# REFINITIV CALLOUTS TRADE HISTORY API

**DRAFT**

Callouts Trade History API 1.0

REFINITIV®

# Version history

| Software version | Document version | Summary of Changes |
|---|---|---|
| Beta1 | 0.1 | Initial version of the Refinitiv Callouts Trade History API |

# Contents

About this document ..... 5
Intended readership ..... 5
Developer Environments ..... 6
Developer Community ..... 6
API Playground ..... 6
About the Trade History API ..... 7
Callouts Demo / UAT ..... 7
How to Use the Callouts Trade History API ..... 8
Environment Access ..... 8
Authentication ..... 8
Trade History Search Request Parameters ..... 8

# About this document

## Intended readership

This document is intended for Refinitiv Callouts client developers intending to use the Refinitiv Data Platform APIs to receive Trade History data for trading activity at their organisations on the Callouts platform.

It is intended for developers who are familiar with the general principles of APIs and assumes they are aware of their programming language, and accessing REST APIs.

# Developer Environments

Clients responsible for using Refinitiv Data Platform API services will require access to Refinitiv's Developer Community portal and API Playground website.

For access rights, please see your account manager.

## Developer Community

The Refinitiv Developer Community provides API support. It also offers self-service learning for performing authentication and setting up a development environment using Python and Postman. This is through documentation and tutorials.

The portal enables you to connect with other API experts and users, and exchange information through community Q & A forums. You can browse or track issues and solutions for your relevant

products.

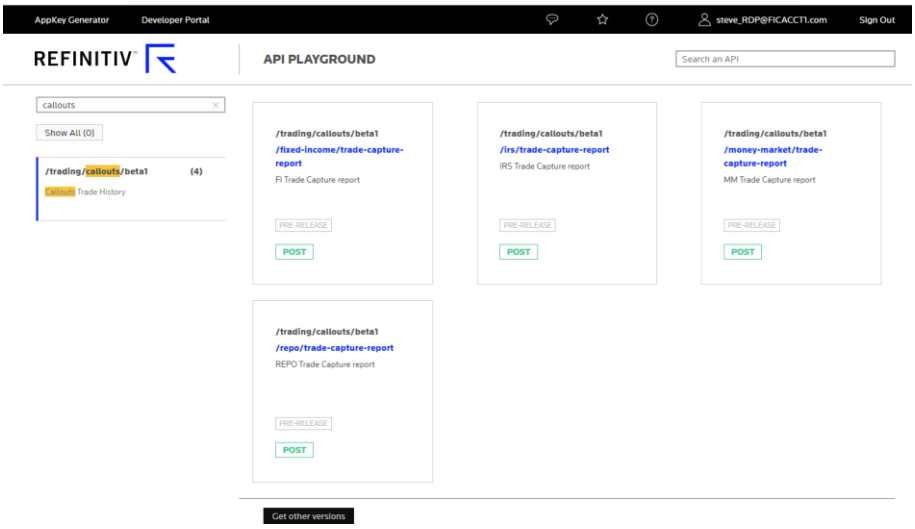
## API Playground

The API Playground provides access to Refinitiv's RDP APIs via a Graphical User Interface (GUI).

It acts as an interactive reference guide and tool, which allows you to test and explore the APIs, send requests, view responses, and review the latest API documentation and download swagger definitions.

For details on the Developer Community and API Playground, as well as for general information on using RDP, please see the RDP API Getting Started Guide. This is located on the Developer Community website.

To access the Callouts Trade History api in the API playground, open the url in a web browser and search for 'Callouts' – this will return the Trade History interfaces for each of the Callouts asset classes currently supporting the Trade History service.



# About the Trade History API

The Trade History API is a REST API that allows programmatic access to executed trade reports at the available organisations within the Callouts platform. It allows for simple querying of executed tickets by date/time range, or by specific ticket IDs where required.

The output can be downloaded in several different formats, and currently supports JSON, XML and CSV data types.

The trade history data that is returned is in the Callouts Platform TradeCaptureReport structure as used by other FIX-based post trade feeds available on the Callouts platform – and generally follows the FIX 5.0 TradeCaptureReport specification with some custom fields needed for the specific Callouts trade data.

## Callouts Demo / UAT

The Callouts platform operates separate DEMO and UAT environments, which have their own separate API interfaces to access test trading data separately to production.

Access to those non-production environments is available separately through your Account Manager if required, and can be used to test your applications in a non-live trading environment.

# How to Use the Callouts Trade History API

The Trade History API is available for each of the Callouts venues, currently Fixed Income, Money Markets, IRS and Repos. Each of the APIs supports the same set of query options and parameters.

## Environment Access

There are several Callouts environments available depending on your accounts available products. Swapping between the different environments involves changing the base path to the Trade History API – the authentication mechanism, network access and data retrieval workflows are the same for all environments.

| Environment | Base URL |
|---|---|
| **Production** | https://api.refinitiv.com/trading/callouts/ |
| **Demo** | https://api.refinitiv.com/trading/callouts-demo/ |
| **UAT** | https://api.refinitiv.com/trading/callouts-uat/ |

## Authentication

Access to the RDP API service is managed by retrieving an oAuth token for your account, and providing that as an Authorization header on following requests.

The request to authenticate will require the credentials for your service access account, and an ApplicationID for your application that your Account Manager will help provide.

The authentication request will provide an access token for use on future requests, as well as a refresh token that can be used to renew the access token when it expires. For full details on how to authenticate and manage your applications connection to the API Gateway please consult the Refinitiv Data Platform Getting Started Guide on the developer portal.

## Trade History Search Request Parameters

The Trade History API is designed to support two primary workflows for retrieving data – either search by data, or search by ID.

### Selecting Output Format

The API allows the caller to choose the requested data format by setting an ‘Accept’ header on the request.

| Output Format | Accept Header |
|---|---|
| **JSON** | application/json |
| **XML** | text/xml |
| **CSV** | text/csv |

## Paging through large result sets

If the specified search results contains a large number of results, they will be chunked into individual pages. The default number of results per-page is currently set to 100, but can be overridden in individual search requests by setting the desired number of results per page in the search request, up to a maximum (currently 1000).

When there are more results that can fit into a given page, the responses to a JSON search will include a _links section that contains the urls to call to retrieve the additional results. The urls provided will indicate how to navigate to the first and last pages, as well as the previous and next set of results using a HTTP GET request. The relevant parameters from those URLs can be transposed into HTTP POST options as required.

If there are no further pages, the next link will be set to null to indicate that there are no more remaining. An example of a search that has additional pages is as follows:

```
    "_links": {
        "self": null,
        "first": {
            "href": "http://api.refinitiv.com/trading/callouts/fixed-income/v1/trade-capture-report?startDate=20210101000000&endDate=20210301000000&”
        },
        "prev": {
            "href": " http://api.refinitiv.com/trading/callouts/fixed-income/v1/trade-capture-report?startDate=20210101000000&endDate=20210301000000&"
        },
        "next": {
            "href": " http://api.refinitiv.com/trading/callouts/fixed-income/v1/trade-capture-report?startDate=20210101000000&endDate=20210301000000&page=1"
        },
        "last": {
            "href": " http://api.refinitiv.com/trading/callouts/fixed-income/v1/trade-capture-report?startDate=20210101000000&endDate=20210301000000&page=1"
        }
    }
```

Paging information is currently only available on JSON search types due to limitations in the other supported datatypes.

## Searching Tickets by Date Range

Searching for tickets by date range requires a start and end datetime for the date range. If the number of results is more than can fit on a single page of results, the calling application can page through the results until all have been retrieved.

The parameters can be passed in a HTTP POST body, or as URL parameters to any of the product search endpoints.

### HTTP Get or Post

| Parameter | Format | Example Value | Description |
|---|---|---|---|
| **startDate** | yyyyMMddHHmmss | 20201101000000 | The start date/time for a date range search |

| Parameter | Format | Example Value | Description |
|---|---|---|---|
| **endDate** | yyyyMMddHHmmss | 20201201000000 | The end date/time for a date range search |
| **pageNumber** | Integer | 2 | The page number to return |
| **pageSize** | Integer | 100 | Number of results to return in each page |

Examples:

```
curl --location --request POST
'https://api.refinitiv.com/trading/callouts/v1/fixed-income/trade-capture-
report' --header 'Accept: application/json' --header 'Content-Type:
application/json' --header 'Authorization: Bearer API_ACCESS_TOKEN' --data-
raw '{

  "startDate": "20201001000000",

  "endDate": "20201031235959"

}'
```

```
curl --location 'https://api.refinitiv.com/trading/callouts/v1/fixed-
income/trade-capture-report?startDate=20201001000000&endDate=20201031235959'
--header 'Accept: application/json' --header 'Content-Type: application/json'
--header 'Authorization: Bearer API_ACCESS_TOKEN'
```

## Searching Tickets by ID

Searching for tickets by ticket ID requires a list of the ticket IDs to return. If the number of results is more than can fit on a single page of results, the calling application can page through the results until all have been retrieved.

### HTTP Post

| Parameter | Format | Example Value | Description |
|---|---|---|---|
| **tradeId** | Array | [“EUFI1234”,”EUFI3456”] | A list of individual ticket ids to search for |

Examples:

```
curl --location --request POST
'https://api.refinitiv.com/trading/callouts/v1/fixed-income/trade-capture-
report' --header 'Accept: application/json' --header 'Content-Type:
application/json' --header 'Authorization: Bearer API_ACCESS_TOKEN' --data-
raw '{

  "tradeId": ["EUFI1234","EUFI3456"]

}'
```

## HTTP Get

| Parameter | Format | Example Value | Description |
|---|---|---|---|
| **tradeId** | Comma-separated list | EUFI1234,EUFI3456 | A list of individual ticket ids to search for |

Examples:

```
curl --header 'Accept: application/json' --header 'Content-Type:
application/json' --header 'Authorization: Bearer API_ACCESS_TOKEN' -location
'https://api.refinitiv.com/trading/callouts/v1/fixed-income/trade-capture-
report?tradeId=EUFI1234,EUFI3456'
```

## Response Errors

The api uses HTTP status codes to indicate the overall status of a request, some errors may include a body that provides more detailed information about the failure. The following error codes and conditions are supported:

| Http Status Code | Description | Comments |
|---|---|---|
| **200** | Success | Response body should contain matching results (if any) |
| **400** | Bad Request | There was a problem processing the request. Further details may include:<br>Invalid Request – request parameters were invalid, this may include invalid paging parameters or a data range which is too large or invalid |
| **401** | Unauthorised | Access is denied, or the provided authentication token is not valid. |
| **403** | Forbidden | Access is denied – check with your account manager to make sure your application account has been onboarded fully. |
| **404** | Not Found | The url, or data, requested could not be found. |
| **405** | Request Method Not Allowed | The API supports GET and POST methods only |
| **406** | Not Acceptable | The application currently supports only application/json, text/xml and text/csv content type – requests that do not include one of those types will be rejected |
| **422** | Unprocessable Request | The request as provided could not be performed, please contact Refinitiv API support for assistance. |
| **429** | Too Many Requests | Trade history searches are rate limited, applications which exceed those limits will be rejected |
| **500** | Internal Server Error | An error occurred processing the request. Please try again later or contact Refinitiv API support. |

## Response Properties

A full description of the latest properties in the response is included in the API Playground 'Reference' section, and is largely based on the FIX 5.0 TCR specification with additional customer fields for additional information provided by the Callouts platform.

Major Properties Common to all Products

| Field Name | Req'd | Type | Comments |
|---|---|---|---|
| **rootParties** | Y | | List of counterparties involved in the ticket with combinations of **RootPartyID**, **RootPartyIDSource**, and **RootPartyRole** |

| Field Name | Req'd | Type | Comments |
|---|---|---|---|
| → r**ootPartyID** | C | String | Used to identify a party in the trade (e.g. Dealer). |
| → **rootPartyIDSource** | C | Char | Used to identify class or source of **RootPartyID** value (e.g. BIC).<br><br>Supported values:<br>'GeneralIdentifier' - Generally accepted market participant identifier |
| → **rootPartyRole** | C | Int | Identifies the type of **RootPartyID**<br><br>Supported values:<br>'EnteringFirm'<br>'ContraFirm' |
| **rootSubParties** | Y | | List of RootParty sub-identifiers. |
| → r**ootPartySubID** | C | String | Used to identify a party in the trade (e.g. Dealer). |
| → r**ootPartySubIDType** | C | Enum | Type of the counterparty sub-identifier<br>'LocationDesk'<br>'Firm'<br>'StartDealerId'<br>'StartDealerLoginEmail'<br>'ClientUserName'<br>'LastDealerName' |
| **aggressorIndicator** | Y | Boolean | Used to identify whether the order initiator is an aggressor or not in the trade.<br>true = Taker<br>false = Maker |
| **currency** | Y | String | Trade Reporting Currency |
| **settlCurrency** | Y | String | Trade Settlement Currency |
| **execID** | Y | String | Trade ID (Max size 45 char) |
| **tradeID** | Y | String | Trade ID.<br>Currently the same as ExecID |

| Field Name | Req'd | Type | Comments |
|---|---|---|---|
| **tradeReportId** | Y | String | Unique ID for the TradeCaptureReport received |
| **settlDate** | Y | Date | Settlement date |
| **settlType** | Y | Enum | Settlement type<br><br>'Regular'<br>'Cash'<br>'NextDay'<br>'FXSpotNextSettlement'<br>'WeekX' – X is 1 to 52<br>'MonthX' – X is 1 to 79<br>'YearX' – X is 1 to 99<br>'BrokenDate' |
| **transactTime** | Y | DateTime | Date and time of the quote execution. |
| **trdType** | Y | Enum | The type of trade that was executed<br>'OTC' |
| **tradeDate** | C | LocalMktDate | Indicates date of trading day. Absence of this field indicates current day (expressed in local time at place of trade). |
| **product** | Y | Enum | Identifies the product of the trade.<br>'FIXEDINCOME'<br>'MONEYMARKET'<br>'FINANCING' - Repo<br>'IRS' - INTEREST RATE SWAP |
| **securityType** | N | Enum | Currently always set to:<br>'CASH' |
| **symbol** | Y | String | Product identifier (symbol) |
| **lastPx** | N | Price | Last price received in the negotiation |
| **lastQty** | N | Quantity | Last quantity received in the negotiation |
| **quoteID** | N | String | ID for the accepted quote (if provided) |

| Field Name | Req'd | Type | Comments |
|---|---|---|---|
| **clientTradeID** | N | String | Client-provided trade ID (if available) |
| **market** | Y | String | Name of the market where the trade occurred |
| **settlType** | Y | | Indicates order settlement period. If present, **LegSettlDate** overrides this field. |
| trdCapRptSideGrp | Y | | List of Trade Capture Report side groups |
| →**Side** | Y | Char | Trade Side where Side and LegSide should be same.<br>Supported values:<br>Buy<br>Sell |
| **trdInstrumtLegGrp** | Y | | List of leg group items associated with this trade |
| → **instrumentLeg** | Y | | A leg |
| →→**legSide** | Y | Char | Trade Side where Side and LegSide should be same.<br>Supported values:<br>Buy<br>Sell |
| →→**legOrderQty** | Y | Qty | Trade Size. This represents the par, face or nominal value for FI instruments. |
| →→**securityAltID** | Y | | List of unique combinations of **legSecurityAltID** and **legSecurityAltIDSource** that identify this product |
| →→→**legSymbol** | N | String | Security symbol. |
| →→→**legSecurityID** | N | String | Security Identifier.<br>This identifier also exists in the SecurityAltIDGrp with the corresponding type |
| →→→**legSecurityIDType** | N | Enum | The type of identifier given in LegSecurityID.<br>Uses same values as SecurityAltIDSource |
| →→→ **legSecurityAltID** | Y | String | Multileg instrument's individual security identifier value of **LegSecurityAltIDSource** type (e.g. ISIN, RIC). |

| Field Name | Req'd | Type | Comments |
|---|---|---|---|
| →→→**legSecurityAltIDSource** | Y | String | Identifies class or source of the multileg instrument's individual security's a**ecurityAltIDSource** value. Required if s**ecurityAltID** is specified.<br><br>Supported values:<br>'ISIN'<br>'RIC'<br>'EJV' Asset ID |
| →→**legCurrency** | N | String | Currency associated with a particular Leg's quantity |
| →→**legSettlDate** | C | LocalMktDate | Specific date of trade settlement (Settlement Date) in YYYYMMDD format.<br>If present, this field overrides **SettlType** . This field is required if the value of **SettlType** is *Future*. |
| **SettlPartySubIDs** | N | | List of settlement subparties (if available) |
| →**SettlPartySubID** | N | Char | PartySubID value within a settlement parties component (e.g. Clearing Account for PartyRole 'Clearing Firm') |
| →**SettlPartySubIDType** | N | Char | Type of SettlPartySubID<br><br>Supported values:<br>'Cash Account'<br>'Security Account'<br>'Portfolio' |
| **trCfxReq** | N | | Refinitiv TCRs associate each leg with one or more allocations. For callouts there will always be one allocation. User defined fields are populated at allocation level. |
| → **trCfxReqDealtAmount** | N | Qty | The allocation quantity, should match the legOrderQty |
| → **trCfxReqUDFGrp** | N | | Number of User Defined Fields |
| →→ **trCfxReqUDFName** | N | String | Name of User Defined Field |

| Field Name | Req'd | Type | Comments |
|---|---|---|---|
| →→ trCfxReqUDFValue | N | String | Value assigned to User Defined Field of the associated name |
| →settlDetails | N | | List of settlement details provided |
| →→settlObligSource | N | Enum | Which side the settlement details refer to<br>'BUYER'<br>'SELLER' |
| →→settlParties | N | | Number of settlement parties |
| →→→settlPartyID | N | String | ID of the countparty that the settlement details refer to |

## Fixed Income Specific Items

| | | | |
|---|---|---|---|
| **trdInstrumtLegGrp** | Y | | List of leg group items associated with this trade |
| → instrumentLeg | Y | | A leg |
| →→legPrice | Y | Price | Traded Price Clean |
| →→numDaysInterest | N | Int | Accrued Days. Number of Days of Interest for convertible bonds and fixed income. Note value may be negative. |
| →→legIssuer | N | String | Multileg instrument's individual security's Issuer. Name of security issuer (e.g. International Business Machines, GNMA). |
| →→legIssueDate | N | LocalMktDate | Multileg instrument's individual leg security's IssueDate. The date on which a bond or stock offering is issued. It may or may not be the same as the effective date ("Dated Date") or the date on which interest begins to accrue ("Interest Accrual Date") |
| →→legCouponRate | N | Percentage | The rate of interest that, when multiplied by the principal, par value, or face value of a bond, provides the currency amount of the periodic interest payment. The coupon is always cited, along with maturity, in any quotation of a bond's price. |

| | | | |
|---|---|---|---|
| →→legMaturityDate | N | LocalMktDate | Multileg instrument's individual security's MaturityDate. |
| →→legRedemptionDate | N | LocalMktDate | Coupon Date. Return of investor's principal in a security. Bond redemption can occur before maturity date. |
| →→legCouponDayCount | N | Int | Day Count Basis. The day count convention used in interest calculations for a bond or an interest bearing security. |
| →→countryOfIssue | Y | String | Country of bond issuer |
| →→numDaysInterest | N | Number | Number of days interest |
| →→legCouponDate | N | Date | Coupon date |
| →→legCouponDayCount | N | Enum | Security coupon day basis type<br>OneOne<br>30/360 USA<br>30/360 SIA<br>30/360 M<br>30E/360<br>30E/360 ISDA<br>Act/360<br>Act/365 Fixed<br>Act/Act AFB<br>Act/Act ICMA<br>Act/Act ICMA Ultimo<br>Act/Act ISDA<br>Bus 252<br>30E+/360<br>Act/365L<br>NL/365<br>NL/360<br>Act/364 |
| →→legCouponRate | N | Price | Leg coupon rate |
| →→legCouponType | N | String | Coupon type<br>e.g. 'Fixed' |
| →→legFactor | N | Price | Leg factor |

| | | | |
|---|---|---|---|
| →→**legIssueDate** | N | Date | Issue date |
| →→**legIssuer** | N | String | Issuer |
| →→**legMaturityDate** | N | Date | Maturity date |
| →→**legSecurityDesc** | N | String | Displayable description of the security as would be shown on the ticket display header<br>e.g. 'GOOGL  3.375  02/25/24' |