

Использование блокчейна и концепта децентрализованной автономной организации для управления организацией

Дмитрий Гончаров

План работ на прошедший семестр

Внедрить реальную систему генерации NFT в DAO

Разработать концептуальную часть организации

Увеличить функционал приложения и организации

Как будет работать приложение

thirdweb для генерации контракта NFT

Quicknode для бродкастинга

Goerli - тестовая сеть Ethernet

Инициализация SDK

```
1 import { ThirdwebSDK } from "@thirdweb-dev/sdk";
2 import ethers from "ethers";
3 import dotenv from "dotenv";
4
5 dotenv.config();
6
7 if (!process.env.PRIVATE_KEY || process.env.PRIVATE_KEY === "") {
8   console.log("Private key not found");
9 }
10
11 if (!process.env.QUICKNODE_API_URL || process.env.QUICKNODE_API_URL == "") {
12   console.log("Quicknode API URL not found");
13 }
14
15 if (!process.env.WALLET_ADDRESS || process.env.WALLET_ADDRESS === "") {
16   console.log("Wallet Address not found");
17 }
18
19 const provider = new ethers.providers.JsonRpcProvider(process.env.QUICKNODE_API_URL);
20 const wallet = new ethers.Wallet(process.env.PRIVATE_KEY, provider);
21 const sdk = new ThirdwebSDK(wallet);
22
23 (async () => {
24   try {
25     const address = await sdk.getSigner().getAddress();
26     console.log("👋 SDK initialized by address:", address);
27   } catch (error) {
28     console.error("Failed to get apps from the SDK", error);
29     process.exit(1);
30   }
31 })();
32
33 export default sdk;
```

Деплой контракта

```
1 import { AddressZero } from "@ethersproject/constants";
2 import sdk from "../1-initialize-sdk.js";
3 import { readFileSync } from "fs";
4
5 (async () => {
6   try {
7     const editionDropAddress = await sdk.deployer.deployEditionDrop({
8       name: "ThisDAO Membership",
9       description: "A DAO for people who read this",
10      image: readFileSync("scripts/assets/DAOlogo.svg"),
11      primary_sale_recipient: AddressZero,
12    });
13
14    const editionDrop = sdk.getEditionDrop(editionDropAddress);
15    const metadata = await editionDrop.metadata.get();
16
17    console.log("✅ Successfully deployed editionDrop contract, address:", editionDropAddress);
18    console.log("✅ editionDrop metadata:", metadata);
19  } catch (error) {
20    console.log("Failed to deploy editionDrop contract", error);
21  }
22})();
```

```
1 import sdk from './1-initialize-sdk.js';
2 import { readFileSync } from 'fs';
3
4 const editionDrop = sdk.getEditionDrop('0x66916F4eB59c762eD10582e33dF94E3376bF7439');
5
6 (async () => {
7   try {
8     await editionDrop.createBatch([
9       {
10         name: 'ThisDAO Membership',
11         description: 'This NFT grants you membership in ThisDAO',
12         image: readFileSync('scripts/assets/DAOLogo.svg'),
13       }
14     ]);
15     console.log('✅ Successfully created a new NFT in the drop');
16   } catch (error) {
17     console.error('Failed to create NFT:', error);
18   }
19 })();
```

Генерация NFT

```
1 import sdk from './1-initialize-sdk.js';
2 import { MaxUint256 } from '@ethersproject/constants';
3
4 const editionDrop = sdk.getEditionDrop('0x66916F4eB59c762eD10582e33dF94E3376bF7439');
5
6 (async () => {
7   try {
8     const claimConditions = [{
9       startTime: new Date(),
10       maxQuantity: 50_000,
11       price: 0,
12       quantityLimitPerTransaction: 1,
13       waitInSeconds: MaxUint256,
14     }]
15
16     await editionDrop.claimConditions.set('0', claimConditions);
17     console.log('✅ Successfully set claim condition');
18   } catch (error) {
19     console.error('Failed to claim conditions:', error);
20   }
21 })();
```

Установка условий

Что дальше

airdrop токенов

voting-механизм

роли

Использование блокчейна и концепта децентрализованной автономной организации для управления организацией

Дмитрий Гончаров