Web 3.0 Cheatsheet

An informative document about Web 3.0 and its terminology.

By Albert Arakelyan

1. Wallets

There should be some wallet to hold your ETH which is MetaMask.
Metamask is mostly for ETH.

2. Networks

To hold your ETH in MetaMask you should choose your network. There are different networks which can be real and test. for example

2.1. Real Networks

- **Ethereum Mainnet**
- **Linea Mainnet**

2.2. Test Networks

Currently most popular ones are

- **Sepolia**
- **Helosky**

3. Faucets

Faucets are places where we can get free ETH on our test network Ethereum balance.
At this moment (26.05.2024) most pupolar faucets are provided by Alchemy and Google.
Links to Faucets:

- **Alchemy:** *https://www.alchemy.com/faucets/ethereum-sepolia*
- **Google Cloud:** *https://cloud.google.com/application/web3/faucet/ethereum/sepolia*

4. Transaction Address

After every transaction you should see transaction address something like this 0x0d49d9c7e0e59a6a59f1fc41d41d7543bcbbd4c10c2731d732019192911ee37b and have its own address in the networks website (Sepolia, Holesky) like this *https://sepolia.etherscan.io/tx/0x0d49d9c7e0e59a6a59f1fc41d41d7543bcbbd4c10c2731d732019192911ee37b*

5. Your Address

You also have a HEX address which appears at the top of you MetaMask page or extension looking like 0xf23abefd....392ef5.
You need this others to send you ETH or you for sending others, also you should give this to Faucets to gain some test ETH.
In addition there is an *address* type in Solidity langauge.

5.1. Private Key

Except for your address you have a Private key which you need for deploying your smart contracts on some platforms (e.g. Alchemy).
You can find it by clicking **three dots** at the top of your MetaMask page or extension then click *Account Details* and then click *Show Private* Key.
Never share with others.

5.2. Secret Recovery Phrase

This is a 12 words combination which you must keep in secret and never lose, otherwise you can never get access back to your account.
It gives you right after registration.

5.3. Login on another computer

You just need to click “Import account” instead of Creating one and then login through you 12 words phrase.

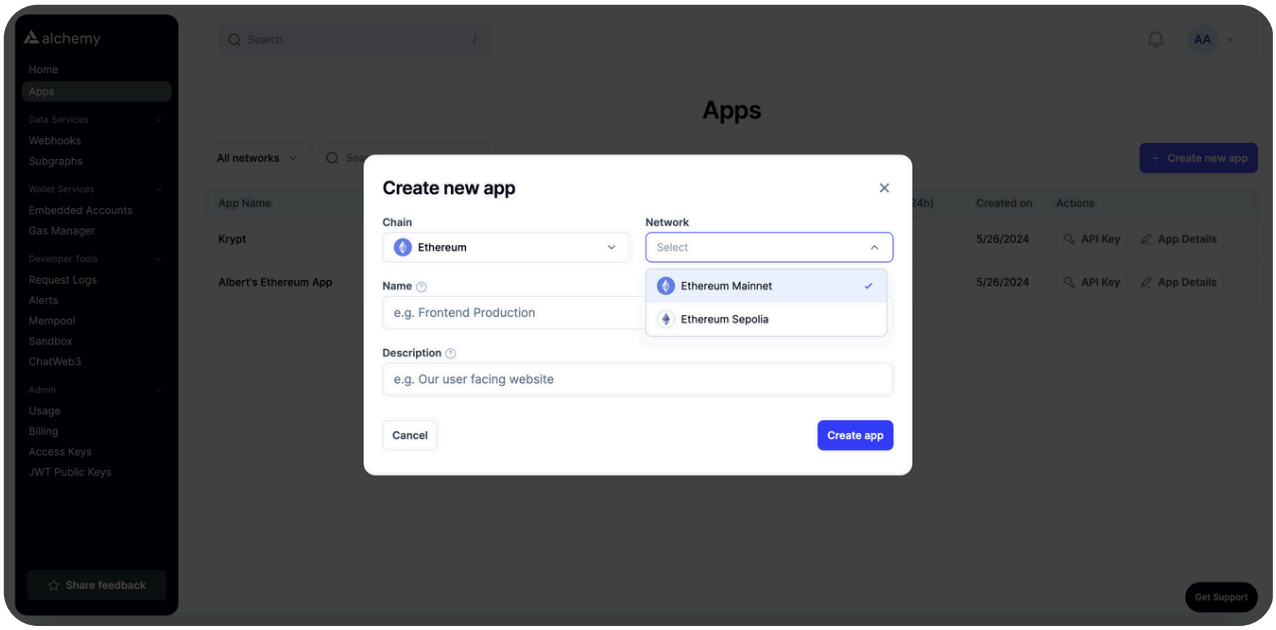
6. Deploying Smart Contracts

As I mentioned we need our private key for deploying Smart Contracts.
Smart Contracts are what based on which transactions are made, means you send some amount of ETH to another person for some reason. And that remains between you two and no bank or other centralized service doesn’t hold it.

6.1. Login on another computer

The most popular one is Alchemy.

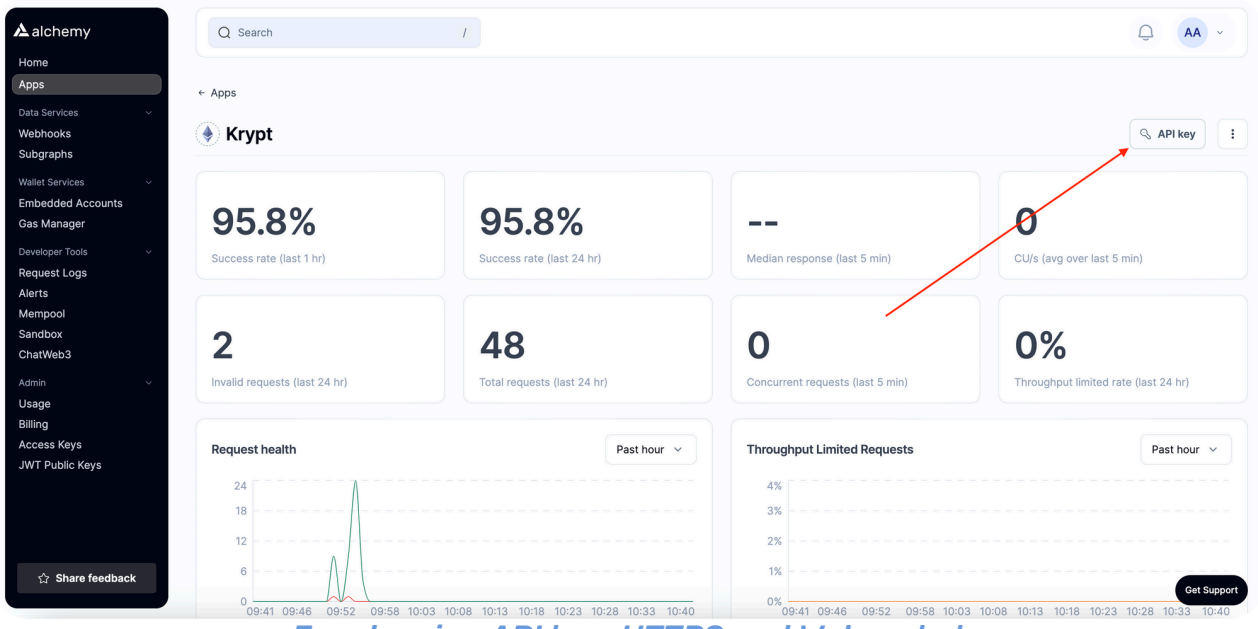
For deploying in Alchemy we need to sign up and create an application on Alchemy.



Alchemy app creating and network choosing process

Which asks us to choose network (which can be real and test) and then gives credentials that we need to use in our Smart Contract designed by hardhat.

Credentials look like we used to know, some API key also HTTPS and Websocket addresses for making requests (requests aren't held by us).



For showing API key, HTTPS and Websocket

After creation a **Smart Contract** and **deployment Script** we need to configure **hardhat** in *hardhat.config.js* where we put the HTTPS address of our app created in **Alchemy**, also our own **MetaMask Private Key**.
And that's it after running our deployment script with *node.js* it is deployed to **Alchemy** (in the application we created before).
Everything related to the codes above can be seen in this repo.

**Note: We need to have non zero ETH balance on the network that we are using that the contract can be deployed.*
After deployment you receive your **Contract Address** which looks like your address 0x32b4f2e4….34ab24, also available in the app that has been created in **Alchemy**.

6.2. Smart Contracts Language

Solidity is considered as Smart Contracts language for ETH and similar with Java, Javascript, C++ and Go.
hardhat is a package that allows us to work with Smart contracts with simple Javascript and deploy them eventually.