

# TOC source sample and description

## 1. General Information

### 1.1. Blockchain description, distinctive features

The TouchCon is a Blockchain project driven by advertising scan compensation and is a platform that builds public trust in mining and Airdrop through decentralization and supports ecosystem. In particular, it provides a unique Blockchain for building smart contract functions and Distributed Ledger Technology of advertising scan compensation.

TouchCon's Blockchain framework supports SQC (Smart QR Code) Blockchain system, and TOC Ad Scan application can customize different cryptocurrency mining and Airdrop.

By providing cooperation between various Blockchains with different mining and Airdrop structures, it is possible to increase transparency and reliability through advertisement scanning, and to provide Big data service by matching the user information and SQC data, thus creating a decentralized infrastructure that contributes to ecosystem stability. The TouchCon uses a separate TOC coin model and is a cryptocurrency based on the Ad Scan compensation driven by PoW and Re-mining, which operates with the conventional consensus algorithm.

Since launching the Mainnet in June 2018, TouchCon has launched its own explorer and pursued its infrastructure expansion through TAA (TouchCon Advertiser Alliance), TJS (TouchCon Jury System) and AMRP (Advertising Marketing Reward Program). We will pursue the expandable payment ecosystem by expanding the Scan Platform and gradually transforming to DPoS.

The portion of the total TOC that is distributed is 9.62%, which is the portion that is not Lock-Up. Tokens assigned to communities were distributed without Lock-Up. The following is the distribution structure of the TOC, and the portion described in the parentheses is the Circulating Supply.

- TouchCon Community : 13%(8.62%)
- Global Business : 1% (1%)
- TouchCon Foundation : 5% (0%)
- Development Team : 1% (0%)
- PoW Mining : 10%(0%)
- Ad Scan Reward : 70% (0%)

\*Twitter: <https://twitter.com/TouchconInfo>

\*Coin Burn: 4.38%(2018.12.17. 21:50)

<http://www.touchcon.org/30/?q=YToyOntzOjEyOiJrZXI3b3JkX3R5cGUiO3M6MzoiYWxsIjtzOjQ6InBhZ2UiO2k6Mjt9&bmode=view&idx=1451725&t=board>

터치콘은 광고 스캔보상으로 구동되는 블록체인 프로젝트이며, 채굴 및 에어드랍 분산화 신뢰를 구축하는 생태계 지원 플랫폼입니다. 특히 광고 스캔보상에 대한 완전한 분산원장과 스마트 컨트랙트 기능을 구축할 수 있는 독자적인 블록체인을 제공합니다. 터치콘 블록체인 프레임워크는 SQC(Smart QR Code) 블록체인 시스템을 지원하고, TOC Ad Scan 애플리케이션을 통해 서로 다른 암호화폐의 채굴과 에어드랍을 커스터마이징할 수 있습니다.

서로 다른 채굴 및 에어드랍 구조를 가진 다양한 블록체인간의 협력을 서비스하면, 분산 광고스캔을 통해 투명성과 신뢰성 증가, User 및 SQC 데이터 매칭에 의한 빅데이터 서비스와 함께 생태계 안정에 기여하는 분산화 인프라가 구축됩니다. 터치콘은 독립된 TOC 코인 모델을 사용하며, 전통적인 합의 알고리즘으로 작동하는 PoW 와 Re-mining 으로 구동시키는 Ad Scan Reward 를 기반으로 하는 암호화폐입니다.

터치콘은 2018 년 6 월 메인넷 출시이후 독자적인 익스플로어를 구동시키며, TAA(터치콘광고주동맹), TJS(터치콘배심원단), AMRP(광고마케팅보상프로그램)를 통해 인프라 확산을 추구하고, Ad Scan Platform 의 확장과 DPoS 로의 점진적 전환을 통해 확장형 지불결제 생태계를 추구해 나갑니다.

전체 TOC 중 유통되는 비중은 9.62%으로 락업되지 않은 부분에 해당합니다. 커뮤니티에 할당되는 토큰은 락업 없이 배포되었습니다. 다음은 TOC 의 분배 구조로, 괄호 안은 시장 유통량 (Circulating Supply)입니다.

## 1.2. All available source code repositories

- <https://github.com/touchconDev/go-touchcon>

## 1.3. For blockchain forks - Original project name and its version or tag

- go-ethereum version : 1.8.3

## 1.4. Recommendations for integration

# 2. Deposits

## 2.1. Dockerfile for building and running an RPC node (Base system Ubuntu 18.04)

- Not supported (Coming soon)

## 2.2. Dockerfile with an application that syncs blockchain into a database

- Not supported (Coming soon)

## 2.3. Code sample for address generation (private key and derived address)

-- code sample --

```
const wallet = require('ethereumjs-wallet');
const myWallet = wallet.generate();

console.log('privateKey :: ' + myWallet.getPrivateKeyString());
console.log('address :: ' + myWallet.getAddressString());
```

## 3. Withdrawals:

### 3.1. Code samples (node-js) for

#### 3.1.1. Preparing raw transaction data (gathering required transaction parameters from the blockchain)

```
const Web3 = require('web3');
const web3 = new Web3(new Web3.providers.HttpProvider('http://localhost:8545'));

// variable nonce, gasPrice is the one of the data that will be used in raw transaction
let nonce;
let gasPrice;

//parameter address is value of from address that data will be used to get nonce variable
web3.eth.getTransactionCount(address).then( result => {
    nonce = result;
}).catch( e => {
    throw e;
});

web3.eth.getGasPrice().then( result => {
    gasPrice = result;
}).catch( e => {
    throw e;
})
```

#### 3.1.2. Generation raw transaction (combining data into a data structure, serializing into a raw transaction)

```
const tx = require('ethereumjs-tx');

const txData = {
  nonce : String,      // get nonce used by toc core
  gasPrice : String,    // gasPrice
  gasLimit : String,    // gasLimit
  to : String,          // from address
  value : String        // value of send amount
};
```

### 3.1.3. Signing and broadcasting raw transaction

```
const tx = require('ethereumjs-tx');

// sample object
const txData = {
  nonce : '0x1',
  gasPrice : '0xba43b7400',
  gasLimit : '0x61a80',
  to : 'toAddress',
  value : '0x1'
};

const rawTx = new tx(txData);

let pk = "from address privateKey";
let privKey = Buffer.from(pk, 'hex');

rawTx.sign(privKey);    // signing transaction

let serializedTx = tx.serialize();
let rawTxData = '0x'+serializedTx.toString('hex');

console.log(rawTxData);
```

### 3.1.4. Getting list of transactions by user address

```
const mongoose = require('mongoose');
const Transaction = mongoose.model('Transaction');

let transactionList = Transaction.find( { from : ( user address } } ).lean(true).sort('-blockNumber')
.then( docs => {
  return docs;
}).catch( e => {
  throw e;
});

console.log(transactionList);
```