How to handle C2C-messages

How to use websocket to handle messages and reply:

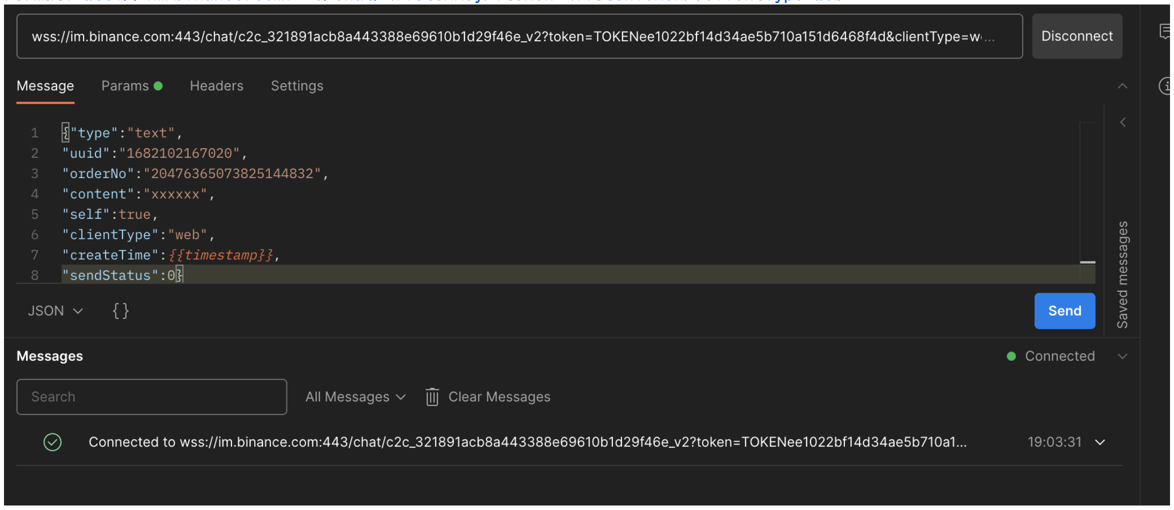
You can use GET /sapi/v1/c2c/chat/retrieveChatCredential to get credentials.

Response:

```
{
    "code": "000000",
    "message": "success",
    "data": {
        "chatWssUrl": "wss://im.binance.com:443/chat",
        "listenKey": "c2c_321891acb8a443388e69610b1d29f46e_v2",
        "listenToken": "TOKEN4e75573ca1e44e96ba809840d05d3e03"
    },
    "success": true
}
```

Using those endpoints you can create a websocket connection (on postman for example)

Format: wss://im.binance.com:443/chat/<listenKey>?token=<listenToken>&clientType=web



wss://im.binance.com:443/chat/c2c_321891acb8a443388e69610b1d29f46e_v2?token=TOKENee1022bf14d34ae5b710a151d6468f4d&clientType=w...

```
{"type":"text",
"uuid":"1682102167020",
"orderNo":"20476365073825144832",
"content":"xxxxxx",
"self":true,
"clientType":"web",
"createTime":{{timestamp}},
"sendStatus":0}
```

Connected to wss://im.binance.com:443/chat/c2c_321891acb8a443388e69610b1d29f46e_v2?token=TOKENee1022bf14d34ae5b710a1... 19:03:31

To send a new message (or reply income messages) you need follow this format:

```
{
"type":"text",
"uuid":"1682102167020",
"orderNo":"20476365073825144832",
"content":"xxxxxx",
"self":true,
"clientType":"web",
"createTime":{{timestamp}},
"sendStatus":0
}
```

Python code implementation:

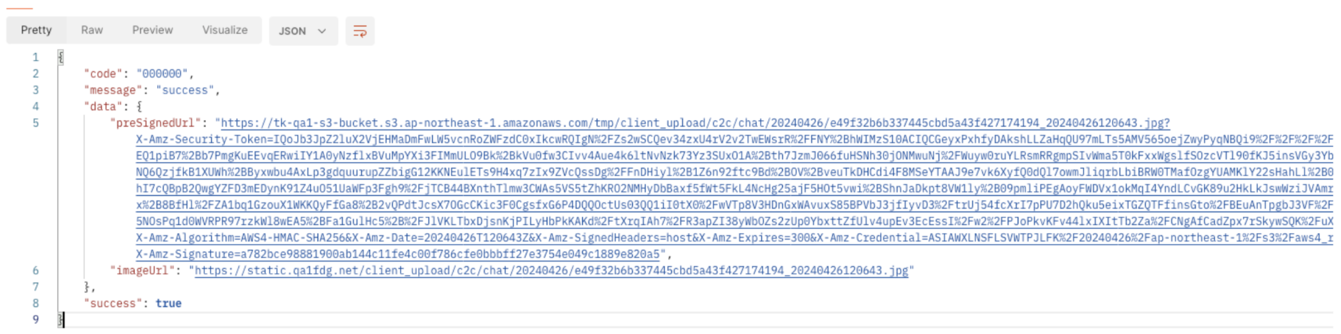
```
import websocket
import json
import hmac
import time
import hashlib
import requests
from urllib.parse import urlencode
import json

BASE_URL = "https://api.binance.com"  # production base url
```

```
KEY = "--"
SECRET = "----"

def hashing(query_string):
    return hmac.new(
        SECRET.encode("utf-8"), query_string.encode("utf-8"), hashlib.sha256
    ).hexdigest()


def get_timestamp():
    return int(time.time() * 1000)


def dispatch_request(http_method):
    session = requests.Session()
    session.headers.update(
        {"Content-Type": "application/json;charset=utf-8", "X-MBX-APIKEY": KEY, "clientType":"WEB"}
    )
    return {
        "GET": session.get,
        "DELETE": session.delete,
        "PUT": session.put,
        "POST": session.post,
    }.get(http_method, "GET")


# used for sending request requires the signature
def send_signed_request(http_method, url_path, payload={},dataLoad={}):
    query_string = urlencode(payload)
    # replace single quote to double quote
    query_string = query_string.replace("%27", "%22")
    if query_string:
        query_string = "{}&timestamp={}".format(query_string, get_timestamp())
    else:
        query_string = "timestamp={}".format(get_timestamp())

    url = (
            BASE_URL + url_path + "?" + query_string + "&signature=" + hashing(query_string)
    )


    # print("{} {}".format(http_method, url))
    params = {"url": url, "params": {},"data":dataLoad}

    response = dispatch_request(http_method)(**params)
    return response



uri_path = "/sapi/v1/c2c/chat/retrieveChatCredential"

dataLoad = {}
param = {}
response = send_signed_request("GET", uri_path,param,dataLoad)


print(response)
response = response.json()


wss_url = response['data']['chatWssUrl'] + "/"+ response['data']['listenKey'] + "?token=" +\
          response['data']['listenToken'] + "&clientType=web"


# websocket.enableTrace(True)


def on_message(ws, message):

    print(json.loads(message))
    return
```

```python
def on_close(ws, close_status_code, close_msg):
    print("### closed ###")

def on_ping(ws,message):
    print(message)

if __name__ == "__main__":
    websocket.enableTrace(False)
    ws = websocket.WebSocketApp(wss_url,
                                on_message=on_message,
                                on_close=on_close,
                                on_ping=on_ping)


    ws.run_forever()
```

How to upload an image using SAPI:

You can use POST /sapi/v1/c2c/chat/image/pre-signed-url to get “preSignedUrl” and "imageUrl"(Limited to 36 times per minute per user id)



```json
{
    "code": "000000",
    "message": "success",
    "data": {
        "preSignedUrl": "https://tk-qa1-s3-bucket.s3.ap-northeast-1.amazonaws.com/tmp/client_upload/c2c/chat/20240426/e49f32b6b337445cbd5a43f427174194_20240426120643.jpg?
            X-Amz-Security-Token=IQoJb3JpZ2luX2VjEHMaDmFwLW5vcnRoZWFzdC0xIkcwRQIgN%2FZs2wSCQev34zxU4rV2v2TwEWsrR%2FFNY%2BhWIMzS10ACIQCGeyxPxhfyDAkshLLZaHqQU97mLTs5AMV565oejZwyPyqNBQi9%2F%2F%2F%2F
            EQ1piB7%2Bb7PmgKuEEvqERwiIY1A0yNzflxBVuMpYXi3FIMmULO9Bk%2BkVu0fw3CIvv4Aue4k6ltNvNzk73Yz3SUxO1A%2Bth7JzmJ066fuHSNh30jONMwuNj%2FWuyw0ruYLRsmRRgmpSIvWma5T0kFxxWgslfSOzcVTl90fKJ5insVGy3Yb
            NQ6QzjfkB1XUWh%2BByxwbu4AxLp3gdquurupZZbigG12KKNEulETs9H4xq7zIx9ZVcQssDg%2FFnDHiyl%2B1Z6n92ftc9Bd%2BOV%2BveuTkDHCdi4F8MSeYTAAJ9e7vk6XyfQ0dQl7owmJliqrbLbiBRW0TMafOzgYUAMKlY22sHahLl%2B0
            hI7cQBpB2QwgYZFD3mEDynK91Z4uO51UaWFp3Fgh9%2FjTCB44BXnthTlmw3CWAs5VS5tZhKRO2NMHyDbBaxf5fWt5FkL4NcHg25ajF5HOt5vwi%2BShnJaDkpt8VW1ly%2B09pmliPEgAoyFWDVx1okMqI4YndLCvGK89u2HkLkJswWziJVAmr
            x%2B8BfHl%2FZA1bq1GzouX1WKKQyFfGa8%2B2vQPdtJcsX7OGcCKic3F0CgsfxG6P4DQQOctUs03QQ1iI0tX0%2FwVTp8V3HDnGxWAvuxS85BPVbJ3jfIyvD3%2FtrUj54fcXrI7pPU7D2hQku5eixTGZQTFfinsGto%2FBEuAnTpgbJ3VF%2F
            5NOsPq1d0WVRPR97rzkWl8wEA5%2BFa1GulHc5%2B%2FJlVKLTbxDjsnKjPILyHbPkKAKd%2FtXrqIAh7%2FR3apZI38yWbOZs2zUp0YbxttZfUlv4upEv3EcEssI%2Fw2%2FPJoPkvKFv44lxIXItTb2Za%2FCNgAfCadZpx7rSkywSQK%2FuX
            X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20240426T120643Z&X-Amz-SignedHeaders=host&X-Amz-Expires=300&X-Amz-Credential=ASIAWXLNSFLSVWTPJLFK%2F20240426%2Fap-northeast-1%2Fs3%2Faws4_r
            X-Amz-Signature=a782bce98881900ab144c11fe4c00f786cfe0bbbff27e3754e049c1889e820a5",
        "imageUrl": "https://static.qa1fdg.net/client_upload/c2c/chat/20240426/e49f32b6b337445cbd5a43f427174194_20240426120643.jpg"
    },
    "success": true
}
```

Then use the following sample code to upload picture to “preSignedUrl” (Python example)

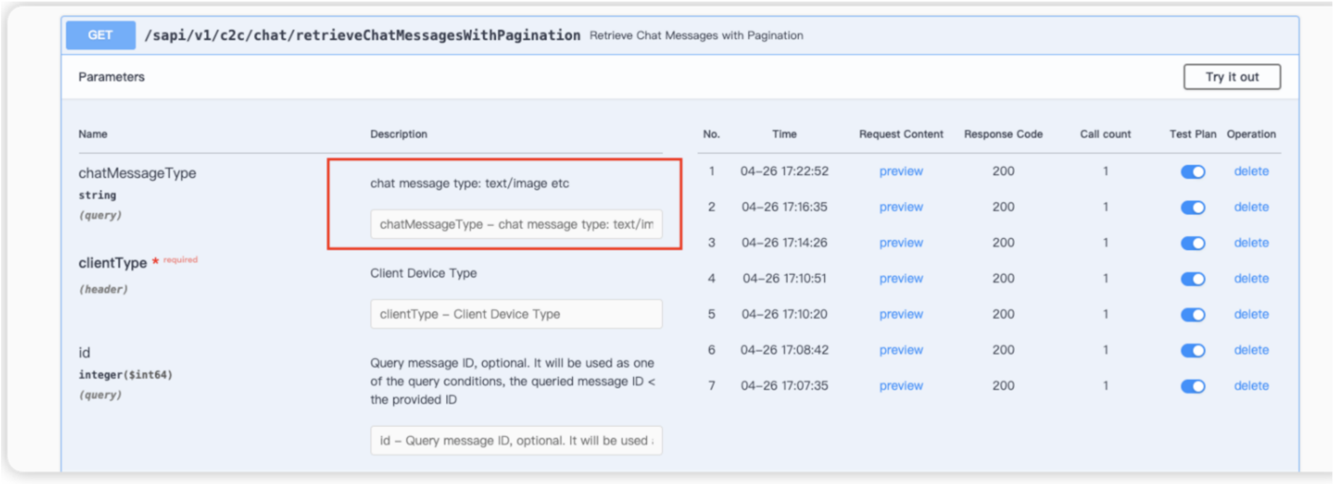
```
import requests

def upload_file_using_presigned_url(presigned_url, file_path):
        with open(file_path, 'rb') as file:
                response = requests.put(presigned_url, data=file)
                if response.status_code == 200:
                        print("File uploaded successfully.")
                else:
                        print("Failed to upload file.")

if __name__ == "__main__":
        file_to_upload = 'test.jpg'
        presigned_url = 'https://tk-qa1-s3-bucket.s3.ap-northeast-1.amazonaws.com/tmp/client_upload/c2c/chat
/20240426/e49f32b6b337445cbd5a43f427174194_20240426120643.jpg?X-Amz-Security-
Token=IQoJb3JpZ2luX2VjEHMaDmFwLW5vcnRoZWFzdC0xIkcwRQIgN%2FZs2wSCQev34zxU4rV2v2TwEWsrR%2FFNY%
2BhWIMzS10ACIQCGeyxPxhfyDAkshLLZaHqQU97mLTs5AMV565oejZwyPyqNBQi9%2F%2F%2F%2F%2F%2F%2F%2F%2F%
2F8BEAUaDDQ2MjQ3NTYzNTQyOSIMoRnEQ1piB7%2Bb7PmgKuEEvqERwiIY1A0yNzflxBVuMpYXi3FIMmULO9Bk%
2BkVu0fw3CIvv4Aue4k6ltNvNzk73Yz3SUxO1A%2Bth7JzmJ066fuHSNh30jONMwuNj%
2FWuyw0ruYLRsmRRgmpSIvWma5T0kFxxWgslfSOzcVTl90fKJ5insVGy3YbPCXpvveJcRDM9%
2BtD2AeB6gOBgddxnkCpmbMBSQ3lrg9CcNQ6QzjfkB1XUWh%2BByxwbu4AxLp3gdquurupZZbigG12KKNEulETs9H4xq7zIx9ZVcQssDg%
2FFnDHiyl%2B1Z6n92ftc9Bd%2BOV%2BveuTkDHCdi4F8MSeYTAAJ9e7vk6XyfQ0dQl7owmJliqrbLbiBRW0TMafOzgYUAMKlY22sHahLl%
2B04UGv1odIcTXaghCzdlVxkRzMbFNJZw7X%2BqbuBeqQctXo6hI7cQBpB2QwgYZFD3mEDynK91Z4uO51UaWFp3Fgh9%
2FjTCB44BXnthTlmw3CWAs5VS5tZhKRO2NMHyDbBaxf5fWt5FkL4NcHg25ajF5HOt5vwi%2BShnJaDkpt8VW1ly%
2B09pmliPEgAoyFWDVx1okMqI4YndLCvGK89u2HkLkJswWziJVAmrrxlBw42IiGsQkl%2FFlpZN%2FO55YloCbTOlRsjptGNxCwqx%2B8BfHl%
2FZA1bq1GzouX1WKKQyFfGa8%2B2vQPdtJcsX7OGcCKic3F0CgsfxG6P4DQQOctUs03QQ1iI0tX0%
2FwVTp8V3HDnGxWAvuxS85BPVbJ3jfIyvD3%2FtrUj54fcXrI7pPU7D2hQku5eixTGZQTFfinsGto%2FBEuAnTpgbJ3VF%
2FTkYMMOWcrrEGOpoBBlMxIgSCkLMO0KiIRs%2BEjC74AvKlg5NOsPq1d0WVRPR97rzkWl8wEA5%2BFa1GulHc5%2B%
2FJlVKLTbxDjsnKjPILyHbPkKAKd%2FtXrqIAh7%2FR3apZI38yWbOZs2zUp0YbxttZfUlv4upEv3EcEssI%2Fw2%
2FPJoPkvKFv44lxIXItTb2Za%2FCNgAfCadZpx7rSkywSQK%2FuXEorTuMJTY73w%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20240426T120643Z&X-Amz-SignedHeaders=host&X-Amz-Expires=300&X-Amz-Credential=ASIAWXLNSFLSVWTPJLFK%
2F20240426%2Fap-northeast-1%2Fs3%2Faws4_request&X-Amz-
Signature=a782bce98881900ab144c11fe4c00f786cfe0bbbff27e3754e049c1889e820a5'
        upload_file_using_presigned_url(presigned_url, file_to_upload)
```

Lastly, send the corresponding "imageUrl" in p2p chat.

How to download an image using SAPI:

Use /sapi/v1/c2c/chat/retrieveChatMessagesWithPagination

Filter "image" type in chatMessageType



/sapi/v1/c2c/chat/retrieveChatMessagesWithPagination Retrieve Chat Messages with Pagination

Parameters

Name	Description
chatMessageType string (query)	chat message type: text/image etc chatMessageType – chat message type: text/im
clientType * required (header)	Client Device Type clientType – Client Device Type
id integer($int64) (query)	Query message ID, optional. It will be used as one of the query conditions, the queried message ID < the provided ID id – Query message ID, optional. It will be used

No.	Time	Request Content	Response Code	Call count	Test Plan	Operation
1	04-26 17:22:52	preview	200	1	●	delete
2	04-26 17:16:35	preview	200	1	●	delete
3	04-26 17:14:26	preview	200	1	●	delete
4	04-26 17:10:51	preview	200	1	●	delete
5	04-26 17:10:20	preview	200	1	●	delete
6	04-26 17:08:42	preview	200	1	●	delete
7	04-26 17:07:35	preview	200	1	●	delete

Then use the following sample code to download picture (Python example)

```
import requests

def download_file(url, save_path):
        response = requests.get(url)
        if response.status_code == 200:
                with open(save_path, 'wb') as file:
                        file.write(response.content)
                        print("File downloaded successfully.")
        else:
                print(f"Failed to download file. Status code: {response.status_code}")

if __name__ == "__main__":
        file_url = "https://static.qa1fdg.net/client_upload/c2c/chat/20240426
/e49f32b6b337445cbd5a43f427174194_20240426120643.jpg"
        save_file_path = "1_downloaded.jpg"
        download_file(file_url, save_file_path)
```