SkinCoin payment service API v1.1

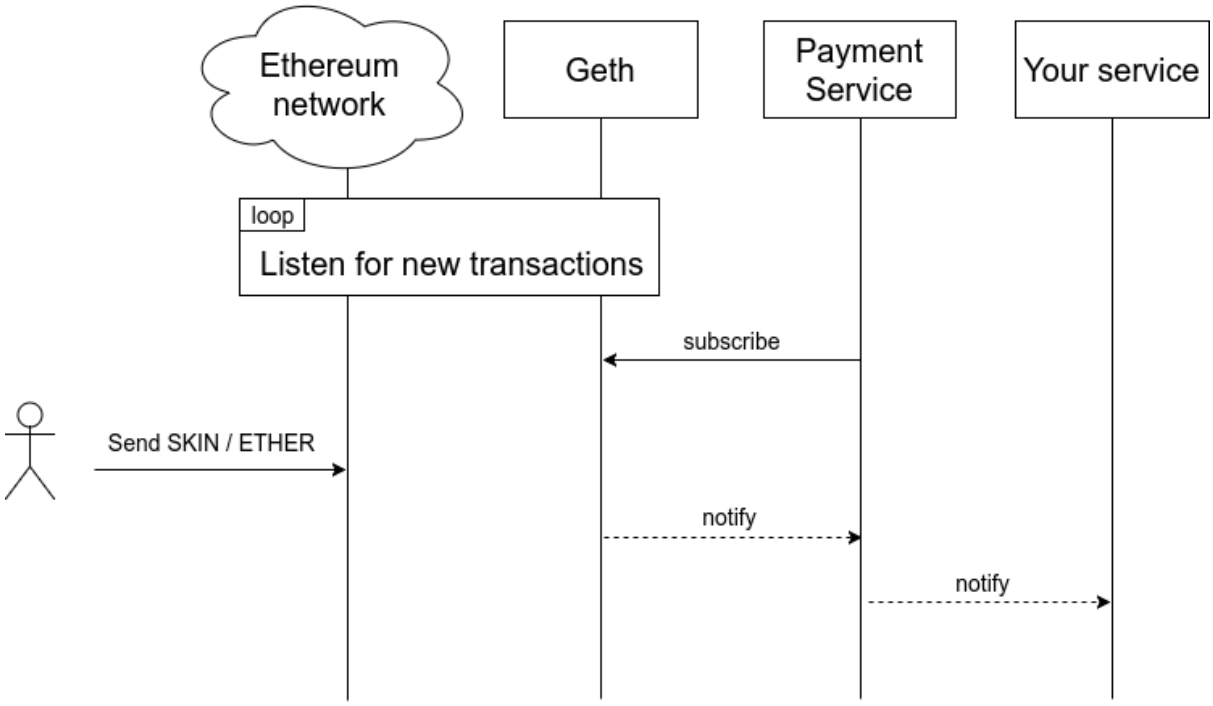
Description

Payment service provides an API to accept and pay with SkinCoin (SKIN) and ETHER

Type	URL	Description
GET	/api/ethereum/getInfo	Get Ethereum network info
POST	/api/ethereum/personal/createAccount	Create a new account
POST	/api/ethereum/createPayout	Create a withdrawal transaction
GET	/api/ethereum/getStatus	Get payment or withdrawal status
GET	/api/ethereum/getBalance	Get account balance

Accept payments

The process to accept payments depicted below:



To accept payments you need:

1) Create SkinCoin wallet address for the user. (See “/api/ethereum/personal/createAccount”)

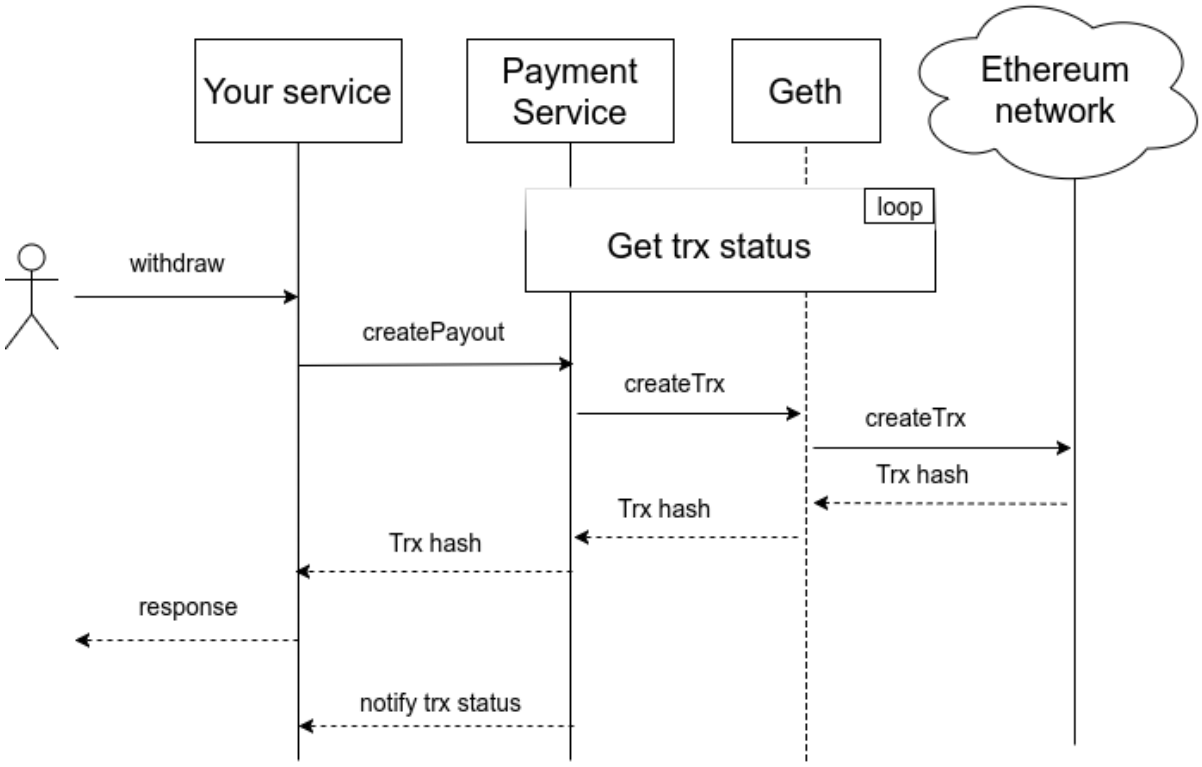
NOTE: Best practice is to create one SkinCoin wallet per user. Also you may create one wallet per payment. It depends from your business needs.

2) The user sends the SkinCoin to this address. It doesn't matter from where the user send the SkinCoin. Payment service listens all new pending transactions of Ethereum network so all transactions which belong to the payment service will be catch and processed.
3) Payment service notifies your service when the payment transaction will be detected in Ethereum network by REST interface.

NOTE: If for some reason the payment service could not send a notification It will try do it in 1 minute and so on, until the notification is successfully delivered

Withdrawal

All withdrawals make from the one payout wallet which have to be configured in Admin console. The process of withdrawal is depicted below. To make a withdrawal see “/api/ethereum/createPayout” example.



Request / Response examples

Get Ethereum network info

To get Ethereum network info invoke the following request:

Request:

GET http://localhost:8189/api/ethereum/getInfo

Response:


```
{
  "network": "PRIVATE_NET",
  "skincoin_address": "0x6e71346880e357a83ae24edd499f4a3fa2cb1825",
```

```
  "client_version": "Geth/v1.6.7-stable/linux-amd64/go1.7.3"
}
```

Create a new account

To create a new account invoke the following request:

Request:

POST http://localhost:8189/api/ethereum/personal/createAccount?api_key=

```
{
  "request_id": "string value"
}
```

Request params:

- **api_key** - API-ключ

Response:

```
{
  "address": "0x"
}
```

Create a withdrawal transaction

To create a withdrawal transaction invoke the following request:

Request:

POST http://localhost:8189/api/ethereum/createPayout?api_key=

```
{
  "request_id": "1234",
  "to": "0x",
  "amount": 1000,
  "currency": "SKIN"
}
```

Response:

```
{
  "request_id": "1234",
  “creating_time”: 12345,
  “from”: “0x..”
  “to”: “0x...”
  “type”: “payment”,
```

```
  “amount”: 1000,
  "status": "created",
  “error”: “”                  // In case when status == “error”
}
```

Where, status may has the following values:

- **created** - Withdrawal was created but not confirmed by needed number of Ethereum network participants. Default number of confirmations is 30.
- **accepted** - Withdrawal is accepted. Successful status for completed withdrawals
- **error** - Error was happened during withdrawal

Get payment / withdrawal status

To get payment / withdrawal status invoke the following request:

Request:

GET http://localhost:8189/api/ethereum/getStatus?api_key=

Параметры запроса:

- **api_key** - API key
- **type** - “payment” / “payout”
- **request_id** - Unique request id

Response:

```
{
  "request_id": "1234",
  “creating_time”: 12345,
  “type”: “payment”,
  “from”: “0x..”
  “to”: “0x...”
  “amount”: 1000,
  "status": "accepted",
  “error”: “”                  // In case when status == “error”
}
```

or

```
{
  "request_id": "1234",
  “creating_time”: 12345,
  “type”: “payout”,
  “from”: “0x..”
  “to”: “0x...”
```

```
  “amount”: 1000,
  "status": "accepted",
  “error”: “”                  // In case when status == “error”
}
```

Get an account balance

To get an account balance invoke the following request:

GET http://localhost:8189/api/ethereum/web3/getBalance?api_key=

Request params:

- api_key - API key
- address - Account address
- currency - SKIN / ETHER. By default ETHER.

Accept REST callback notifications

To accept REST notifications your service must implement POST callback interface to accept the following data:

```
{
  "request_id": "1234",
  “creating_time”: 12345,
  “type”: “payment” / “payout”,
  “hash”: “0x”,  // Transaction hash
  “from”: “0x..”,
  “to”: “0x...”,
  “amount”: 1000,
  "status": "created",
  "currency": "SKIN",   // "SKIN" / "ETHER"
  “error”: “”                  // In case when status == “error”
}
```

In response to the payment service you need send:

```
{
  “success”: true
}
```

Security

Each notification contains HTTP "Authorization" header with a hash of secret key "auth_secret" (you may set up a secret key in Admin console). SHA256 algorithm is used to hash the secret key. For example:

auth_secret = "K746X5XMjE6qPDAAtQstfUHkgGUfqDdnYx4TTq"

SHA256 = "2a612c1246bd04cc3e16da495598abb335eeb6f7c4e7058fae78018fd771bd2e"

HTTP header is:

"Authorization: 2a612c1246bd04cc3e16da495598abb335eeb6f7c4e7058fae78018fd771bd2e"

Service that accepts the callback notification must calculate SHA256 hash from "auth_secret" and compare with the "Authorization" header value before processing the notification