

ThunderChain Test Environment APIs

Test access APIs

baseUrl: <https://sandbox-walletapi.onethingpcs.com>

1. Developer registration

****Functionality****

This test registration interface uses an email address. If the user registers for the first time, service_id/secret is generated and emailed to the developer. If the email address has **already** been registered, an error code is returned.

****Request****

Method: POST

URL: /api/linktest/regist

BODY: JSON

Argument description:

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

--	--	--	--

email	string	yes	This is the email used for the test. Its primary purpose is for receiving test messages
-------	--------	-----	---

callback	string	no	This is the callback URL address for contract test or LinkToken exchange
----------	--------	----	--

****Response****

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

--	--	--	--

code	int	yes	Error code, 0: Success or Non-0: Failure
------	-----	-----	--

msg	string	yes	Error prompt message
-----	--------	-----	----------------------

|data|object|no|Data returned for the request|

2. Test account recharge

****Functionality****

It requests to recharge a designated account with LinkToken. The developer recharges his/her own account before he/she may use it for test.

Each request transfers 1 LinkToken to the account.

****Request****

Method: POST

URL: /api/linktest/recharge

BODY: JSON

Argument description:

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

--	--	--	--

email	string	yes	This is the email used for the test. Its primary purpose is for receiving test messages
-------	--------	-----	---

address	string	yes	This is the address of the account to be recharged
---------	--------	-----	--

sign	string	yes	Signature md5(email=xxx&address=xxx&secret=xxx), where xxx are the actual values of the request
------	--------	-----	---

****Response****

Argument name	Argument type	Required	Notes
-	-	-	-

code	int	yes	Error code, 0: Success or Non-0: Failure
msg	string	yes	Error prompt message
data	object	no	Data returned for the request

3. Contract publishing

****Functionality****

The user publishes contract to the test environment by providing the compiled bytecode.

****Request****

Method: POST

URL: /api/linktest/contract/deploy

BODY: JSON

Argument description:

Argument name	Argument type	Required	Notes
-	-	-	-

email	string	yes	This is the email used for the test. Its primary purpose is for receiving test messages
bytecode	string	yes	This is the compiled contract bytecode in hexadecimal ABI format
params	string	yes	This is the initialization parameter of the constructor in hexadecimal ABI format
sign	string	yes	Signature md5(email=xxx&bytecode=xxx&secret=xxx), where xxx are the actual values of the request

****Response****

Argument name	Argument type	Required	Notes
code	int	yes	Error code, 0: Success or Non-0: Failure
msg	string	yes	Error prompt message
data	object	no	Data returned for the request

****data****

Argument name	Argument type	Required	Notes
id	int	yes	This is the id created by contract deployment. It allows one to query the address of the deployed contract

4. Contract address query

****Functionality****

You may query the address of contract account using the id created by contract deployment.

****Request****

Method: POST

URL: /api/linktest/contract/address

BODY: JSON

Argument description:

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

-	-	-	-
---	---	---	---

email	string	yes	This is the email used for the test. Its primary purpose is for receiving test messages
-------	--------	-----	---

id	int	yes	This is the id returned by contract deployment
----	-----	-----	--

sign	string	yes	Signature md5(email=xxx&id=xxx&secret=xxx), where xxx are the actual values of the request
------	--------	-----	--

****Response****

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

-	-	-	-
---	---	---	---

code	int	yes	Error code, 0: Success or Non-0: Failure
------	-----	-----	--

msg	string	yes	Error prompt message
-----	--------	-----	----------------------

data	object	no	Data returned for the request
------	--------	----	-------------------------------

****data****

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

-	-	-	-
---	---	---	---

address	string	yes	account address
---------	--------	-----	-----------------

5. Query the address of the last deployed contract

****Functionality****

You may query the id and address of the last deployed contract.

****Request****

Method: POST

URL: /api/linktest/contract/last

BODY: JSON

Argument description:

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

--	--	--	--

email	string	yes	This is the email used for the test. Its primary purpose is for receiving test messages
-------	--------	-----	---

sign	string	Yes	Signature md5 (email=xxx&secret=xxx), where xxx are the actual values of the request
------	--------	-----	--

****Response****

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

--	--	--	--

code	int	yes	Error code, 0: Success or Non-0: Failure
------	-----	-----	--

msg	string	yes	Error prompt message
-----	--------	-----	----------------------

data	object	no	Data returned for the request
------	--------	----	-------------------------------

****data****

Argument name	Argument type	Required	Notes
---------------	---------------	----------	-------

-	-	-	-
---	---	---	---

id	int	yes	Contract ID
----	-----	-----	-------------

address	string	yes	account address
---------	--------	-----	-----------------

6. Invoking LinkToken Pocket Protocol

Invoking protocol

A contract must be executed through LinkToken Pocket. You should either invoke the LinkToken App from a third-party app or, alternatively, scan a QR code from the LinkToken App.

The invoking protocol of LinkToken Pocket is scheme://host?

Description of request arguments

Argument	Type	Required	Notes
-----	-----	----	-----
scheme	string	yes	LinkToken Pocket scheme otst
host	string	yes	The host value of LinkToken Pocket is: payment
tx-data	byte[]	yes	This is a Base64 code primarily consisting of order information about the payment. It is in the format of key=value joined by &.
resource	byte[]	yes	This is a Base64 code of the sourceapp
cb-data	byte[]	yes	This is a Base64 code representing the additional information that the payment sender asks the Pocket to return
x-source	string	yes	source app scheme eg. wky-app (optional callback upon calling on iOS; neglected on Android)
x-success	string	yes	Callback upon success. If you leave it blank, return to the source app will not be made (optional callback upon calling on iOS; neglected on Android)
x-error	string	Yes	Callback upon failure. If you leave it blank, return to the source app will not be made. It consists of parameters: error-Code and errorMessage (optional callback upon calling on iOS; neglected on Android)
x-cancel	string	yes	Callback upon cancellation. If you leave it blank, return to the source app will not be made (optional callback upon calling on iOS; neglected on Android)

****Descriptions of tx-data arguments****

Argument	Type	Required	Notes
desc	string	yes	Description of contract execution. A prefix of "Contract Execute-" is required.
callback	string	no	A url code as the link for backend callback
to	string	yes	Receiving Pocket address of the transfer
value	string	yes	Number of LinkTokens in wei
order_id	string	yes	Order ID (to be obtained from the backend)
prepay_id	string	yes	Prepayment order ID
service_id	string	yes	Service ID. Contact Trade Center for service_id and private signature key (to be obtained from backend). The field is reserved (converted to integer upon app's submission to geth)
data	string	yes	Code of the contract to be executed. It is a hexadecimal string starting with 0x and consists of the address and call parameters of the function. It is NULL if only transfer is initiated.
gas_limit	string	yes	This is the max. Gas payment used to calculate the fee of contract execution (converted to integer upon app's submission to geth)
tx_type	string	yes	This is the transaction type. Use value contract to indicate a contract or tx_third for a third-party transaction. A default for NULL value is supported for third-party transactions. Compatibility with existing clients is available.
sign=md5(sha512(callback=xxx&prepay_id=xxx&service_id=xxx&to=xxx&value=xxx&key=私钥))			
sign	string	yes	Transaction signature
sign=md5(sha512(callback=xxx&prepay_id=xxx&service_id=xxx&to=xxx&value=xxx&key=private key))			

****Response****

1. Success

Use x-success as the argument for return after successful payment. The field is defined by payment service.

Argument name	Type	Required	Notes
hash	string	yes	This is the hash after the initiated transaction is successful. Query on subsequent status may be available.

2. Failure

Use the URL of x-error for return. The field is defined by x-callback-url

Argument name	Type	Required	Notes
-----	-----	----	-----
error-Code	string	yes	Error code. Note: it is error-Code.
errorMessage	string	yes	Error message.

3. Cancel

Use the URL of x-cancel for return.