Request For Stream API Documentation

Table of contents:

Introduction

Overview

In today's fast-paced financial markets, efficient and accurate trade quoting is critical for both traders and liquidity providers. One solution that has become increasingly popular is the **Request for Stream (RFS)** service.

The RFS allows market participants to receive real-time price level updates for specific trading instruments, enabling better decision-making and more dynamic market engagement.

Some advantages of using RFS

Real-Time Pricing

Your will receive continuous, real-time price levels. This ensures that you can react immediately to favorable market conditions.

Transparency

What you see is what you can get from our liquidity providers.

Improved Liquidity

As we are merging all the liquidity from different providers, you will get access to a deeper liquidity pools, resulting in a better execution.

How it works

- Users are granted a balance to use the RFS called `Exposure`
- Users stream all the liquidity currently available through our websockets by subscribing to the `User order book` channel.
- Users will need to also subscribe to the `User Trade` channel to receive the futures events on any trade done.

- Users can send orders through our REST api and get the result of the execution.
- If an order get matches in the order book, the RFS system will send out a trade event on the `User Trade` channel.
- Each filled order will impact the `Exposure`. Users can check their current exposure with `Account Exposure` endpoint.
- Users will need to settle their exposure with Flowdesk before reaching the limit to avoid any interruption of service.

Quick start

Today our service is only available in REST and Websocket protocol.

Please contact Flowdesk's team to get your credentials. You will receive 4 information:

- client_id
- client_secret
- audience
- client_name

How to use the credentials

`client_id`, `client_secret` and the `audience` are used in the authentication part. `client_name` will be used along the `jwt` you will get from authentication part to stream or execute with the API.

Authentication

Base URLs

Here are the base URLs for the authentication API environments:

PRODUCTION

```
https://rfs-prod-authentication-auth-domain.auth.eu-west-1.amazoncognito.com
```

DEVELOPMENT

```
https://rfs-dev-authentication-auth-domain.auth.eu-west-1.amazoncognito.com
```

Endpoint

If you don't have your credentials yet, please contact us.

GET

```
/oauth2/token
```

Request parameters

Parameter	Type	Required	Description
client_id	string	Yes	Your client id
client_secret	string	Yes	Your client secret
audience	string	Yes (Production only)	Your audience. Not required on Development

PRODUCTION AUDIENCE

```
https://rfs-customer-1.flowdesk.com
```

Example

```
 curl --request POST \
  --url 'https://rfs-prod-authentication-auth-domain.auth.eu-
west-1.amazoncognito.com/oauth2/token' \
  --header 'content-type: application/x-www-form-urlencoded' \
  --data grant_type=client_credentials \
  --data client_id=<CLIENT_ID> \
  --data client_secret=<CLIENT_SECRET> \
  --data audience=https://rfs-customer-1.flowdesk.com
```

Response

Success

```
{
    "access_token": "<your_access_token>",
    "expires_in": 86400,
    "token_type": "Bearer"
}
```

Failure

```
{
    "error": "access_denied"
}
```

General information

Base URLs

Here are the base URLs for the Websocket API environments:

PRODUCTION

```
wss://ws.trading.flowdesk.co
```

DEVELOPMENT

```
wss://ws.trading.dev.flowdesk.co
```

📄 Account Exposure

Request parameters

📄 RFS Trades History

Returns trades executed specifically through the RFS (Request for Stream) service. This endpoint provides a subset ...

📄 OTC Trades

Returns all OTC trades regardless of execution method. This endpoint provides a broader scope than RFS-only trade...

📄 OTC Settlements

Returns settlement data for OTC trades, including transfer information and settlement status. This endpoint helps tra...

Account Exposure

GET

`/api/v1/query/clients/<client_name>/exposure`

Request parameters

No parameters needed. Make sure to send the Bearer token in the header.

Example

```
curl --location
'https://api.rfs.flowdesk.co/api/v1/query/clients/<client_name>/e
\
--header 'accept: application/json, text/plain, */*' \
--header 'authorization: Bearer XXXX' \
--header 'content-type: application/json' \
--data ''
```

Response

Success ack

```
{
    "data": {
        "customer_infra_name": "<client_name>",
        "locked": "2",
        "available": "999998",
        "limit": "1000000"
    }
}
```


Failure


```
{
    "exp": "token expired"
}
```

RFS Trades History

Returns trades executed specifically through the RFS (Request for Stream) service. This endpoint provides a subset of OTC trades that were processed via the RFS workflow.

USE CASE

Use this endpoint when you need to track trades that originated from RFS requests, offering detailed execution information including matches and order details.

GET

```
/api/v1/query/clients/<client_name>/rfs-trades
```

Authentication

You will need your `client_name` and the Bearer token you can get from the authentication section.

Request Parameters

Parameter	Type	Required	Description
startTime	Long	No	Timestamp in milliseconds to filter trades from (INCLUSIVE). Default: 24 hours ago
currency1	String	No	Base currency (e.g., BTC)
currency2	String	No	Quote currency (e.g., USDT)
exchangeType	String	No	Market type, only `spot` supported
exchange	String	No	Exchange identifier
channelType	String	No	The channel type, only `public` and `private` are supported. Defaults to `private` for retro-compatibility
limit	Int32	No	The size of data result page. Default `100`. Max `500`.
page	Int32	No	The number of the page returned

Example Request

```
curl --location
'https://api.rfs.flowdesk.co/api/v1/query/clients/<client_name>/r
trades?currency1=BTC&currency2=USDT' \
--header 'accept: application/json, text/plain, */*' \
--header 'authorization: Bearer XXXX' \
--header 'content-type: application/json' \
--data ''
```

Response

Success Response (channelType=private)


```
{
    "client_id": "{client_id}",
    "data": [
        {
            "id": "0196198b-fa76-7e70-8321-30d5acf398b3",
            "price": "77326.9",
            "size": "0.2",
            "side": "buy",
            "timestamp": 1744728706089195638,
            "marketKey": {
                "exchange": "internal",
                "exchange_type": "spot",
                "currency1": "BTC",
                "currency2": "USDT"
            },
            "tradeType": "maker",
            "orderId": "0196198b-fa76-7e70-8321-30b67740f72e",
            "status": "SETTLED",
            "matches": [
                {
                    "id": "0196198b-fa76-7e70-8321-
30cfcdc96896",
                    "price": "77326.9",
                    "size": "0.2",
                    "order_id": "9c31a349-f60a-465b-a8f6-
ec6b5b74dab2",
                    "provider": "flowdesk"
                }
            ]
        }
    ]
}
```


Success Response (channelType=public)

```
{
    "client_id": "{client_id}",
    "data": [
        {
                "currency1": "BTC",
                "currency2": "USDT",
                "price": "115955.26",
                "size": "0.1",
                "side": "buy",
                "timestamp": 1758041876802887357
        },
        {
                "currency1": "XRP",
                "currency2": "USDC",
                "price": "2.66",
                "size": "25",
                "side": "buy",
                "timestamp": 1758027757464091310
        },
        {
                "currency1": "BTC",
                "currency2": "USDT",
                "price": "115870.91",
                "size": "0.1",
                "side": "buy",
                "timestamp": 1758041280061009384
        },
        {
                "currency1": "BTC",
                "currency2": "USDT",
                "price": "115384.61",
                "size": "0.1",
                "side": "buy",
                "timestamp": 1758027757578955000
        }
        ]
}
```


Error Response

```
{
    "exp": "token expired"
}
```

Response Fields (private)

Field	Type	Description
`id`	String	Unique trade identifier
`price`	String	Execution price as string
`size`	String	Trade size/quantity
`side`	String	Trade side: `buy` or `sell`
`timestamp`	Long	Trade execution timestamp in nanoseconds
`marketKey`	Object	Market information including exchange and currency pair
`tradeType`	String	Trade type: `maker` or `taker`
`orderId`	String	Associated order identifier
`status`	String	Status of the trade. Possible values: `BOOKED` (trade is registered but settlement process has not started yet), `PENDING` (settlement process has started but is not finished yet), `SETTLED` (settlement process is finished)
`matches`	Array	Array of matching orders that filled this trade

Response Fields (public)

Field	Type	Description
`currency1`	String	Base currency
`currency2`	String	Quote currency
`price`	String	Execution price as string
`size`	String	Trade size/quantity
`side`	String	Trade side: `buy` or `sell`
`timestamp`	Long	Trade execution timestamp in nanoseconds

OTC Trades

Returns all OTC trades regardless of execution method. This endpoint provides a broader scope than RFS-only trades, including all OTC trading activity.

USE CASE

Use this endpoint when you need a complete view of all OTC trading activity for analysis, reporting, or reconciliation purposes.

GET

```
/api/v1/query/clients/<client_name>/otc-trades
```

Authentication

You will need your `client_name` and the Bearer token you can get from the authentication section.

Request Parameters

This endpoint returns all OTC trades for the specified client without filtering options.

Example Request

```
curl --location
'https://api.rfs.flowdesk.co/api/v1/query/clients/<client_name>/o
trades' \
--header 'accept: application/json, text/plain, */*' \
--header 'authorization: Bearer XXXX' \
--header 'content-type: application/json'
```

Response

Success Response


```
{
    "otcTrades": [
        {
            "side": "buy",
            "price": 77326.9,
            "size": 0.2,
            "customerName": "client_name",
            "currency1": "BTC",
            "currency2": "USDT",
            "status": "filled",
            "bookedAt": "2025-01-24T14:25:06.089195638Z"
        }
    ]
}
```


Error Response


```
{
    "exp": "token expired"
}
```


PERSPECTIVE

> All fields are returned from the client's perspective. For example, `side: sell` on a BTC trade means the client sold BTC to Flowdesk.

Response Fields

Field	Type	Description
`side`	String	Trade side from the client's perspective: `buy` (client buys from Flowdesk) or `sell` (client sells to Flowdesk)
`price`	Number	Execution price as number
`size`	Number	Trade size/quantity as number
`customerName`	String	Client identifier name
`currency1`	String	Base currency (e.g., BTC)
`currency2`	String	Quote currency (e.g., USDT)
`status`	String	Trade status (see Trade Status values below)
`bookedAt`	String	Trade booking timestamp in ISO 8601 format

Enum Values

Trade Status

- **`BOOKED`** - The OTC trade is booked but not yet associated to a settlement, or has been associated to a rejected settlement
- **`PENDING`** - The OTC trade is associated to a pending settlement
- **`SETTLED`** - The OTC trade is associated to a closed settlement

- `REJECTED` - The OTC trade has been rejected

OTC Settlements

Returns settlement data for OTC trades, including transfer information and settlement status. This endpoint helps track the settlement process for completed trades.

USE CASE

Use this endpoint to monitor settlement status, track transfers, and reconcile trade settlements for accounting and compliance purposes.

GET

```
/api/v1/query/clients/<client_name>/otc-settlements
```

Authentication

You will need your `client_name` and the Bearer token you can get from the authentication section.

Request Parameters

Parameter	Type	Required	Description
startDate	DateTime	Yes	Start date to filter settlements from (ISO 8601 format: `2025-01-01T00:00:00Z`)
endDate	DateTime	Yes	End date to filter settlements to (ISO 8601 format: `2025-01-31T23:59:59Z`)

Example Request

```
curl --location
'https://api.rfs.flowdesk.co/api/v1/query/clients/<client_name>/o
settlements?startDate=2025-01-01T00:00:00Z&endDate=2025-01-
31T23:59:59Z' \
--header 'accept: application/json, text/plain, */*' \
--header 'authorization: Bearer XXXX' \
--header 'content-type: application/json'
```

Response

Success Response

```
{
    "data": [
        {
            "id": "ff1cbd61-f949-4f7b-9eb1-53df018ad9ca",
            "customerName": "client_name",
            "status": "CLOSED",
            "trades": [
                "39ca97e8-a674-4ad1-8605-b37a610f4d9f"
            ],
            "transfers": [
                {
                    "externalReferenceId": "ref-001",
                    "executionDate": "2025-01-24T14:25:06Z",
                    "amount": "15465.38",
                    "currency": "USDT",
                    "direction": "OUTGOING",
                    "transferType": "ONCHAIN"
                },
                {
                    "externalReferenceId": "ref-002",
                    "executionDate": "2025-01-24T14:25:06Z",
                    "amount": "0.2",
                    "currency": "BTC",
                    "direction": "INCOMING",
                    "transferType": "ONCHAIN"
                }
            ],
            "createdAt": "2025-01-24T14:25:06.089195638Z",
            "updatedAt": "2025-01-24T14:30:12.123456789Z"
        }
    ]
}
```


Error Response


```
{
    "exp": "token expired"
}
```


PERSPECTIVE

> All fields are returned from the client's perspective. An `OUTGOING` transfer means the client sends funds to Flowdesk; an `INCOMING` transfer means the client receives funds from Flowdesk.

Response Fields

Settlement Object

Field	Type	Description
`id`	String	Unique settlement identifier
`customerName`	String	Client name associated with the settlement
`status`	String	Settlement status (see Settlement Status values below)
`trades`	Array[String]	List of trade IDs included in this settlement
`transfers`	Array[Transfer]	List of transfer objects for this settlement
`createdAt`	String	Settlement creation timestamp in ISO 8601 format
`updatedAt`	String	Settlement last update timestamp in ISO 8601 format

Transfer Object

Field	Type	Description
`externalReferenceId`	String	External reference identifier for the transfer
`executionDate`	String | null	Transfer execution date (may be null if not executed)
`amount`	String	Transfer amount as string
`currency`	String	Currency of the transfer
`direction`	String	Transfer direction (see Transfer Direction values below)
`transferType`	String	Type of transfer (see Transfer Type values below)

Enum Values

Settlement Status

- **`OPEN`** - Settlement is still in progress
- **`CLOSED`** - Settlement has been completed successfully
- **`REJECTED`** - Settlement was rejected and will not be processed

Transfer Direction

From the client's perspective:

- **`INCOMING`** - Client receives funds from Flowdesk
- **`OUTGOING`** - Client sends funds to Flowdesk

Transfer Type

- `ONCHAIN` - On-chain cryptocurrency transfer
- `OFFCHAIN` - Off-chain transfer (internal system)
- `BANK` - Bank wire transfer
- `PREFUNDINGDEBIT` - Pre-funding debit adjustment

Create Order

If you don't have your credentials yet, please contact us.

POST

```
/api/v1/rfs/clients/<client_name>/rfs-orders
```

Request parameters

You will need your client_name and the Bearer token you can get from the authentication part.

Parameter	Type	Required	Description
side	string	Yes	the side or the order: `sell` or `buy`
price	string	No	the price of your order (required for limit only)
size	string	Yes	the size of your order
order_type	string	Yes	only `limit` or `market` supported
time_in_force	string	No	only `fok` and `ioc` supported for limit order
market	Market	Yes	the Market of your order
-- base	string	Yes	the base currency
-- quote	string	Yes	the quote currency
-- type	string	Yes	the market type, only `spot` supported

Example

Limit Order (with time_in_force)

```
curl -X POST --location
'https://api.rfs.flowdesk.co/api/v1/rfs/clients/<client_name>/rfs
orders' \
--header 'Content-Type: application/json' \
 --header 'Authorization: Bearer <token_here>' \
--data '{
    "side": "sell",
    "price": "0.4",
    "size": "50",
    "order_type": "limit",
    "time_in_force": "fok",
    "market": {
        "base": "XRP",
        "quote": "USDT",
        "type": "spot"
    }
}'
```

Market Order (without time_in_force)

```
curl -X POST --location
'https://api.rfs.flowdesk.co/api/v1/rfs/clients/<client_name>/rfs
orders' \
--header 'Content-Type: application/json' \
 --header 'Authorization: Bearer <token_here>' \
--data '{
    "side": "buy",
    "size": "50",
    "order_type": "market",
    "market": {
        "base": "XRP",
        "quote": "USDT",
        "type": "spot"
    }
}'
```

Response

Limit Order Response

Success ack (Limit Order)


```
{
  "message": {
    "id": "01924d61-8f11-7e22-a742-0635a20676f4",
    "side": "sell",
    "price": "0.5248800000",
    "avg_price": "0.5248800000",
    "size": "50",
    "order_type": "limit",
    "filled_size": "0"
  },
  "code": 201
}
```

Market Order Response

Success ack (Market Order)


```
{
  "message": {
    "id": "01924d61-8f11-7e22-a742-0635a20676f4",
    "side": "buy",
    "avg_price": "0.5248800000",
    "size": "50",
    "order_type": "market",
    "filled_size": "50"
  },
  "code": 201
}
```


Error Responses

Failure FOK


```
{
    "message": "Order not fully executed",
    "code": 400
}
```


Failure IOC


```
{
    "message": "Order not executed",
    "code": 400
}
```


Failure exposure

```
{
    "message": "Exposure limit reached",
    "code": 422
}
```


Failure ack


```
{
  "message": {
    "time_in_force": [
      {
        "code": "Wrong time_in_force param",
        "message": "Only FOK or IOC are allowed",
        "params": {
          "value": "post_only"
        }
      }
    ]
  },
  "code": 400
}
```

General information

Base URLs

Here are the base URLs for the Websocket API environments:

PRODUCTION

```
wss://ws.trading.flowdesk.co
```

DEVELOPMENT

```
wss://ws.trading.dev.flowdesk.co
```

📄 Orderbook

How to maintain local order book

📄 Trades

Request parameters

Orderbook

How to maintain local order book

The first message you will receive is the acknowledgment message if the subscription is successful.

Then you will receive a second message with the snapshot of all the orders in the current order book.

Starting from this message, you will receive only **update events**.

You will need to apply these updates on the initial order book snapshot received.

Request parameters

You will need your client_name and the Bearer token you can get from the authentication part.

Parameter	Type	Required	Description
jwt	string	Yes	jwt you get in the authentication section

Subscription Message

```
{
  "event": "subscribe",
  "data": {
    "channels": [
      {
        "currency1": "<currency1>",
        "currency2": "<currency2>",
        "marketData": "<market_data>",
        "exchangeType": "<exchange_type>",
        "service": "rfs",
        "clientName": "<client_name>"
      }
    ]
  }
}
```

Body	Type	Required	Description
currency1	string	Yes	base currency
currency2	string	Yes	quote currency
marketData	string	Yes	the data you want to subscribe to, `orderbook` or `trades`
exchangeType	string	Yes	the market type, only `spot` supported now
service	string	Yes	service to call, only `rfs` supported now
clientName	string	Yes	the client name you received from us

Example

```
wscat -c 'wss://ws.trading.flowdesk.co/?jwt=XXXXXXX' -x
'{"event":"subscribe","data":{"channels":[{"currency1":"
<currency1>","currency2":"<currency2>","marketData":"
<market_data>","exchangeType":"
<exchange_type>","service":"rfs","clientName":"
<client_name>"}]}}' -w 60
```

Response

Subscription Success ack


```
{
  "event": "subscribe-ack",
  "data": [
    {
      "requestedChannel": {
        "currency1": "XRP",
        "currency2": "USDT",
        "marketData": "orderbook",
        "exchangeType": "spot",
        "service": "rfs",
        "clientName": "test-name"
      },
      "resolvedChannels": [
        {
          "currency1": "XRP",
          "currency2": "USDT",
          "marketData": "orderbook",
          "exchangeType": "spot",
          "service": "rfs",
          "clientName": "test-name"
        }
      ]
    }
  ]
}
```

Failure ack

```
{
  "error": "bad request, could not identify user channels"
}
```

Snapshot Message

```
{
  "event": "snapshot",
  "channel": {
    "currency1": "XRP",
    "currency2": "USDT",
    "marketData": "orderbook",
    "exchangeType": "spot",
    "service": "rfs",
    "clientName": "test-name"
  },
  "data": {
    "asks": [
      [
        "0.64075",
        "8316"
      ],
      [
        "0.64086",
        "27754"
      ]
    ],
    "bids": [
      [
        "0.53667",
        "144165"
      ],
      [
        "0.53658",
        "126887"
      ]
    ]
  }
}
```


Update Message with only an ask

```
{
  "event": "update",
  "channel": {
    "currency1": "XRP",
    "currency2": "USDT",
    "marketData": "orderbook",
    "exchangeType": "spot",
    "service": "rfs",
    "clientName": "test-name"
  },
  "data": {
    "asks": [
      [
        "0.67705",
        "5417"
      ]
    ],
    "bids": []
  }
}
```

Trades

Request parameters

You will need your client_name and the Bearer token you can get from the authentication part.

Parameter	Type	Required	Description
jwt	string	Yes	jwt you get in the authentication section

Subscription Message

```
{
  "event": "subscribe",
  "data": {
    "channels": [
      {
        "currency1": "<currency1>",
        "currency2": "<currency2>",
        "marketData": "<market_data>",
        "channelType": "<channel_type>",
        "exchangeType": "<exchange_type>",
        "service": "rfs",
        "clientName": "<client_name>"
      }
    ]
  }
}
```

Body	Type	Required	Description
currency1	string	Yes	base currency
currency2	string	Yes	quote currency
marketData	string	Yes	the data you want to subscribe to, `orderbook` or `trades`
channelType	string	No	the channel type, only `public` and `private` are supported. Defaults to `private` for retro-compatibility
exchangeType	string	Yes	the market type, only `spot` supported now
service	string	Yes	service to call, only `rfs` supported now
clientName	string	Yes	the client name you received from us

Example

```
wscat -c 'wss://ws.trading.flowdesk.co/?jwt=XXXXXXX' -x
'{"event":"subscribe","data":{"channels":[{"currency1":"
<currency1>","currency2":"<currency2>","marketData":"
<market_data>","exchangeType":"
<exchange_type>","service":"rfs","clientName":"
<client_name>"}]}}' -w 60
```

Response

Subscription Success ack

```
{
  "event": "subscribe-ack",
  "data": [
    {
      "requestedChannel": {
        "currency1": "XRP",
        "currency2": "USDT",
        "marketData": "trades",
        "exchangeType": "public",
        "exchangeType": "spot",
        "service": "rfs",
        "clientName": "test-name"
      },
      "resolvedChannels": [
        {
          "currency1": "XRP",
          "currency2": "USDT",
          "marketData": "trades",
          "exchangeType": "public",
          "exchangeType": "spot",
          "service": "rfs",
          "clientName": "test-name"
        }
      ]
    }
  ]
}
```


Failure ack


```
{
  "error": "bad request, could not identify user channels"
}
```


Update Message (Private)

```
{
  "event": "update",
  "channel": {
    "service": "rfs",
    "clientName": "test-name",
    "marketData": "trades",
    "channelType": "private",
    "exchangeType": "spot",
    "currency1": "XRP",
    "currency2": "USDT"
  },
  "data": {
    "createdAt": "2024-10-02T14:58:44.000Z",
    "executedAt": "2024-10-02T14:58:44.000Z",
    "internalID": "toto",
    "price": "100",
    "side": "buy",
    "size": "5",
    "source": "flowdesk",
    "type": "taker"
  }
}
```


Update Message (Public)

```
{
  "event": "update",
  "channel": {
    "service": "rfs",
    "clientName": "test-name",
    "marketData": "trades",
    "channelType": "public",
    "exchangeType": "spot",
    "currency1": "XRP",
    "currency2": "USDT"
  },
  "data": {
    "createdAt": "2024-10-02T14:58:44.000Z",
    "price": "120",
    "side": "buy",
    "size": "3",
  }
}
```