JustSwap API Documentation

by Just team

v 1.0.0

Table of Contents

I. Backend API

1.1 Get access to API for all trading pairs

GET: https://api.justswap.io/v2/allpairs

Parameters:

page_size : int, size of each page, max 50
page_num: int, number of the page, starting from 0

Format of return value:


```
{
  "data":
    [{
    "0_TR7NHqjeKQxGTCi8q8ZY4pL8otSzgjLj6t": {//key: ID of TRX and token
    "quote_id": "0", // 0 refers to TRX's id
    "quote_name": "TRX", // only TRX is accepted for the moment
    "quote_symbol": "TRX", // only TRX is accepted for the moment
    "quote_decimal": "6", // TRX allows up to 6 decimals
    "base_id": "TR7NHqjeKQxGTCi8q8ZY4pL8otSzgjLj6t", // contract address of the token
    "base_name": "Tether USD", // name of base_token, e.g. USDT
    "base_symbol": "USDT", // symbol for token
    "base_decimal": "6", // max decimal of token
    "price": "37.450887374011593590", // price of the base_token, 1 USDT = 37.45 TRX at
the moment
    "quote_volume": "16928742252878", // total amount of TRX traded in the last 24 hours,
unit: 1 SUN
    "base_volume": "452025130508" // total amount of base_token traded in the last 24
hours, unit: min unit of the base_token
    }],
  "total_num": 25, // total entries
  "err_no":0,
  "err_msg":""
}
```


Note:

Maximum number of requests to the API is 1 request per user per second.

II. Smart Contract API

2.1 Smart contract address

Name	Address	Note
factory contract	TXk8rQSAvPvBBNtqSoY6nCfsXWCSSpTVQF	Factory contract is designed to create trading pairs and to manage lists of trading pairs
exchange contract	TQn9Y2khEsLJW1ChVWFMSMeRDow5KcbLSE	Each trading pair comes with an exchange contract. This is an exchange contract address for USDT/TRX.

2.2 List of contract interface

2.2.1 TRC20 Token

```
interface ITRC20 {
  function transfer(address to, uint256 value) external returns (bool);
  function approve(address spender, uint256 value) external returns (bool);
  function transferFrom(address from, address to, uint256 value) external returns (bool);
  function totalSupply() external view returns (uint256);
  function balanceOf(address who) external view returns (uint256);
  function allowance(address owner, address spender) external view returns (uint256);
  event Transfer(address indexed from, address indexed to, uint256 value);
  event Approval(address indexed owner, address indexed spender, uint256 value);
}
```

2.2.2 Factory

```
interface IJustswapFactory {
  event NewExchange(address indexed token, address indexed exchange);

  function initializeFactory(address template) external;
  function createExchange(address token) external returns (address payable);
  function getExchange(address token) external view returns (address payable);
  function getToken(address token) external view returns (address);
  function getTokenWihId(uint256 token_id) external view returns (address);
}
```

2.2.3 Exchange

```
interface IJustswapExchange {

  event TokenPurchase(address indexed buyer, uint256 indexed trx_sold, uint256 indexed
tokens_bought);
  event TrxPurchase(address indexed buyer, uint256 indexed tokens_sold, uint256
indexed trx_bought);
  event AddLiquidity(address indexed provider, uint256 indexed trx_amount, uint256
indexed token_amount);
  event RemoveLiquidity(address indexed provider, uint256 indexed trx_amount, uint256
indexed token_amount);

  function () external payable;

  function getInputPrice(uint256 input_amount, uint256 input_reserve, uint256
output_reserve) external view returns (uint256);

  function getOutputPrice(uint256 output_amount, uint256 input_reserve, uint256
output_reserve) external view returns (uint256);

  function trxToTokenSwapInput(uint256 min_tokens, uint256 deadline) external payable
returns (uint256);

  function trxToTokenTransferInput(uint256 min_tokens, uint256 deadline, address
recipient) external payable returns(uint256);

  function trxToTokenSwapOutput(uint256 tokens_bought, uint256 deadline) external
payable returns(uint256);

  function trxToTokenTransferOutput(uint256 tokens_bought, uint256 deadline, address
recipient) external payable returns (uint256);

  function tokenToTrxSwapInput(uint256 tokens_sold, uint256 min_trx, uint256 deadline)
```

```
external returns (uint256);

  function tokenToTrxTransferInput(uint256 tokens_sold, uint256 min_trx, uint256
deadline, address recipient) external returns (uint256);

  function tokenToTrxSwapOutput(uint256 trx_bought, uint256 max_tokens, uint256
deadline) external returns (uint256);

  function tokenToTrxTransferOutput(uint256 trx_bought, uint256 max_tokens, uint256
deadline, address recipient) external returns (uint256);

  function tokenToTokenSwapInput(uint256 tokens_sold, uint256 min_tokens_bought,
uint256 min_trx_bought, uint256 deadline, address token_addr) external returns (uint256);

  function tokenToTokenTransferInput(uint256 tokens_sold, uint256 min_tokens_bought,
uint256 min_trx_bought, uint256 deadline, address recipient, address token_addr) external
returns (uint256);

  function tokenToTokenSwapOutput(uint256 tokens_bought, uint256 max_tokens_sold,
uint256 max_trx_sold, uint256 deadline, address token_addr) external returns (uint256);

  function tokenToTokenTransferOutput(uint256 tokens_bought, uint256 max_tokens_sold,
uint256 max_trx_sold, uint256 deadline, address recipient, address token_addr) external
returns (uint256);

  function tokenToExchangeSwapInput(uint256 tokens_sold, uint256 min_tokens_bought,
uint256 min_trx_bought, uint256 deadline, address exchange_addr) external returns
(uint256);

  function tokenToExchangeTransferInput(uint256 tokens_sold, uint256
min_tokens_bought, uint256 min_trx_bought, uint256 deadline, address recipient, address
exchange_addr) external returns (uint256);

  function tokenToExchangeSwapOutput(uint256 tokens_bought, uint256
max_tokens_sold, uint256 max_trx_sold, uint256 deadline, address exchange_addr)
external returns (uint256);

  function tokenToExchangeTransferOutput(uint256 tokens_bought, uint256
max_tokens_sold, uint256 max_trx_sold, uint256 deadline, address recipient, address
exchange_addr) external returns (uint256);

  function getTrxToTokenInputPrice(uint256 trx_sold) external view returns (uint256);

  function getTrxToTokenOutputPrice(uint256 tokens_bought) external view returns
(uint256);

  function getTokenToTrxInputPrice(uint256 tokens_sold) external view returns (uint256);

  function getTokenToTrxOutputPrice(uint256 trx_bought) external view returns (uint256);
```

```
  function tokenAddress() external view returns (address);

  function factoryAddress() external view returns (address);

  function addLiquidity(uint256 min_liquidity, uint256 max_tokens, uint256 deadline)
external payable returns (uint256);

  function removeLiquidity(uint256 amount, uint256 min_trx, uint256 min_tokens, uint256
deadline) external returns (uint256, uint256);
}
```

2.3 Detail description of contract interface

2.3.1 Factory

2.3.1.1 Query interface

1. getExchange

function getExchange(address token) external view returns (address payable);

Function description: Use TRC20 token to obtain the corresponding exchange address in JustSwap.

Parameter description:

Parameter	Type	Description
token	address	TRC20 token address

Returns:

address	Trading pair address in JustSwap

2. getToken

function getToken(address token) external view returns (address)

Function description: Use the exchange address in JustSwap to obtain the address of TRC20 token.

Parameter description:

Parameter	Type	Description
token	address	exchange address in JustSwap

Returns:

address	TRC20 token address

2.3.1.2 Modification interface

1. createExchange

function createExchange(address token) external returns (address payable);

Function description: create JustSwap trading pair.Each TRC20 token creates only one exchange address.

Parameter description:

Parameter	Type	Description
token	address	TRC20 token address

Returns:

address	Trading pair address in JustSwap

2.3.1.3 Contract Event

1. NewExchange

event NewExchange(address indexed token, address indexed exchange);

Function description: The interface sends an event when creating trading pair with createExchange.

Parameter description:

Parameter	Type	Description
token	address	TRC20 token address
exchange	address	TRC20 token's corresponding exchange address in JustSwap

2.3.2 Exchange

2.3.2.1 Query interface

1. getTrxToTokenInputPrice

function getTrxToTokenInputPrice(uint256 trx_sold) external view returns (uint256);

Function description: To know the amount of TRC20 token available for purchase through the amount of TRX sold

Parameter description:

Parameter	Type	Description
trx_sold	uint256	amount of TRC sold

Returns:

uint256	amount of TRC20 token available to purchase

2. getTrxToTokenOutputPrice

function getTrxToTokenOutputPrice(uint256 tokens_bought) external view returns (uint256);

Function description: To know the amount of TRX to be paid through the amount of TRC20 token bought

Parameter description:

Parameter	Type	Description
tokens_bought	uint256	Amount of TRC20 token bought

Returns:

uint256	Amount of TRX to be paid

3. getTokenToTrxInputPrice

function getTokenToTrxInputPrice(uint256 tokens_sold) external view returns (uint256);

Function description: To know the amount of TRX available for purchase through the amount of TRC20 token sold

Parameter description:

Parameter	Type	Description
tokens_sold	uint256	amount of TRC20 token sold

Returns:

uint256	amount of TRX available for purchase

4. getTokenToTrxOutputPrice

function getTokenToTrxOutputPrice(uint256 trx_bought) external view returns (uint256);

Function description: To know the amount of TRC20 token to be paid through the amount of TRX bought

Parameter description:

Parameter	Type	Description
trx_bought	uint256	amount of TRX bought

Returns:

uint256	Amount of TRC20 token to be paid

5. tokenAddress

function tokenAddress() external view returns (address);

Function description: Obtain the address of TRC20 tokens available for trade within this trading pair in JustSwap

Parameter description: N/A

Returns:

address	address of TRC20 token available to be traded in this trading pair

6. factoryAddress

function factoryAddress() external view returns (address);

Function description: Access the Factory contract address that creates the JustSwap trading pairs

Parameter description: N/A

Returns:

address	factory contract address for JustSwap

2.3.2.2 Modification interface

1. addLiquidity

function addLiquidity(uint256 min_liquidity, uint256 max_tokens, uint256 deadline) external payable returns (uint256);

Parameter description:
Add liquidity to the current exchange.Deduct callvalue and corresponding token and add them into contracts.Then mint liquidity token as marker.
Approve before adding liquidity.

Parameter description:

Parameter	Type	Description
min_liquidity	uint256	Minimum additional issuance of liquidity token
max_tokens	uint256	Maximum additional amount of TRC20 token
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit
msg.value	uint256	Amount of TRX deposited when adding liquidity

Returns:

uint256	Additional amount of liquidity token issued to callers

Note:
1). msg.value parameter is the callvalue that comes with the function call (namely amount of TRX) and is not included on the parameter list of function signature. Other “msg.value” below are defined as the same.

2. removeLiquidity

function removeLiquidity(uint256 amount, uint256 min_trx, uint256 min_tokens, uint256 deadline) external returns (uint256, uint256)

Function description: remove liquidity; transfer the amount of TRX and TRC20 token removed to callers.

Parameter description:

Parameter	Type	Description
amount	uint256	Amount of liquidity token removed
min_trx	uint256	Minimum amount of TRX removed
min_tokens	uint256	Minimum amount of TRC20 token removed
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit

Returns:

uint256	Amount of TRX removed
uint256	Amount of TRC20 tokens removed

3. trxToTokenSwapInput

function trxToTokenSwapInput(uint256 min_tokens, uint256 deadline) external payable returns (uint256);

Function description: sell TRX (fixed amount) to buy token

Parameter description:

Parameter	Type	Description
min_tokens	uint256	Minimum amount of token needed; calculated from the slippage selected by the user
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit
msg.value	uint256	Amount of TRX the user pays for buying token

Returns:

uint256	Amount of TRC20 token purchased

4. trxToTokenSwapOutput

function trxToTokenSwapOutput(uint256 tokens_bought, uint256 deadline) external payable returns(uint256);

Function description: sell TRX to buy token (fixed amount)

Parameter description:

Parameter	Type	Description
tokens_bought	uint256	Amount of token purchased (fixed amount)
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit
msg.value	uint256	Maximum amount that can be paid to buy the fixed amount of token; calculated from the slippage

Returns:

uint256	Amount of TRX sold

5. tokenToTrxSwapOutput

function tokenToTrxSwapOutput(uint256 trx_bought, uint256 max_tokens, uint256 deadline) external returns (uint256)

Function description: sell token to buy TRX (fixed amount)

Parameter description:

Parameter	Type	Description
trx_bought	uint256	Amount of TRX purchased (fixed amount)
max_tokens	uint256	Maximum amount of token that can be deducted from the user account; calculated from the slippage
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit

Returns:

uint256	Amount of TRC20 token sold

6. tokenToTrxSwapInput

function tokenToTrxSwapInput(uint256 tokens_sold, uint256 min_trx, uint256 deadline) external returns (uint256);

Function description: sell token to buy TRX (token is in a fixed amount)

Parameter description:

Parameter	Type	Description
tokens_sold	uint256	Amount of token sold (fixed amount)
min_trx	uint256	Minimum amount of TRX needed, which is calculated from the slippage
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit

Returns:

uint256	Amount of TRX purchased

7. trxToTokenTransferInput

function trxToTokenTransferInput(uint256 min_tokens, uint256 deadline, address recipient) external payable returns(uint256);

Function description: sell TRX to buy token (TRX is in a fixed amount)Then, transfer the purchased token to the recipient address.

Parameter description:

Parameter	Type	Description
min_tokens	uint256	Minimum amount of token required; calculated from the slippage selected by the user
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit
recipient	address	Address receiving token
msg.value	uint256	Amount of TRX needed to buy the token

Returns:

uint256	Amount of TRC20 tokens purchased

8. trxToTokenTransferOutput

function trxToTokenTransferOutput(uint256 tokens_bought, uint256 deadline, address recipient) external payable returns (uint256);

Function description: sell TRX and buy token; token is in a fixed amount. Then, transfer the purchased token to the recipient address.

Parameter description:

Parameter	Type	Description
tokens_bought	uint256	Amount of token purchased (fixed amount)
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit
recipient	address	Address receiving token
msg.value	uint256	Maximum amount that can be paid to buy the fixed amount of token, which is calculated from slippery percentage

Returns:

uint256	Amount of TRX sold

9. tokenToTrxTransferInput

function tokenToTrxTransferInput(uint256 tokens_sold, uint256 min_trx, uint256 deadline, address recipient) external returns (uint256);

Function description: sell tokens and buy TRX; token is in a fixed amount. Then, transfer the purchased TRX to the recipient address.

Parameter description:

参数	类型	描述
min_liquidity	uint256	Minimum additional issuance of liquidity token
max_tokens	uint256	Maximum additional amount of TRC20 token
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit

msg.value	uint256	Amount of TRX deposited when adding liquidity

Returns:

uint256	Amount of TRX purchased

10. tokenToTrxTransferOutput

function tokenToTrxTransferOutput(uint256 trx_bought, uint256 max_tokens, uint256 deadline, address recipient) external returns (uint256);

Function description: sell tokens and buy TRX; TRX is in a fixed amount. Then, transfer the purchased TRX to the recipient address.
Parameter description:

Parameter	Type	Description
trx_bought	uint256	Amount of TRX purchased (fixed amount)
max_tokens	uint256	Maximum amount of tokens that can be deducted from the user account; calculated from the slippage
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit
recipient	address	Address receiving TRX

Returns:

uint256	Amount of TRC20 token sold

11. tokenToTokenSwapOutput

function tokenToTokenSwapOutput(uint256 tokens_bought, uint256 max_tokens_sold, uint256 max_trx_sold, uint256 deadline, address token_addr) external returns (uint256);

Function description: sell token1 and buy token2 (token2 is in a fixd amount).Since TRX functions as intermediary, both token1 and token2 need to have exchange addresses.

Parameter description:

Parameter	Type	Description
tokens_bought	uint256	Amount of token2 purchased (fixed amount)

max_tokens_sold	uint256	Maximum amount of token1 that can be deducted from the user account; calculated from the slippage
max_trx_sold	uint256	Amount of TRX converted in the interim(suggested value: -1)
deadline	uint256	unix timestamp; transaction will revert if exceeding this time limit
token_addr	address	Token2 contract address (not the exchange address, because it is the token1 exchange address that is called, and the contract will automatically fetch the exchange2 address according to token2)

Returns:

uint256	Amount of token1 sold

12. tokenToTokenSwapInput

function tokenToTokenSwapInput(uint256 tokens_sold, uint256 min_tokens_bought, uint256 min_trx_bought, uint256 deadline, address token_addr) external returns (uint256);

Function description: sell token1 and buy token2 (token1 is in a fixd amount).

Parameter description:

Parameter	Type	Description
tokens_sold	uint256	Amount of token1 sold (fixed amount)
min_tokens_bought	uint256	Minimum amount of token2 needed, which is calculated from the slippage
min_trx_bought	uint256	Amount of TRX converted in the interim(suggested value: 1)
deadline	uint256	Unix timestamp; transaction will revert if exceeding this time limit
token_addr	address	Token2 contract address

Returns:

uint256	Amount of token2 purchased

13. tokenToTokenTransferInput

function tokenToTokenTransferInput(uint256 tokens_sold, uint256 min_tokens_bought, uint256 min_trx_bought, uint256 deadline, address recipient, address token_addr) external returns (uint256);

Function description: sell token1 and buy token2 (token1 is in a fixd amount).Then, transfer the purchased token2 to the recipient's address.

Parameter description:

Parameter	Type	Description
tokens_sold	uint256	Amount of token1 sold (fixed amount)
min_tokens_bought	uint256	Minimum amount of token2 needed, which is calculated from the slippage
min_trx_bought	uint256	Amount of TRX converted in the interim(suggested value: 1)
deadline	uint256	unix timestamp; transaction will revert if exceeding this time limit
recipient	address	Address receiving token2
token_addr	address	Token2 contract address

Returns:

uint256	Amount of token2 purchased

14. tokenToTokenTransferOutput

function tokenToTokenTransferOutput(uint256 tokens_bought, uint256 max_tokens_sold, uint256 max_trx_sold, uint256 deadline, address recipient, address token_addr) external returns (uint256);

Function description: sell token1 and buy token2 (token2 is in a fixd amount).Then, transfer the purchased token2 to the recipient's address.
Parameter description:

Parameter	Type	Description
tokens_bought	uint256	Amount of token2 purchased (fixed amount)

max_tokens_sold	uint256	Maximum amount of token1 that can be deducted from the user account; calculated from the slippage
max_trx_sold	uint256	Amount of TRX converted in the interim(suggested value: -1)
deadline	uint256	unix timestamp; transaction will revert if exceeding this time limit
recipient	address	Address receiving token2
token_addr	address	Token2 contract address

Returns:

uint256	Amount of token1 sold

2.3.2.3 Contract event

1. TokenPurchase

event TokenPurchase(address indexed buyer, uint256 indexed trx_sold, uint256 indexed tokens_bought)

Function description: trigger the event in trxToTokenSwapInput, trxToTokenTransferInput, trxToTokenSwapOutput and trxToTokenTransferOutput

Parameter description:

Parameter	Type	Description
buyer	address	User who triggers the transaction
trx_sold	uint256	Amount of TRX paid
tokens_bought	uint256	Amount of TRC20 tokens purchased

2. TrxPurchase

event TrxPurchase(address indexed buyer, uint256 indexed tokens_sold, uint256 indexed trx_bought)

Function description: trigger the event in tokenToTrxSwapInput, tokenToTrxTransferInput, tokenToTrxSwapOutput, tokenToTrxTransferOutput, tokenToTokenSwapInput, tokenToTokenTransferInput, tokenToTokenSwapOutput and tokenToTokenTransferOutput

Parameter description:

Parameter	Type	Description
buyer	address	User who triggers the transaction
tokens_sold	uint256	Amount of TRC20 tokens paid
trx_bought	uint256	Amount of TRX purchased

3. AddLiquidity

event AddLiquidity(address indexed provider, uint256 indexed trx_amount, uint256 indexed token_amount)

Function description: trigger the event in addLiquidity

Parameter description:

Parameter	Type	Description
provider	address	User who triggers the transaction (the one who provides the liquidity)
trx_amount	uint256	Amount of TRX provided by the liquidity
token_amount	uint256	Amount of TRC20 tokens provided by the liquidity

4. RemoveLiquidity

event RemoveLiquidity(address indexed provider, uint256 indexed trx_amount, uint256 indexed token_amount)

Function description: trigger the event in removeLiquidity

Parameter description:

Parameter	Type	Description
provider	address	User who triggers the transaction (the one who removes the liquidity)

trx_amount	uint256	Amount of TRX removed
token_amount	uint256	Amount of TRC20 tokens removed