WallStreetNinja

Documentation

Table of Contents

1 Architecture

1.1 Backend

The backend has three functions.

1. Store persistent user data, like notifications and user settings
2. Call any 3th party API, which requires an API key
3. Fetch and cache any 3th party API, which is to slow to be called in real time from the frontend.

The 3th party APIs which are called from the backend are Coingecko, CovalentHQ and 1Inch. Also RPC calls are done via an arbitrary RPC provider.

1.1.1 Coingecko

Coingecko provides token pricing information. As the Coingecko API is quite slow in response and offers a very limited number of calls, the data is cached via the backend. (https://www.coingecko.com/en/api/documentation)

1.1.2 CovalentHQ

CovalentHQ provides the data about transactions and balances of each connected wallet. The good response time allows real time usage. However, it requires an API key which is the reason why the calls are relayed via the backend. (https://www.covalenthq.com/docs/api)

1.1.3 1Inch

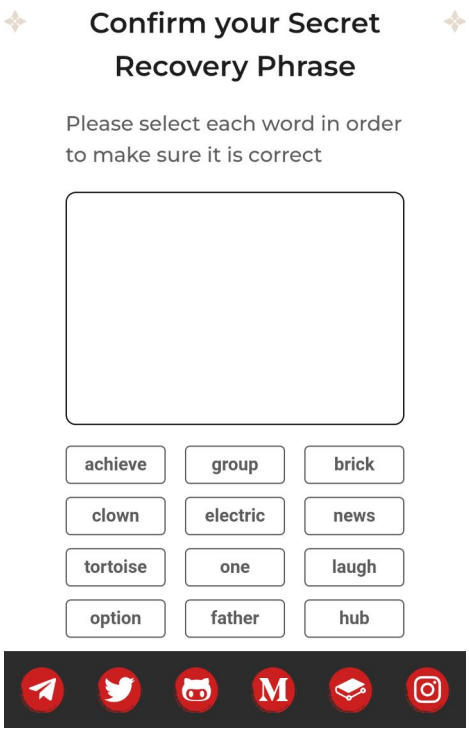
1Inch is an DEX aggregator for multiple token. (https://docs.1inch.io/docs/aggregation-protocol/api/swagger)

1.2 Frontend

The frontend provides an easy-to-use crypto wallet. Other Ethereum DApps can be connected via the custom implementation of WalletConnect. Beside that, it offers a token exchange via 1Inch. To interact with Bitcoin, the frontend communicates directly with the Blockcypher API (https://www.blockcypher.com/dev/bitcoin).

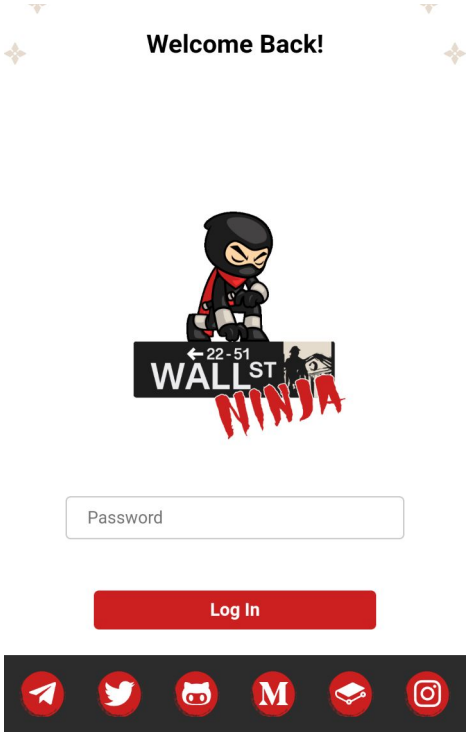
1.2.1 Account Creation

During the account creation the user is given a mnemonic. This mnemonic is used to derive any private key which is used to sign transactions. Also an encrypted file (keyfile) which contains the mnemonic is provided to the user too. The encryption is done with a user chosen password.



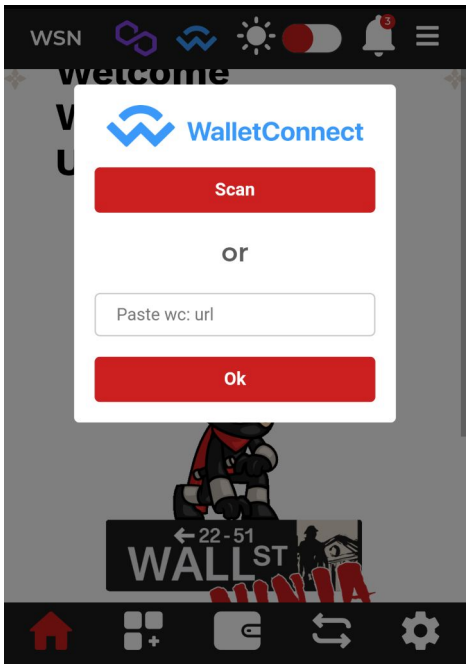
1.2.2 Login

In order to provide convenience, the encrypted mnemonic is stored in the browser. This allows an easy login via password. If the user has cleared the browser cache or is at an entirely new machine, the user can login by providing the mnemonic or the keyfile created via account creation.



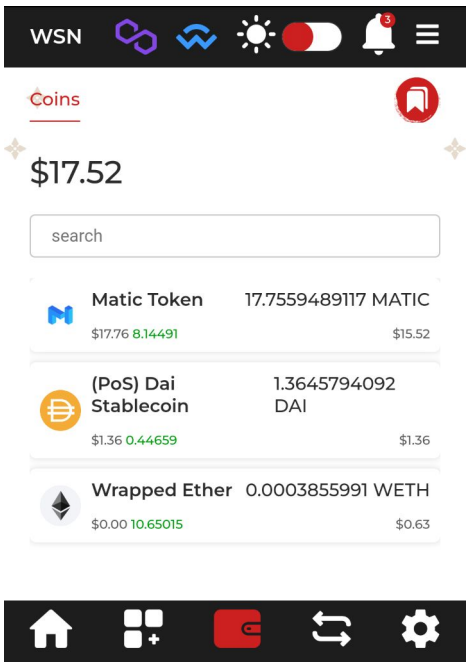
1.2.3 WalletConnect

WalletConnect is used to connect WallStreetNinja to any WalletConnect compatible DApp. On a desktop browser this is done via a link, which needs to be copied from the DApp. Smartphones offer the additional option to scan the WalletConnect QR code, provided by the DApp.



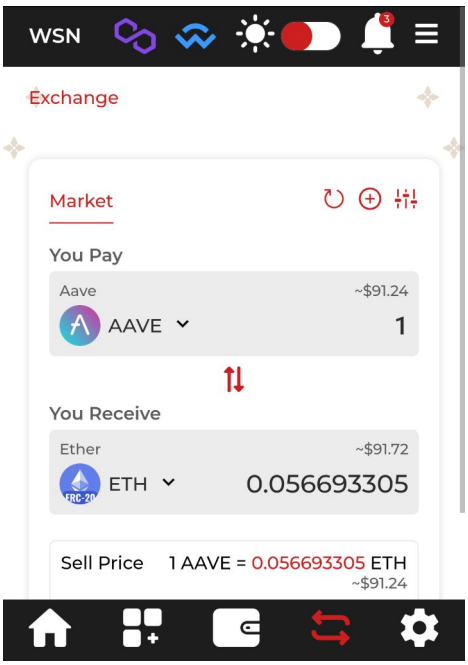
1.2.4 Wallet Overview

Provides a list of owned tokens, transactions for each token and allows to send tokens. To provide the wallet address to other, the address can be viewed, copied via a copy button or be displayed as QR code.



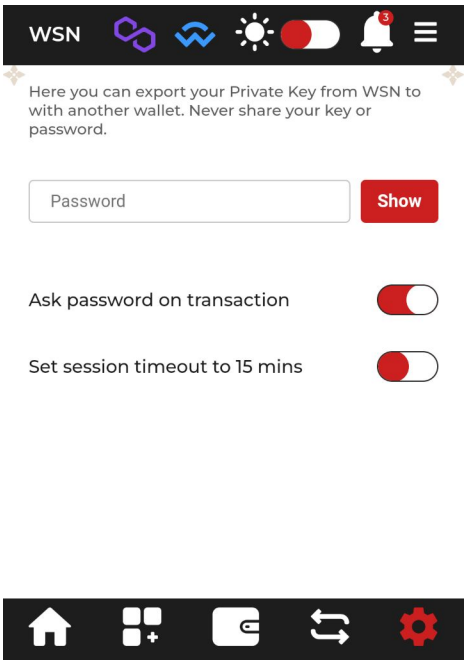
1.2.5 Exchange

WSN has a custom UI to access the 1Inch API, to exchange tokens. By default only a few tokens are in the lists. Tokens can be added but they need to be known to 1Inch to be displayed and exchangeable.



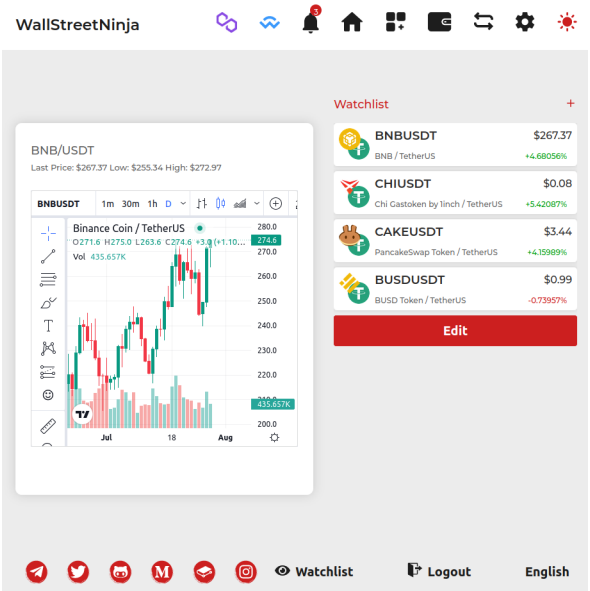
1.2.6 Settings

Offers a way to regain the mnemonic if it should be lost and the user is still logged in and knows the password. To avoid accidental transactions, it is possible to set a required password before every transaction. As well, a 15 minutes session timeout can be activated, for security reasons.



1.2.7 Watchlist

Shows charts of selected tokens via the Tradingview Widget.



2 Database Backup

In order to create a backup of the database, please go to: https://k3xcc-liaaa-aaaad-qckqq-cai.ic.fleek.co/maintenance

The password for downloading and uploading a database backup is: “c9f33df795a7c35cb6958f7288b89924e953548d3a16391770c924ef80b3b9b6”

3 Deployment

3.1 Backend

Please take a backup of the database before applying any changes to the backend.

Please clone the repository. This will download the repository.

```
$ git clone https://github.com/wsninja/wsn-backend.git
```

Enter the repository’s local root directory.

```
$ cd wsn-backend
```

Install the dependencies.

```
$ npm install
```

Build the docker image (replace <tag> with the version tag).

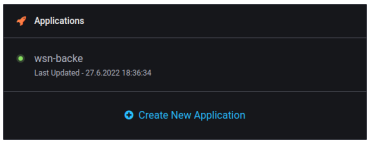
```
$ docker build -t wsndevop3/wsn-backend:<tag> .
```

Upload the docker image (replace <tag> with the version tag).

```
$ docker push wsndevop3/wsn-backend:<tag>
```

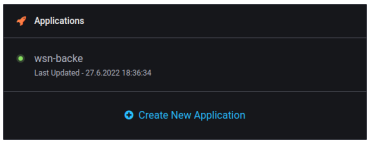
Please go to https://stackos.io and login to the cluster “Authority”.

If you want to update the docker image, please click on “wsn-backe”.

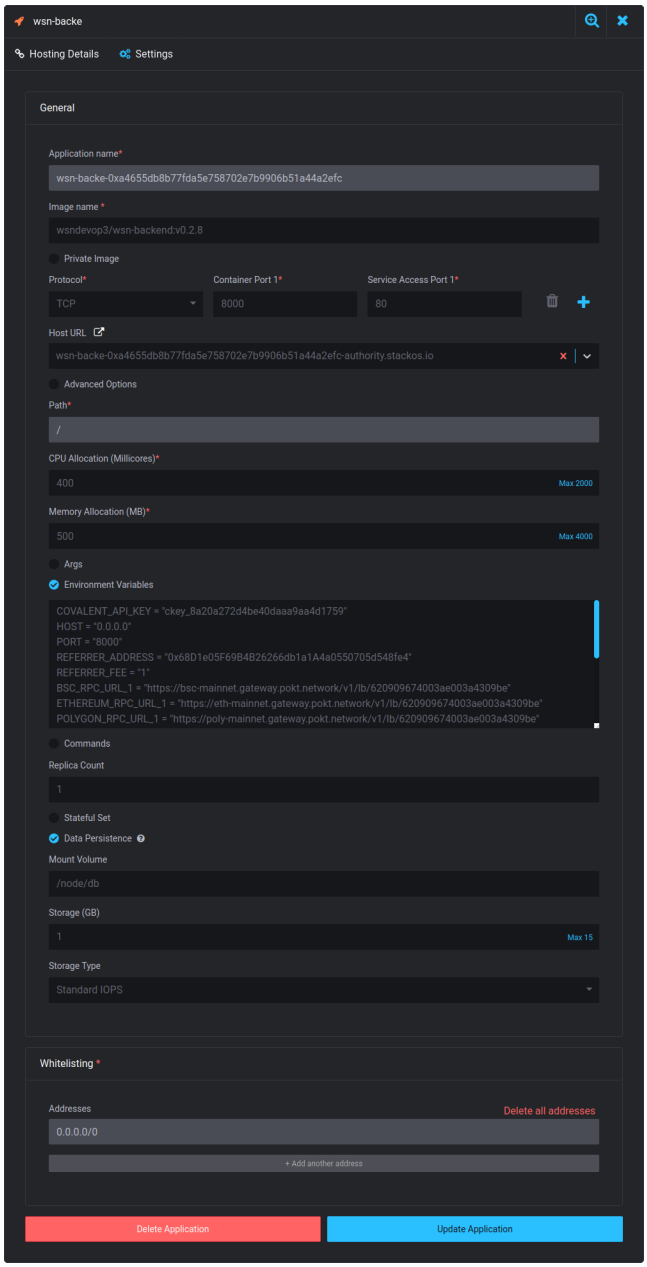


Then you can change the field “Image Name” to the new docker image and click “Update Application”. It might take some for the update to take affect.

If you want do deploy an entirely new instance, click on “Create New Application”. Please provide sufficient resources beforehand.



The following screenshot shows the current settings of the backend. They can be used as a template. The field “Environment Variables” is not completely visible on the screenshot. Therefor they are added below the screenshot as text.



The environment variables:

COVALENT_API_KEY = "ckey_8a20a272d4be40daaa9aa4d1759"

HOST = "0.0.0.0"

PORT = "8000"

REFERRER_ADDRESS = "0x68D1e05F69B4B26266db1a1A4a0550705d548fe4"

REFERRER_FEE = "1"

BSC_RPC_URL_1 = "https://bsc-mainnet.gateway.pokt.network/v1/lb/620909674003ae003a4309be"

ETHEREUM_RPC_URL_1 = "https://eth-mainnet.gateway.pokt.network/v1/lb/620909674003ae003a4309be"

POLYGON_RPC_URL_1 = "https://poly-mainnet.gateway.pokt.network/v1/lb/620909674003ae003a4309be"

BSC_RPC_URL_2 = "https://bsc-mainnet.gateway.pokt.network/v1/lb/620909674003ae003a4309be"

ETHEREUM_RPC_URL_2 = "https://eth-mainnet.gateway.pokt.network/v1/lb/620909674003ae003a4309be"

POLYGON_RPC_URL_2 = "https://poly-mainnet.gateway.pokt.network/v1/lb/620909674003ae003a4309be"

DB_PATH = "/node/db/wsn.sq3"

DEBANK_API_KEY = "a12a32c0ecf3714f90272d02009344b382b279b8"

SIGNING_PASSWORD = "RG4JnCj9r9TtErXrNarPSFmJZ8Pyj54CudxnnV8MRXSHNJN2VMdVjpZEtuSBqZEHsX"

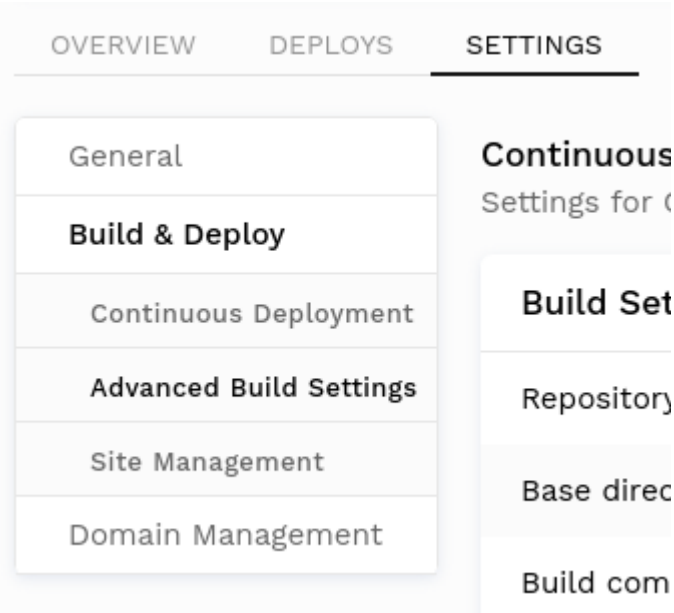
ADMIN_PASSWORD_HASH = "0bfdb50ab223410491ef9257d65029faaec852ac2ca7181ea662f8690b54875e"

ADMIN_PASSWORD_SALT = "64e3d7ad2fc2219efd1cf0fd738466f65e02d384da32b5ce026f6c14218be157"

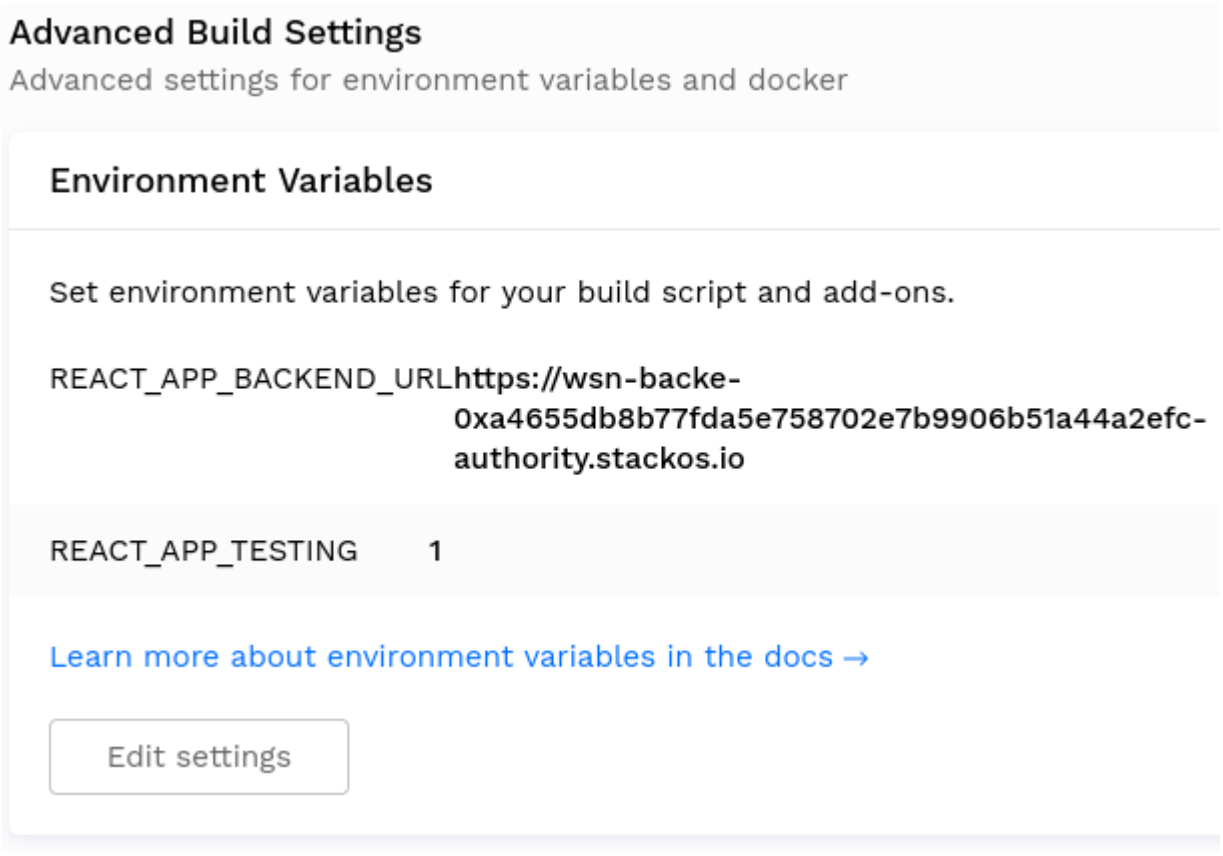
3.2 Frontend

The frontend is deployed and updated automatically by Fleek, when changes are pushed to the main branch of the wsn-frontend repository.

The environment variables of the frontend can be set in Fleek by going to “SETTINGS”→”Build & Deploy”→“Advanced Build Settings”



The environment variables can be set by clicking on “Edit settings”.



The current environment variables as text for easier copy & paste.

REACT_APP_BACKEND_URL = "https://wsn-backe-0xa4655db8b77fda5e758702e7b9906b51a44a2efc-authority.stackos.io"

REACT_APP_TESTING = 1