

Fuspay Wallet API Documentation

Description

This doc describes how to **connect, authenticate** and **transmit** data via Fuspay Wallet System

Note: The Fuspay wallet system uses **Rivest-Shamir-Adleman (RSA)** asymmetric encryption to encrypt and decrypt data via REST API

Connect

To access the Fuspay wallet system, you need to download your RSA Digital certificates from your dashboard. The certificates are public and private certificate. To send data to Fuspay you need to encrypt the request payload with your public certificate and to get response from us, you need to decrypt the response with your private certificate.

Also, you must whitelist your server IP address on our system before a successful call can be made.

Authentication

To authenticate, you need to add the header - 'your-fuspay-secret-key'.

Remember to encrypt your request with your public key and decrypt it with your private key

Available Endpoints

1. Wallet Balance

HTTP 1/1 GET <https://exchanger-api.fuspay.finance/gateway/balance>

HEADER: your-fuspay-secret-key

Request

```
{  
}
```

Successful Response 200 OK

```
{  
  status: "success",  
  balance: Number,  
  wallet_address: String  
  chain: "usdt-trc20"  
}
```

Failed Response 400

```
{  
  status: "error",  
  message: String  
}
```

2. Withdrawal Payout

HTTP 1/1 POST <https://exchanger-api.fuspay.finance/gateway/payout>

HEADER: your-fuspay-secret-key

Request

```
{
amount: Number,
chain: "usdt-trc20"
}
Successful Response 201 OK
{
status: "success",
withdrawal_reference: String
}
Failed Response 400
{
status: "error",
message: String
}
```

3. Get Withdrawal Payout Fee

HTTP 1/1 POST <https://exchanger-api.fuspay.finance/gateway/fee>

HEADER: your-fuspay-secret-key

Request

```
{
amount: Number,
chain: "usdt-trc20"
}
```

Successful Response 200 OK

```
{
status: "success",
withdrawal_reference: String
}
```

Failed Response 400

```
{
status: "error",
message: String
}
```

Sample Encryption Code [REQUEST] - NodeJS

```
1
2 require('dotenv').config()
3 ///  
4 const fs = require("fs")
5 ///  
6 const crypto = require("crypto")
7 ///  
8 const path = require("path")
9 const path_to_public_key = path.join(__dirname, "..", "sample", "public-rsa.pem")
10 const path_to_private_key = path.join(__dirname, "..", "sample", "private-rsa.pem")
11 ///  
12 const read_stream = fs.createReadStream(path_to_public_key, 'utf8');
```

```

13 let public_key_str = '';
14 let private_key_str = '';
15 read_stream.on('data', (chunk) => {
16 // Concatenate each chunk to the public_key_str string
17 public_key_str += chunk;
18 });
19 read_stream.on('error', (error) => {
20 console.error('Error reading file:', error);
21 });
22 read_stream.on('end', () => {
23 // console.log('File reading complete.');
24 // console.log('Concatenated string:', public_key_str); // Output the final concatenated string
25 //: convert post data to string
26 const data_to_post = JSON.stringify({
27 amount: 1000,
28 chain: "usdt-trc20"
29 })
30 const buffer = Buffer.from(data_to_post, 'utf8');
31 const encrypted_data = crypto.publicEncrypt(public_key_str, buffer);
32 const save_enc_str = encrypted_data.toString('base64');
33 //: use axios or any http client to post data to fuspay
34 const axios = require('axios');
35 const url = 'https://exchanger-api.fuspay.finance/gateway/payout';
36 // const url = 'http://localhost:5000/gateway/payout';
37 const secret_key = process.env.FUSPAY_LIVE_SECRET_DEMO;
38 axios.post(url, {
39 enc_str: save_enc_str
40 }, {
41 headers: {
42 'your-fuspay-secret-key': secret_key
43 }
44 })
45 .then(response => {
46 //: RESPONSE FROM FUSPAY
47 // - Decrypt the response with your private key certificate since its encrypted
48 const encrypted_response = response.data
49 // console.log('Response:', response.data);
50 //: read it as string using stream api
51 const read_private_stream = fs.createReadStream(path_to_private_key, 'utf8');
52 read_private_stream.on('data', (chunk) => {
53 // Concatenate each chunk to the data string
54 private_key_str += chunk;
55 });
56 read_private_stream.on('error', (error) => {
57 console.error('Error reading file:', error);
58 });
59 read_private_stream.on('end', () => {
60 //console.log("End", private_key_str)
61 //: decrypt the response now and do something with it
62 const decrypted_data = crypto.privateDecrypt(
63 {
64 key: private_key_str,
65 passphrase: '',
66 },
67 Buffer.from(encrypted_response, "base64")
68 );
69 const decrypted_response = decrypted_data.toString()
70 // parse back to json

```

```

71 const get_res_obj = JSON.parse(decrypted_response)
72 console.log("Decrypted Response JSON from FusPay", get_res_obj)
73 })
74 })
75 .catch(error => {
76 console.error('Error:', error);
77 });
78 });
79

```

Sample PHP code

```

1 <!--
2 * @module encryption and decryption Restful API - Fuspay Wallet
3 * @description Sample Code in NodeJS. It uses RSA asymmetric encryption
4 * This can also be translated to other programming languages
5 -->
6
7 <?php
8 //require 'vendor/autoload.php'; // Assuming you have Composer autoloading
9
10 use \GuzzleHttp\Client;
11
12 // Load environment variables from .env file
13 $dotenv = Dotenv\Dotenv::createImmutable(__DIR__);
14 $dotenv->load();
15
16 // Get the path where you stored the public and private keys
17 $path_to_public_key = __DIR__ . '/../sample/public-rsa.pem';
18 $path_to_private_key = __DIR__ . '/../sample/private-rsa.pem';
19
20 // Read the public key file
21 $public_key_str = file_get_contents($path_to_public_key);
22
23 // Read the private key file
24 $private_key_str = file_get_contents($path_to_private_key);
25
26 // Convert post data to string
27 $data_to_post = json_encode(['amount' => 1000, 'chain' => 'usdt-trc20']);
28
29 // Encrypt the data using the public key
30 $is_valid = openssl_public_encrypt($data_to_post, $encrypted_data, $public_key_str);
31 if ($is_valid) {
32     $save_enc_str = base64_encode($encrypted_data);
33
34     // Make the HTTP POST request to FusPay
35     $client = new Client();
36     $url = 'https://exchanger-api.fuspay.finance/gateway/payout'; // Update with the correct URL
37     $secret_key = $_ENV['FUSPAY_LIVE_SECRET_DEMO'];
38
39     $headers = [
40         'your-fuspay-secret-key' => $secret_key,
41     ];
42     $data = [
43         'enc_str' => $save_enc_str,
44     ];
45

```

```
46 $response = $client->post($url, [  
47     'headers' => $headers,  
48     'json' => $data,  
49 ]);  
50  
51 $encrypted_response = $response->getBody()->getContents();  
52  
53 // Decrypt the response with the private key  
54 openssl_private_decrypt(base64_decode($encrypted_response), $decrypted_data, $private_key_str);  
55  
56 // Parse the decrypted response as JSON  
57 $get_res_obj = json_decode($decrypted_data, true);  
58 echo "Decrypted Response JSON from FusPay: " . print_r($get_res_obj, true);  
59 }  
60
```