

---

# **FIN4Xplorer**

*Release 0.7.0*

**23.07.2020**



---

## Inhaltsverzeichnis

---

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	Technical infrastructure . . . . .	3
1.2	Digital Wallet . . . . .	4
1.3	Getting Ether . . . . .	4
<b>2</b>	<b>Applikationsübersicht</b>	<b>5</b>
2.1	Startseite . . . . .	5
2.2	Tokens . . . . .	6
2.3	Claims . . . . .	7
<b>3</b>	<b>Tokens erstellen</b>	<b>9</b>
3.1	Assistent zur Token-Erstellung . . . . .	9
<b>4</b>	<b>Tokens bekommen</b>	<b>11</b>
4.1	Beweisen . . . . .	12
<b>5</b>	<b>More functionalities</b>	<b>15</b>
5.1	Wallet . . . . .	15
5.2	Settings . . . . .	16
5.3	Inbox . . . . .	19
5.4	Token curation . . . . .	20
5.5	Sources of value . . . . .	20





Willkommen zur Dokumentation der FIN4Xplorer Plattform.

The latest live version can be found at [demo.finfour.net](https://demo.finfour.net) and more general information are available at [finfour.net](https://finfour.net).

The basic idea is: anyone can create tokens and anyone can claim these tokens. Once a claim is made, the user has to provide proof that the associated positive action actually happened. Which proof or proofs these are, is decided by the token creator when designing the token. If all proofs on a claim are automatically (e.g. a location to be within a certain circle) or manually (e.g. someone involved in the action is asked to approve a photo) approved, the claimer gets an amount „minted“ on that token. We say someone „gets tokens“, which basically means that that persons public address gets an entry in the balance-sheet on the respective token with a positive number.



### 1.1 Technical infrastructure

**FIN4Xplorer** consists of two parts:

- Smart contracts that are deployed to the Ethereum blockchain. The entire functionality is in these smart contracts. Everything could also be done by interacting with them via a command line.
- A frontend or client in the form of a *Web DApp* that acts as an user interface for convenient interaction with said smart contracts. Web apps run directly in the browser, both on desktop computers and on smartphones. DApp stands for distributed app, meaning they connect to a blockchain.

#### 1.1.1 Infrastructure, programming languages and frameworks used

We decided to use the [Ethereum](#) blockchain for now. It satisfies our requirements of an open-source and public blockchain that supports smart contracts. Furthermore it is used widely these days and therefore plenty support resources are available.

The smart contracts are written in [Solidity](#) and we deploy them via [truffle](#) to the [Rinkeby test network](#).

Rinkeby is one of a few Ethereum test networks and behaves similar to the main network. The difference is, that Ether, the „currency“ used to pay for transactions, is not worth „real“ money. On the main network Ether has to be earned via mining or bought on trading platforms. On Rinkeby it can be requested from „faucets“ for free, see section [Getting Ether](#).

During development we use [Ganache](#) from truffle locally to simulate an Ethereum blockchain.

For storing media files that certain proof types require, we use [IPFS](#) via the [Infura](#) gateway.

As digital wallet to connect desktop or mobile browsers to the Ethereum network we use [MetaMask](#), more about that in the following section [Digital Wallet](#).

The frontend is a [React app](#) that uses [drizzle](#) from truffle to connect to the smart contracts on the Ethereum blockchain. For now we don't have native mobile apps. Mobile users have to use DApp browser apps and navigate to the same site as desktop users. The layout of the frontend works well on different screen sizes as relatively narrow boxes are arranged

per page. Depending on the width of the browser window, they will either be rendered next to each other or below one another.

## 1.2 Digital Wallet

The only, yet significant, difference a visitor of a Web DApp experiences compared to a Web App, is the need for a „bridge to the blockchain“. Also called a digital wallet. Its first task is to establish the connection to the blockchain network - either via a [full node](#) that users run themselves or via a gateway service like Infura. Its second task is to pop up whenever the user seeks to write data to the blockchain and enable the convenient signing of such transactions with the users private key.

There are many organizations and projects out there offering digital wallets in different stages of maturity and compatibility. While the trend goes towards deeper and native integration into browser (or even operation systems), for now the market appears scattered.

The digital wallet we are using ourselves, and that seems to be the currently best supported one out there, is MetaMask: [metamask.io](#). MetaMask is available both as extension for desktop browsers (Chrome, Firefox, Opera and Brave) and as early-stage mobile app (iOS and Android).

Once you have a digital wallet, either restore an existing account or follow the process to create a new account. In most digital wallets, creating a new account involves the randomized mathematical creation of a new and unique mnemonic (also called „seed phrase“). From this, one or more accounts, each with a public address and a matching private key, can be derived deterministically. Meaning you only have to remember/store (the digital wallet will probably remind you thoroughly to do so) the mnemonic to restore your account(s) anytime on any machine. Usually only the first account per mnemonic will be used.

After having created your new account, don't forget to switch to the *Rinkeby Test Network*. By default you will be on the *Main Ethereum Network*.

## 1.3 Getting Ether

After installing a digital wallet and creating an account, the last step before being ready to use FIN4Xplorer is to get some Rinkeby Ether (ETH). All transactions on the Ethereum (test) network cost Ether, the so called gas fee. The price depends on the current workload of the network as well as how many additions/changes to data on smart contracts your transaction causes. On test networks, Ether can be obtained for free from „faucets“. On the main network, Ether has either to be earned by being an active participant in the network (running a full node and doing mining), has to be traded or has to be bought on exchanges.

For obtaining Ether on the Rinkeby test network we suggest two options:

- Using the „authenticated faucet“ at [faucet.rinkeby.io](#) requires a social media post
- We build our own little faucet server that gives a small amount of ETH per click if the user has less then a threshold value: click *Request Ether* on the landing page of [demo.finfour.net](#) and wait a little until you get a confirmation message.



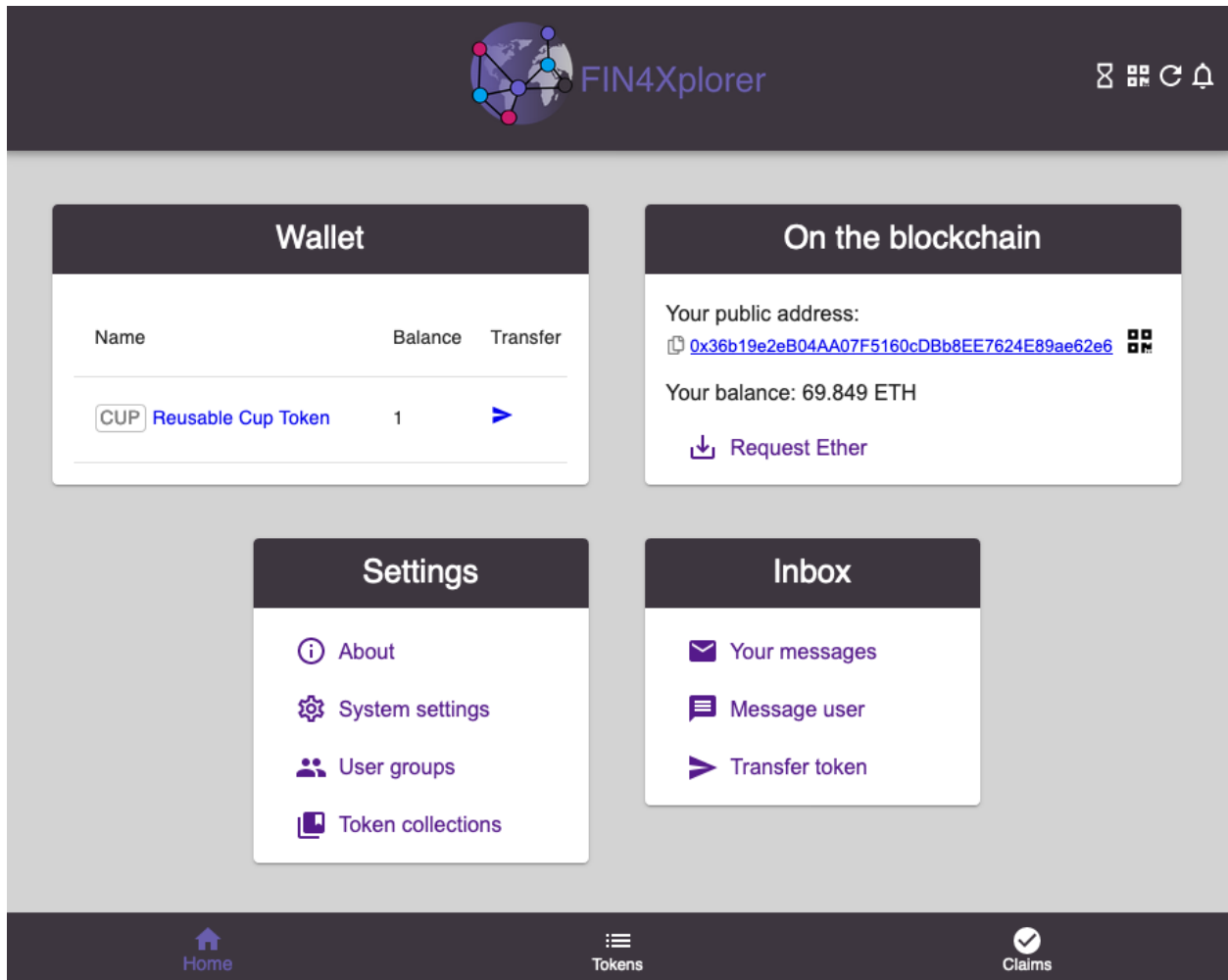
---

## Applikationsübersicht

---

### 2.1 Startseite

*Home* acts as landing page and is the central place of the DApp.



The two main subsites are accessible in the bottom bar: *Tokens* and *Claims*.

The icons in the top bar are for showing pending transactions, the users QR code, reloading the page and for getting notified about new messages.

The various boxes shown are explained in the chapter [More functionalities](#).

## 2.2 Tokens

Under *Tokens* new tokens can be created as described in [Create tokens](#). On the right side all existing tokens are listed with direct links to claiming them or to view more details about them.

### Create a new token

- + Start token creation wizard
- ↑↓ Upload token draft (JSON)
- 📄 Copy an existing token

### All tokens

Name	Supply	Actions
<span style="border: 1px solid #ccc; padding: 2px;">CUP</span> Reusable Cup Token	1	<a href="#">View</a> <a href="#">Claim</a>

Clicking on *View* on a Token brings up a dedicated site showing details about the design and some performance statistics of the token.

CUP **Reusable Cup Token**

[0x1B72852e4767eD3917d94087eB35D8d2A6f73950](#)

Short description: -  
Long description: -

---

Creation: Today at 3:08 PM

Token creator: [0x36b19e2eB04AA07F5160cDBb8EE7624E89ae62e6](#)

Verifier types: Claimable only N times per user,  
Specific address

Total number of claims: 1

Total supply: 1

---

is transferable  
 is mintable  
 is burnable  
 is capped

---

Decimals: 0  
Initial supply: 0

Minter role owners:  
Fin4Claiming contract

---

Fixed minting quantity per claim: 1  
Unit of measurement: quantity  
Claimable actions: -

Token Profile

Your balance: 1

---

Token actions: [Claim](#), [Transfer](#), [Download claims](#)

## 2.3 Claims

The layout of the *Claims* site is analog to the *Tokens* site: submitting new claims can be done on the left side and a list of previous claims is displayed on the right side. If users get to this site via a *Claim-Link* on a token directly, the *Token Type* will be prefilled. More info about the claiming process can be found in [Claim tokens](#).

### Claim Tokens


Token  
Select... | v

Comment

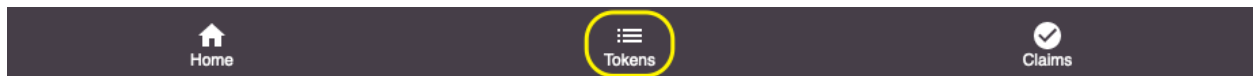
+ SUBMIT

### My previous claims

Reusable Cup Token 1 CUP

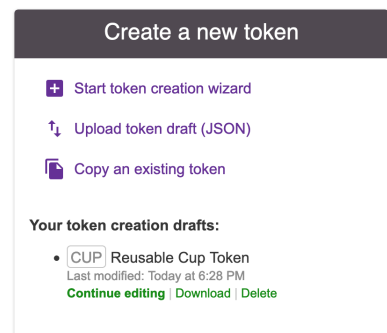
Today at 3:08 PM  APPROVED

## Tokens erstellen



Die Möglichkeit für jeden, neue Token auf der FIN4Xplorer-Plattform zu erstellen, steht im Mittelpunkt seines Konzepts. Sobald ein Token erstellt wurde, kann es von jedem beansprucht werden. Ein erfolgreicher Anspruch führt zu einem Saldo auf dem jeweiligen Token. Da wir unsere Token-Verträge auf dem [ERC-20 standard](#) basieren, wird dieser Kontostand in jeder digitalen Ethereum-Wallet-App angezeigt, die den ERC-20-Standard unterstützt. Beachten Sie, dass Sie auch in solchen Apps zum Rinkeby-Testnetz wechseln müssen.

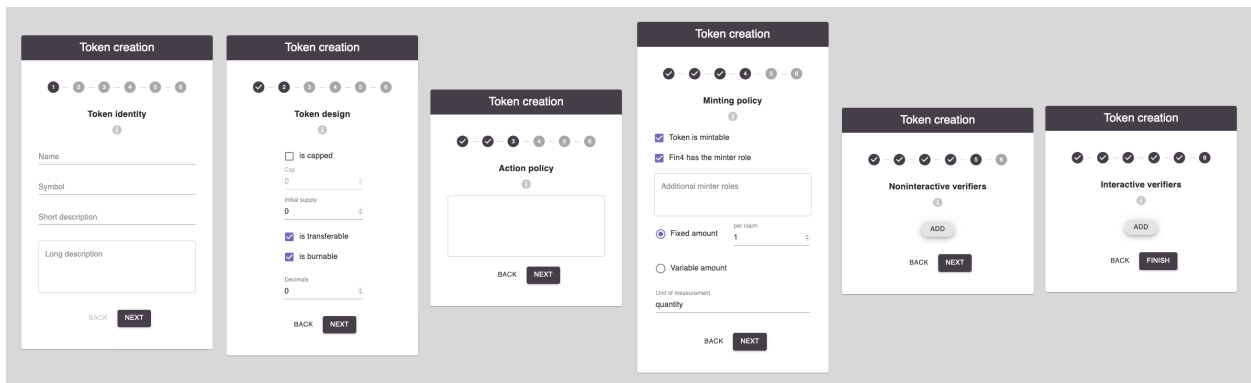
### 3.1 Assistent zur Token-Erstellung



Der Aufwand für die Erstellung eines Tokens kann von einem schnellen 1-Minuten-Klick bis zu einer langwierigen Planung in einem multidisziplinären Team reichen. Die möglichen Einstiegspunkte für den Token-Erstellungsprozess sind:

- Erstellen eines neuen Tokens von Grund auf neu
- Hochladen eines Token-Entwurfs im JSON-Format
- Kopieren eines vorhandenen Token-Designs als Vorlage

Die letzten beiden importieren einen Entwurf zur Token-Erstellung, der am unteren Rand des Felds angezeigt wird. Die erste Option führt Sie direkt zum Assistenten zum Erstellen von Token. Es besteht aus 6 Schritten: *Token-Identität*, *Token-Design*, *Aktionsrichtlinie*, *Münzrichtlinie*, *Nicht interaktive Verifizierer* und *Interaktive Verifizierer*. In jedem Schritt kann eine Infobox geöffnet werden, die Erklärungen enthält.



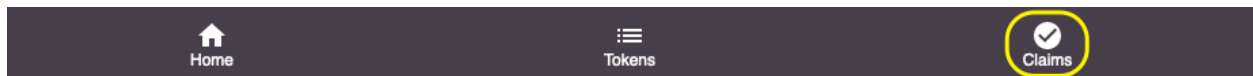
Beachten Sie, dass Sie den Assistenten jederzeit verlassen können und Ihr Fortschritt als Entwurf gespeichert wird. Entwürfe können verwendet werden, um den Vorgang später fortzusetzen oder um sie herunterzuladen, freizugeben und zu importieren.

Sobald die 6 Schritte abgeschlossen sind, kann das Token in der Blockchain erstellt werden. Dafür müssen zwei oder mehr Transaktionen bestätigt werden. Zwei für die Token-Erstellung selbst und eine für jeden Proof mit Parametern. Bis alle Proof-Parametrisierungstransaktionen bestätigt sind, bleibt der Token deaktiviert.

---

## Tokens bekommen

---

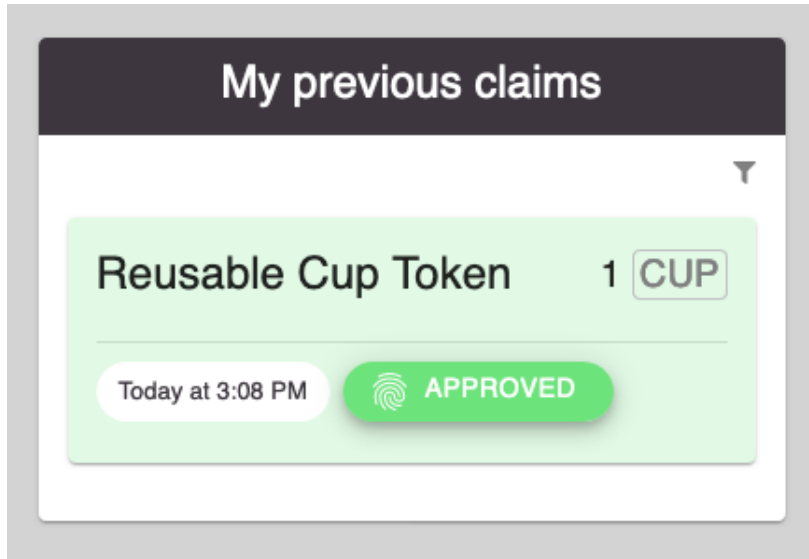


Tokens, die wie in [Tokens erstellen](#) beschrieben erstellt wurden, können beansprucht werden.

A form titled 'Claim Tokens' with a dark header. It contains a 'Token Type' dropdown menu with 'CUP Reusable Cup Token' selected, a 'Comment' text input field, and a '+ SUBMIT' button at the bottom.

Das Einreichen eines Anspruchs ist der erste Schritt, um eine Guthaben für den betreffenden Token zu erhalten. Einmal gemacht zeigt es sich unter *My previous claims*. Ansprüche dort können in einer von drei Stufen sein:

- In rot mit einer Aufforderung an *Submit proof*, dies ist die Standardeinstellung nach Einreichung eines neuen Anspruchs
- In grün der genehmigte Zustand, was auch bedeutet, dass dem Antragsteller ein Saldo geprägt werden muss
- In grau der abgelehnte Zustand, wenn einer oder mehrere der Beweise manuell oder automatisch abgelehnt wurden



Ein Klick auf das Filtersymbol in der oberen rechten Ecke öffnet ein Menü, in dem Sie Ansprüche basierend auf ihren drei Stufen ausblenden oder anzeigen können.

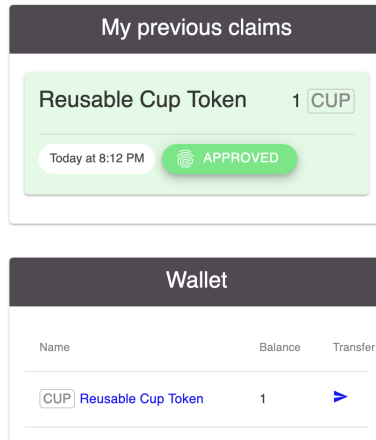
## 4.1 Beweisen

Damit ein Anspruch erfolgreich ist, müssen alle vom Token-Ersteller angegebenen Prüfer vom Benutzer einen Nachweis erhalten und genehmigt werden - automatisch oder von der ausgewählten Person, Gruppe oder dem ausgewählten Sensor. Eine Proof-Submission-Site könnte so aussehen.

Hier wird ein Verifizierer angezeigt (*ClaimableOnlyNTimes*), der nicht manuell initiiert werden musste. Es gehört zur Kategorie der Prüfer, die automatisch ihre Zustimmung oder Ablehnung erteilen, sobald ein Anspruch geltend gemacht wird.

Sobald alle Prüfer Beweise erhalten und genehmigt haben, wird der Anspruch grün angezeigt und der Antragsteller erhält einen Saldo auf diesem Token.





Wenn ein Prüfer eines Anspruchs den vorgelegten Nachweis ablehnt, wird der gesamte Anspruch abgelehnt. Es ist nicht möglich, diesen Anspruch durch erneutes Einreichen von Beweisen wiederherzustellen. Ein neuer Anspruch muss geltend gemacht werden, wenn der Benutzer es erneut versuchen möchte.

Wenn Verifizierer von der Genehmigung anderer Benutzer abhängig sind, haben sie die Möglichkeit, dem Antragsteller zusammen mit ihrer Genehmigung oder Ablehnung eine Nachricht hinzuzufügen. Diese Nachricht wird auf der jeweiligen Proof-Submission-Site angezeigt.



---

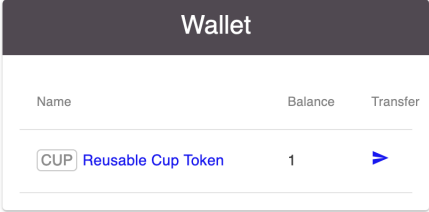
## More functionalities

---

All the functionalities presented below are accessible via the [Home site](#).

### 5.1 Wallet

The wallet box shows the users balance on all tokens with approved claims. The token name links to its respective detailed site (TODO ref).




Name	Balance	Transfer
<a href="#">CUP Reusable Cup Token</a>	1	<a href="#">▶</a>

The transfer icon links to a site allowing the *transfer* of balances on transferable tokens to other users.


#### 5.1.1 Transfer Tokens

Note that once a transfer has been made, there is no way to undo it from the senders site. Only the recipient could choose to send it back.


## Transfer tokens to user



Token Type

Reusable Cup Token 

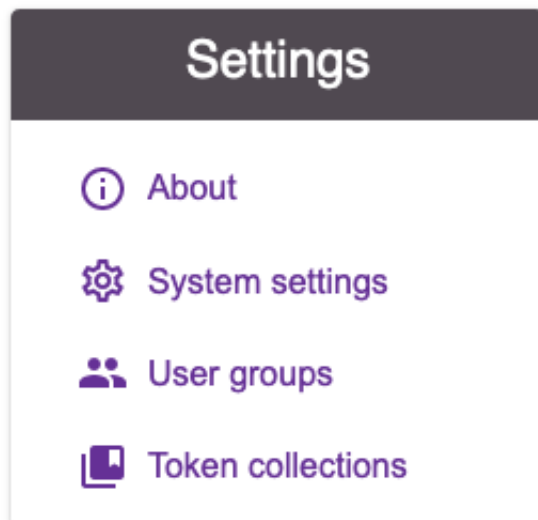
---

Amount 

**SEND**

## 5.2 Settings

*Settings* contains Links to the *About* and *System settings* sites. The latter contains the language preference and a list of relevant smart contract addresses.



Furthermore, *User groups* and *Token collections* can be found there.

## 5.2.1 User Groups

User groups can be used for two purposes. Firstly to define „administrators“ of a *token collection* who can add or remove tokens from a collection. Secondly, certain proof types can notify members of a user group about a pending approval that one of them should approve or reject.

Under *My groups* users can message the owner (unless they are the owner) and leave groups (also possible for owners). Upon leaving a group, users can choose to notify the group owner about it or not.

Create a group

Group name

Add yourself as member

**SUBMIT**

Groups you created

Group name	ID	Action
Reusable Cup Approvers	0	<a href="#">Edit</a>

My groups

Group name	ID	Action
Reusable Cup Approvers	0	You are owner <a href="#">Leave group</a>
Sustainable Zurich collection admins	1	<a href="#">Message owner</a> <a href="#">Leave group</a>

Editing a group as owner comprises the options to add or remove members and to transfer ownership of the group.

Edit group

Reusable Cup Approvers

You are the creator of this group

Edit members

Member	Action
0xA63FD7032AEA19FE728498D970dF8AB938bc764	<a href="#">Remove yourself</a>
0x97Bd320109859833784d3cB720e32653f5e93464	<a href="#">Remove</a>

Removing yourself as member does not change your ownership of this group

Add a member    Add multiple members

Public address of new member QR

**ADD**

Transfer ownership

TRANSFER OWNERSHIP

## 5.2.2 Token collections

Token collections are a way to curate a list of tokens with a much lower effort than the community-curated approach of the *token curated registry*.

### Create token collection

Collection-Name

---

Short-name (e.g. "ethz" for "ETH Zürich")

---

Description

---

**+ SUBMIT**

### Token collections

Name	Actions
Sustainable Zurich	<a href="#">View</a> <a href="#">Edit</a>

They can be used for organisations to promote their tokens in one place, for individuals to create portfolios or other use cases.

### Collection details

**Sustainable Zurich**

Description: Tokens focused on sustainability in Zurich

Number of tokens: 1

[Edit collection](#)

### Tokens in collection

Name	Supply	Actions
<span>CUP</span> Reusable Cup Token	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Claim</a>

The collection creator can add or remove tokens from the collection, appoint an admin group to do the same or transfer ownership of the collection.

**Edit Collection**

Edit **Sustainable Zurich**

[View collection](#)

You are the creator of this collection

**Manage tokens**

Token	Action
<div style="border: 1px solid #ccc; display: inline-block; padding: 2px 5px; margin-right: 5px;">CUP</div> Reusable Cup Token	<a href="#" style="color: blue; text-decoration: underline;">Remove</a>
Select... <span style="float: right;">  ▾</span>	
<div style="background-color: #333; color: white; border-radius: 15px; padding: 5px 15px; display: inline-block;">+ ADD TOKENS</div>	

**Manage admins**

+ SELECT USER GROUP

**Transfer ownership**

TRANSFER OWNERSHIP

## 5.3 Inbox

**Inbox**

- ✉
[Your messages](#)
- 💬
[Message user](#)


*Your messages* shows all messages sent to you. These can be

- Proof contracts notifying you about pending approvals: for you directly or a group you are a member of
- System notifications like a user leaving a group you created
- Messages from other users

If you get a new message while being on the DApp, the notification bell in the top right corner of the site will turn yellow and take you to the messages site upon clicking.


*Message user* takes you to the option to message other users directly. The QR icon can be used to scan the public address of the recipient instead of having to type or paste it in.

### Send message to user

Public address of message-receiver 

Message text

Note that messages are *not* encrypted. With some effort they can be retrieved in plain text from the blockchain.



## 5.4 Token curation

The token curation feature (Token Curated Registry, TCR) is currently disabled. If you are interested how it would look like, see the previous version of this documentation.

## 5.5 Sources of value

In an upcoming version we will activate a new feature we are working on. It will allow token creators to attach underlying sources of value to their token. This could be Bitcoin-backed ERC20 tokens, other tokens with special properties or just free text for something the token creator promises off-chain to redeem a token for.

The [FIN4Documentation repository](#) that this documentation is being build from is licensed under „Creative Commons Attribution Share Alike 4.0 International“ (CC-BY-SA-4.0). Both the [FIN4Xplorer repository](#) as well as the [FIN4Contracts repository](#) are licensed under „GNU Affero General Public License v3.0“ (AGPL-3.0).