한 번에 끝내는
블록체인 개발 A to Z

Chapter 2

Blockchain 2.0 - Ethereum

Chapter 2

Blockchain 2.0 - Ethereum

Node와 Client

Go Ethereum

❶ 공식 Full Node

Ethereum Full Node는 총 3가지 존재한다. Go-Ethereum(geth), cppEthereum(aleth)(현재 미지원), pyethereum(현재 미지원) 으로 존재한다. 공식적이지는 않지만 많이 사용했던 openethereum(parity)(현재 미지원)이 존재한다.

❷ 다양한 통신방안

Socket통신만 지원하는 bitcoind와는 달리 geth는 IPC, HTTP, Websocket등 다양한 방법으로 Client와 통신할 수 있는 방안을 제공한다.

❸ 사용방안에 따른 용량관리

Bitcoind prune와 달리 geth는 사용 방안에 따라 여러가지 모드를 제공한다. Snap, Full, light 모드등을 제공한다. 여기에 archive 모드를 on/off 하느냐에 따라 접근 가능한 블록체인 데이터가 달라진다.

Geth 실행
주요 Option

❶ --http.api

JSON RPC API(admin,clique, debug, eth, les, miner, personal, txpool,web3) 중 Client에 허락되는 API 만 지정하여 Open이 가능하다.

❷ --networkid

테스트넷과 메인넷 그리고 Private Blockchain 중 선택하여 운영이 가능하도록 한다.

❸ --syncmode

블록체인 데이터를 어느정도 가지고 있을지 정하는 것이다. Snap 모드는 약 400GB정도이고, Full 모드의 경우 1000GB정도이다.

❹ --gcmode

Full, Archive 모드로 구분되며 Archive의 경우 약 11TB정도의 데이터 저장공간이 필요하다. 모든 History 데이터가 조회가 가능하다.

Geth 설치 및 실행

- Geth는 Binary파일을 다운 받아서 실행하면 자동으로 default 값으로 설정된 값으로 실행이 된다.
- Geth 설치 후 Node에 접근하는 방법은 2가지 방법이 있다.
- >> geth console 은 node실행과 console 실행이 동시에 진행된다.
- >> geth attach 은 노드를 실행한 상태에서 진행해야한다.

Geth API 예제

- Geth Attach에서 가장 먼저 진행해야하는 것은 Block Syncing이 완료되어야 한다는 것이다.
- 전체 블록체인이 Sync가 완료되어야 해당 계정의 State를 정확하게 알 수 있고, Transaction을 실행시킬 수 있게 된다.

```
#geth 접속
>>geth attach
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.1-stable-1e67410e/windows-386/go1.9.2
 modules: admin:1.0 debug:1.0 eth:1.0 net:1.0 personal:1.0 rpc:1.0
 txpool:1.0 web3:1.0

> eth.syncing
false
> eth.blockNumber
5160607
> eth.coinbase
0x6afb711d9ec4a47496df4e7e5173a23e898bb264
> eth.getBalance(personal.listAccounts[0])
0
> eth.getBalance(eth.coinbase)
0
```

JSON-RPC API

❶ eth_getBlockByNumber

Ethereum 블록 정보를 조회합니다.
Number, Header Hash 기준으로 조회가 가능합니다.

❷ eth_newfilter

Geth에서 제공 받을 Event 정보를 등록합니다. 특정 주소, 이벤트 형식에 따라서 Dapp에서 조회가 가능합니다.

❸ eth_getTransactionCount

사용자가 거래 생성을 위해서 현재 nonce 정보를 조회합니다.

❹ eth_sendTransaction

거래의 Sign과 Network 배포를 동시에 진행하는 함수입니다. 서명된 Transaction의 경우 sendRawTransaction을 사용합니다.

GraphQL

- 2015년 개발된 새로운 방식의 어플리케이션 Query Language이다.
- 클라이언트에서 서버에 필요한 데이터를 지정해서 요청하는 방식으로, REST API와는 달리 하나의 URL로 데이터 처리가 가능하다.

```
>> curl -X POST http://test-
server/getUserInfo
-d {userName : "testUser"}

{"result": {"userName":
"testUser", "userId" : "100",
"userPhone": 010-1121-1111}}
```

```
>> curl -X POST http://test-
server/graphql
-d { "query": "{ UserInfo(userName:
"testUser") { userName userId
userPhone }

{"data": {"UserInfo": {"userName":
"testUser", "userId" : "100",
"userPhone": 010-1121-1111}}}
```

Infura Node

❶ Full Node 운영 대행

24시간 운영이 되어야 하는 Full Node는 운용이 힘들기 때문에, 최근 Infura와 같은 노드 운영 대행 업체들이 많이 등장하였다.

❷ 대행 운영의 장점

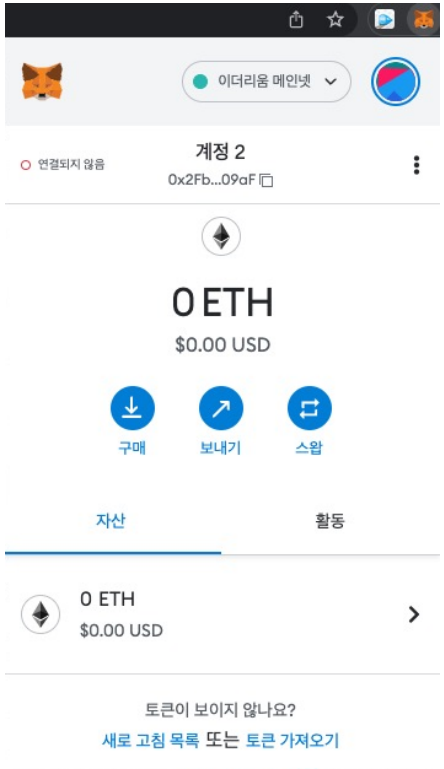
대행운행을 하게되면, 사용자가 24시간 운영해야하는 서버를 준비할 필요도 없으며, 많은 데이터 용량 관리나 자주 있는 업데이트를 신경쓰지 않아도 되는 운영상의 장점이 있다.

❸ 대행 운영의 단점

중앙화된 기관의 문제점이 발생할 수 있다. Infura Node의 데이터가 위변조 되는 경우, 잘못된 거래를 생성할 수 있다. 또한 Infura Node가 중지되는 경우 서비스가 중단될 수 있는 문제가 있다.

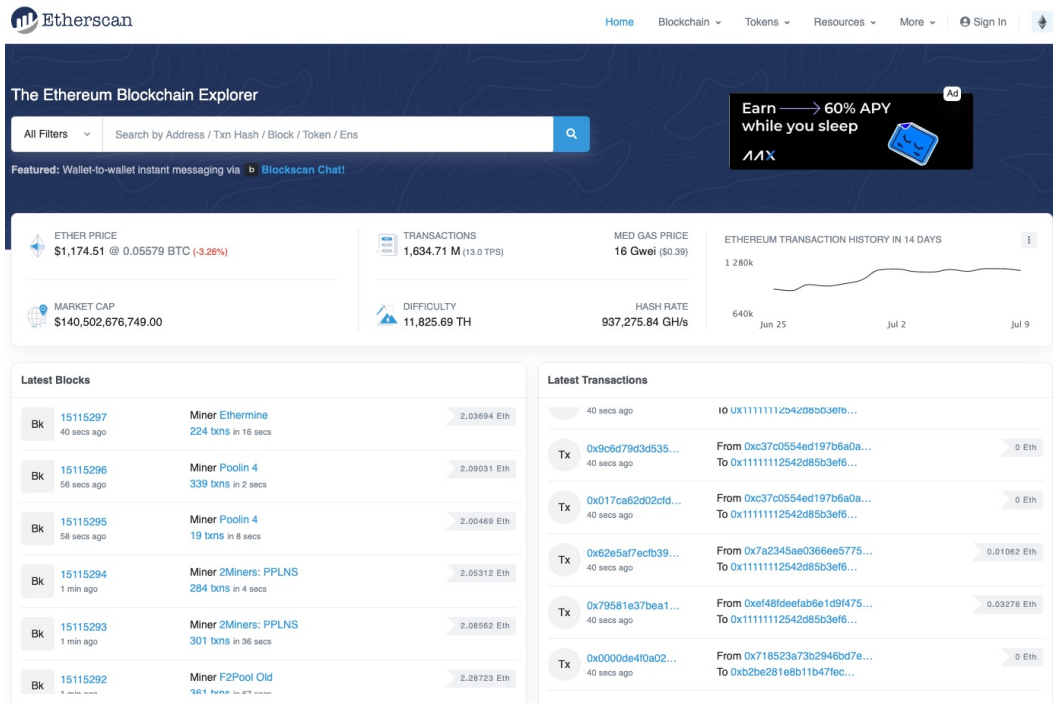
Metamask

- Chrome Extension 기반 Wallet
- 다양한 네트워크(메인넷, 테스트넷, Private Blockchain)에 접근가능
- 일반 사용자 뿐 아니라 Remix IDE를 통한 웹을 통한 개발 시 사용
- Defi 서비스 이용가능
- 안전한 Key 관리



Etherscan

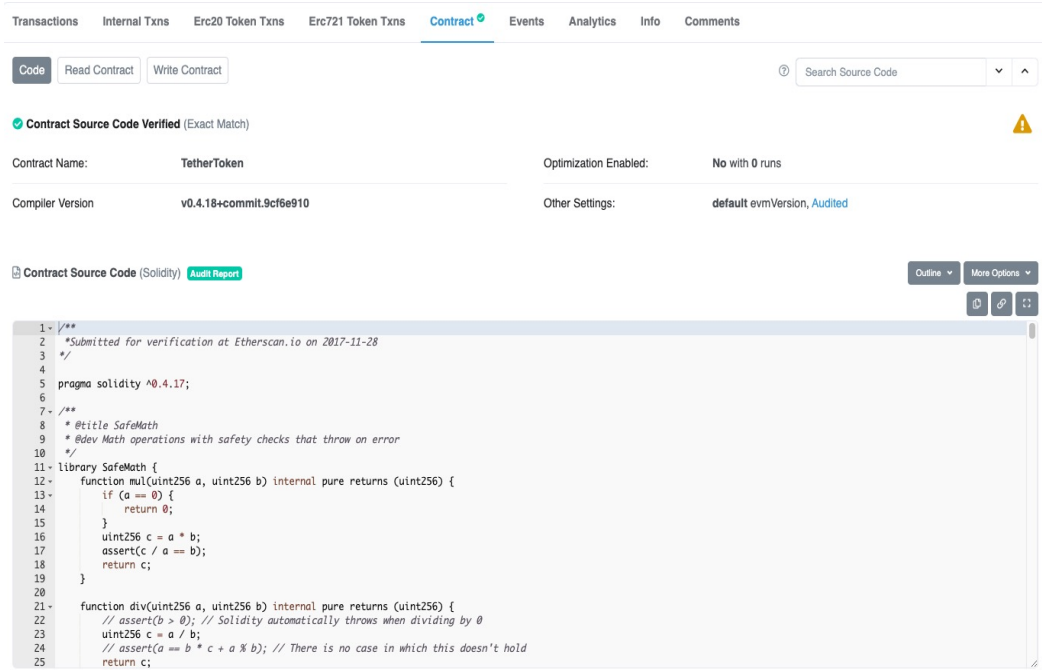
- Ethereum의 대표적인 Explorer
- Block, Transaction등 블록체인 전체 정보 확인 가능
- 테스트넷 개발 시 테스트 결과를 확인하는 용도로 활용
- 일반 서비스의 컨트랙트 소스코드 검증용으로도 활용



(출처 : etherscan.io)

Etherscan Source 검증

- Smart Contract 배포 시 Source Code가 네트워크에 배포되는 것이 아닌 Bytecode만 배포
- Source Code상에 악성 코드 존재 여부를 확인 불가
- Bytecode에서 source code로 decompile은 거의 불가능에 가까움
- Explorer에서 검증 시스템 제공



(출처 : etherscan.io)