

uWallet API

For auth you have to use TOKEN in headers like: { authorization: TOKEN }

Get wallet address		
POST	https://api.uwallet.biz/v1/wallet	
Param Name	Type	Description
Inputs		
coin	string	Coin name
Outputs as OBJECT		
address	string	The address of wallet

Get ALIAS wallet address		
POST	https://api.uwallet.biz/v1/wallet/alias	
Param Name	Type	Description
Inputs		
coin	string	Coin name
Outputs as OBJECT		
address	string	The address of alias wallet

Get wallet balances		
GET	https://api.uwallet.biz/v1/wallet	
Param Name	Type	Description
Outputs as ARRAY		
coin	string	Coin name
amount	number	Amount of total balance
free	number	Amount of available balance
blocked	number	Amount of blocked balance

Send coin to ADDRESS

POST	https://api.uwallet.biz/v1/wallet/send	
Param Name	Type	Description
Inputs		
coin	string	Coin name
amount	string	Amount you want to send
receiver	string	Receiver wallet address
Outputs as OBJECT		
hash	string	The hash of network response

Create a voucher

POST	https://api.uwallet.biz/v1/voucher	
Param Name	Type	Description
Inputs		
coin	string	Coin name
amount	string	Amount you want to create
Outputs as OBJECT		
transactionId	string	The ID of TRANSACTIONS
code	string	The voucher code

Activate a voucher

POST	https://api.uwallet.biz/v1/voucher/use	
Param Name	Type	Description
Inputs		
coin	string	Coin name
code	string	The voucher code
Outputs as OBJECT		
transactionId	string	The ID of TRANSACTIONS
status	string	Status of request (confirm, reject, ...)
receive	number	The amount of voucher
batchId	number	Reference number of source (PM, cVoucher, ...)

Get transactions		
GET	https://api.uwallet.biz/v1/transaction	
Query Name	Type	Description
Inputs as URL QUERY		
fromDate	timestamp	Filter transactions start from date (optional)
toDate	timestamp	Filter transactions start to date (optional)
coin	string	Filter transactions with coin name (optional)
id	string	Filter transactions with ID (optional)
Param Name	Type	Description
Outputs as ARRAY		
id	string	ID of transaction
coin	string	Coin name
amount	number	Amount of transaction
fee	number	Fee of transaction
receive	number	Final amount of transaction (amount – fee)
status	string	Status of request (confirm, reject, ...)
side	string	Side of transaction: SEND/RECEIVE
type	string	Type of transaction: DIRECT/VOUCHER
data	object	More detail like sender, receiver, hash, ...

Steps of activate voucher:

- 1- **POST** https://api.uwallet.biz/v1/voucher/use

```
curl --location 'v1/voucher/use' \
--header 'Authorization: {TOKEN}' \
--header 'Content-Type: application/json' \
--data '{
  "coin": "PM",
  "code": "xxxx-xxxxx"
}'
```

- 2- If status 'confirm', your voucher activates successful, otherwise go to step 3
- 3- **GET** https://api.uwallet.biz/v1/transaction?id={transactionId} every 30 seconds until status changed to 'reject' or 'confirm'

```
curl --location 'v1/transaction?transactionId=xxxxxxxxx' \
--header 'authorization: {TOKEN}'
```

Steps of create voucher:

- 1- **POST** <https://api.uwallet.biz/v1/voucher>

```
curl --location 'v1/voucher' \  
--header 'authorization: {TOKEN}' \  
--header 'Content-Type: application/json' \  
--data '{  
  "coin": "TRX",  
  "amount": "1.4"  
}'
```

Steps of send token (coin):

- 1- **POST** <https://api.uwallet.biz/v1/wallet/send>

```
curl --location 'v1/wallet/send' \  
--header 'authorization: {TOKEN}' \  
--header 'Content-Type: application/json' \  
--data '{  
  "coin": "TRX",  
  "amount": 0.4,  
  "receiver": "xxxxxxx"  
}'
```

Steps of receive token (coin):

- 1- **GET** <https://api.uwallet.biz/v1/transaction> every 30 seconds

```
curl --location 'v1/transaction' \  
--header 'authorization: {TOKEN}'
```

- 2- If you see new transaction, check the ID in your database; if does not exist, go to step 3
- 3- If status is 'confirm' or 'reject', update or insert your database, otherwise check transactions 30 seconds later

Note: you can also filter transactions via [dateFrom](#), [dateTo](#), [coin](#), [side](#) and [type](#)

```
curl --location 'v1/transaction?dateFrom={timestamp}&dateTo={timestamp}&coin=TRX&side=receive&type=direct' \  
--header 'authorization: {TOKEN}'
```