

# 4blockz project work presentation

Mattia De Vivo

18/06/2022

Titolo: Project Work di Algorand1

Team: 4blockz

Membri: - Alessandro Farina - Mattia De Vivo - Giuseppe Cristillo - Simone Piscitelli

## Algofund

Crowdfunding platform using Algorand.

### Usage

Clone sandbox node repository: `git clone https://github.com/algorand/sandbox.git`  
Switch directory: `cd sandbox` Execute sandbox containers with private network  
with following command: `./sandbox up dev`

Go to the main directory of this project (where this README.md file is stored) and then follow the steps: - create your venv named **algofund-venv** `python -m venv algofund-venv` - activate your venv with `source algofund-venv/bin/activate` - install all the project dependencies with `pip install -r requirements.txt` - go to algofund directory `cd algofund` - create migrations `python manage.py migrate` - run the server `python manage.py runserver` - backend url is `http://127.0.0.1:8000/`

To see documentation about REST API use `swagger.yaml` file in the root directory of the project.

Web frontend built with Flutter (by backend dev) is in `algofund/web` folder: - `cd web` - `python -m http.server 8080` - go to your browser at `http://localhost:8080`

Admin info:

Admin portal at `http://127.0.0.1:8000/admin/login/?next=/admin/``  
username: admin  
password: admin

email: admin@api.com

## Development information

### Python venv

- to create a Python venv use the command `python -m venv <name-of-venv>` where `-m` indicates to Python to execute `venv` module as a script
- to activate a `venv` you need to use the `source <name-of-venv>/bin/activate` command
- to deactivate a `venv` use `deactivate` command

### Pip with environments

- after you activated the `venv` you can install all the dependencies with `pip install -r requirements.txt`
- `requirements.txt` is the file containing all the project dependencies, you can generate the file by running the command `pip freeze > requirements.txt`

## Django

### Guide

- start Django project `django-admin startproject algofund`
- run the project `python manage.py runserver`
- create a new app for our API `python manage.py startapp api`
- register the new app inside the main project `algofund/setting.py`

```
INSTALLED_APPS = [  
    'api.apps.ApiConfig',  
    ...  
]
```

- migrate already existing databases `python manage.py migrate`
- create **superuser** with `python manage.py createsuperuser`
- whenever we make changes or define a new model we need to tell Django to migrate those changes `python manage.py makemigrations && python manage.py migrate`
- remember to register models you want to manage with **admin account** inside `admin.py` file

## Django REST framework

Now we want to serialize data from our database via endpoints: - install Django REST framework `pip install djangorestframework` - tell Django the REST framework inside `mysite/settings.py` has been installed

```

INSTALLED_APPS = [
    'rest_framework',
    ...
]

```

- create `serializers.py` inside `api` folder, here we need to create the serializer
- create a serializer class inside `api/views.py` file
- `router` imported from `rest_framework` is responsible of dynamically routing requests to the right resource in a dynamic way, so if we add or delete items from the db the URLs will update to match. A router works with a `viewset` to dynamically route requests.

## TEST

GET list of pools `http://localhost:8000/pools/`, example of response:

```

[
  {
    "id": 5,
    "name": "4",
    "description": "this is the 4th test",
    "applicationIndex": "dsd1237493875493274983279",
    "minAmount": 100,
    "expiryTime": "2022-06-11",
    "image": "http://127.0.0.1:8000/media/pictures/newrelic.png"
  },
  {
    "id": 6,
    "name": "6",
    "description": "6",
    "applicationIndex": "6",
    "minAmount": 100,
    "expiryTime": "2022-06-11",
    "image": "http://127.0.0.1:8000/media/pictures/firma.png"
  },
  {
    "id": 8,
    "name": "989",
    "description": "89989",
    "applicationIndex": "98978766",
    "minAmount": 100,
    "expiryTime": "2022-06-11",
    "image": "http://127.0.0.1:8000/media/pictures/Screenshot_from_2021-11-10_17-58-40.png"
  },
  {

```

```

    "id": 7,
    "name": "ezara",
    "description": "ezara",
    "applicationIndex": "123456576",
    "minAmount": 100,
    "expiryTime": "2022-06-11",
    "image": "http://127.0.0.1:8000/media/pictures/ezara.jpeg"
  },
  {
    "id": 9,
    "name": "fewfewfewfewf",
    "description": "fedwsfwfew",
    "applicationIndex": "23232543453453453452",
    "minAmount": 300,
    "expiryTime": "2022-06-23",
    "image": null
  },
  {
    "id": 2,
    "name": "test",
    "description": "The very first pool test",
    "applicationIndex": "123456789",
    "minAmount": 100,
    "expiryTime": "2022-06-11",
    "image": "http://127.0.0.1:8000/media/pictures/ezara.jpeg"
  },
  {
    "id": 10,
    "name": "test16",
    "description": "faaaaa",
    "applicationIndex": "2",
    "minAmount": 200,
    "expiryTime": "2022-06-23",
    "image": null
  },
  {
    "id": 3,
    "name": "test2",
    "description": "the second test",
    "applicationIndex": "987654321",
    "minAmount": 100,
    "expiryTime": "2022-06-11",
    "image": "http://127.0.0.1:8000/media/pictures/ezara.jpeg"
  },
  {
    "id": 11,

```

```

        "name": "testdatetime",
        "description": "Test datetime",
        "applicationIndex": "3",
        "minAmount": 300,
        "expiryTime": "2022-07-24",
        "image": null
    },
    {
        "id": 4,
        "name": "third test",
        "description": "this is the third test",
        "applicationIndex": "1237493875493274983279",
        "minAmount": 100,
        "expiryTime": "2022-06-11",
        "image": "http://127.0.0.1:8000/media/pictures/cic.png"
    }
]

```

GET data of a single pool by primary key (id) <http://localhost:8000/pools/<id>>:

```

{
    "id": 11,
    "name": "testdatetime",
    "description": "Test datetime",
    "applicationIndex": "3",
    "minAmount": 300,
    "expiryTime": "2022-07-24",
    "image": "http://127.0.0.1:8000/media/pictures/cic.png"
}

```

POST pool <http://localhost:8000/pools/> with body:

```

{
    "name": "test33",
    "description": "JOE DESCRIPTION",
    "minAmount": 300,
    "expiryTime": "2022-06-23",
    "creatorMnemonic": "fine april congress twelve cave welcome slogan salt nice domain camp",
    "image": "base64encodedimage"
}

```

POST withdraw funds from pool <http://localhost:8000/pools/<id>/withdraw> with body:

```

{
    "requesterMnemonic": "fine april congress twelve cave welcome slogan salt nice domain ca"
}

```

POST fund on pool <http://localhost:8000/pools/<id>/funds> with body:

```
{
  "senderMnemonic": "oblige later shift bless able draw journey behave offer fox easily po
  "amount": "100000"
}
```

GET all the addresses <http://localhost:8000/addresses/>, example of response:

```
[
  {
    "sk": "troCXrBuEuV/ub6rPJBB8LQg9V045fZOR098wu09G4JKnAKq/RnUGVeim1AGa8bRIT6w8DZn5tbzt
    "addr": "JKOAFKX5DHKBSV5CTNIAM26G2EQT5MHQGZT6NVXTW7WQTDNXMIMDJF2Z4U",
    "mnemonic": "fine april congress twelve cave welcome slogan salt nice domain camp ex
  },
  {
    "sk": "wVSfi3shgAkNbxTLFBeLjgrDx+MMiWHVXAdm07K1Vlu0V2QjI5WSOPPpc8SBHCsutuOFiaHAf78r
    "addr": "WRLWIIZDSWJDR47JOPCICHBLF23OHBMJUHAH7PZLSHPZX4P673XQ76G3YA",
    "mnemonic": "oblige later shift bless able draw journey behave offer fox easily pot
  },
  {
    "sk": "GUCj+1JEyYpPknHpHp+9I9vswFXxplwC7w5XfMK4XlHbSa984raq3ddu9CNgW275MDQoscUcknCG
    "addr": "3NE267HCW2VN3V306QRWAW307EYDIKFRYU0JE4EGTZKEZE05DELE2MG4WE",
    "mnemonic": "add minimum lava early myth tired mystery immense tag ladder tell rate
  }
]
```

## Algorand sandbox commands

```
./sandbox up dev # start Algorand containers with dev data
./sandbox goal account list
./sandbox goal account balance -a <wallet-address> # show balance of person/contract
./sandbox goal app read --app-id <smartContractApplicationIndex> -d data
./sandbox goal account export -a <wallet-address> # export mnemonic
```