



My TON wallet submission for #137 Contest: Free
TON wallet as a Chrome extension

(March 10 - May 12, 2021, 22:59 UTC)

Our team with unshadowed pleasure wants to present our web-extension on judges review. We called it "My TON wallet" that describes that all our aspirations were directed to the best user experience. We have used only the best practises in security area for the web extension developing process, that can be found on [the official chrome website for developers](#)

As we all know the most important thing that can hold back all cryptocurrencies innovations are - user experience and security. People don't want to trust any external tools or wallet if they can't have human support for them with incidents. By this reason all cryptocurrencies software must have the highest security level. The risks can appear from external libraries. For this reason we have reduced the use of any external libraries that could have any vulnerability source code in the future. In the near future we will remove all dependencies completely. For now we use only 11 (exclude @tonclient).

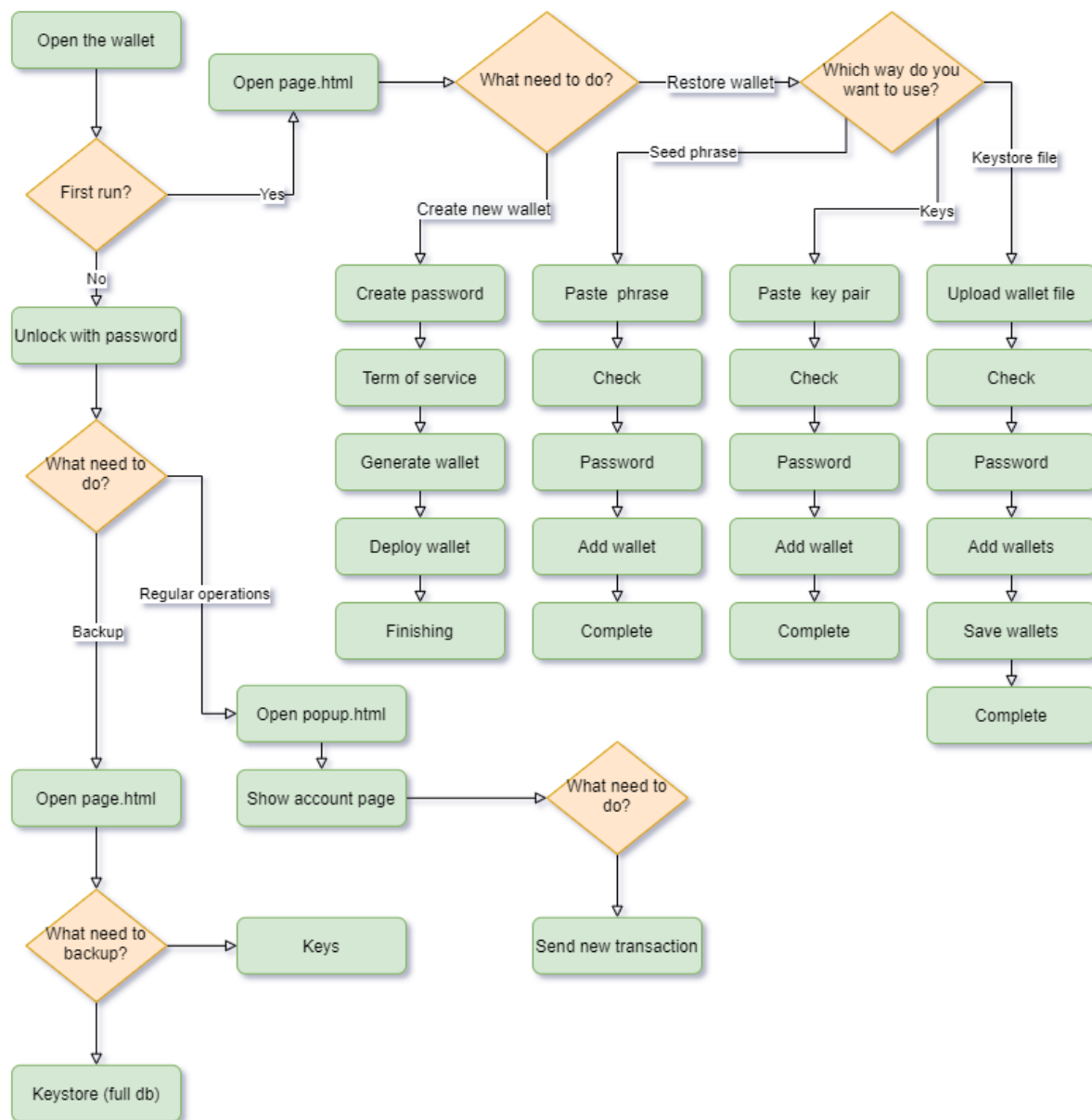
We use practise when the wallet password is stored as a single variable in browser memory. This in-memory approach allows minimizing the chance to grab access for restoring passwords for the wallet. Later we want to try to apply a technique for time encryption password in memory.

The password checking for the unlock operation is provided by the comparison of a random hex key with length 256 that is stored in indexedDb and its encrypted hash that was obtained by AES-GCM with password inclusion. In this case to receive the master password needs to decrypt encrypted data that must be equal to the hex string by 256 length. Also, IndexedDb stores only common information, like transaction history, etc. key pair is encrypted with AES-GCM

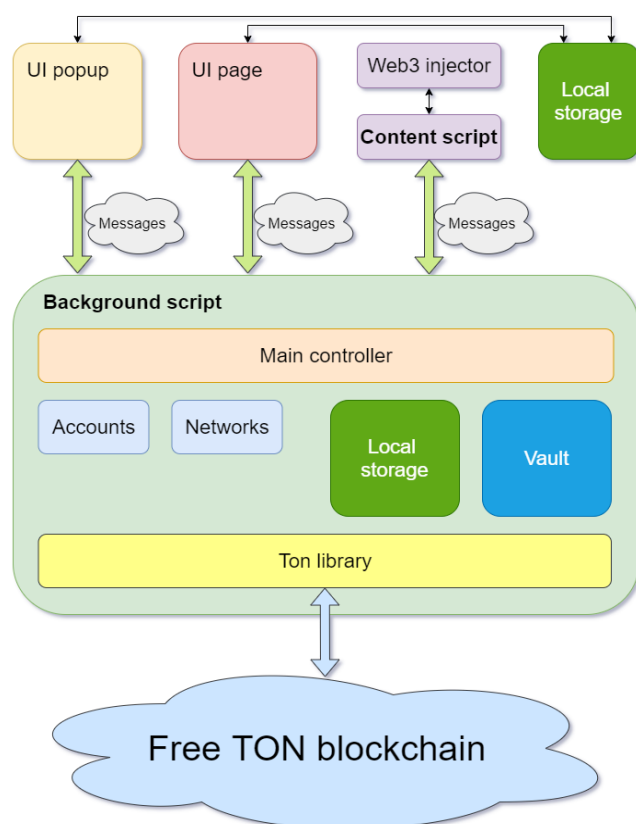
Let's review the second point that holds mass propagation of decentralized software - the user experience.

Here we can see user flows that are possible to the web extension.

Everything is extremely simple and concise. We will add additional flows as we evolve.



Here is the common architecture:



The web extension contains two main UI points. Popup for common operation and the page that allow providing multisteps operations, like backup/restore, etc. Later we plan to integrate to the page ABI interaction by html forms and smart contract composing in a rich code editor. All communications happen by standard messages mechanism with strong

origin source checking. The content script can't reach to the popup or to the page bypass the background script. This script manages all operations between the free TON blockchain and web extension. The background script has the one main controller and all operations are split on several sub controllers that run specific functions that relate with certain areas (account methods, network methods, token methods, etc.) The local storage has a custom wrapper and doesn't store any secure information.

The storages between the popup/page and background script are isolated during interaction, but have the common browser local storage. This approach allows synchronizing preference for theme between the popup and the page, for example. The vault is the indexedDb storage that contains several dbs that store accounts/networks, etc. Each account record contains encrypted keys that can be decrypted by in-memory password when they are needed for signing or another operation, after this operation they wipe from the browser memory.

Common review of technical stack

For building processes we use <https://rollupjs.org/guide/en/>, because it allows saving time on compilation drastically.

For javascript framework we use <https://svelte.dev/> that was developed in the main idea working on devices with low memory, such as salepoint terminals, etc.

For UI style we use a low size library <https://alexnb.github.io/svelte-chota>, that we plan to integrate into web extension source code to reach zero-dependency aim.

To work easily with indexedDb that allow storing data up to 80% of disk space we uses a low size wrapper <https://www.npmjs.com/package/idb>

For interaction with Free TON blockchain we use the official ton-client-js library.

For multisig wallet smart contract uses the most recommended and most tested variant

<https://github.com/tonlabs/ton-labs-contracts/tree/master/solidity/safemultisig>

<https://github.com/mozilla/webextension-polyfill> allows supporting cross browser buildings

We use <https://gulpjs.com/> for building distributives for browsers

Checklist by the criterias

Hard criteria

- **Generic**

- ***English language of the interface;***

Support english/russian languages (initially depends on PC locales settings)

- ***Support of Google Chrome;***

We use Chrome as the main browser for developing, but also web extension works in IE, Firefox, Opera, Brave.

- ***Absence of analytical trackers (Google Analytics, Yandex Metrika, etc.);***

We don't track any user activities, we don't collect any user actions even anonymized data, like Metamask does.

- ***Support of mainnet and testnet(s);***

These both + local network for developers, that is based on [tonlabs/local-node](https://github.com/tonlabs/local-node) docker container. Port 7777

- ***On-chain activity history (transactions, messages, contract interactions, etc.);***

Each account has transaction history, that contains information about deploy/transfer actions (another type can be added during evolution process)

- ***Any calls that require the user's keys must ask for the password input to decrypt them from the local storage.***

To reduce any annoying experiences we have the master password that stores by in-memory approach and uses it for the decryption process. All the time before and after signing operation all keys are encrypted. We have added the auto sign out feature, that allows adjusting a time when web extension will wipe the master password from the browser memory and by this moment will lock the wallet.

- **Wallet features**

- ***Native support of any open-sourced non-custodial Free TON wallets, e.g.:***
 - ***Original TON wallets (Wallet v. 3);***
 - ***TON Labs' wallets (SafeMultisig, SetCodeMultisig);***

We have selected the most tested and recommended wallet SafeMultisig

- ***Random seed phrase generation;***

A user can see this phrase on the initial step of adding a new account in the wallet, can download in the file or print. We avoid any additional checking like a game in memorizing , etc. in consensus that it is annoying for the user. But instead in this case the user must confirm that the seed phrase is saved by checkbox checking.

- ***12 or 24 words wallet initialization (based on wallet contract);***

We decided that 12 words will be well enough for SafeMultisig smart contract

- ***Wallet seed phrase backup and restoration;***

A user has several options how to backup and restore access to the wallet. We have the seed phrase backup process (where the user must confirm that the phrase is saved) on the initial step and keys/keystore(full or limited) backups by demand. For the restoration process the user can use any of these backup types. But we recommend using the keystore backup type as it is more comfortable. This backup type is encrypted by password and has a hint that allows saving some memorable phrase to remind the password after a long time when the wallet is unused.

- ***Public and private keys generation, backup, and restoration;***

This step is the second after seed phrase generation. All keys are generated by seed phrase.

- ***Encrypted local key storage;***

We use AES-GCM encrypted keys that store into indexDb, not in the browser local storage. All additional information that is useful for better UI, caches, transaction histories, balances stores without any encryption to improve performance. This information can't give any access to user assets. But later we plan to provide "paranoid mode" in settings.

- ***Password protection;***

Web extension has the master password that gives access to the wallet information. This password is used to access for any wallet functionally and for keys decrypting/encrypting.

- ***Support of sending a memo with messages (or encoded payload).***

A user can send a transaction with some amount and message to another address that supports SafeMultisig smart contract

Soft criteria

- ***Multilanguage support;***

Supporting English/Russian languages from the box, any other languages can be added in quick mode. Just need to translate the one file. The browser independent i18n implementation which is not hard bound with PC locale.

- ***The extension is published in the Chrome store;***

The link can be found on forum or by searching in [chrome web store](#) (we can't be sure, on the moment of evaluation, about that the web extension will be available, because there is inject.js content script (dummy layer for further web3 like library) that inserts itself on each page). Archives for manual installation are available from [this page](#).

- ***Support of additional browsers (Firefox, Brave, Edge, Safari, Opera);***

The common build process.

- ***Browser notifications on events;***

Notifications happen when transaction adds to the local db

- ***Detailed and easily understandable charts explaining the architecture and business processes;***

In judges taste. The all additional documentation is available on the official website <https://mytonwallet.com>

- ***Brevity;***

To be involved in the development process you don't need to learn any special libraries, just read a code and documentation.

- ***Mostly everyday English to facilitate understanding;***

We have tried to use simple English, because our team includes people from all world and not all read English classic literature.

- ***Readiness to participate in the implementation of the solution in the next stage;***

We have a big list of requested features and desires to implement them all. Check below.

- ***Verifiable extension security along with the process to verify the equality of published version with source code.***

We use

<https://developer.chrome.com/docs/extensions/mv3/security/> guide.

Performance

Among aims that were set up for developers by themselves was the performance. We had the strong understanding that good user experience will demand speed and quick response for the product. That's why we have selected a svelte.js that compiles into vanilla javascript code. This decision allowed our team to create a web extension with a size around 6.7 Mb, where the tonclient.wasm library takes up 3.95Mb. In comparison with Metamask, where there is no big external library, the size is 28.4Mb and our web extension is smaller at 76%. This fact will allow us to implement a small sized mobile application or even create a version for the salepoint terminal some.

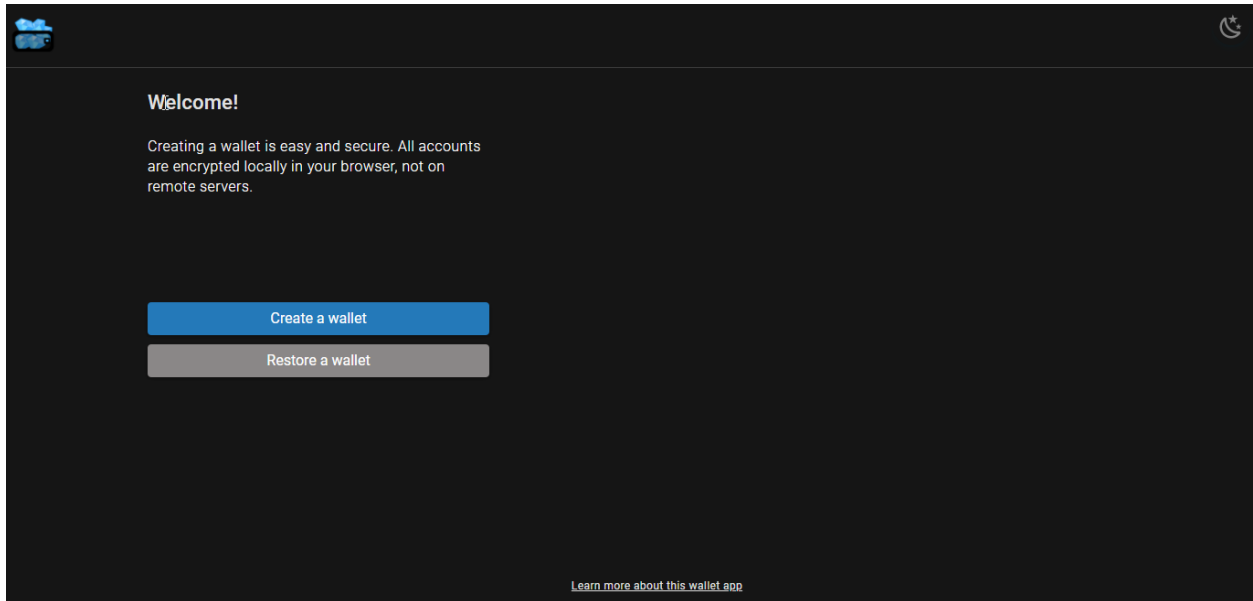
User flows

For every page can be applied a light or dark theme. The dark theme is set by default.

First run

After web-extension installation , the user can select what a process must be invoked - create a wallet or restore a wallet.

Here we show the creating process.



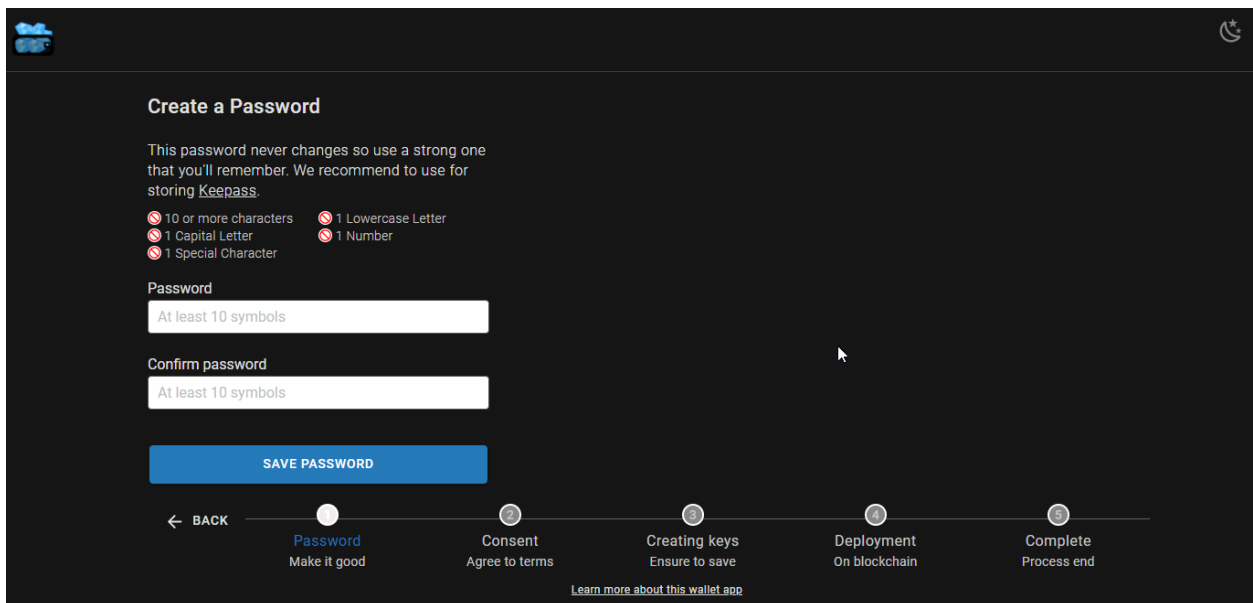
Welcome!

Creating a wallet is easy and secure. All accounts are encrypted locally in your browser, not on remote servers.

Create a wallet

Restore a wallet

[Learn more about this wallet app](#)



Create a Password

This password never changes so use a strong one that you'll remember. We recommend to use for storing [Keepass](#).

- 10 or more characters
- 1 Capital Letter
- 1 Special Character
- 1 Lowercase Letter
- 1 Number

Password

At least 10 symbols

Confirm password

At least 10 symbols

SAVE PASSWORD

← BACK

1 Password
Make it good



2 Consent
Agree to terms

3 Creating keys
Ensure to save

4 Deployment
On blockchain

5 Complete
Process end

[Learn more about this wallet app](#)



Create a Password

This password never changes so use a strong one that you'll remember. We recommend to use for storing [Keepass](#).

✔ 10 or more characters

✔ 1 Lowercase Letter

✔ 1 Capital Letter

✔ 1 Number

✔ 1 Special Character

Password

Confirm password

SAVE PASSWORD

← BACK

1

Password

Make it good

2

Consent

Agree to terms

3

Creating keys

Ensure to save

4

Deployment



On blockchain

5

Complete

Process end

[Learn more about this wallet app](#)



Remember

Storing your password and backing up your wallet is **YOUR RESPONSIBILITY**. This is important to keeping your cryptocurrency safe.

I understand

go back

← BACK

✔

Password

Make it good

2

Consent

Agree to terms

3

Creating keys

Ensure to save

4

Deployment



On blockchain

5


Complete

Process end

[Learn more about this wallet app](#)



Your seed phrase



ankle	bus	gospel	evoke	pulp	goddess
midnight	spray	amount	siren	fat	slight

Print

Save as file

☐ I have saved in strong place

Deploy wallet on blockchain

[← BACK](#)

✓

Password

Make it good

✓

Consent

Agree to terms

●

Creating keys

Ensure to save

④

Deployment



On blockchain

⑤


Complete

Process end

[Learn more about this wallet app](#)



Your seed phrase



ankle	bus	gospel	evoke	pulp	goddess
midnight	spray	amount	siren	fat	slight

Print

Save as file

☒ I have saved in strong place

Deploy wallet on blockchain

[← BACK](#)

✓

Password

Make it good

✓

Consent

Agree to terms

●

Creating keys

Ensure to save

④

Deployment

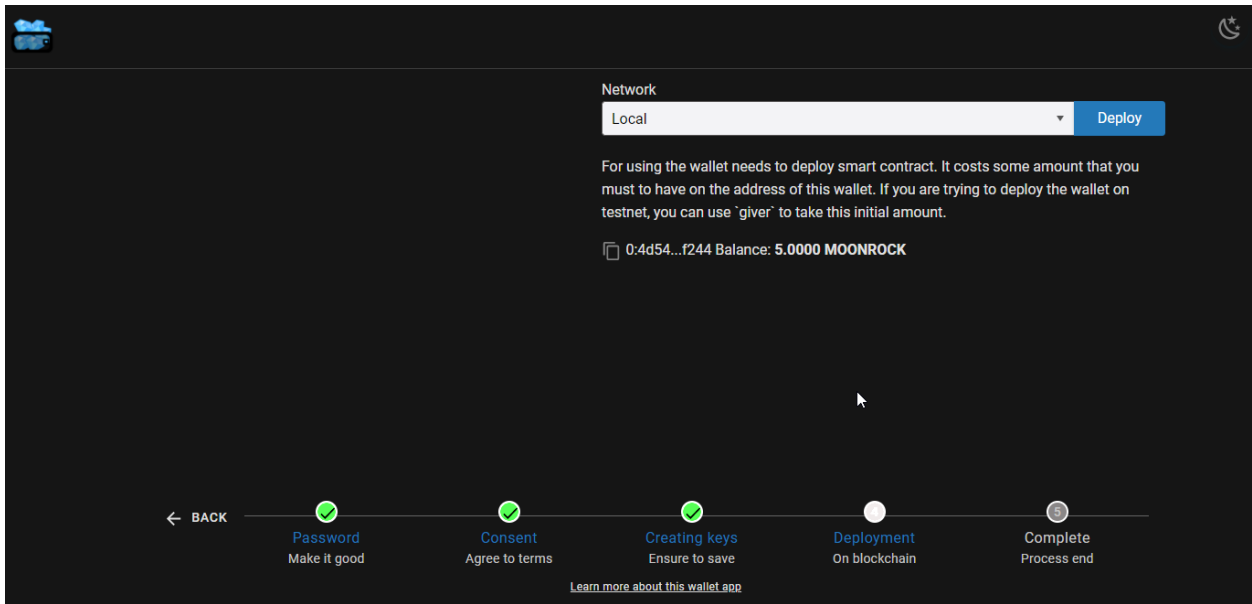
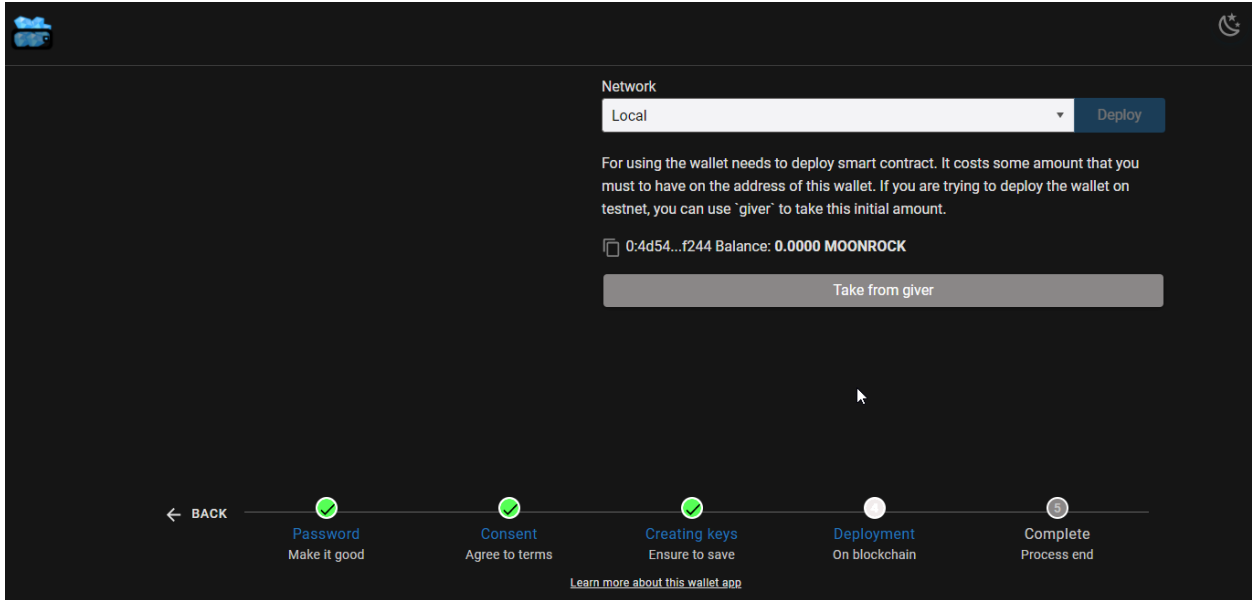
On blockchain

⑤

Complete

Process end

[Learn more about this wallet app](#)





Deployment process

← BACK



Password
Make it good



Consent
Agree to terms



Creating keys
Ensure to save

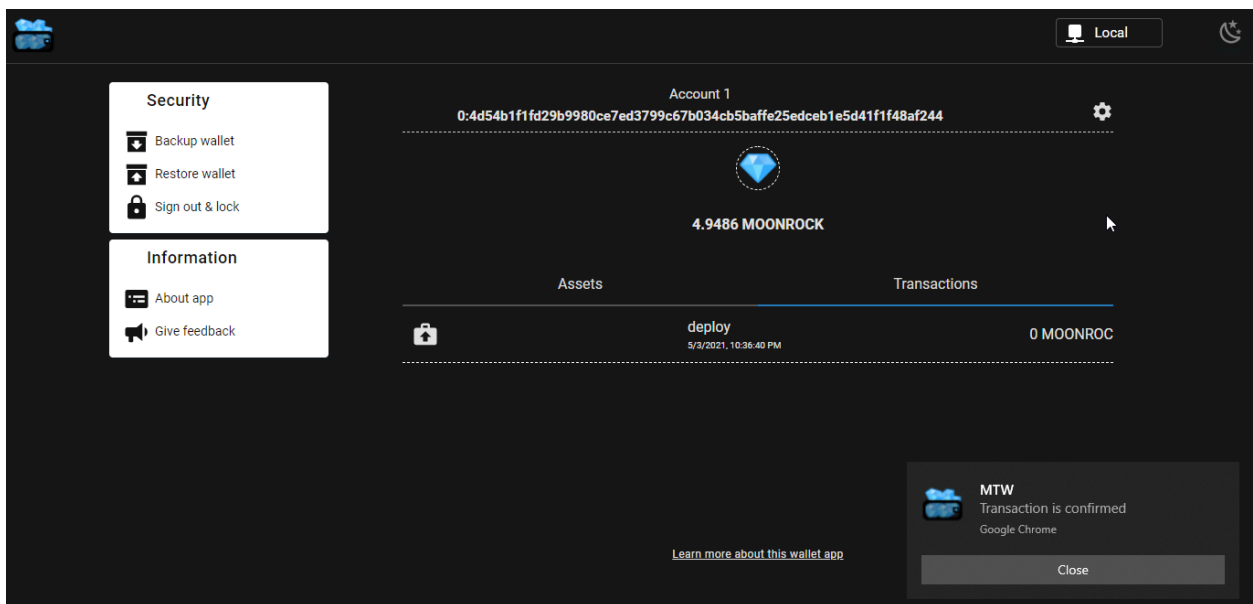
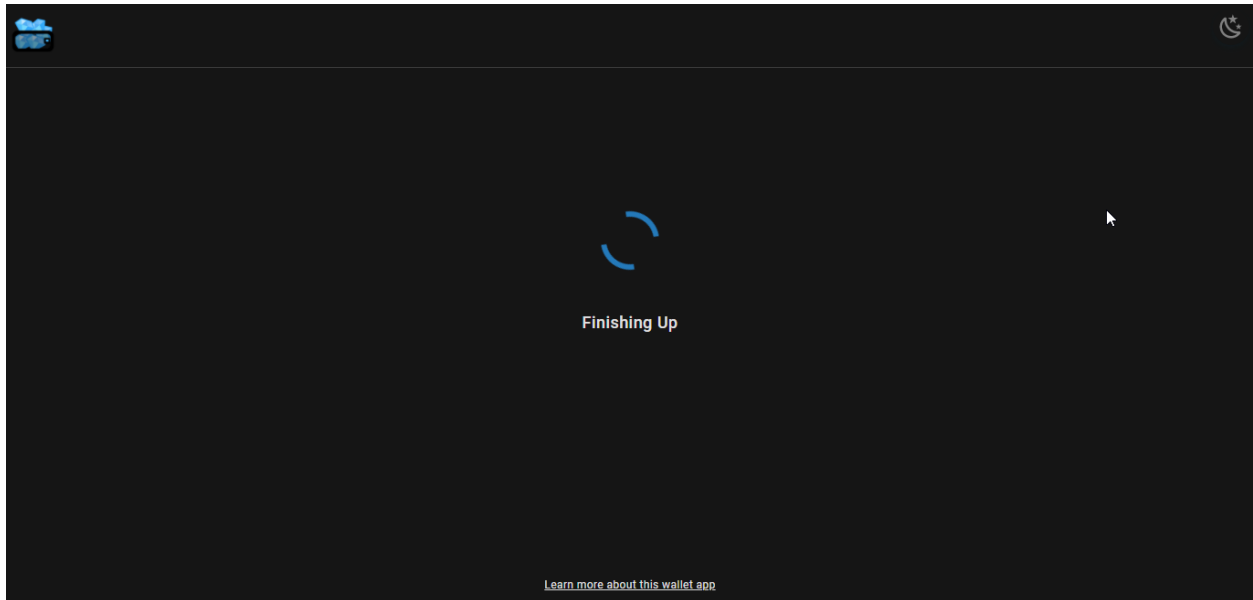


Deployment
On blockchain



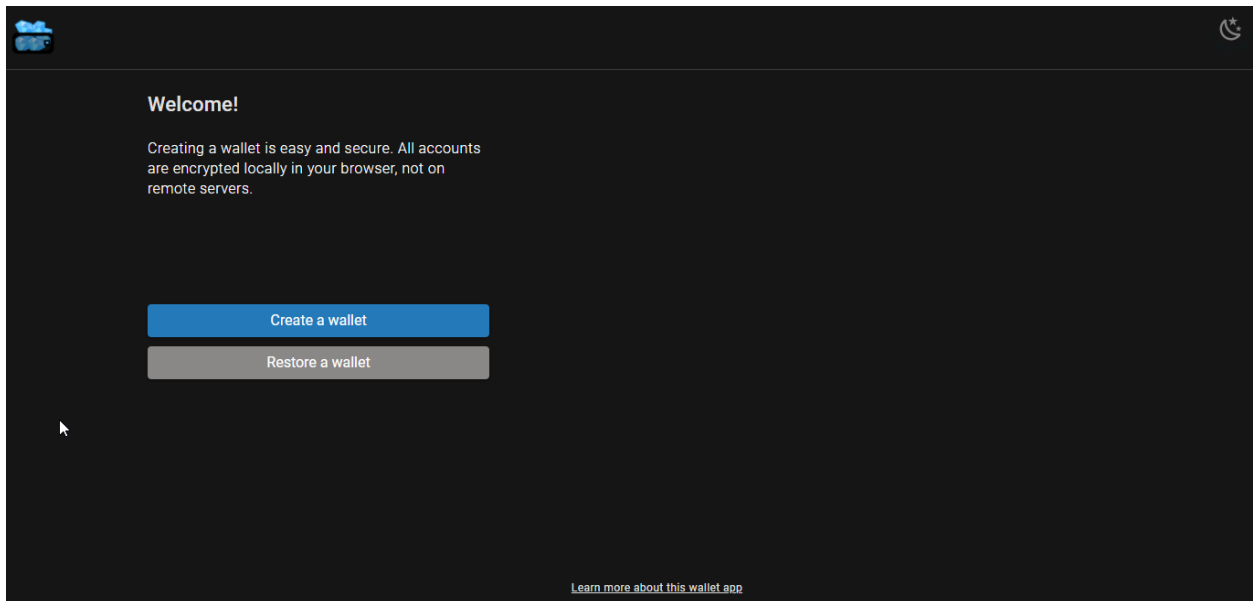
Complete
Process end

[Learn more about this wallet app](#)

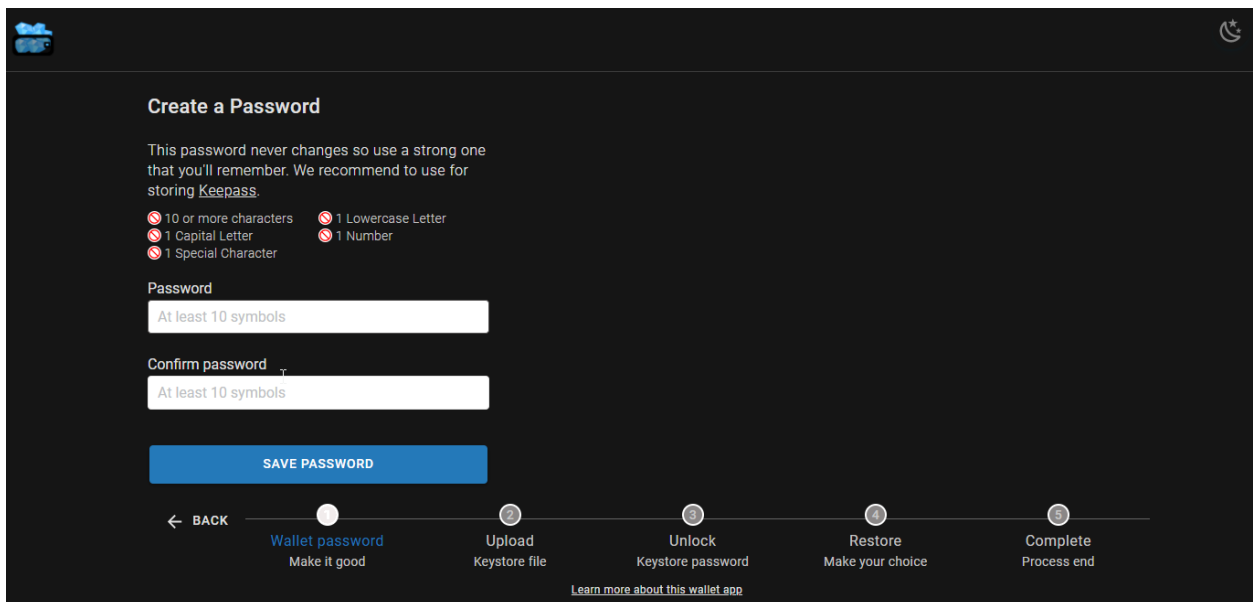


The user also can select “restore a wallet”. This process to allow restoring a wallet from the keystore file, that can store full exported data (address, keys, contacts list, contracts list, transactions, etc.) or limited (address, keys) for all accounts. The user can’t restore from seed phrase or keys pair on the initial step, because these are legacy methods that must be avoided.



As soon as the user will have at least one account in the wallet, there will be available to restore from seed phrase, keys pair.



The image shows the 'Welcome!' screen of a wallet application. At the top left is a logo with three blue circles and a white 'C'. At the top right is a moon icon. The main text reads: 'Welcome! Creating a wallet is easy and secure. All accounts are encrypted locally in your browser, not on remote servers.' Below this are two buttons: 'Create a wallet' (blue) and 'Restore a wallet' (grey). At the bottom right is a link: 'Learn more about this wallet app'.



The image shows the 'Create a Password' screen of the wallet application. At the top left is a logo with three blue circles and a white 'C'. At the top right is a moon icon. The main text reads: 'Create a Password This password never changes so use a strong one that you'll remember. We recommend to use for storing Keepass.' Below this are four requirements, each with a red circle and a white 'X': '10 or more characters', '1 Lowercase Letter', '1 Capital Letter', and '1 Number'. Below these are two input fields: 'Password' and 'Confirm password', both with a placeholder 'At least 10 symbols'. Below the input fields is a blue button labeled 'SAVE PASSWORD'. At the bottom is a progress bar with five steps: 'Wallet password' (Make it good), 'Upload' (Keystore file), 'Unlock' (Keystore password), 'Restore' (Make your choice), and 'Complete' (Process end). The first step is highlighted with a blue circle. At the bottom right is a link: 'Learn more about this wallet app'.



Create a Password

This password never changes so use a strong one that you'll remember. We recommend to use for storing [Keepass](#).

✔ 10 or more characters

✔ 1 Lowercase Letter

✔ 1 Capital Letter

✔ 1 Number

✔ 1 Special Character

Password

Confirm password

SAVE PASSWORD

← BACK

1

Wallet password

Make it good

2

Upload

Keystore file

3

Unlock

Keystore password

4

Restore



Make your choice

5

Complete

Process end

[Learn more about this wallet app](#)



Restore accounts

To restore your accounts, please upload the keystore file created during your backup.

[Click here to choose a file](#) or drag and drop your file below.

Drop file here

Confirm keystore

← BACK

✔ 1

Wallet password

Make it good

2

Upload

Keystore file

3

Unlock

Keystore password

4

Restore



Make your choice

5

Complete

Process end

[Learn more about this wallet app](#)



Restore accounts

To restore your accounts, please upload the keystore file created during your backup.

[Click here to choose a file](#) or drag and drop your file below.

My_Ton_Wallet_21.04.2021, 13_34_54.keystore

Confirm keystore

← BACK

1

Wallet password

Make it good

2

Upload

Keystore file

3

Unlock

Keystore password

4

Restore



Make your choice

5

Complete

Process end

[Learn more about this wallet app](#)



Keystore file confirmed

[Enter your keystore file password.](#)

last modified date:
Wed Apr 21 2021 13:34:55 GMT+0300 (Moscow Standard Time)

Password hint
my hint

Keystore password

CONFIRM PASSWORD

← BACK

1

Wallet password

Make it good

2

Upload

Keystore file

3

Unlock

Keystore password

4

Restore



Make your choice

5

Complete

Process end

[Learn more about this wallet app](#)



Keystore file confirmed

Enter your keystore file password.

last modified date:
Wed Apr 21 2021 13:34:55 GMT+0300 (Moscow Standard Time)

Password hint
my hint

Keystore password

.....

CONFIRM PASSWORD

← BACK

✓

Wallet password

Make it good

○

Upload

Keystore file

○

Unlock

Keystore password

○

Restore



Make your choice

○

Complete

Process end

[Learn more about this wallet app](#)



Password confirmed

Almost there! Now let's select which accounts you'd like to restore

	Nickname	Address
<input checked="" type="checkbox"/>	All wallets	
<input checked="" type="checkbox"/>	Main Account	0:f888562b1bb89105bcd39d47959483140aa54...

Restore accounts

Cancel

← BACK

✓

Wallet password

Make it good

✓

Upload

Keystore file

○

Unlock

Keystore password

○

Restore

Make your choice

○

Complete

Process end

[Learn more about this wallet app](#)



Accounts restored

You've added the following wallets successfully!
You may now perform transactions using these addresses.



Nickname	Address
Main Account	0:f888562b1bb89105bcd39d47959483140aa54...

Finish



Wallet password
Make it good



Upload
Keystore file



Unlock
Keystore password

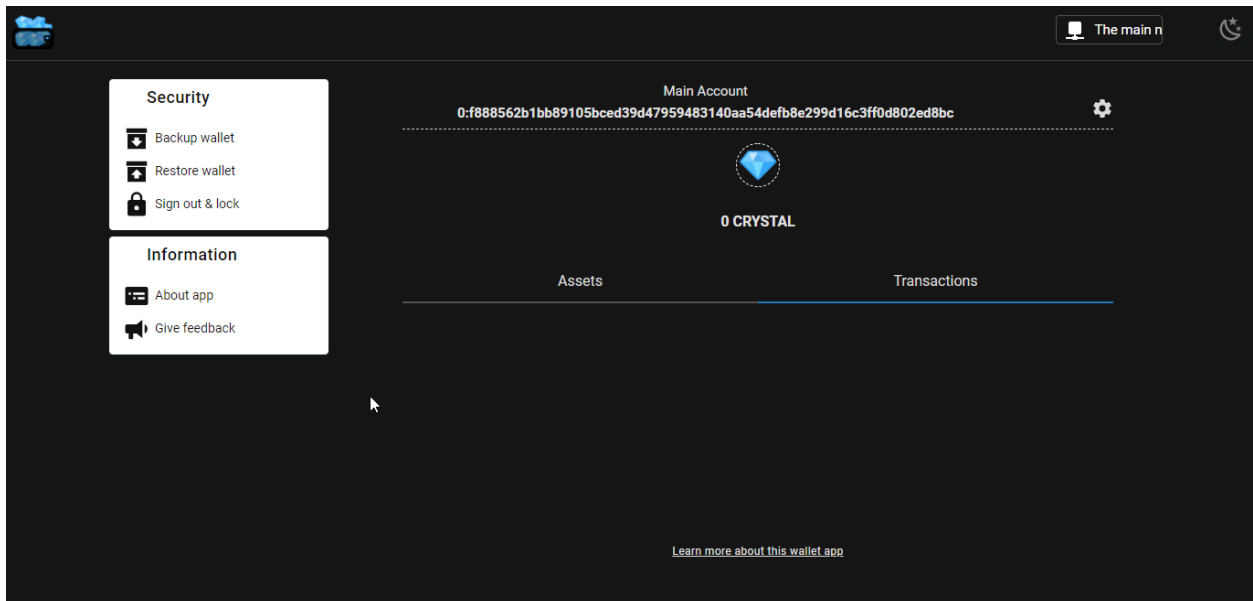
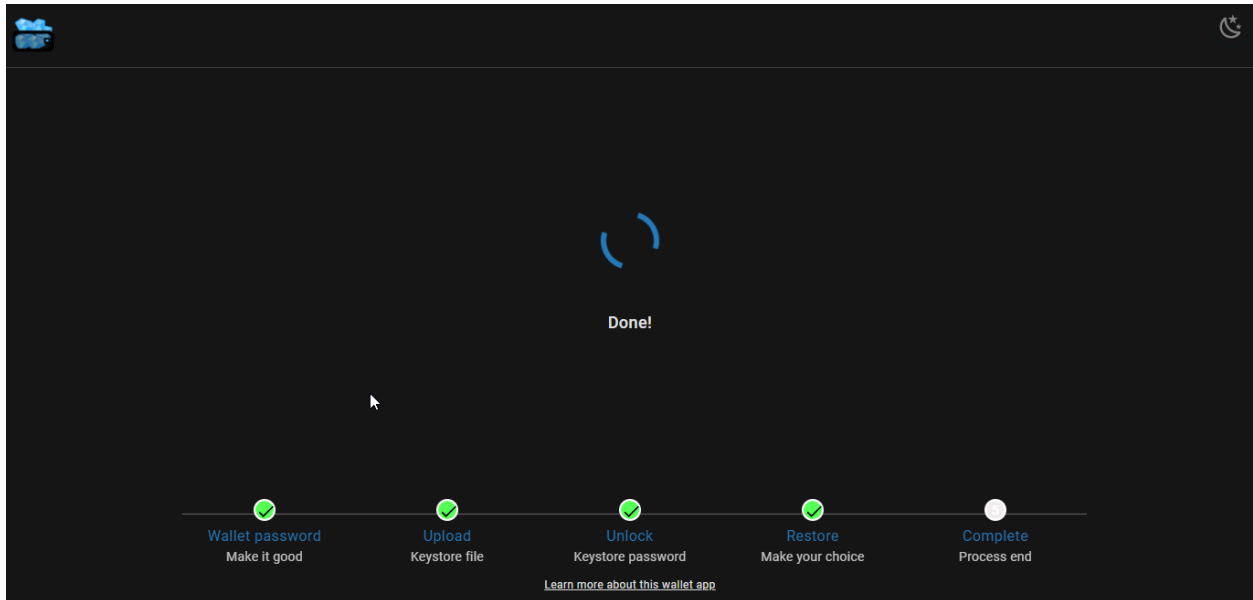


Restore
Make your choice



Complete
Process end

[Learn more about this wallet app](#)





Backup

The user can do the backup procedure for own wallet by several ways:

1. Keystore
2. Keys

Below shows the keystore creating process:



Backup wallet



This process will allow you to either create an encrypted keystore or decrypt and view your all of your secret keys.

Keeping backups safe is your responsibility.

Create backup file

View account keys

Back to home



Keystore password

For maximum security needs to store in a password manager such as [Keepass](#).

10 or more characters

1 Lowercase Letter

1 Capital Letter

1 Number

1 Special Character

Password

At least 10 symbols

Confirm password

At least 10 symbols

CREATE KEYSTORE

Team is not responsible for lost or stolen passwords

Password hint (Optional)

Create a password hint

☐ Full export

If you want to save all contracts added on accounts, all transactions related with accounts, then tick the checkbox.

[Help](#)

← BACK

1

Set password

Make it good

2



Generate file

Just a second

3

Download

Keep it safe



Keystore password

For maximun security needs to store in a password manager such as [Keepass](#).

☒ 10 or more characters

☒ 1 Lowercase Letter

☒ 1 Capital Letter

☒ 1 Number

☒ 1 Special Character

Password

Confirm password

CREATE KEYSTORE

Team is not responsible for lost or stolen passwords

Password hint (Optional)

my hint

☒ Full export

If you want to save all contracts added on accounts, all transactions related with accounts, then tick the checkbox.

[Help](#)

← BACK

1

Set password

Make it good

2

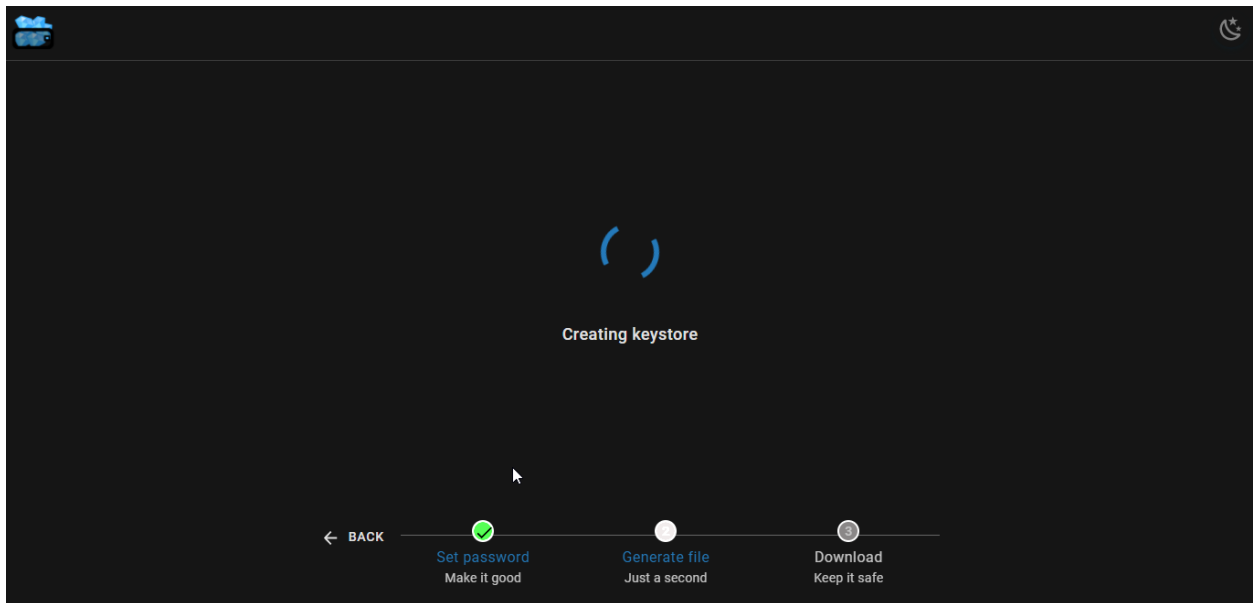
Generate file

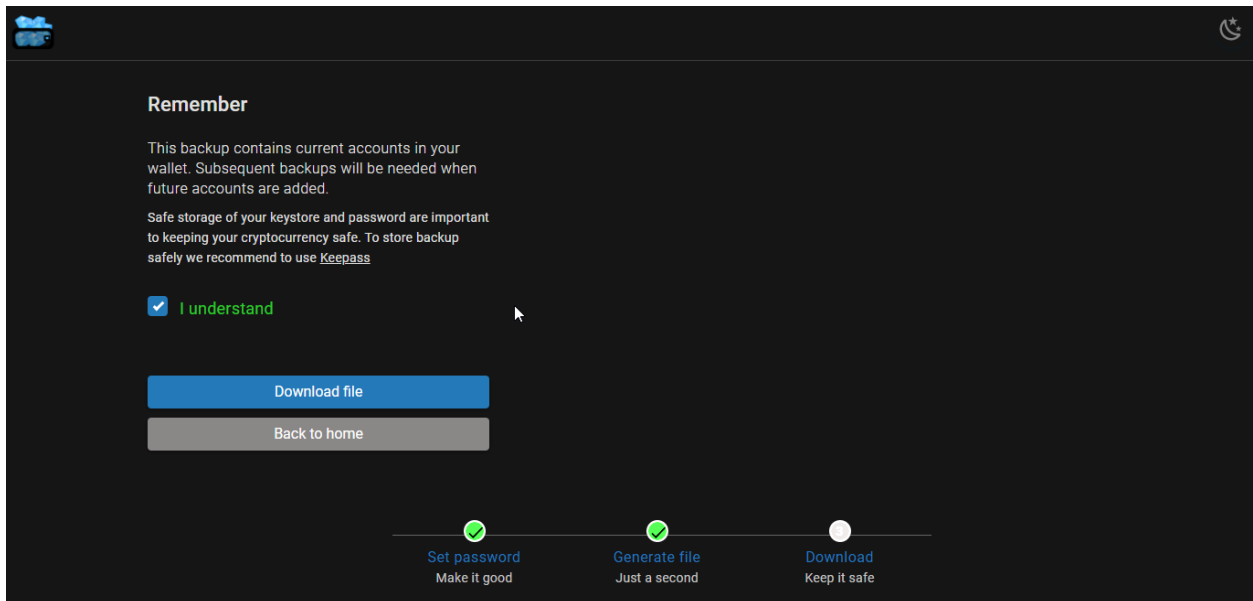
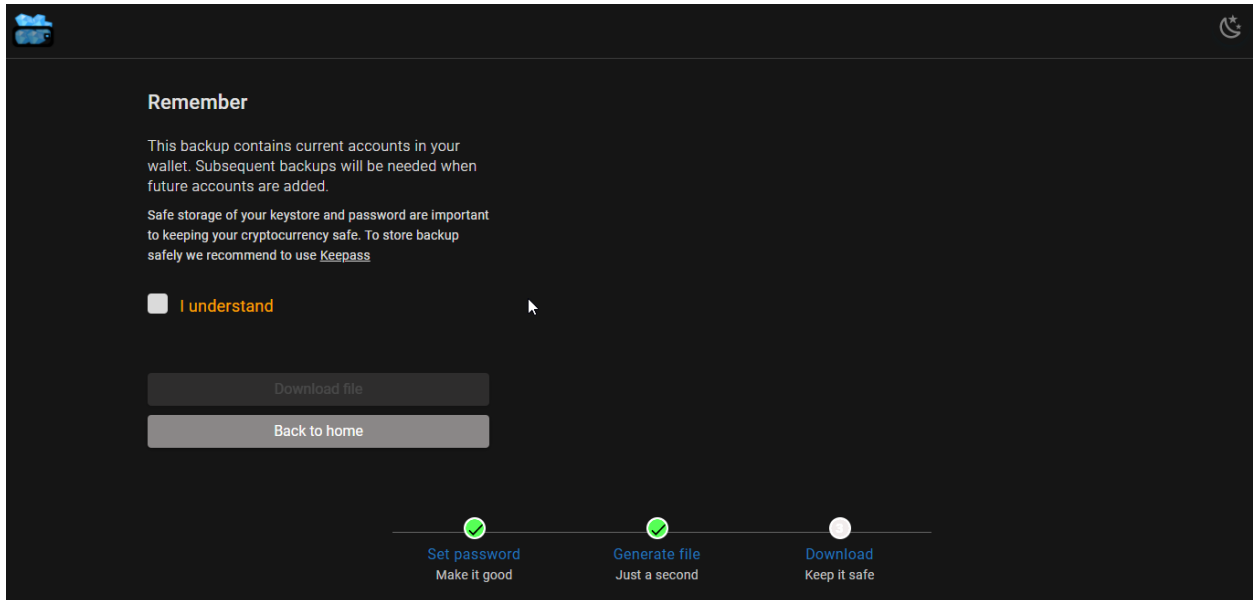
Just a second

3

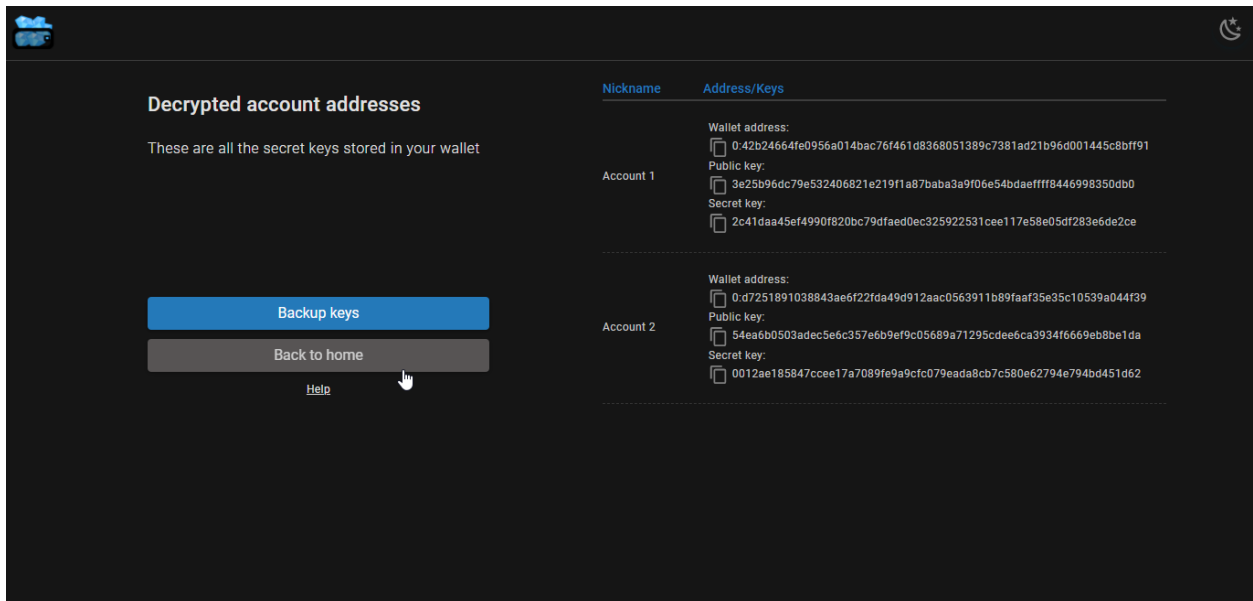
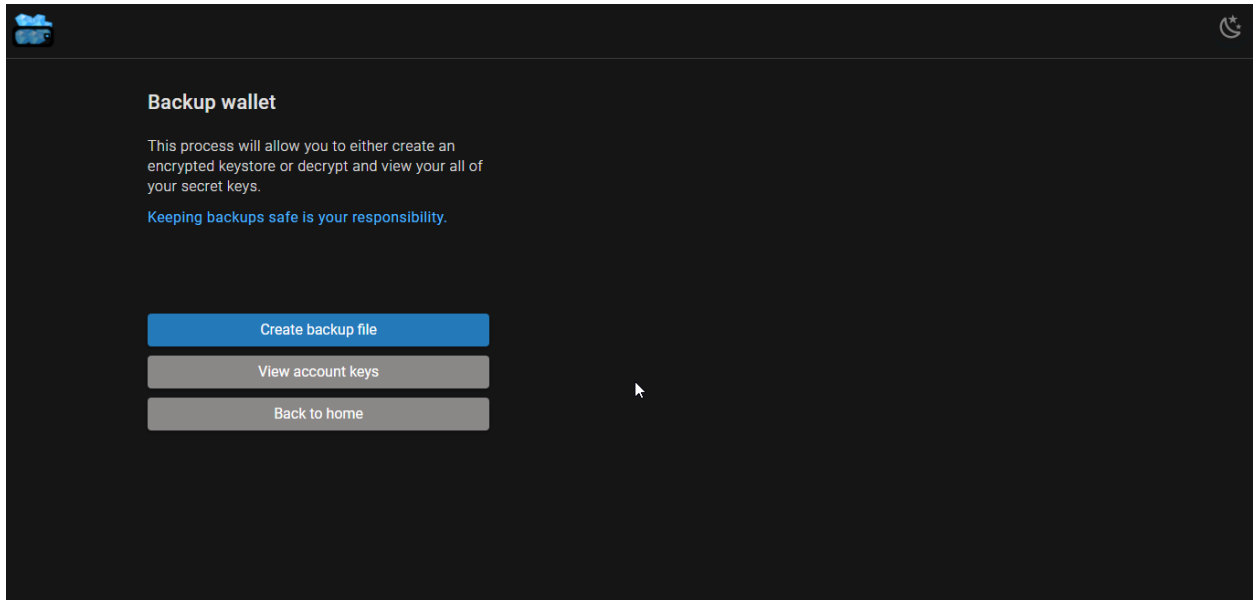
Download

Keep it safe





If the user wants to see own kyes, then it is possible to do via the second way:



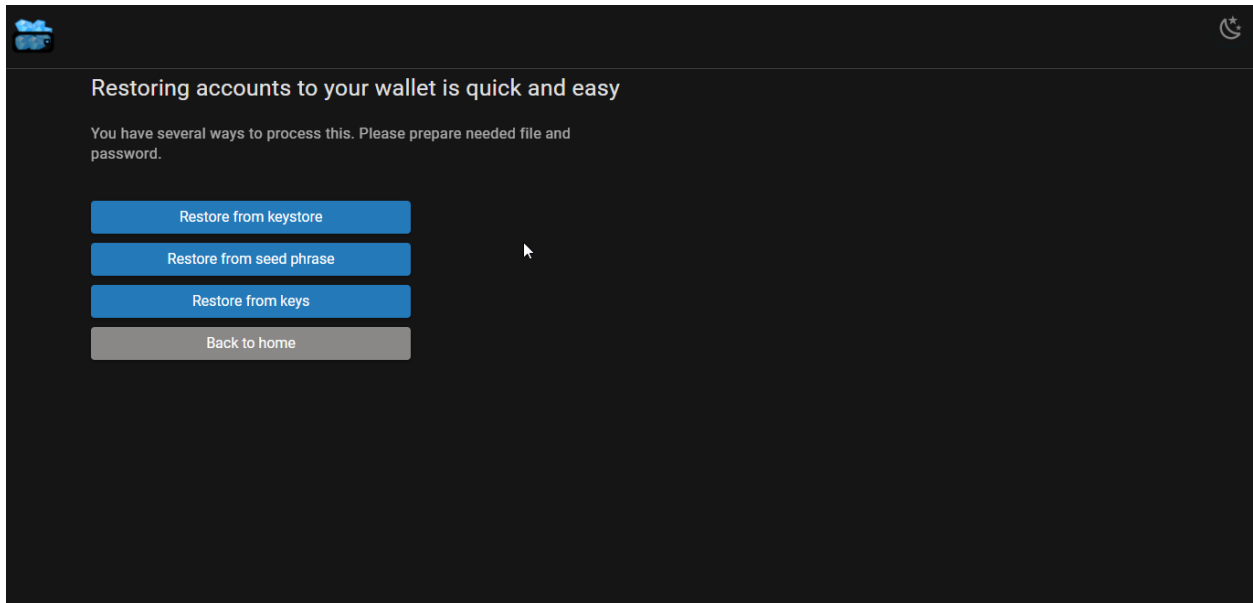
Restore

Web extension allow restoring wallet by 3 ways:

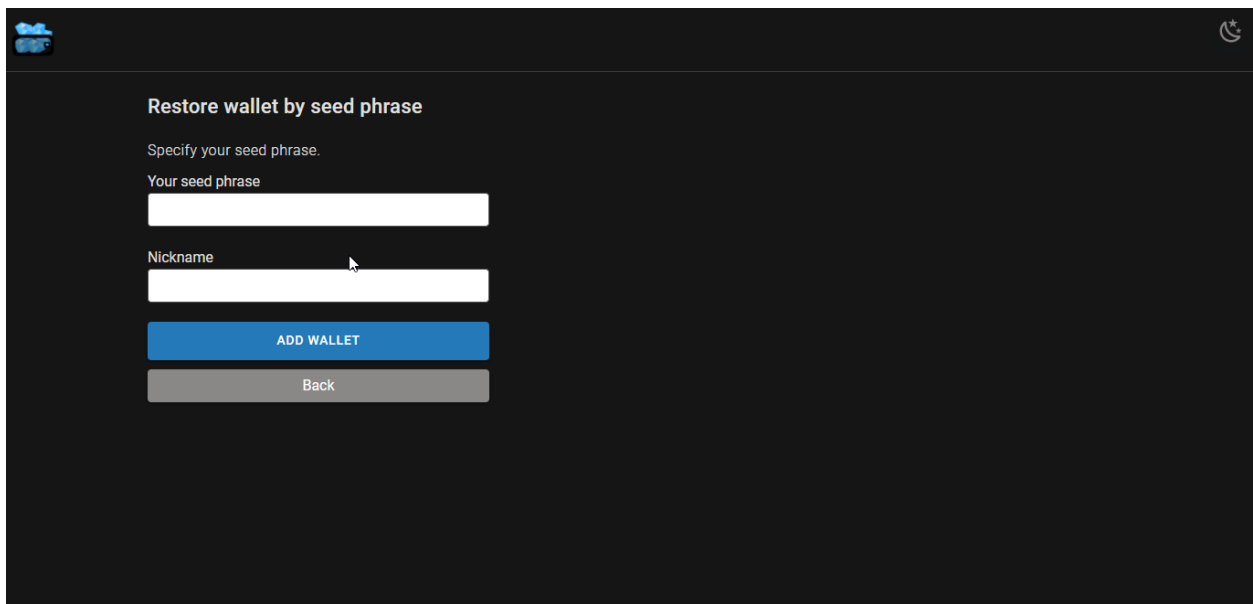
1. From the keystore
2. From the seed phrase
3. From the key pair

UI for restoring by first method (from the keystore) is the same as on the first run step.

UI for restoring by second method (from the seed phrase):



The screenshot shows a dark-themed mobile application interface for restoring a wallet. At the top left is a logo with three blue cubes, and at the top right is a moon icon. The main heading is "Restoring accounts to your wallet is quick and easy". Below it, a subtitle reads: "You have several ways to process this. Please prepare needed file and password." There are four buttons stacked vertically: "Restore from keystore", "Restore from seed phrase", "Restore from keys", and "Back to home". The first three buttons are blue, and the last one is grey.



The screenshot shows a dark-themed mobile application interface for restoring a wallet by seed phrase. At the top left is a logo with three blue cubes, and at the top right is a moon icon. The main heading is "Restore wallet by seed phrase". Below it, a subtitle reads: "Specify your seed phrase." There are two input fields: "Your seed phrase" and "Nickname". Below the input fields are two buttons: "ADD WALLET" (blue) and "Back" (grey).



Restore wallet by seed phrase

Specify your seed phrase.

Your seed phrase

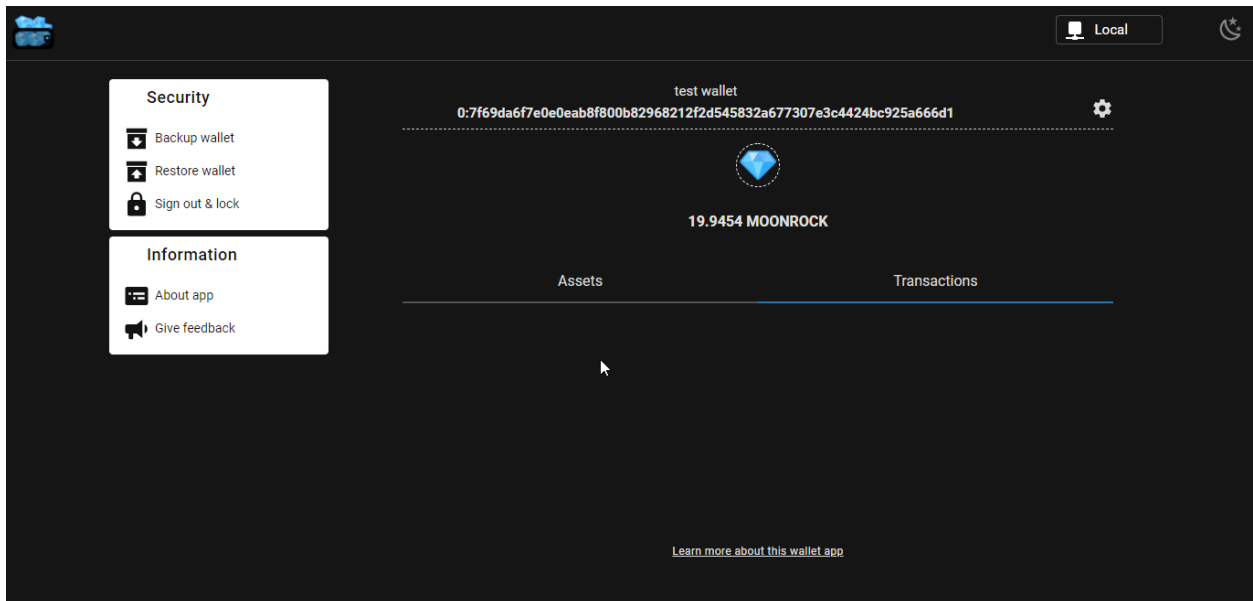
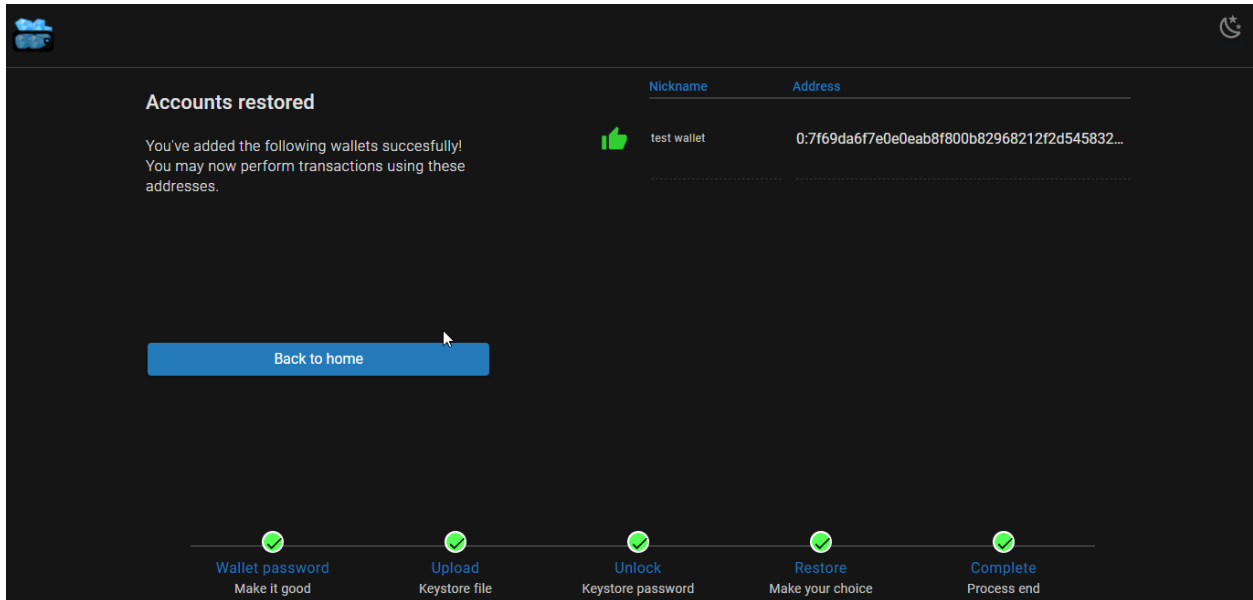
capable service large never drum parrot matter pa

Nickname

test wallet

ADD WALLET

Back



UI for restoring by third method (from the key pair):



Restoring accounts to your wallet is quick and easy

You have several ways to process this. Please prepare needed file and password.

Restore from keystore

Restore from seed phrase

Restore from keys

Back to home



Restore wallet by keys

Specify your keys.

Public key

Secret key

Nickname

ADD WALLET

Back



Accounts restored

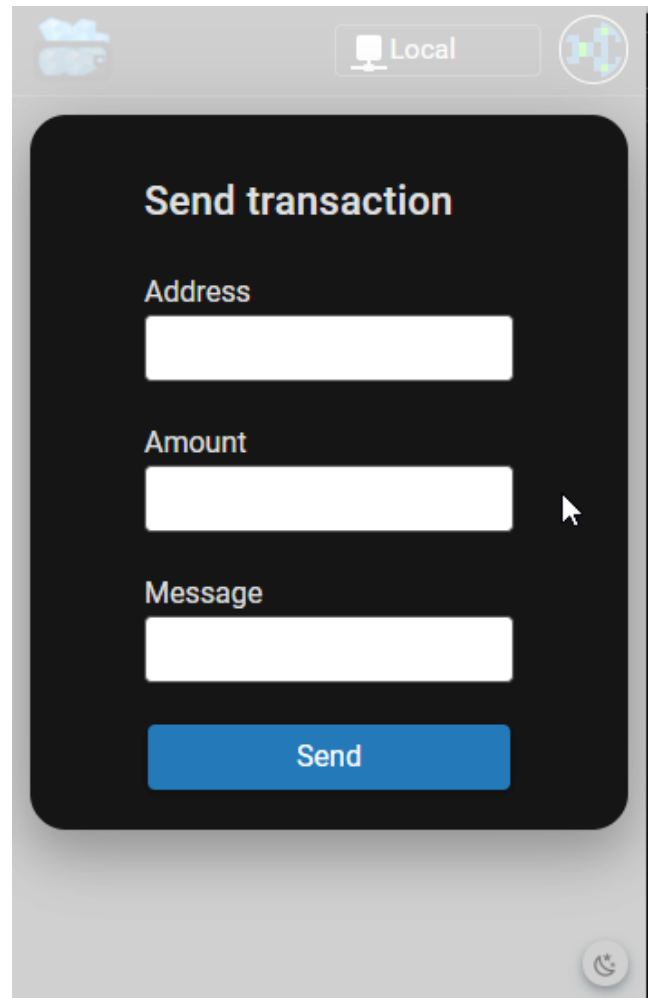
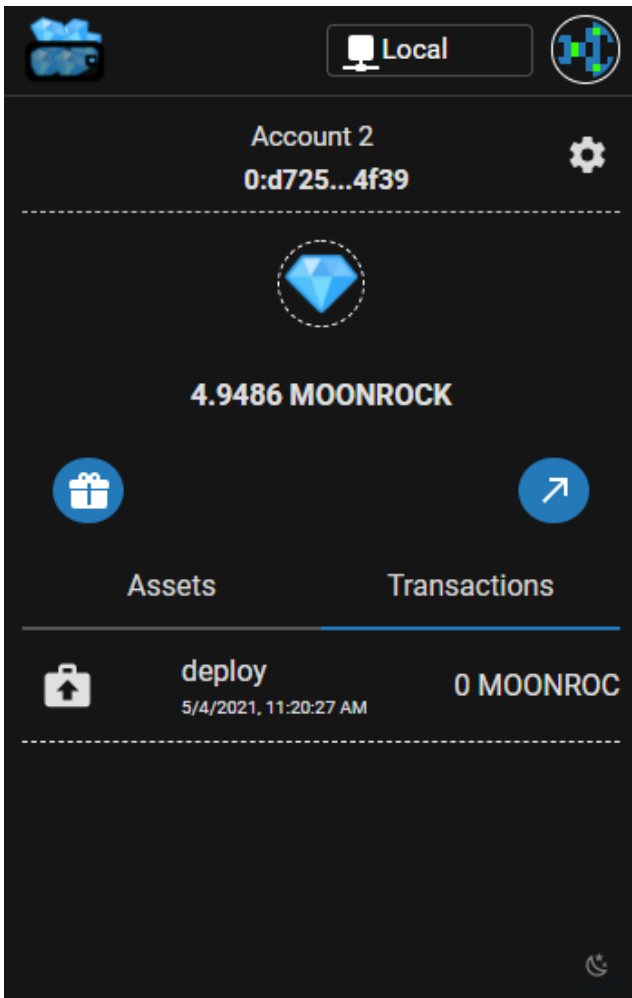
You've added the following wallets succesfully!
You may now perform transactions using these addresses.





Nickname	Address
keys account	0:42b24664fe0956a014bac76f461d8368051389c...

[Back to home](#)

Transaction sending





Local


Send transaction


Address


Amount


Message


Send









Local


Account 2
0:d725...4f39

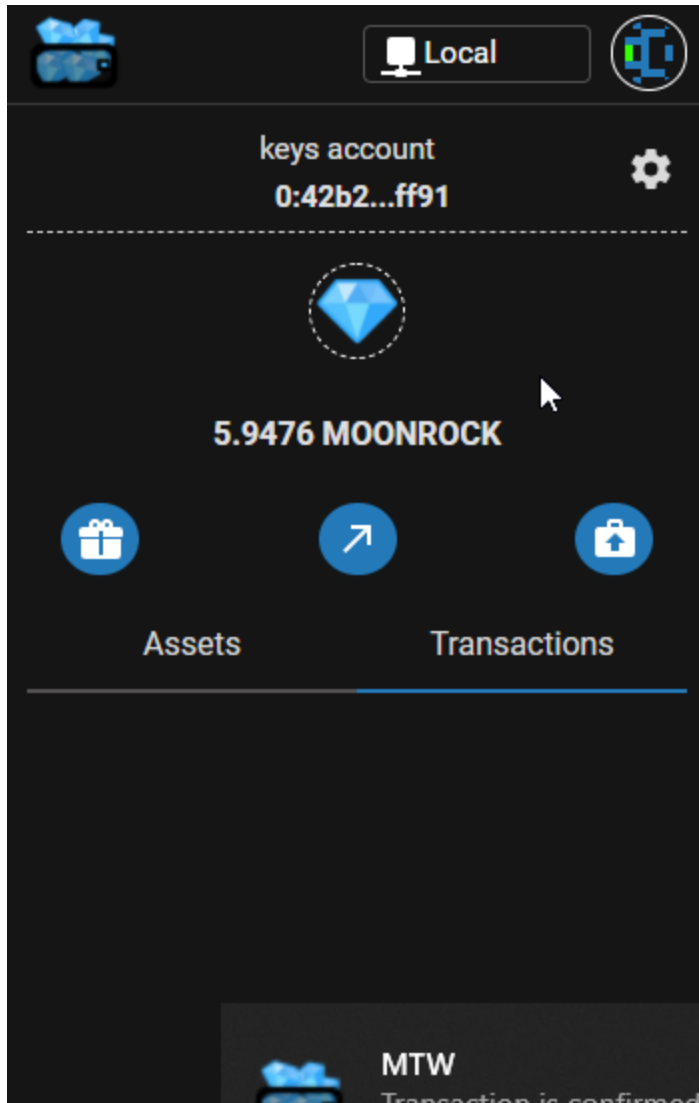

3.9283 MOONROCK



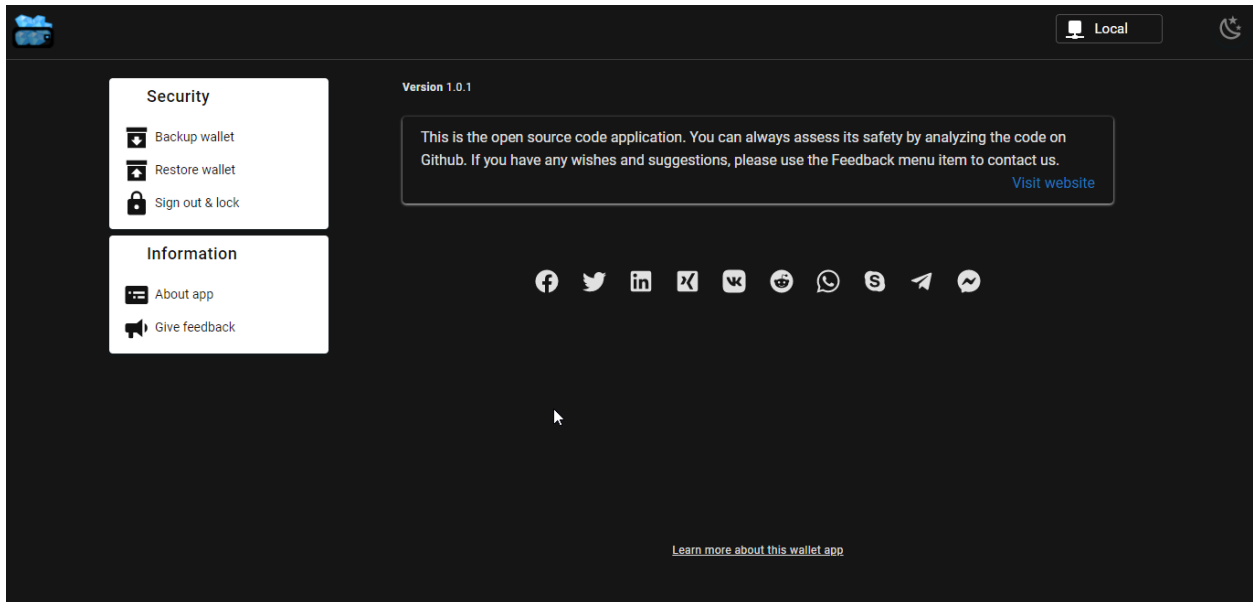


Assets	Transactions
	deploy 5/4/2021, 11:20:27 AM 0 MOONROC
	transfer 5/4/2021, 2:06:16 PM 1 MOONROC





About us



License

The source code provides by Apache License 2.0

Documentation

All documentation you can find by this link (<https://mytonwallet.com>)

Requested features

1. Dapp connector for the signing transactions, the signing messages, the encrypt/decrypt messages
2. Creating multisign wallet (many owners, requirement count more than 0)
3. Adding tokens and interactions with them
4. DePools supporting
5. Loading ABI for the smart contract interaction

6. Smart contract editing/compiling/deploying right from the web extension
7. Mobile version for IOS/Android platforms with the same functionality as in browsers
8. Qr codes for the payment system via the mobile version (the payment sending, ask payment, the deep link)
9. Onboarding library for Dapp for the quick installation of the web extension or the mobile version
10. Swipe operations
11. DEX supporting
12. Buying on external exchanges
13. Supporting of easy conversion between currencies

Source code

<https://github.com/mytonwallet/web-extension>

<https://github.com/mytonwallet/mytonwallet.github.io>

Contacts

Telegram: @telepulos (not corporate account, just the coordinator of the team members)

Email: support@mytonwallet.com