

---

# OPALS

THE DEFI TOKEN





Opals - A TRC20 token in the Tron Block Chain with 32,000,000 (Thirty Two Million) maximum supply.

# TRON

Tron launched in 2017 is a blockchain similar to Ethereum. But it is much faster and the transaction fee is very lesser compared to the Ethereum.

In 2018, TRON switched its protocol from an ERC-20 token on top of Ethereum to an independent peer-to-peer network. After that, marketed to rival Ethereum, TRON boomed in market capitalization during the early part of 2018, becoming a top 10 cryptocurrency. The market cap of Tron is \$1,145,333,571 as on 1 Sep, 2020.

Like Ethereum, one can deploy smart contracts on the Tron Blockchain and use it for any purpose.

# TRC20 TOKEN

TRC20 (TRC20 Token Standard) is a token standard based on the implementation of smart contract when using TRON network to issue a token. In order to swap, transfer, share token and be supported by digital wallets, all the TRC20 tokens follow a series of specifications which include 3 optional items and 6 required items.

## 3 Optional Items:

### 1) Token Name

string public constant name = "OPALS";

### 2) Token Abbreviation

For example, OPL is the abbreviation of OPALS. OPALS and OPL represent the same token, but OPL is more simple.

string public constant symbol = "OPL";

### 3) Token Precision

Token precision is the minimum divisible unit. Precision 0 means the minimum divisible unit is 1. Precision 2 means the minimum divisible unit is 0.01. The maximum value of precision is 18.

uint8 public constant decimals = 6;

## 6 Required Items:

The code below is a brief TRC20 smart contract:

```
contract TRC20 {  
function totalSupply() constant returns (uint theTotalSupply);  
function balanceOf(address _owner) constant returns (uint balance);  
function transfer(address _to, uint _value) returns (bool success);  
function transferFrom(address _from, address _to, uint _value) returns (bool success);  
function approve(address _spender, uint _value) returns (bool success);  
function allowance(address _owner, address _spender) constant returns (uint remaining);  
event Transfer(address indexed _from, address indexed _to, uint _value);  
event Approval(address indexed _owner, address indexed _spender, uint _value);  
}
```

## 1) totalSupply()

This function returns the total supply of the token.

```
contract MyTRCToken {
    uint256 _totalSupply = 1000000;
    function totalSupply() constant returns (uint256 theTotalSupply) {
        theTotalSupply = _totalSupply;
        return theTotalSupply;
    }
}
```

## 2) balanceOf()

This function returns the token balance of the specific account.

```
contract MyTRCToken {
    mapping(address => uint256) balances;
    address public owner; // Owner of this contract
    function balanceOf(address _owner) constant returns (uint256 balance) {
        return balances[_owner];
    }
}
```

### 3) transfer()

This function is used to transfer an amount of tokens from the smart contract to a specific address.

```
contract OPALS {
    mapping(address => uint256) balances;
    function transfer(address _to, uint256 _amount) returns (bool success) {
        if (balances[msg.sender] >= _amount && _amount > 0 && balances[_to] + _amount
> balances[_to]) {
            balances[msg.sender] -= _amount;
            balances[_to] += _amount;
            Transfer(msg.sender, _to, _amount); // trigger event
            return true;
        } else {
            return false;
        }
    }
}
```

#### 4) approve()

This function is used to authorize the third party (like a DAPP smart contract) to transfer token from the token owner's account.

```
contract OPALS {
    mapping(address => mapping (address => uint256)) allowed;
    function approve(address _spender, uint256 _amount) returns (bool success) {
        allowed[msg.sender][_spender] = _amount;
        // msg.sender is the third party's account
        Approval(msg.sender, _spender, _amount);
        // trigger event
        return true;
    }
}
```



## 5) transferFrom()

This function is used to allow the third party to transfer token from an owner account to a receiver account. The owner account must be approved to be called by the third party.

Compared to transfer(), what's the meaning of transferFrom()?

By using transferFrom(), the third party can transfer token from your account to another account automatically.

```
contract OPALS {
    mapping(address => uint256) balances;
    function transferFrom(address _from, address _to, uint256 _amount) returns (bool
success) {
        if (balances[_from] >= _amount && allowed[_from][msg.sender] >= _amount &&
_amount > 0 && balances[_to] + _amount > balances[_to]) {
            balances[_from] -= _amount;
            balances[_to] += _amount;
            Transfer(_from, _to, _amount);
            return true;
        } else {
            return false;
        }
    }
}
```

## 6) allowance()

This function is used to query the remaining amount of tokens the third party can transfer.

Approve() can authorize the third party to transfer token from the owner account, then through transferFrom() it can transfer token to a receiver account.

For example:

Alice owns 1000 TRX, if he wants to allow the third party contract C to manipulate 100 TRX from his account:

Call approve(C, 100) using Alice's account

Contract C wants to transfer 10 TRX to Bob from Alice. Call transferFrom(Alice, Bob, 10) using contract C's account.

Call allowance(Alice, C) to query the remaining tokens that the contract C can transfer from Alice's account.

## 2 Event Functions

```
event Transfer(address indexed _from, address indexed _to, uint256 _value)
```

When token is successfully transferred, it has to trigger Transfer Event.

```
event Approval(address indexed _owner, address indexed _spender, uint256 _value)
```

When approval() is successfully called, it has to trigger Approval Event.

# WHAT IS OPALS?

OPALS (OPL) is a TRC20 token created in the Tron Block Chain with a maximum supply of 32,000,000. [TCyIHMVFNj6exwxtZT6g2NrCWNQyLIy742](https://tronscan.org/#/contract/TCyIHMVFNj6exwxtZT6g2NrCWNQyLIy742) is the contract address of the OPL token. The pre-sale happened between Sep 3, 2020 to Oct 3, 2020. Instead of burning the unsold tokens, the unsold tokens were added to the staking smart contract. [Click here](#) to see the proof. The tokens added to the staking smart contract cannot be used for any purpose, other than the staking rewards. 18,000,000 OPL were added to the staking smart contract to pay the staking rewards. Once 60% of the 18 Million OPL is paid as rewards, staking will be disabled permanently. People who have already staked, will continue to earn their staking rewards, until they earn the maximum allowed rewards. This assures a fair reward to everyone and the contract will not run out of OPL to pay the rewards. OPL can be traded on the JustSwap at the moment. In future, OPL will be listed in other leading exchanges.

# WHAT IS THE USE CASE OF OPALS (OPL)?

- 1) Stake OPL and earn staking rewards.
- 2) Advertise on the Opals Social Network.
- 3) Advertise on the Opals Video Sharing Portal.
- 4) Play on the Opals casino.
- 5) Provide liquidity in JustSwap.
- 6) More to come...

# OPALS (OPL) & SALES DISTRIBUTION

## **OPALS (OPL) Distribution:**

Founders and Developers - 10%

Marketing & Promotions - 10%

Liquidity Reserve - 5%

Public - 75%

## **Sales Distribution:**

Founders and Developers - 10%

Marketing & Promotions - 10%

Liquidity Reserve - 15%

Referral Commission - 20%

Dividend Pool - 45%

# TRADE / STAKE OPALS (OPL)

## **Where to trade OPL?**

You can trade OPL on JustSwap (<https://www.justswap.io/>) at the moment. In future we will list OPL in other exchanges.

## **Where to stake OPL?**

Stake OPL on <https://www.opalstaking.io/> and earn staking rewards.

# PROJECTS / TEAM

## Projects

- 1) Opals Social Network: The model is similar to Facebook, but without much of restrictions and limitations.
- 2) Opals Video Sharing Portal: The model is similar to YouTube, but without much of restrictions and limitations.
- 3) Opals Casino: Addictive and Attractive provably fair casino games with a lowest house edge.

## Team:

The team is headed and founded by Arkaitz Badaya, stock market analyst and a day trader. Adam Henry - a technical administrator heads the team of 12 who handles different divisions of the project.

# ROADMAP

June 2020 - Market Research

July 2020 - Team Development

Aug 2020 - TRC20 Smart Contract Development

Sep 2020 - Presale followed by Regular Sale

Oct 2020 - Opals Staking / JustSwap Listing

Nov 2020 - Listing on PoloniDex

Dec 2020 – Opals Social Network Launch

Jan 2021 – Opals Video Sharing Portal Launch

Feb 2021 – Opals Casino Launch

March 2021 – Marketing and Promotions

April 2021 - Further development and expansion



## DISCLAIMER & RISK WARNING:

Crypto or any Financial Investment has its own risk. Crypto Market is highly volatile and have more risks than any other financial markets. Plan your investment wisely and take only the risks that are affordable to you. Even though the business model is genuine and stable, you have to understand that the market is controlled by the traders and users of OPALS and it's not controlled by the founders or administrators of OPALS. By participating in any of the activities of OPALS, you agree to take the risks and understand and accept that the founders or administrators or developers or any of the team members of OPALS are not responsible for any of the losses you may incur. You also accept and agree that the OPALS or its founders or administrators or developers or any of the team members cannot be held liable (legal or illegal) for any of the losses you may incur by participating in any of the services provide by the OPALS. OPALS and all its services are provided in as-is basis. If you do not accept this disclaimer, please do not participate in any of the services provided by OPALS.



Main Website: <https://www.opals.io/>

Staking Website: <https://www.opalstaking.io>

Telegram: [https://t.me/Opals\\_Tron\\_Defi](https://t.me/Opals_Tron_Defi)