Build Documentation

In order to run the project:

1. the appwrite docker swarm must be started
2. a project must be created in the Appwrite Console, with platform and API key needs
3. environment variables must be set with project-related data from Appwrite Console
4. an Admin teams and database collections should be initialized with provided python scripts
5. cloud functions must be added to the Appwrite project
6. the frontend must be build and started

On some Operating Systems like Ubuntu, you might need to write `sudo` before all the `docker` and "install"-related commands.

1. Starting Appwrite

If you don't have installed appwrite already in the project's root directory, you should do so first. You can use the single CLI command as described in `./frontend/README.md`.

Otherwise, when the `./appwrite/` is available together with `./appwrite/.env` and `./appwrite/docker-compose.yml`, you can start appwrite via one command within the `./appwrite/` folder:

```
docker-compose up -d
```

It will not only start the Appwrite docker containers but also update running containers when `docker-compose.yml` or `.env` was changed.

2. Project Creation

This step can be skipped if this was done one time in the past.

You can check an online tutorial for how to setup a project:

link text

3. Environment variables

NFT-the-World requires some environment variables. You can find a list of them in `./frontend/.env`. If missing, the frontend defines own default values set in `.fronted/utils/config.js`.

So before any scripts or the application can properly work, these environment variables must be set with values that you can obtain from the Appwrite console that was shown in step 2. Typical environment variables specifically used for our project start with `APPWRITE_...` or `REACT_APP_...`. Environment variables can be exported like so (replacing the `<project-ID>` with an actual string):

```
export APPWRITE_PROJECT=<project-ID>
```

4.1. Admin Team Creation

In order to use the privileged features in the frontend, initial teams of Admins must be added to the project. This process is faciliated by a Python script which is located in `./backend/python_init_script/main.py`.

The python script requires three things of input information. These can either be passed as command line arguments. If not specified, an environment variable is used by the Python script.

- Appwrite Endpoint URL, which is for example accessed by the frontend for API functionality. This typically is `<your-domain/IP>/v1/` including the protocol specification (e.g. `http://`).

  Environment variable: `APPWRITE_ENDPOINT` Command line prefix: `--endpoint=...`

- Appwrite Project ID. Copy the Project ID that is linked to the created project from step 2. You can copy the number from the URL while the project settings are opened. Example: `http://localhost/console/home?project=618eea46b90ef` -> the Project ID is the hexadecimal number behind `?project=`, i.e. `618eea46b90ef`.

  Environment variable: `APPWRITE_PROJECT_ID` (**NOT `APPWRITE_PROJECT`**) command line prefix: `--projectid=...`

- API key. This also needs to be taken from the Appwrite console. Choose an API-Key with enough permissions for the features that you'd want to use. For simplicity, it's recommended to use a master key with all permissions enabled. The given name of your key is the required value. Example: if my key is named `nfttheworld_masterkey` then this will be the value to use.

  Environment variable: `APPWRITE_API_KEY` command line prefix: `--apikey=...`

After you got the necessary project information, define the initial team of admins by editing `./backend/python_init_script/template.csv` (or the `.xlsx` excel file).
Then start the script via following command from the root of the Git project, which does the remaining job for you. If you don't want to use the exported `$APPWRITE_...` environment variables, you can override environment variables with corresponding CLI arguments. You can omit CLI arguments when environment variables should be used.

```
python3 ./backend/python_init_script/main.py --endpoint=<...> --projectid=<...> --ap
```

If the command gives you a "module import error", you need to install the python package `pandas` and `appwrite` with Python's package manager "PIP"

```
pip3 install pandas appwrite
```

If you executed the script for the first time, it should tell you that it didn't find the users that you specified in your `template` file, i.e. they were added.

4.2. Initialization of database collections

There are python scripts which can be executed to initialize the database "collections" (a group of database documents with equal format) which allow the backend to create database documents and add them to that collection. Environment variables are needed again as input for the scripts. Execute following commands from the route directory.

Before execution, environment variables `APPWRITE_ENDPOINT`, `APPWRITE_PROJECT` and `APPWRITE_API_KEY` need to be `export`ed (e.g. `export APPWRITE_ENDPOINT=...`).

- initialize Wallet Collection:

  ```
  python3 ./backend/database-collection-schemas/createWalletsCollection.py
  ```

- initialize Announcement Collection:

  ```
  python3 ./backend/database-collection-schemas/createAnnouncementCollection.py
  ```

After the Wallet Collection, you can copy the "Collection ID" value of the "Wallets" in the Appwrite Console.

database settings for wallet collection

Add this value to an environment variable `REACT_APP_WALLET_COLLECTION_ID` which will be needed for remembering the Wallet connection. You should also add all environment variables related to the Appwrite Project to `./frontend/.env` .

5. Add cloud functions

For each cloud function in the `./backend/appwrite-functions/` directory (which is represented by a subdirectory), the cloud function needs to be installed in the Appwrite project.

Each cloud function directory contains a README.md which explains the deployment. In summary, deployment requires an archive like `.tar.gz` containing the executed script (`main.py`) and all required additional dependency modules that must be installed into a proper subdirectory via `pip` . For example the `appwrite` dependency can be installed into subdirectory `./.appwrite/` by executing this comment in the root of the archived directory.

```
PIP_TARGET=./.appwrite pip install -r ./requirements.txt --upgrade --ignore-installe
```

`requirements.txt` and `main.py` must be located in the same directory.

The archive then is uploaded as function to the appwrite console. In the project page, the `Functions` area can be opened by clicking on the `Functions` tabin the left side ribbon. Choose `python...` as runtime and any name you like and upload the archive.

This manual process can also be automatized via the `appwrite CLI` but which is totally optional to install and use:

```
export APPWRITE_FUNCTION_ID=<13-digit-hexadecimal-string>
appwrite functions createTag --functionId=$APPWRITE_FUNCTION_ID --command=<script-ru
```

After the upload is finished, the "Function ID" on the right side of the "Functions" page should be copied and saved in environment variable `APPWRITE_FUNCTION_ID` . Other required environment variables are `APPWRITE_FUNCTION_PROJECT_ID` (yet another environment variable for the Appwrite Project ID), `APPWRITE_FUNCTION_USER_ID` (which can be retrieved from the "Users" page of the Appwrite Console for the desired user) and `APPWRITE_FUNCTION_DATA` (which contains input for the function).

6 Building and running the frontend

After the environment variables are set in `./frontend/.env` , Admins teams and database collections were initialized, the frontend can be build in the last step. In the root of the git project, you can use following for building

```
docker build -t <name> ./frontend/
```

and running

```
docker run -it -p <port>:80 <name>
```

the application. `<name>` could be for example `nftfrontend` and `<port>` could be `8181` which means that the application would be available at the URL `${APPWRITE_DOMAIN}:8181` .

Done

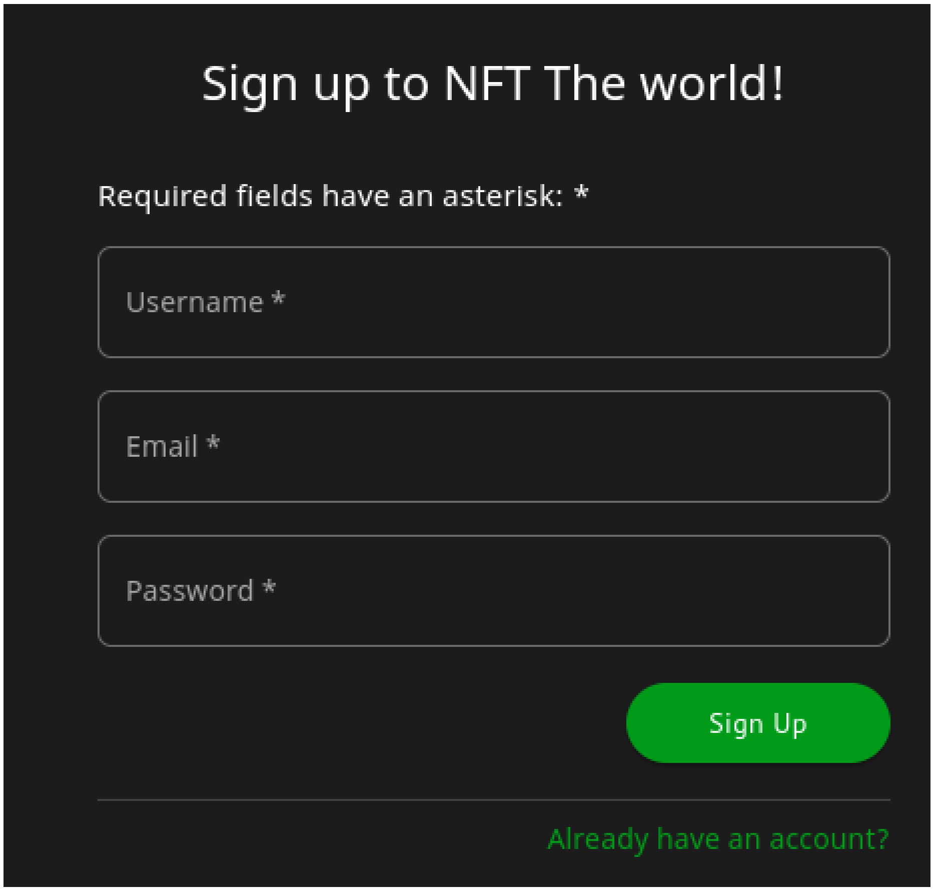
Now you can access *NFT-the-World* at the port to which you bound the frontend's docker container.

User Documentation

Following can be used in the app:

sign up

One can create a new account on the signup page. Enter a username, email and password, and create a new account.

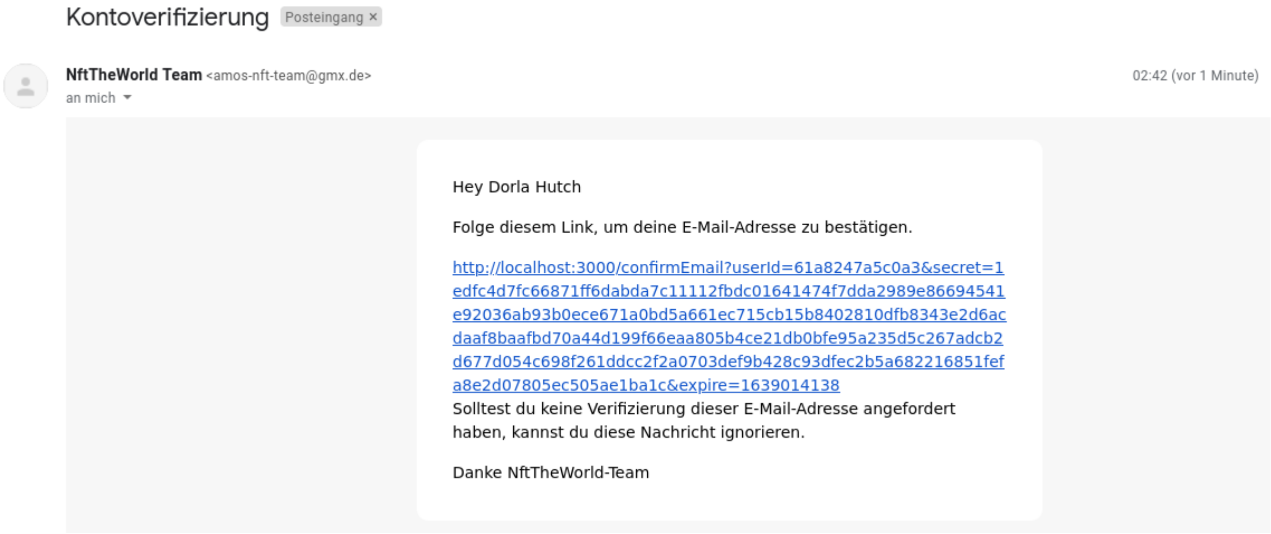

Sign up to NFT The world!

Required fields have an asterisk: *

email verification

After creating an account a email will be sent to the email specified at sign up which can be verified by clicking on the link in the received email.

The email (in German) will look something like this:


Kontoverifizierung

NftTheWorld Team <amos-nft-team@gmx.de>
an mich

02:42 (vor 1 Minute)

Hey Dorla Hutch

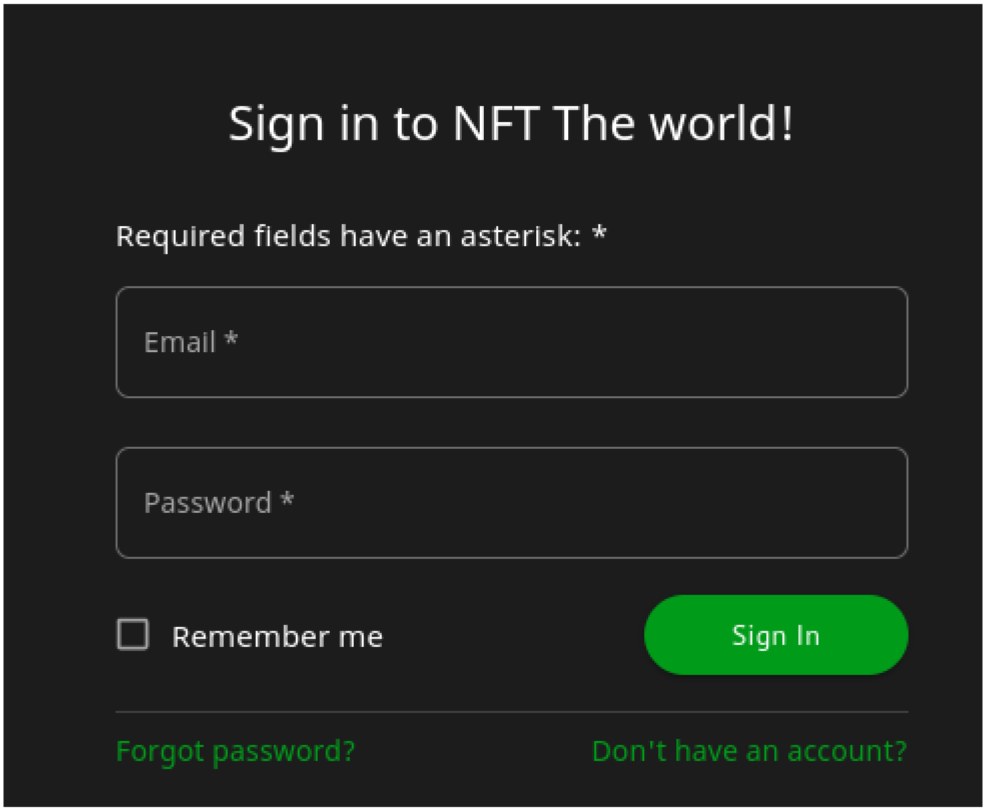
Folge diesem Link, um deine E-Mail-Adresse zu bestätigen.

http://localhost:3000/confirmEmail?userId=61a8247a5c0a3&secret=1edfc4d7fc66871ff6dabda7c11112fbdc01641474f7dda2989e86694541e92036ab93b0ece671a0bd5a661ec715cb15b8402810dfb8343e2d6acdaaf8baafbd70a44d199f66eaa805b4ce21db0bfe95a235d5c267adcb2d677d054c698f261ddcc2f2a0703def9b428c93dfec2b5a682216851fefa8e2d07805ec505ae1ba1c&expire=1639014138
Solltest du keine Verifizierung dieser E-Mail-Adresse angefordert haben, kannst du diese Nachricht ignorieren.

Danke NftTheWorld-Team

Log In

One can create into their account by accessing the login page where they need to enter their email and password.



Sign in to NFT The world!

Required fields have an asterisk: *

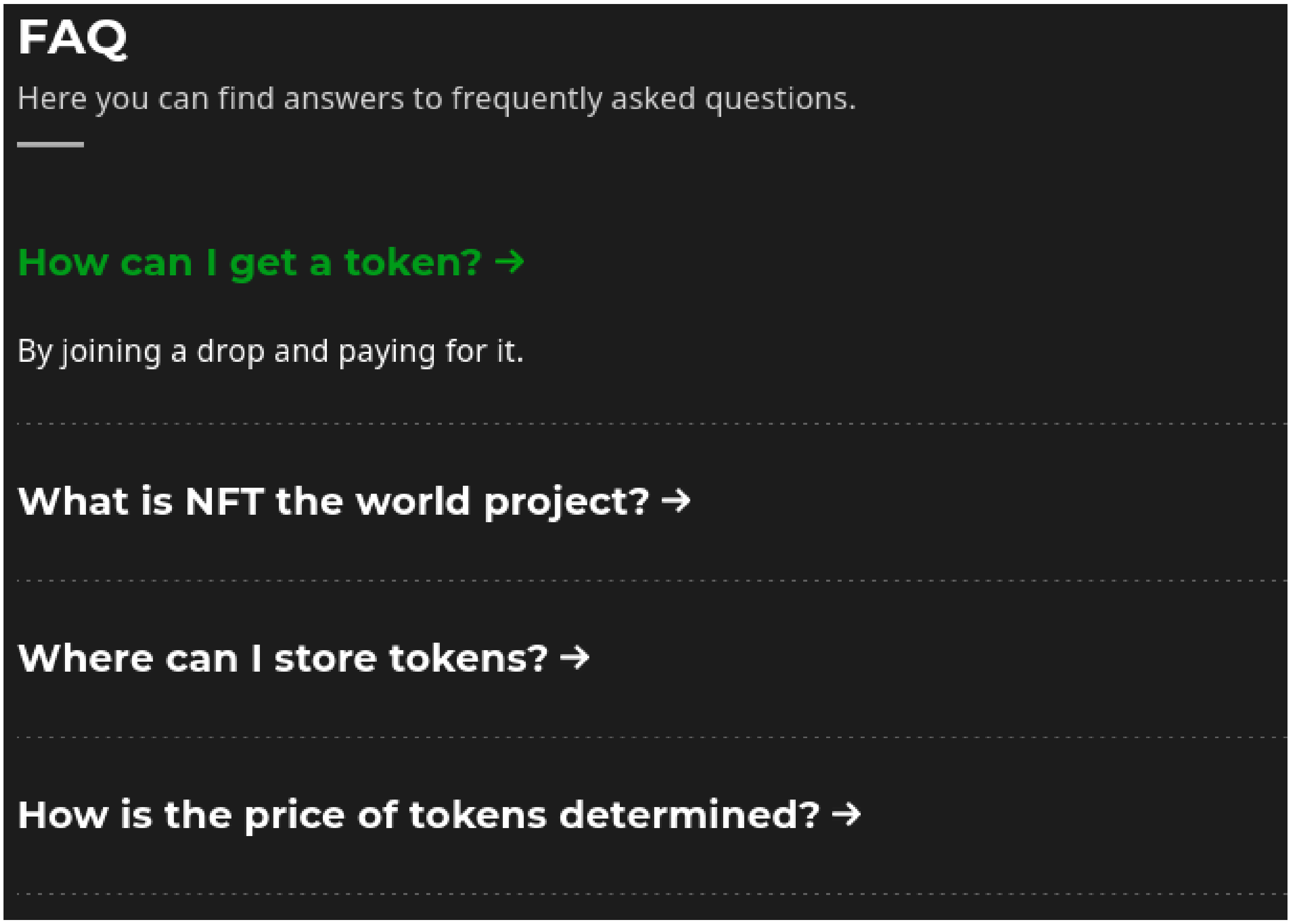
Email *

Password *

☐ Remember me

FAQ page

On the FAQ page users can get answers to frequently asked questions.



FAQ

Here you can find answers to frequently asked questions.

How can I get a token? →

By joining a drop and paying for it.

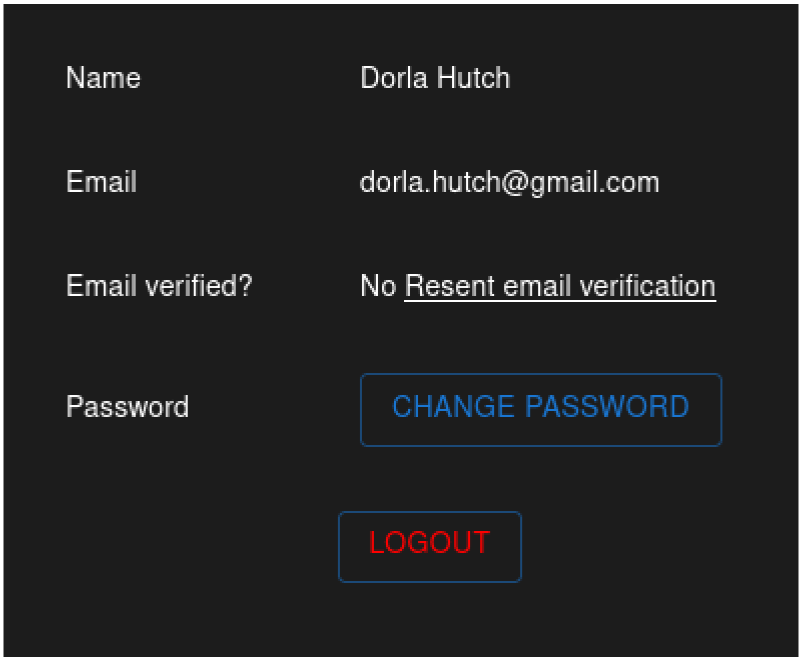
What is NFT the world project? →

Where can I store tokens? →

How is the price of tokens determined? →

User Profile

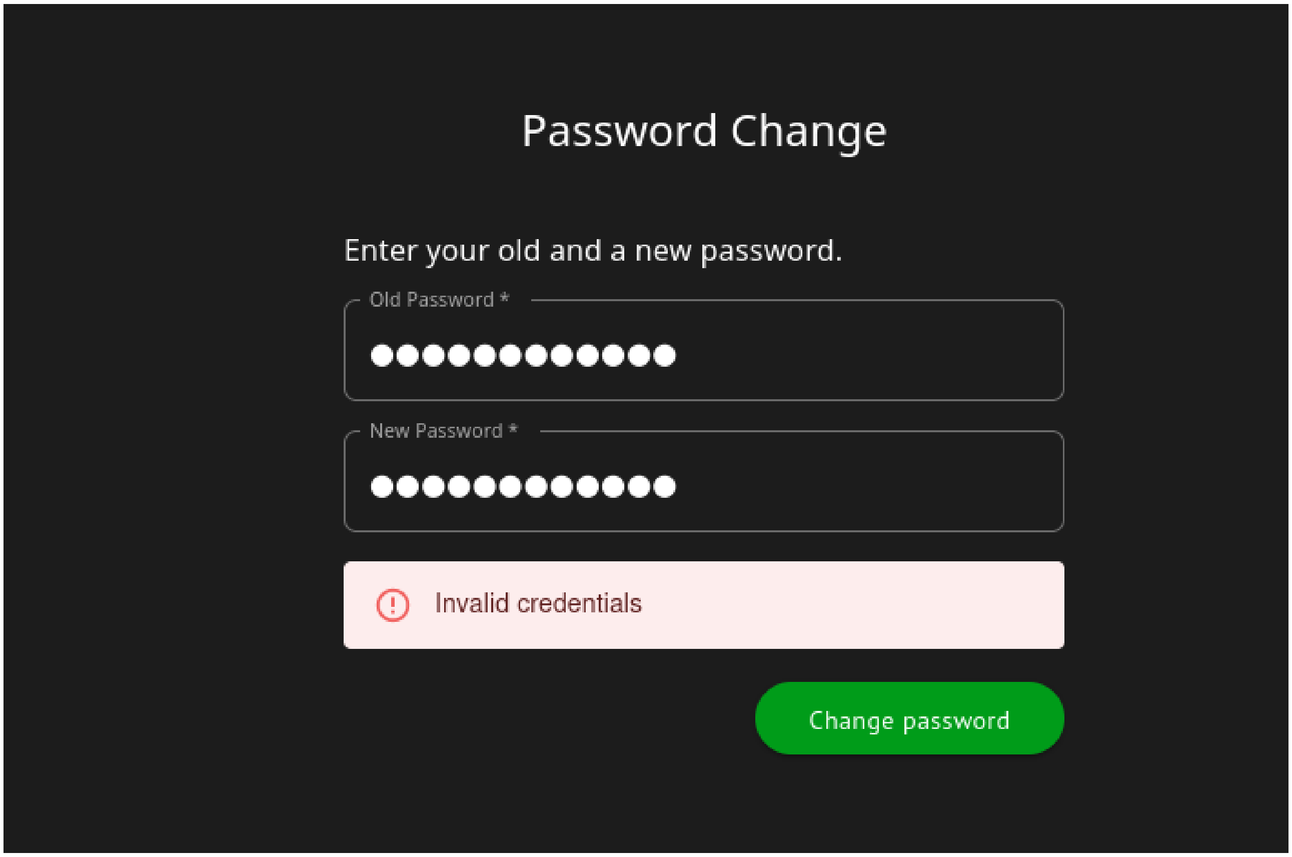
In the profile users can see basic information regarding their profile.



Name	Dorla Hutch
Email	dorla.hutch@gmail.com
Email verified?	No Resent email verification
Password	CHANGE PASSWORD

Password Change

One can change their password through the profile. On the password change page one needs to enter their old and a new password and confirm that they want to chang their password.



Password Change

Enter your old and a new password.

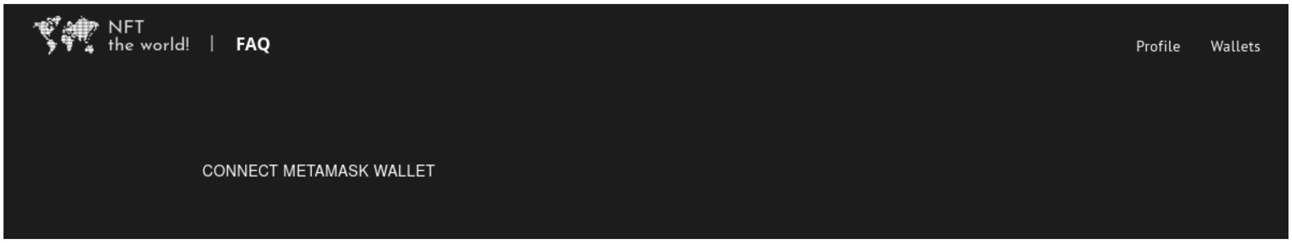
Old Password *
●●●●●●●●●●●●

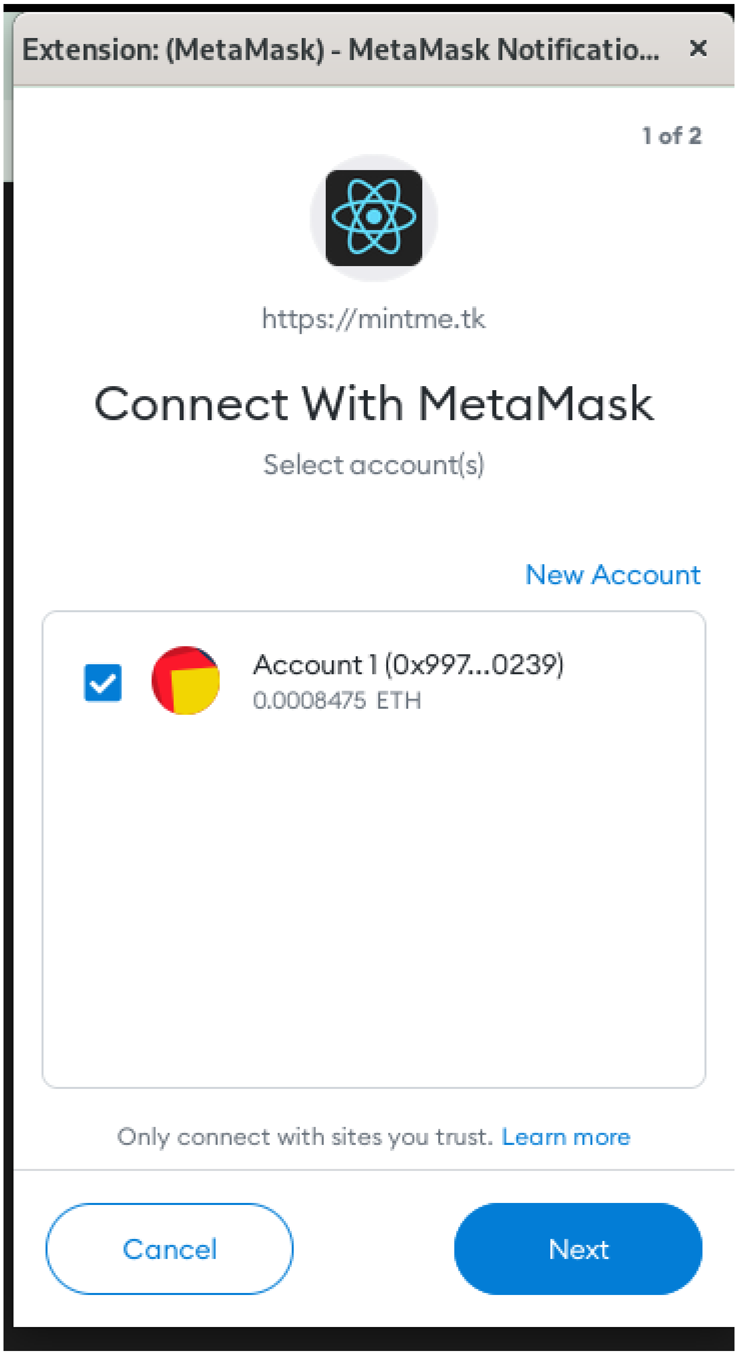
New Password *
●●●●●●●●●●●●

Invalid credentials

Connect Crypto Wallet with account

One can connect your ETH wallet by accessing the profile. Currently, only MetaMask is supported. To this end, one needs to click on the connect wallet button and confirm that they want to connect their wallet in MetaMask.





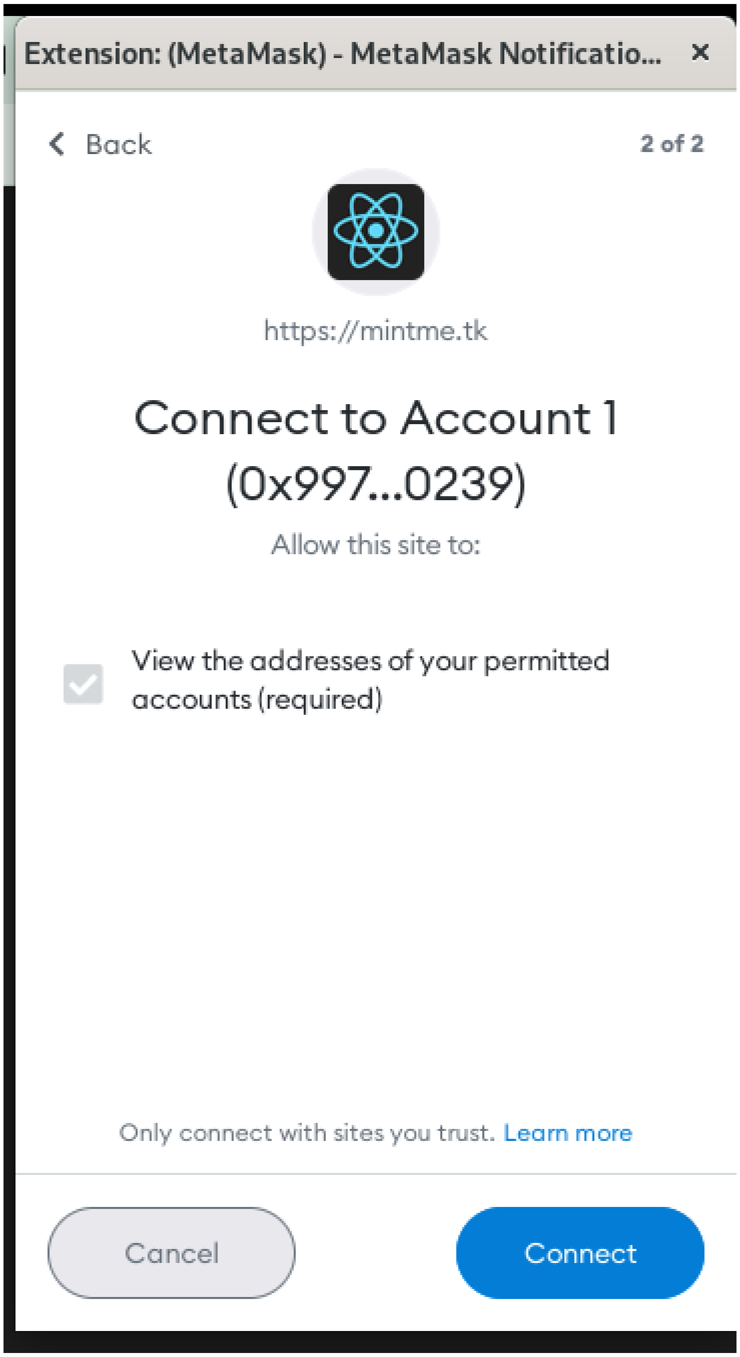
https://mintme.tk

Connect With MetaMask

Select account(s)

☑ Account 1 (0x997...0239)
0.0008475 ETH

Only connect with sites you trust. Learn more



Extension: (MetaMask) - MetaMask Notificatio... ×

< Back 2 of 2

https://mintme.tk

Connect to Account 1 (0x997...0239)

Allow this site to:

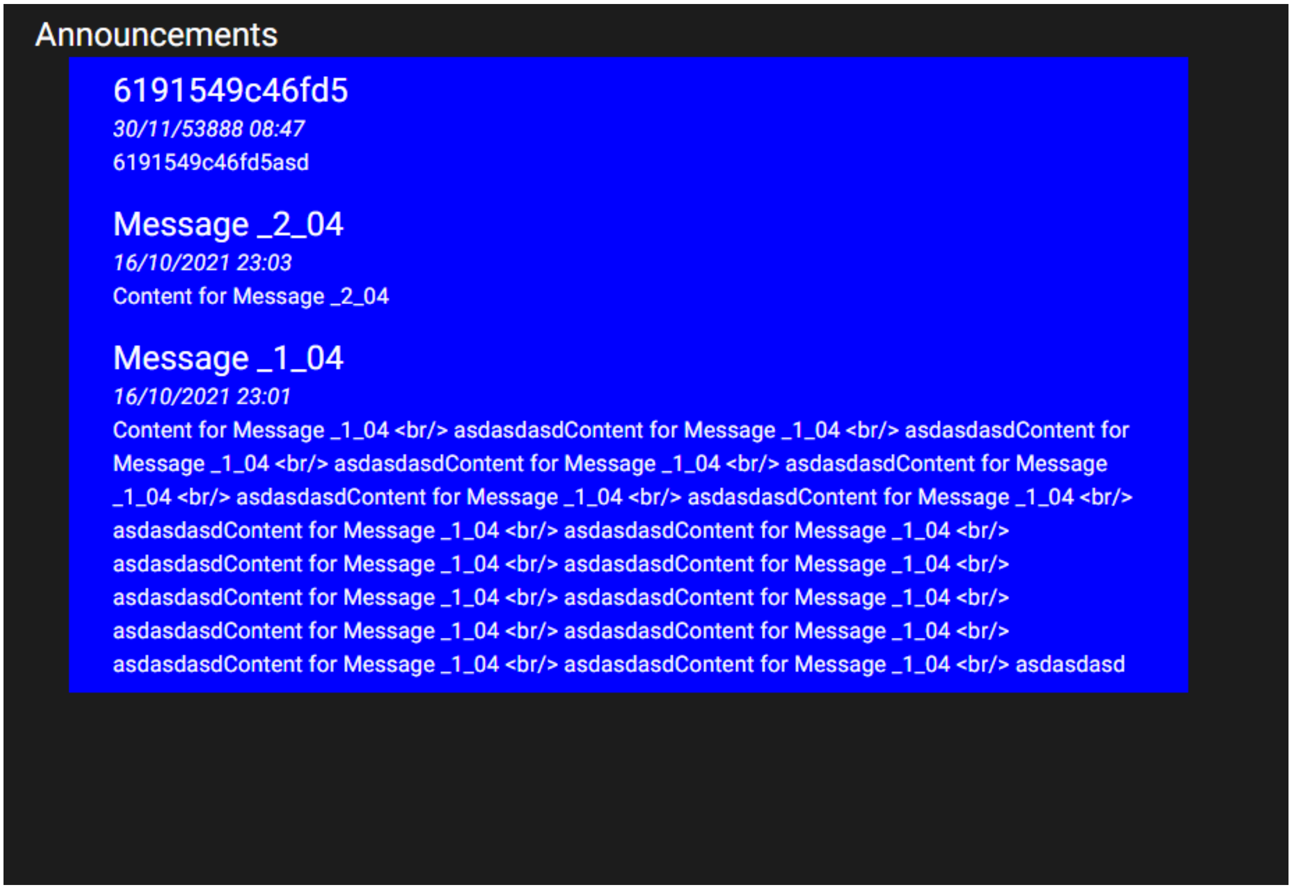
☑ View the addresses of your permitted accounts (required)

Only connect with sites you trust. Learn more

Creation and Modification of Announcements [WIP]

As an admin, announcements can be created. All other users can view announcements e.g. on the landing page. For admins, extra buttons are displayed where they can reach a webpage for editing announcements.

Admin view:



Announcements

6191549c46fd5

30/11/53888 08:47

6191549c46fd5asd

Message _2_04

16/10/2021 23:03

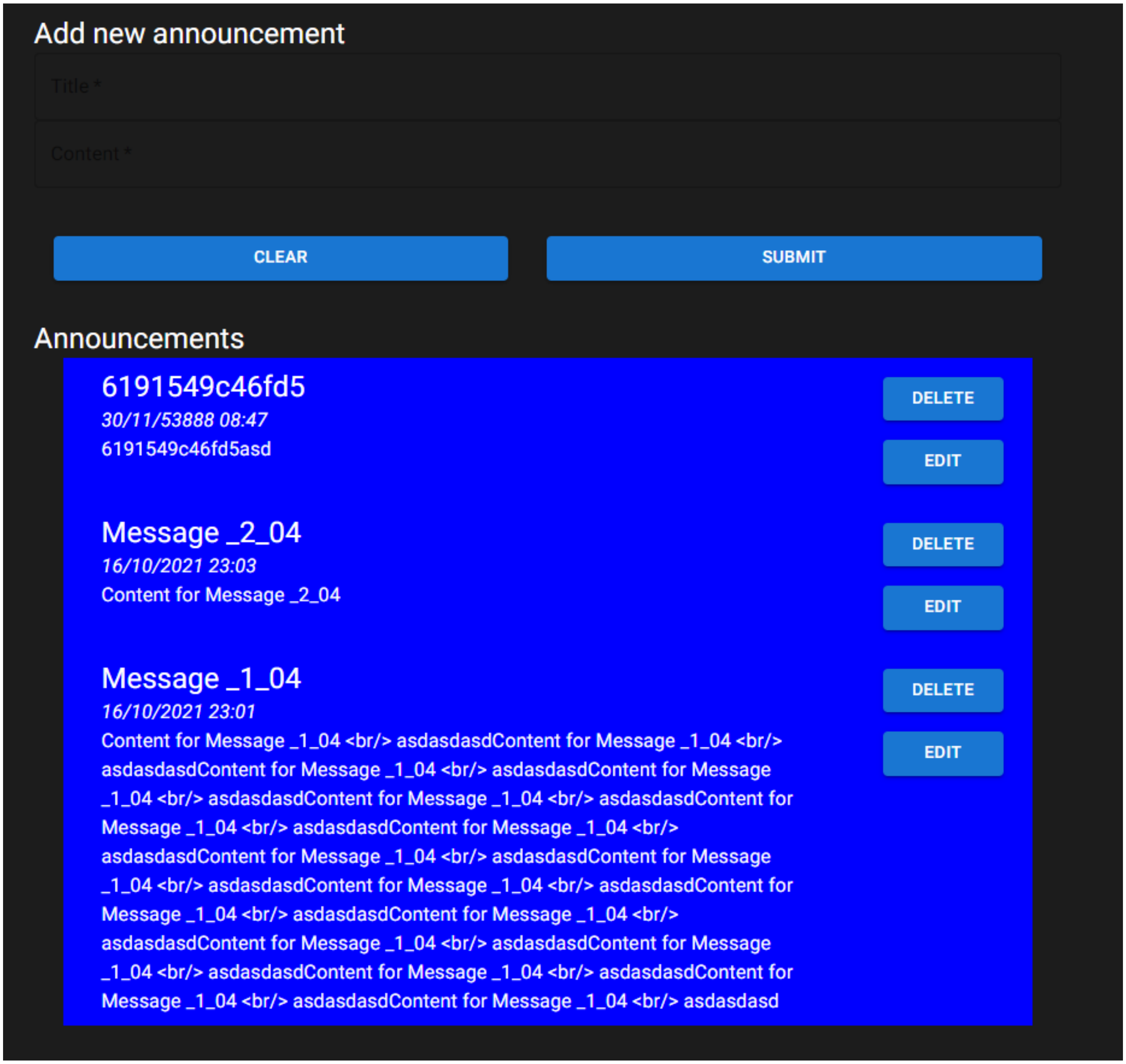
Content for Message _2_04

Message _1_04

16/10/2021 23:01

Content for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasd

User view:



6191549c46fd5
30/11/53888 08:47
6191549c46fd5asd

Message _2_04
16/10/2021 23:03
Content for Message _2_04

Message _1_04
16/10/2021 23:01
Content for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasdContent for Message _1_04 <br/> asdasdasd

Technical documentation

For more details, there are `README.md` files within some directories like `./frontend/` and `./blockchain/` .

Software architecture description

Please have a look in the Wiki for an overview over the repository artifacts.