0332084fa17847579ffda27d5de3c9ab>

这个就是当前使用的

moonbeam rpc (需要续费)

```
Moonbeam
HTTPS
https://moonbeam.api.onfinality.io/rpc?apikey=c78dd454-f16f-4f67-8a03-???
Websocket
```

```
wss://moonbeam.api.onfinality.io/ws?apikey=c78dd454-f16f-4f67-8a03-???
```

部署位置

/opt/app 目录

/opt/app/docker-compose.yml docker-compose 文件

/opt/app/blockscout 本app的目录

/opt/app/docker_folder/ redis与 postgres的数据目录

参考资料 手动部署方式

<https://docs.blockscout.com/for-developers/manual-deployment>

clone代码

```
git clone https://github.com/poanetwork/blockscout
```

使用的moonbeam节点：

<https://docs.moonbeam.network/cn/builders/get-started/endpoints/>

<https://moonbeam.api.onfinality.io/public>

<wss://moonbeam.api.onfinality.io/public-ws>

设置变量

(具体见这个说明页面，里面包含了一个csv: <https://docs.blockscout.com/for-developers/information-and-settings/env-variables>)

```
export NETWORK=Moonbeam
export SUBNETWORK=Coreth
export LOGO=/images/blockscout_logo.svg
export LOGO_FOOTER=/images/blockscout_logo.svg
export ETHEREUM_JSONRPC_VARIANT=parity
export ETHEREUM_JSONRPC_HTTP_URL=moonbeam.api.onfinality.io/public
export ETHEREUM_JSONRPC_TRACE_URL=moonbeam.api.onfinality.io/public
export ETHEREUM_JSONRPC_WS_URL=wss://moonbeam.api.onfinality.io/public-ws
export ETHEREUM_JSONRPC_TRANSPORT=https
export SECRET_KEY_BASE=VTIB3uHDNbvY0+60ZwgUoUBKdn9ppLR8MI4CpRz4/qLyEFs54ktJfaNT6Z221No
export PORT=4000
export COIN=GLMR
export DATABASE_URL=postgresql://postgres:88888888@localhost:5432/blockscout
```

运行elixir命令，编译和安装代码

```
# mix do deps.get, local.rebar --force, deps.compile, compile

Could not find Hex, which is needed to build dependency :tesla
Shall I install Hex? (if running non-interactively, use "mix local.hex --force") [Yn]
* creating /root/.asdf/installs/elixir/1.12.3-otp-24/.mix/archives/hex-1.0.1
Resolving Hex dependencies...
Dependency resolution completed:
Unchanged:
  absinthe 1.6.5
  absinthe_phoenix 2.0.2
```

```

absinthe_relay 1.5.1
accept 0.3.5
bcrypt_elixir 1.1.1

..... 经过漫长的等待 (10分钟?)
==> phoenix_ecto
Compiling 7 files (.ex)
Generated phoenix_ecto app
==> block_scout_web
Compiling 201 files (.ex)
Generating BlockScoutWeb.Cldr for 2 locales named ["en", "root"] with a default locale named "en"

09:24:47.199 [debug] Current global API rate limit 50 reqs/sec

09:24:47.206 [debug] Current API rate limit by key 50 reqs/sec

09:24:47.206 [debug] Current API rate limit by IP 50 reqs/sec
Generated block_scout_web app# mix do deps.get, local.rebar --force, deps.compile, compile
Could not find Hex, which is needed to build dependency :tesla
Shall I install Hex? (if running non-interactively, use "mix local.hex --force") [Yn]
* creating /root/.asdf/installs/elixir/1.12.3-otp-24/.mix/archives/hex-1.0.1
Resolving Hex dependencies...
Dependency resolution completed:
Unchanged:
  absinthe 1.6.5
  absinthe_phoenix 2.0.2
  absinthe_relay 1.5.1
  accept 0.3.5
  bcrypt_elixir 1.1.1

```

保证postgrest 已经启动运行

创建和运行数据库迁移：mix do ecto.create, ecto.migrate

```

The database for Explorer.Repo has been created
The database for Explorer.Repo.Account has already been created

07:19:51.987 [info] == Running 20180117221921 Explorer.Repo.Migrations.CreateAddress.change/0 forward
07:19:51.993 [info] create table addresses
07:19:52.004 [info] == Migrated 20180117221921 in 0.0s
....
10:08:18.105 [info] == Migrated 20211206071033 in 0.0s
10:08:18.107 [info] == Running 20220111085751 Explorer.Repo.Migrations.AddressAddCounters.change/0 forward
10:08:18.107 [info] alter table addresses
10:08:18.107 [info] == Migrated 20220111085751 in 0.0s

```

开始安装nodejs的各种依赖：

运行下列命令：

```

$ cd /opt/app/blockscout/
$ cd apps/block_scout_web/assets
$ npm install
$ node_modules/webpack/bin/webpack.js --mode production (这个时间比较久)
$ cd -

```

结果如下：

```

....
autocomplete (270 KiB)
  autocomplete.js
search-results (1010 KiB)
  ../css/search-results.css
  search-results.js

WARNING in webpack performance recommendations:
You can limit the size of your bundles by using import() or require.ensure to lazy load some parts of your application.
For more info visit https://webpack.js.org/guides/code-splitting/

```

```
webpack 5.75.0 compiled with 3 warnings in 363359 ms
```

然后继续：

```
$ cd apps/explorer  
$ npm install  
$ cd -
```

生成静态文件：

```
$ mix phx.digest
```

这里会报错，说几个路径找不到。我继续了。

```
root@iZj6c676alhqd017zcx9l3Z:/opt/app/blockscout# mix phx.digest  
==> ethereum_jsonrpc  
The input path "priv/static" does not exist  
==> explorer  
The input path "priv/static" does not exist  
==> indexer  
The input path "priv/static" does not exist  
==> block_scout_web  
Check your digested files at "priv/static"
```

生成签名。

phenix 不用https不能访问，据说。

```
cd apps/block_scout_web; mix phx.gen.cert blockscout blockscout.local; cd -
```

修改 /etc/hosts

把 blockscout blockscout.local 加进来

```
127.0.0.1 localhost blockscout blockscout.local
```

```
127.0.0.1 localhost blockscout blockscout.local  
  
# The following lines are desirable for IPv6 capable hosts  
::1 localhost ip6-localhost ip6-loopback blockscout blockscout.local  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
  
172.31.189.236 iZj6c676alhqd017zcx9l3Z iZj6c676alhqd017zcx9l3Z
```

修改浏览器：

(参考：<https://stackoverflow.com/questions/50819206/is-there-a-equivalent-of-allow-insecure-localhost-flag-of-google-chrome-in-firefox>)

chrome: Enable chrome://flags/#allow-insecure-localhost

firefox: about:config -> network.websocket.allowInsecureFromHTTPS 选择true

终于配置好了。

启动:

```
$ mix phx.server
```

踩过的坑

务必使用香港服务器。

国内服务器的话，很多时候网络是无法访问的。

官方文档过期了。

需要参考Dockerfile 和 官方给的ENV 列表，对照才行。

当然了还必须具备elixir的阅读代码的能力。

配置文件，一定要注意不多不少。

可以填错，但是不可以留空。留空的话elixir会报错。估计是类型转换引起的

对于moonbeam, 这样的网络, 需要使用配置项: ganache

否则 internal transaction 无法解析

```
ETHEREUM_JSONRPC_VARIANT=ganache
```

DATADOG_PORT 的设置, 这里有问题。

```
50 export DATABASE_URL="postgresql://myroot:123456@localhost
51 export POOL_SIZE="20"
52 export ECTO_USE_SSL="TRUE"
53 export DATADOG_HOST="localhost"
54 export DATADOG_PORT=8126
55 export SPANDEX_BATCH_SIZE=10
56 export SPANDEX_SYNC_THRESHOLD=100
57 export HEART_BEAT_TIMEOUT="30"
58 export HEART_COMMAND=""
```

类型转换

报错原因：can not convert string into integer 类似的话。。

解决办法：找到报错的app: apps/explorer/lib/explorer/application.ex

直接把value写死。(或者在System.get_env 之后，把对应的结果手动转换成 int.)

left value not match ...

这里会出现很多次。需要慢慢调试。

找到报错的 ex文件（这个提示中没有，很不友好，需要根据上一条报错的app来找到。）

例如：

```
** (Mix) Could not start application explorer: Explorer.Application.start(:normal, []) returned an error: shutdown: failed to start child: Explorer.Chain.Events.Listener
** (EXIT) an exception was raised:
** (KeyError) key :password not found in: [types: nil, types: Postgrex.DefaultTypes, username: "root", sync_connect: true, hostname: "localhost", database: "blockscout", migration_timestamps: [type: :utc_datetime_usec], url: "postgresql://myroot:123456@localhost:5432/blockscout", pool_size: 20, timeout: 80000]
(elixir 1.12.3) lib/keyword.ex:420: Keyword.fetch!/2
(postgrex 0.15.10) lib/postgrex/protocol.ex:778: Postgrex.Protocol.auth_md5/4
```

光看报错信息是不知道是哪个文件的，所以，需要根据class的名称： Explorer.Application, 找到对应的ex文件：

```
2022-01-25T09:59:36.118 [info] Application explorer exited: Explorer.Application.start(:normal, []) returned an error: shutdown: failed to start child: Explorer.Chain.Events.Listener
** (EXIT) an exception was raised:
** (KeyError) key :password not found in: [types: nil, types: Postgrex.DefaultTypes, username: "root", sync_connect: true, hostname: "localhost", database: "blockscout", migration_timestamps: [type: :utc_datetime_usec], url: "postgresql://myroot:123456@localhost:5432/blockscout", pool_size: 20, timeout: 80000]
```

所以找到对应的 apps/explorer/lib/explorer/chain/events/listener.ex

进行修改即可：

```
{:ok, pid} =
  explorer_repo
  |> Notifications.start_link()
修改成：

temp = explorer_repo
  |> Notifications.start_link()
IO.puts "=== temp is: #{inspect temp}"
{:ok, pid} = temp
```

这样的话，报错信息就会可读可理解了，变成下图所示：

```
** (KeyError) key :password not found in: [types: nil, types: Postgrex.DefaultTypes, username: "root", sync_connect: true, hostname: "localhost", database: "blockscout", migration_timestamps: [type: :utc_datetime_usec], url: "postgresql://myroot:123456@localhost:5432/blockscout", pool_size: 20, timeout: 80000]
```

我们就知道，原来缺少了password，就可以针对性的做代码的调整了（下面截图是把password写死）

```
26   temp = explorer_repo
27     |> Keyword.put(:database, extract_db_name(db_url))
28     |> Keyword.put(:hostname, extract_db_host(db_url))
29     |> Keyword.put(:password, '123456')
30     |> Keyword.put(:username, 'myroot')
31     IO.puts "=== temp: #{inspect temp}"
```

访问入口

<https://blockscout.raindrop.link/>