Dive into Ethereum – Clients and EVM

Ethereum was co-founded by Vitalik Buterin and a team of co-founders including Gavin Wood, Charles Hoskinson, Jeffrey Wilcke, Mihai Alisie, Amir Chetrit, and Anthony Di Iorio. While Buterin is widely recognized as the primary visionary and author of the Ethereum whitepaper, the project's development and success were a collaborative effort.

Ethereum:

- Ethereum is a blockchain with a computer embedded in it.
- It is the foundation for building apps and organizations in a decentralized, permissionless, censorship-resistant way.
- In the Ethereum universe, there is a single, canonical computer (called the Ethereum Virtual Machine, or EVM) whose state everyone on the Ethereum network agrees on.
- Everyone who participates in the Ethereum network (every Ethereum node) keeps a copy of the state of this computer.
- Additionally, any participant can broadcast a request for this computer to perform arbitrary computation.
- Whenever such a request is broadcast, other participants on the network verify, validate, and carry out ("execute") the computation.
- This execution causes a state change in the EVM, which is committed and propagated throughout the entire network.
- Requests for computation are called transaction requests; the record of all transactions and the EVM's present state gets stored on the blockchain, which in turn is stored and agreed upon by all nodes.

❖ **Ethereum client** is a software that:

- Implements the Ethereum protocol.
- Connects to other nodes in the network.
- Verifies and propagates transactions.
- Maintains the blockchain state.
- Allows developers/users to interact with the blockchain via RPC APIs.

  Think of it as:

  The “browser” for the Ethereum network — without it, you can’t see or interact with the blockchain.

❖ **Types of Ethereum Clients:**

Ethereum has multiple client implementations — all follow the same protocol but are written in different languages.

Execution Clients (EVM-focused)

- Geth (Go Ethereum) — written in Go.

- Nethermind — written in C#.
- Besu — Java-based.
- Erigon — Go-based, optimized for archival nodes.

They execute transactions, maintain state, and run the EVM.

❖ **Consensus Clients (Proof-of-Stake era)**

Since The Merge (2022), Ethereum uses two layers:

- Execution layer (runs the EVM, processes transactions).
- Consensus layer (validates blocks, runs PoS).

Consensus clients:

- Prysm — Go.
- Lighthouse — Rust.
- Teku — Java.
- Nimbus — Nim.

These talk to execution clients via Engine API (secured with JWT tokens).

❖ **EVM – Ethereum Virtual Machine**

The EVM is:

- A runtime environment that executes smart contract code.
- Deterministic: given the same inputs, all nodes get the same output.
- Stack-based architecture.
- Works with EVM bytecode compiled from Solidity, Vyper, etc.
- Runs in an isolated sandbox (no direct access to OS, file system, or internet).

Key EVM Features

- Gas model: Each computation costs "gas" to prevent infinite loops & spam.
- Deterministic execution: All nodes run the same operations for consensus.
- State management:
  - Account state (balances, nonces, code, storage).
  - World state (global mapping of addresses → account data).

❖ **How Clients and EVM Work Together**

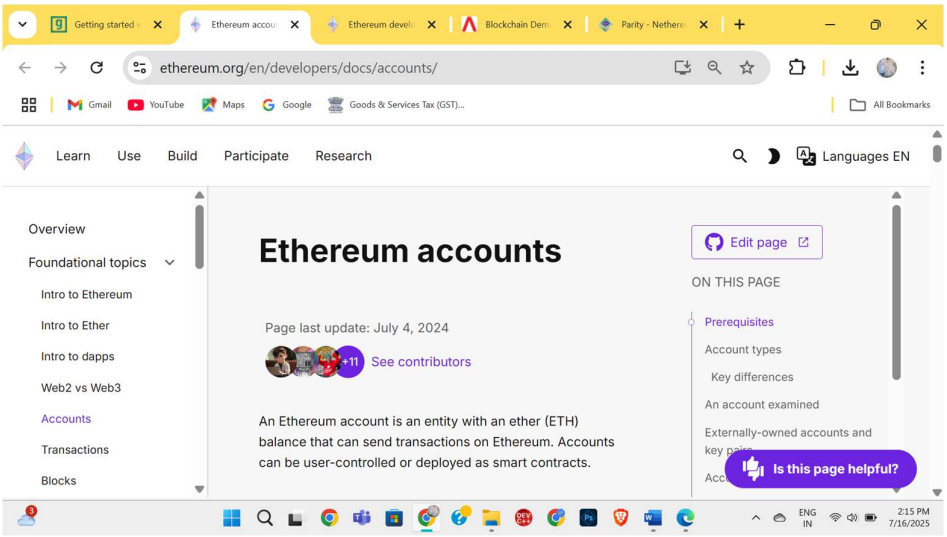
Flow of a transaction:

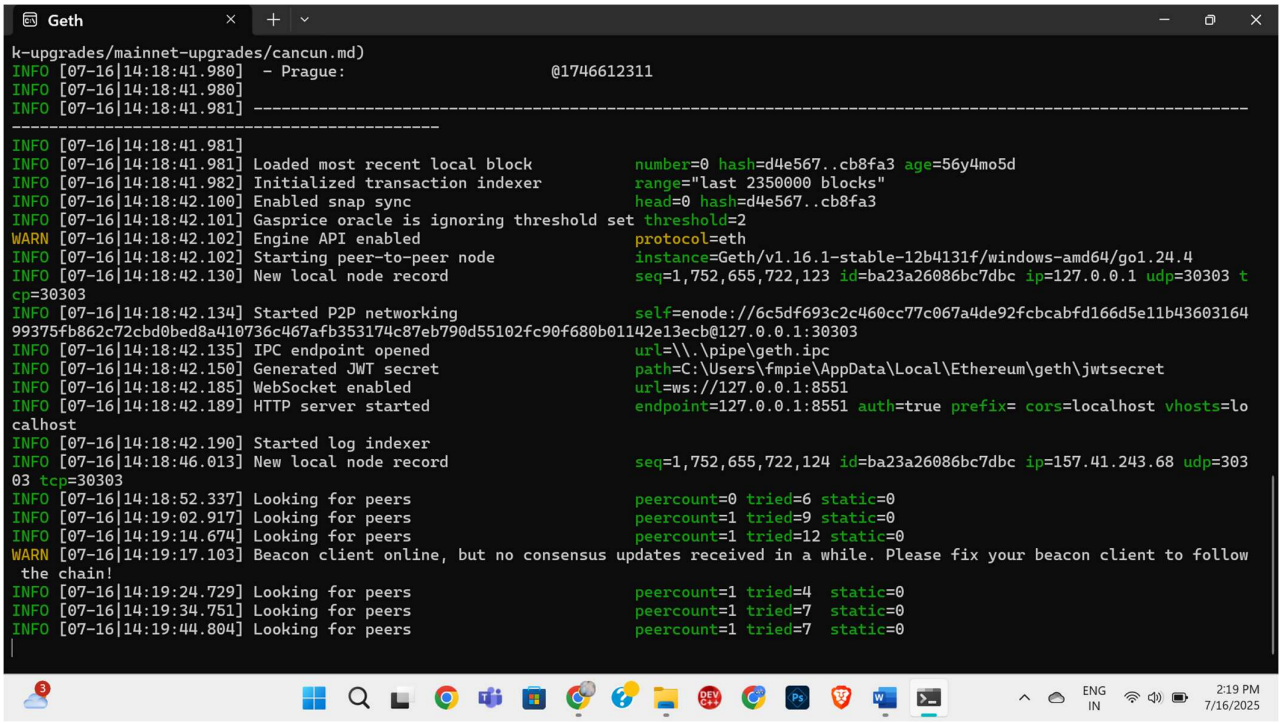
1. User/DApp sends transaction and Execution client receives it.
2. Execution client:
   - Verifies transaction (signature, nonce, gas).
   - Executes it in EVM.
   - Updates state (balances, storage, contract code).
3. Execution client passes block info to Consensus client.
4. Consensus client:
   - Runs Proof-of-Stake validation.
   - Finalizes block on-chain.

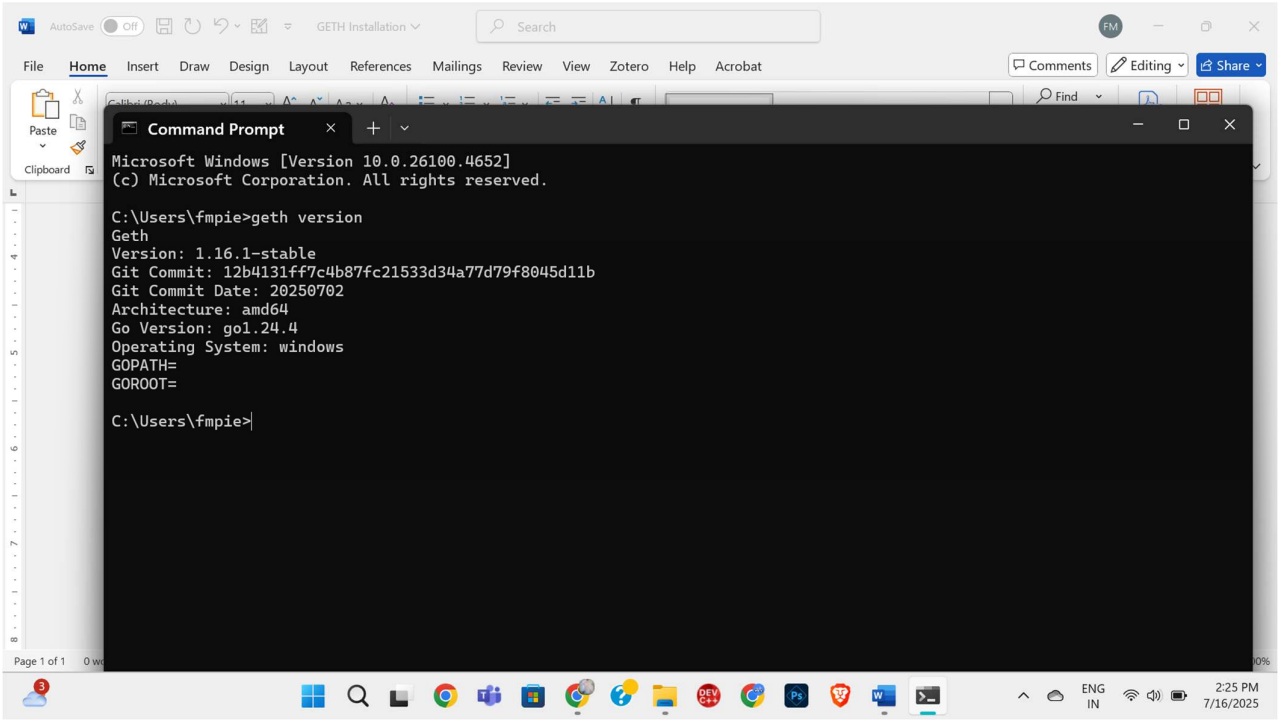
Ether:

- Ether (ETH) is the native cryptocurrency of Ethereum.
- The purpose of ETH is to allow for a market for computation. Such a market provides an economic incentive for participants to verify and execute transaction requests and provide computational resources to the network.
- Any participant who broadcasts a transaction request must also offer some amount of ETH to the network as a bounty.
- The network will burn part of the bounty and award the rest to whoever eventually does the work of verifying the transaction, executing it, committing it to the blockchain, and broadcasting it to the network.
- The amount of ETH paid corresponds to the resources required to do the computation.
- ETH is also used to provide crypto-economic security to the network in three main ways:
    1) it is used as a means to reward validators who propose blocks or call out dishonest behavior by other validators;
    2) It is staked by validators, acting as collateral against dishonest behavior—if validators attempt to misbehave their ETH can be destroyed;
    3) it is used to weigh 'votes' for newly proposed blocks, feeding into the fork-choice part of the consensus mechanism.

Geth Installation Process :



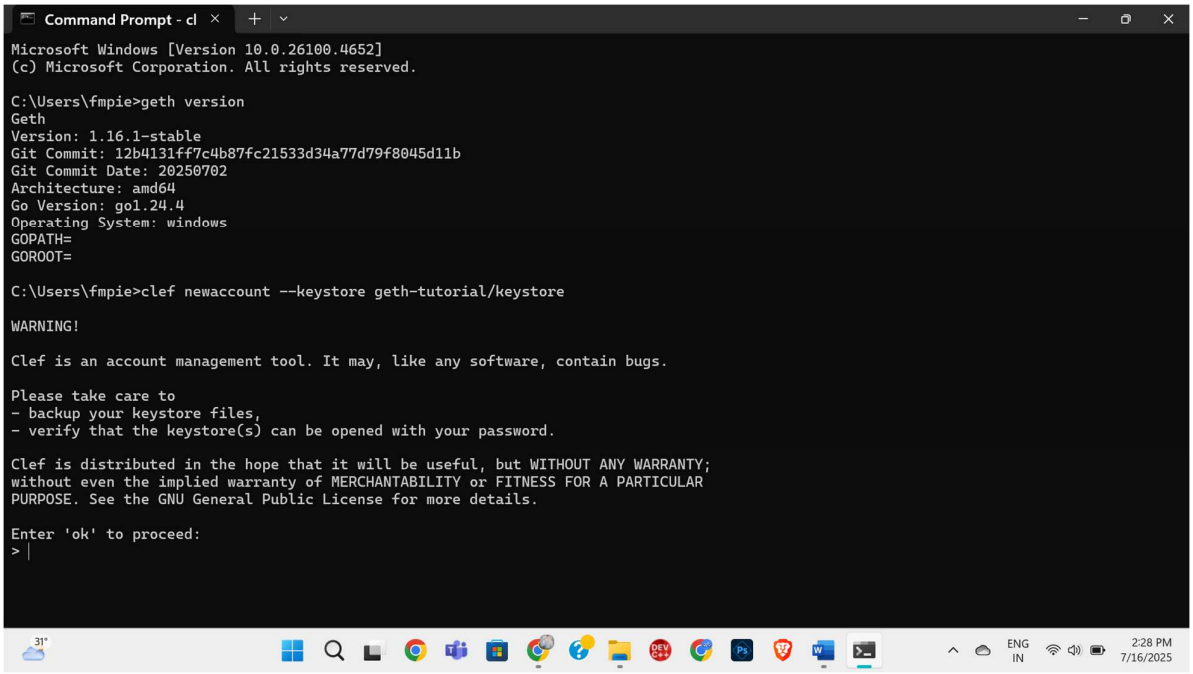

Geth
k-upgrades/mainnet-upgrades/cancun.md)
INFO [07-16|14:18:41.980] - Prague: @1746612311
INFO [07-16|14:18:41.980]
INFO [07-16|14:18:41.981] ----------
INFO [07-16|14:18:41.981]
INFO [07-16|14:18:41.981] Loaded most recent local block   number=0 hash=d4e567..cb8fa3 age=56y4mo5d
INFO [07-16|14:18:41.982] Initialized transaction indexer   range="last 2350000 blocks"
INFO [07-16|14:18:42.100] Enabled snap sync   head=0 hash=d4e567..cb8fa3
INFO [07-16|14:18:42.101] Gasprice oracle is ignoring threshold set threshold=2
WARN [07-16|14:18:42.102] Engine API enabled   protocol=eth
INFO [07-16|14:18:42.102] Starting peer-to-peer node   instance=Geth/v1.16.1-stable-12b4131f/windows-amd64/go1.24.4
INFO [07-16|14:18:42.130] New local node record   seq=1,752,655,722,123 id=ba23a26086bc7dbc ip=127.0.0.1 udp=30303 t
cp=30303
INFO [07-16|14:18:42.134] Started P2P networking   self=enode://6c5df693c2c460cc77c067a4de92fcbcabfd166d5e11b43603164
99375fb862c72cbd0bed8a410736c467afb353174c87eb790d55102fc90f680b01142e13ecb@127.0.0.1:30303
INFO [07-16|14:18:42.135] IPC endpoint opened   url=\\.\pipe\geth.ipc
INFO [07-16|14:18:42.150] Generated JWT secret   path=C:\Users\fmpie\AppData\Local\Ethereum\geth\jwtsecret
INFO [07-16|14:18:42.185] WebSocket enabled   url=ws://127.0.0.1:8551
INFO [07-16|14:18:42.189] HTTP server started   endpoint=127.0.0.1:8551 auth=true prefix= cors=localhost vhosts=lo
calhost
INFO [07-16|14:18:42.190] Started log indexer
INFO [07-16|14:18:46.013] New local node record   seq=1,752,655,722,124 id=ba23a26086bc7dbc ip=157.41.243.68 udp=303
03 tcp=30303
INFO [07-16|14:18:52.337] Looking for peers   peercount=0 tried=6 static=0
INFO [07-16|14:19:02.917] Looking for peers   peercount=1 tried=9 static=0
INFO [07-16|14:19:14.674] Looking for peers   peercount=1 tried=12 static=0
WARN [07-16|14:19:17.103] Beacon client online, but no consensus updates received in a while. Please fix your beacon client to follow
 the chain!
INFO [07-16|14:19:24.729] Looking for peers   peercount=1 tried=4 static=0
INFO [07-16|14:19:34.751] Looking for peers   peercount=1 tried=7 static=0
INFO [07-16|14:19:44.804] Looking for peers   peercount=1 tried=7 static=0
ENG
IN
2:19 PM
7/16/2025

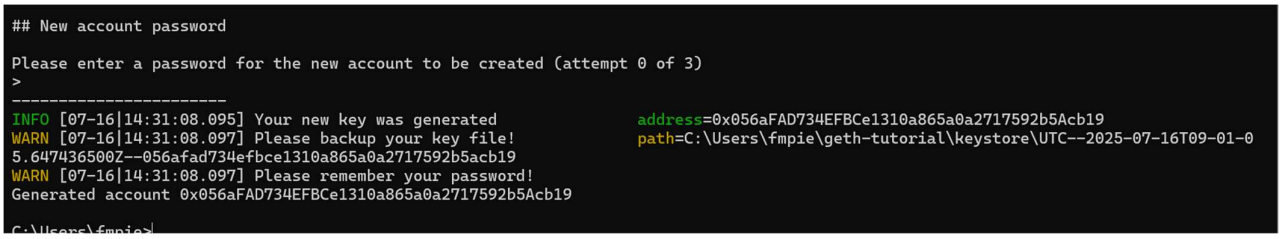

AutoSave
GETH Installation
Search
File
Home
Insert
Draw
Design
Layout
References
Mailings
Review
View
Zotero
Help
Acrobat
Comments
Editing
Share
Paste
Clipboard
Find
Command Prompt
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.
C:\Users\fmpie>geth version
Geth
Version: 1.16.1-stable
Git Commit: 12b4131ff7c4b87fc21533d34a77d79f8045d11b
Git Commit Date: 20250702
Architecture: amd64
Go Version: go1.24.4
Operating System: windows
GOPATH=
GOROOT=
C:\Users\fmpie>
Page 1 of 1
ENG
IN
2:25 PM
7/16/2025

Getting started with Geth

Step 1: Generating accounts:

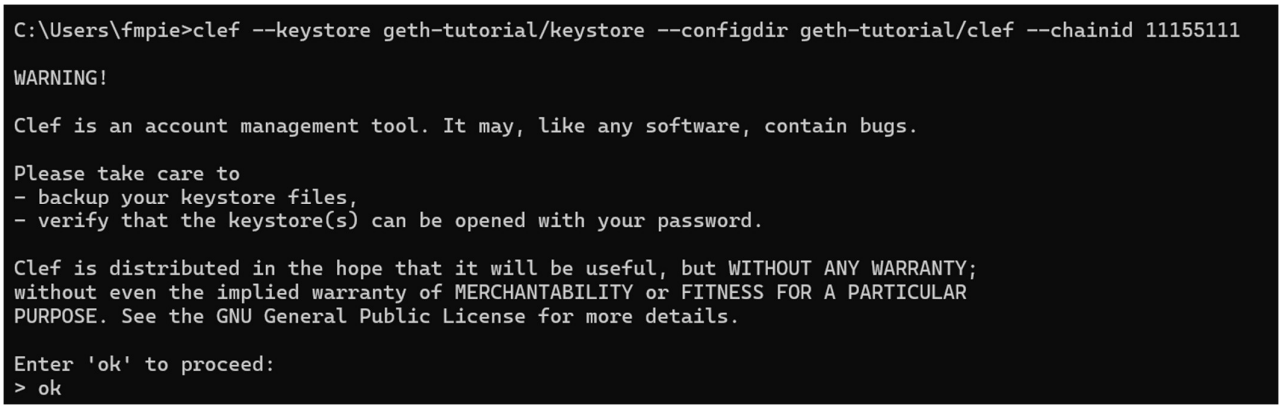
clef newaccount --keystore geth-tutorial/keystore





Step 2: Start Clef:

clef --keystore geth-tutorial/keystore --configdir geth-tutorial/clef --chainid 11155111



```
INFO [07-16|14:34:18.691] Using CLI as UI-channel
INFO [07-16|14:34:19.143] Loaded 4byte database                    embeds=268,621 locals=0 local=./4byte-custom.json
WARN [07-16|14:34:19.144] Failed to open master, rules disabled    err="failed stat on geth-tutorial\clef\masterseed.json: CreateFile
 geth-tutorial\clef\masterseed.json: The system cannot find the path specified."
INFO [07-16|14:34:19.144] Starting signer                          chainid=11,155,111 keystore=geth-tutorial/keystore light-kdf=false
 advanced=false
INFO [07-16|14:34:19.161] Audit logs configured                    file=audit.log
INFO [07-16|14:34:19.163] IPC endpoint opened                      url=\\.\pipe\clef.ipc

------- Signer info -------
* intapi_version : 7.0.1
* extapi_version : 6.1.0
* extapi_http : n/a
* extapi_ipc : \\.\pipe\clef.ipc

------- Available accounts -------
0. 0x056aFAD734EFBCe1310a865a0a2717592b5Acb19 at keystore://C:\Users\fmpie\geth-tutorial\keystore\UTC--2025-07-16T09-01-05.647436500Z
--056afad734efbce1310a865a0a2717592b5acb19
```

Step 3: Start Geth:

geth --sepolia --datadir geth-tutorial --authrpc.addr localhost --authrpc.port 8551 --authrpc.vhosts localhost --authrpc.jwtsecret geth-tutorial/jwtsecret --http --http.api eth,net --signer=geth-tutorial/clef/clef.ipc --http

```
C:\Users\fmpie\geth-tutorial>clef --keystore clef-keystore --configdir clef --ipcpath clef/clef.ipc

WARNING!

Clef is an account management tool. It may, like any software, contain bugs.

Please take care to
- backup your keystore files,
- verify that the keystore(s) can be opened with your password.

Clef is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the GNU General Public License for more details.

Enter 'ok' to proceed:
>
```

```
INFO [07-16|22:03:54.422] Using CLI as UI-channel
INFO [07-16|22:03:54.840] Loaded 4byte database                    embeds=268,621 locals=0 local=./4byte-custom.json
WARN [07-16|22:03:54.841] Failed to open master, rules disabled    err="failed stat on geth-tutorial\clef\masterseed.jso
n: CreateFile geth-tutorial\clef\masterseed.json: The system cannot find the file specified."
INFO [07-16|22:03:54.841] Starting signer                          chainid=11,155,111 keystore=geth-tutorial/keystore li
ght-kdf=false advanced=false
INFO [07-16|22:03:54.845] Audit logs configured                    file=audit.log
INFO [07-16|22:03:54.845] IPC endpoint opened                      url=\\.\pipe\clef.ipc

------- Signer info -------
* intapi_version : 7.0.1
* extapi_version : 6.1.0
* extapi_http : n/a
* extapi_ipc : \\.\pipe\clef.ipc

------- Available accounts -------
0. 0x9efb60Aa2823a8C499ce19D85C2d4e04826323Ef at keystore://C:\Users\fmpie\geth-tutorial\keystore\UTC--2025-07-16T16-32-
30.600342100Z--9efb60aa2823a8c499ce19d85c2d4e04826323ef
```

```
C:\Users\fmpie>curl http://localhost:8545

C:\Users\fmpie>geth attach http://127.0.0.1:8545
Welcome to the Geth JavaScript console!

instance: Geth/v1.16.1-stable-12b4131f/windows-amd64/go1.24.4
at block: 0 (Sun Oct 03 2021 18:54:41 GMT+0530 (IST))
 modules: eth:1.0 net:1.0 rpc:1.0 web3:1.0

To exit, press ctrl-d or type exit
>
```

```
C:\Users\fmpie>curl http://localhost:8545

C:\Users\fmpie>geth attach http://127.0.0.1:8545
Welcome to the Geth JavaScript console!

instance: Geth/v1.16.1-stable-12b4131f/windows-amd64/go1.24.4
at block: 0 (Sun Oct 03 2021 18:54:41 GMT+0530 (IST))
 modules: eth:1.0 net:1.0 rpc:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts;
Error: request timed out
        at web3.js:6020:9(39)
        at get (web3.js:5920:66(14))
        at <eval>:1:5(1)
```

```
-------- List Account request--------------
A request has been made to list all accounts.
You can select which accounts the caller can see
  [x] 0x9efb60Aa2823a8C499ce19D85C2d4e04826323Ef
    URL: keystore://C:\Users\fmpie\geth-tutorial\keystore\UTC--2025-07-16T16-32-30.600342100Z--9efb60aa2823a8c499ce19d85c2d4e04826323
ef
-----------------------------------------
Request context:
        NA -> ipc -> NA

Additional HTTP header data, provided by the external caller:
        User-Agent: ""
        Origin: ""
Approve? [y/N]:
> Y
-----------------------
WARN [07-16|22:15:53.976] Served account_list                    reqid=2 duration=38.5678126s err="request denied"
```

Step 4: Get Testnet Ether

Step 5: Interact with Geth:

geth attach http://127.0.0.1:8545

List of accounts:

eth.accounts;

```
C:\Users\fmpie>geth attach http://127.0.0.1:8545
Welcome to the Geth JavaScript console!

instance: Geth/v1.16.1-stable-12b4131f/windows-amd64/go1.24.4
at block: 0 (Sun Oct 03 2021 18:54:41 GMT+0530 (IST))
 modules: eth:1.0 net:1.0 rpc:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts;
```

```
-------- List Account request--------------
A request has been made to list all accounts.
You can select which accounts the caller can see
  [x] 0x9efb60Aa2823a8C499ce19D85C2d4e04826323Ef
    URL: keystore://C:\Users\fmpie\geth-tutorial\keystore\UTC--2025-07-16T16-32-30.600342100Z--9efb60aa2823a8c499ce19d85c2d4e04826323
ef
-------------------------------------------
Request context:
        NA -> ipc -> NA

Additional HTTP header data, provided by the external caller:
        User-Agent: ""
        Origin: ""
Approve? [y/N]:
> y
```

```
instance: Geth/v1.16.1-stable-12b4131f/windows-amd64/go1.24.4
at block: 0 (Sun Oct 03 2021 18:54:41 GMT+0530 (IST))
 modules: eth:1.0 net:1.0 rpc:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts;
["0x9efb60aa2823a8c499ce19d85c2d4e04826323ef"]
>
```
