

JustSwap 接口文档

— Just 团队

v 1.0.0

目录

一、后端API	3
1.1 获取所有交易对接口	3
二、智能合约接口	4
2.1 合约地址	4
2.2 合约接口列表	4
2.2.1 TRC20 Token	4
2.2.2 Factory	4
2.2.3 Exchange	5
2.3 合约接口详细描述	7
2.3.1 Factory	7
2.3.1.1 查询型接口	7
1、getExchange	7
2、getToken	7
2.3.1.2 修改型接口	8
1、createExchange	8
2.3.1.3 合约事件	8
1、NewExchange	8
2.3.2 Exchange	8
2.3.2.1 查询型接口	8
1、getTrxToTokenInputPrice	8
2、getTrxToTokenOutputPrice	9
3、getTokenToTrxInputPrice	9
4、getTokenToTrxOutputPrice	10
5、tokenAddress	10
6、factoryAddress	10
2.3.2.2 修改型接口	11
1、addLiquidity	11
2、removeLiquidity	11
3、trxToTokenSwapInput	12
4、trxToTokenSwapOutput	12
5、tokenToTrxSwapOutput	13
6、tokenToTrxSwapInput	13
7、trxToTokenTransferInput	14
8、trxToTokenTransferOutput	14
9、tokenToTrxTransferInput	15
10、tokenToTrxTransferOutput	15
11、tokenToTokenSwapOutput	16
12、tokenToTokenSwapInput	16
13、tokenToTokenTransferInput	17

14、tokenToTokenTransferOutput	17
2.3.2.3 合约事件	18
1、TokenPurchase	18
2、TrxPurchase	19
3、AddLiquidity	19
4、RemoveLiquidity	19

一、后端API

1.1 获取所有交易对接口

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

参数:

page_size : int, 每个分页大小, 最大值 50

page_num: int, 第几个分页, 从0 开始

返回值格式:

```
{
  "data":
  [
    "0_TR7NHqjeKQxGTCi8q8ZY4pL8otSzgjLj6t": { //key:TRX的id和token的id
      "quote_id": "0", // 0表示TRX的id
      "quote_name": "TRX", // 目前只支持TRX的名称
      "quote_symbol": "TRX", // 目前只支持TRX的简称
      "quote_decimal": "6", // TRX的精度为6
      "base_id": "TR7NHqjeKQxGTCi8q8ZY4pL8otSzgjLj6t", // token的contract
      address
      "base_name": "Tether USD", // base_token的名称, 这里是 USDT
      "base_symbol": "USDT", // token的简称
      "base_decimal": "6", // token的精度
      "price": "37.450887374011593590", // 当前 base_token 的价格, 当前是 1 USDT
      = 37.45 TRX
      "quote_volume": "16928742252878", // 最近24小时累计交易 TRX 的数量, 单位:
      1 SUN
      "base_volume": "452025130508" // 最近24小时累计交易 base_token 的数量, 单
      位是其最小单位
    }
  ],
  "total_num": 25, // 总条目数
  "err_no": 0,
  "err_msg": ""
}
```

备注:

该接口单访问来源的最大访问频率为 1次 / second

二、智能合约接口

2.1 合约地址

合约名称	合约地址	备注
factory合约	TXk8rQSAvPvBBNtqSoY6nCfsXWCSSpTVQF	factory 合约负责创建交易对, 和维护交易对列表
exchange 合约	TQn9Y2khEsLJW1ChVWFMSMeRDow5KcbLSE	每种交易对有一个 exchange 合约. 这个是 USDT/TRX 的 exchange 合约地址.

2.2 合约接口列表

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 合约接口详细描述

2.3.1 Factory

2.3.1.1 查询型接口

1、getExchange

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

功能描述：通过TRC20 token获取justswap中对应的exchange地址。

参数描述：

参数	类型	描述
token	address	TRC20 token 地址

Returns:

address	justswap的交易对地址
---------	----------------

2、getToken

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

功能描述：通过justswap中的exchange地址获取TRC20 token地址。

参数描述：

参数	类型	描述
token	address	justswap中的exchange地址

Returns:

address	TRC20 token地址
---------	---------------

2.3.1.2 修改型接口

1、createExchange

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

功能描述：创建justswap交易对。每个TRC20 token只能创建一个exchange地址。

参数描述：

参数	类型	描述
token	address	TRC20 token 地址

Returns:

address	justswap的交易对地址
---------	----------------

2.3.1.3 合约事件

1、NewExchange

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

功能描述：每次通过 createExchange 创建交易对时发出。

参数描述：

参数	类型	描述
token	address	TRC20 token 地址
exchange	address	justswap中TRC20 token对应的exchange地址

2.3.2 Exchange

2.3.2.1 查询型接口

1、getTrxToTokenInputPrice

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

功能描述：通过支付TRX的数量获取可购买到的TRC20 token 数量

参数描述：

参数	类型	描述
trx_sold	uint256	卖的TRX数量

Returns:

uint256	可购买到的 TRC20 token 数量
---------	----------------------

2、getTrxToTokenOutputPrice

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

功能描述：通过购买 TRC20 token 数量获取只要支付的TRX数量

参数描述：

参数	类型	描述
tokens_bought	uint256	购买 TRC20 token 数量

Returns:

uint256	需要支付的TRX数量
---------	------------

3、getTokenToTrxInputPrice

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

功能描述：通过卖的TRC20 token数量 获取 可购买到的 TRX 数量

参数描述：

参数	类型	描述
tokens_sold	uint256	卖的TRC20 token数量

Returns:

uint256	可购买到的 TRX 数量
---------	--------------

4、getTokenToTrxOutputPrice

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

功能描述：通过购买的TRX数量 获取 需要支付的TRC20 token数量

参数描述：

参数	类型	描述
trx_bought	uint256	购买 TRX 数量

Returns:

uint256	需要支付的TRC20 token数量
---------	--------------------

5、tokenAddress

function tokenAddress() external view returns (address);

功能描述：获取这个justswap交易对中可进行交易的TRC20 token地址

参数描述：无

Returns:

address	可在该交易对进行交易的TRC20 token地址
---------	--------------------------

6、factoryAddress

function factoryAddress() external view returns (address);

功能描述：获取创建justswap交易对的Factory合约地址

参数描述：无

Returns:

address	justswap的factory合约地址
---------	----------------------

2.3.2.2 修改型接口

1、addLiquidity

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

功能描述：

给当前exchange添加流动性。扣除callvalue和对应的token到exchange合约中。mint出流动性token作为标识。

添加流动性之前需要先进行approve操作。

参数描述：

参数	类型	描述
min_liquidity	uint256	最低增发的liquidity token 数量
max_tokens	uint256	最大增加的TRC20 token数量
deadline	uint256	秒的时间戳，超过该时间，交易将revert
msg.value	uint256	添加流动性时转入的 TRX 数量

Returns:

uint256	增发给调用者的 liquidity token 数量
---------	----------------------------

备注:

1). msg.value 参数是调用函数时，附带的 callvalue(即：TRX 数量)，并不在函数签名的参数列表中，以下 msg.value 定义相同，不再赘述。

2、removeLiquidity

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

功能描述：移除流动性，将移除的 TRX 和 TRC20 token数量转给调用者。

参数描述：

参数	类型	描述
amount	uint256	移除的liquidity token数量
min_trx	uint256	最低移除的 TRX 数量

min_tokens	uint256	最低移除的 TRC20 token数量
deadline	uint256	秒的时间戳, 超过该时间, 交易将revert

Returns:

uint256	移除的 TRX 数量
uint256	移除的 TRC20 token数量

3、trxToTokenSwapInput

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

功能描述：卖TRX购买token, TRX是固定数量

参数描述：

参数	类型	描述
min_tokens	uint256	根据用户选择的滑点计算出的最低需要获得的token数量
deadline	uint256	秒的时间戳, 超过该时间, 交易将revert
msg.value	uint256	用户购买token支付的TRX

Returns:

uint256	购买到的TRC20 token数量
---------	-------------------

4、trxToTokenSwapOutput

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

功能描述：卖TRX购买token, token是固定数量

参数描述：

参数	类型	描述
tokens_bought	uint256	购买的token数量（固定数量）

deadline	uint256	秒的时间戳，超过该时间，交易将revert
msg.value	uint256	根据滑点计算出用户购买固定数量的token，最多可支付的值

Returns:

uint256	卖的TRX数量
---------	---------

5、tokenToTrxSwapOutput

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

功能描述：卖token购买TRX，TRX是固定数量

参数描述：

参数	类型	描述
trx_bought	uint256	购买的TRX数量（固定数量）
max_tokens	uint256	根据滑点计算出最多扣用户的token数量
deadline	uint256	秒的时间戳，超过该时间，交易将revert

Returns:

uint256	卖的TRC20 token 数量
---------	------------------

6、tokenToTrxSwapInput

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

功能描述：卖token购买trx，token是固定数量

参数描述：

参数	类型	描述
tokens_sold	uint256	卖的token数量（固定数量）
min_trx	uint256	根据滑点计算出最低需要得到的TRX数量
deadline	uint256	秒的时间戳，超过该时间，交易将revert

Returns:

uint256	购买到的 TRX 数量
---------	-------------

7、trxToTokenTransferInput

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

功能描述：卖TRX购买token，TRX是固定数量。然后将购买的token转移给recipient地址。

参数描述：

参数	类型	描述
min_tokens	uint256	是根据用户选择的滑点计算出的最低需要获得的token数量
deadline	uint256	秒的时间戳，超过该时间，交易将revert
recipient	address	接收token的地址
msg.value	uint256	用户购买token需支付的TRX

Returns:

uint256	购买到的 TRC20 token 数量
---------	---------------------

8、trxToTokenTransferOutput

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

功能描述：卖TRX购买token，token是固定数量，然后将购买的token转移给recipient地址。

参数描述：

参数	类型	描述
tokens_bought	uint256	购买的token数量（固定数量）
deadline	uint256	秒的时间戳，超过该时间，交易将revert
recipient	address	接收token的地址
msg.value	uint256	根据滑点计算出用户购买固定数量的token，最多可支付的值

Returns:

uint256	卖的TRX数量
---------	---------

9、tokenToTrxTransferInput

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

功能描述：卖token购买TRX，token是固定数量，然后将购买的TRX转移给recipient地址。

参数描述：

参数	类型	描述
tokens_sold	uint256	卖的token数量（固定数量）
min_trx	uint256	根据滑点计算出最低需要购买到的trx数量
deadline	uint256	秒的时间戳，超过该时间，交易将revert
recipient	address	接收trx的地址

Returns:

uint256	购买到的TRX数量
---------	-----------

10、tokenToTrxTransferOutput

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

功能描述：卖token购买TRX，TRX是固定数量，然后将购买的TRX转移给recipient地址。

参数描述：

参数	类型	描述
trx_bought	uint256	购买的TRX数量（固定数量）
max_tokens	uint256	根据滑点计算出最多扣用户的token数量
deadline	uint256	秒的时间戳，超过该时间，交易将revert
recipient	address	接收TRX的地址

Returns:

uint256	卖的 TRC20 token 数量
---------	-------------------

11、tokenToTokenSwapOutput

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

功能描述：卖token1 购买 token2（token2 是固定数量。因为是通过TRX做中转，所以需要 token1和token2分别存在exchange地址。

参数描述：

参数	类型	描述
tokens_bought	uint256	买token2的数量（固定数量）
max_tokens_sold	uint256	根据滑点计算出 最多扣用户的 token1 数量
max_trx_sold	uint256	TRX中间转换的数量。（建议值：-1）
deadline	uint256	秒的时间戳，超过该时间，交易将revert
token_addr	address	token2 的合约地址（不是exchange地址，因为调用的是 token1的exchange地址，合约会根据token2自动获取其 exchange2地址）

Returns:

uint256	卖的token1数量
---------	------------

12、tokenToTokenSwapInput

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

功能描述：卖token1 购买 token2（token1 是固定数量）。

参数描述：

参数	类型	描述
tokens_sold	uint256	卖token1的数量（固定数量）
min_tokens_bought	uint256	根据滑点计算出最低需要得到的 token2 数量

min_trx_bought	uint256	TRX中间转换的数量。（建议值：1）
deadline	uint256	秒的时间戳，超过该时间，交易将revert
token_addr	address	token2 的合约地址

Returns:

uint256	买的token2数量
---------	------------

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);
```

功能描述：卖token1 购买 token2 (token1 是固定数量)。然后将购买的token2转移给recipient 地址。

参数描述：

参数	类型	描述
tokens_sold	uint256	卖token1的数量（固定数量）
min_tokens_bought	uint256	根据滑点计算出最低需要得到的 token2 数量
min_trx_bought	uint256	TRX中间转换的数量。（建议值：1）
deadline	uint256	秒的时间戳，超过该时间，交易将revert
recipient	address	接收 token2 的地址
token_addr	address	token2 的合约地址

Returns:

uint256	买的token2数量
---------	------------

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);
```

功能描述：卖token1 购买 token2 (token2 是固定数量)。然后将购买的token2转移给recipient 地址。

参数描述：

参数	类型	描述
tokens_bought	uint256	买token2的数量（固定数量）
max_tokens_sold	uint256	根据滑点计算出 最多扣用户的 token1 数量
max_trx_sold	uint256	TRX中间转换的数量。（建议值：-1）
deadline	uint256	秒的时间戳，超过该时间，交易将revert
recipient	address	接收 token2 的地址
token_addr	address	token2 的合约地址

Returns:

uint256	卖的token1数量
---------	------------

2.3.2.3 合约事件

1、TokenPurchase

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

功能描述：在 txToTokenSwapInput、txToTokenTransferInput、txToTokenSwapOutput、txToTokenTransferOutput 中触发此事件

参数描述：

参数	类型	描述
buyer	address	触发交易的用户
trx_sold	uint256	支付的TRX数量
tokens_bought	uint256	购买到的 TRC20 token 数量

2、TrxPurchase

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

功能描述：在 tokenToTrxSwapInput、tokenToTrxTransferInput、tokenToTrxSwapOutput、tokenToTrxTransferOutput、tokenToTokenSwapInput、tokenToTokenTransferInput、tokenToTokenSwapOutput、tokenToTokenTransferOutput 中触发此事件

参数描述：

参数	类型	描述
buyer	address	触发交易的用户
tokens_sold	uint256	支付的 TRC20 token 数量
trx_bought	uint256	购买到的 TRX 数量

3、AddLiquidity

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

功能描述：在 addLiquidity 时触发该事件

参数描述：

参数	类型	描述
provider	address	触发交易的用户，即该笔流动性提供者
trx_amount	uint256	该笔流动性提供的 TRX 数量
token_amount	uint256	该笔流动性提供的 TRC20 token 数量

4、RemoveLiquidity

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

功能描述：在 removeLiquidity 中触发该事件

参数描述：

参数	类型	描述
----	----	----

provider	address	触发交易的用户，即移除该笔流动性的用户
trx_amount	uint256	移除的 TRX 数量
token_amount	uint256	移除的TRC20 token 数量