



ETHOS AFP™ – REST API

Version 18.01.22.06.6.v2



DDCAP ETHOS AFP™™ RESTful API Endpoints and Functions

Authentication and authorisation

The ETHOS REST API uses “Basic HTTP Authentication”. Every request to the RESTful web service must include the username and password in the HTTP headers. All connection requests must be made over HTTPS, HTTP requests will be rejected.

Username and passwords for accessing the ETHOS AFP™™ are managed by DDCAP or Condor Trade. The access for using the REST API must be enabled by the ETHOS support team. Please ask your ETHOS support contact for assistance should you require EthosSupport@ddcap.co.uk

Environments

This guide has been written for testing the API on the DDCAP development environment <https://devafp.ddcap.com/api/> however the API can be requested to the same resource URI [/api/*] on the other ETHOS environments, listed below:

Testing / UAT URL

| | |
|--------------------------------------|---|
| DDCAP development environment | https://devafp.ddcap.com/api/ |
| Condor Trade development environment | https://devafp.condortrade.com/api/ |

Demonstration URL

| | |
|--|---|
| DDCAP demonstration environment | https://demoafp.ddcap.com/api/ |
| Condor Trade demonstration environment | https://demoafp.condortrade.com/api/ |

Live / Production URL

| | |
|-------------------------------------|---|
| DDCAP Production environment | https://afp.ddcap.com/api/ |
| Condor Trade Production environment | https://afp.condortrade.com/api/ |

Connections

The system architecture of the ETHOS AFP™ allows the routing of data between the platforms of DDCAP and Condor Trade. What this means in practice is that connections into ETHOS can be made to a single platform regardless of the request they are making as the application has built in logic to pass requests to the correct location.

API requests

To construct a REST API request, combine these components:

| Component | Description |
|----------------------------|--|
| The HTTP method | <ul style="list-style-type: none">• GET. Requests data from a resource.• POST. Submits data to a resource to process.• PUT. Updates a resource.• PATCH. Partially updates a resource.• DELETE. Deletes a resource. |
| The URL to the API service | <ul style="list-style-type: none">• Testing. https://devafp.ddcap.com/api/ & https://demoafp.ddcap.com/api/• Live. https://afp.ddcap.com/api/ |
| The URI to the resource | The resource to query, submit data to, update, or delete. For example, /api/deals/ |
| HTTP request headers | Includes the Authorization header with the "Basic HTTP Authentication". |
| A JSON request body | Required for most POST and some GET, PUT, PATCH, DELETE calls. |

API responses

ETHOS REST API calls return HTTP status codes. Some API calls also return JSON formatted response bodies that include information about the resource including URI details for other requests. You may use these URI links to request more information and construct an API flow that is relative to a specific request.

HTTP status codes

Each REST API request returns a success or error HTTP status code.

Success

In the responses, ETHOS AFP™ returns these HTTP status codes for successful requests:

| <u>Status code</u> | <u>Description</u> | |
|--------------------|--------------------|---|
| 200 | OK | The request succeeded. |
| 201 | Created | A POST method successfully created a resource. If the resource was already created by a previous execution of the same method, for example, the server returns the HTTP 200 OK status code. |
| 202 | Accepted | The server accepted the request and will execute it later. |
| 204 | No Content | The server successfully executed the method but returns no response body. |

Error

In the responses for failed requests, ETHOS AFP™ returns HTTP **4XX** or **5XX** status codes.

Versioning of RESTful API - this document covers the **original version** and **version 2**

When publishing APIs which are consumed by external users is it not always practical to enforce the users to adhere to the most recent updates to the API, in these cases we will version our API release by releasing the updates in different versions. The different versions will usually function in similar ways but may return the data in a new format or with extra data included.

The information on this document covers the **original version** and **version 2**, which can be distinguished in the request URI by the inclusion of the characters "v2/" after the /api/ wording, for example instead of sending a request to /api/deals/ for the original version, a version 2 request would be sent to /api/v2/deals/

With the exception of the /v2/ addition to the URI, the Version 2 requests all use the same formatting as their original version counterparts. Most of the changes seen with Version 2 are in the data of the response's vs those of the original version. Unless detailed on this document you can assume that version 2 syntax is the same original version, with the differences being documented below.

Account structure

In order to better understand how to use this API it is important to note that the records for Banks and Clients and different Deal types are all created separately and then combined to form the unique combinations specific to those banks/entities.

For example when BankA trades with ClientA they may be able to use the commodity X but not commodity Y, whereas, when BankA deals with ClientB they may be able to use the commodity Y but not commodity X. For this reason, the ETHOS system is built up of the Bank+Client+Deal_type in order to capture the specific configuration of the underlying trades which can take place. In the following guide you will see the Bank+Client+Deal_type referred to as BCDT. User accounts will be associated to a specific Bank. DDCAP will configure the Clients and deal types available to your banks account dependent upon the underlying signed agreements and instruction from the bank.

Available URL's

| URL List | | |
|---|--------------------|---|
| URL | Allowed Methods | Allowed Parameters |
| /api/ | GET | |
| /api/bcdts/ | GET | |
| /api/bcdts/<bcdt code> | GET | |
| /api/commodities/ | GET | tran_bank_code, tran_client_code, tran_deal_type, tran_trade_date, tran_settlement_date, tran_principal, tran_currency |
| /api/commodities/<commodity code> | GET | |
| /api/currencies/ | GET | |
| /api/currencies/<ccy code> | GET | |
| /api/deals/ | GET, POST | bank, status, trade_date |
| /api/deals/<deal key> | GET, DELETE, PATCH | |
| /api/deals/<deal key>/documents/ | GET | |
| /api/deals/<deal key>/documents/<document filename> | GET | |

| | | |
|----------------------------------|-----|--|
| /api/structures/<structure code> | GET | |
|----------------------------------|-----|--|

Example request

GET <https://devafp.ddcap.com/api/>

Returns the index of available URL

Response – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
{
  "Commodities": "https://devafp.ddcap.com/api/commodities/",
  "Index": "https://devafp.ddcap.com/api/",
  "Currencies": "https://devafp.ddcap.com/api/currencies/",
  "Deals": "https://devafp.ddcap.com/api/deals/",
  "BCDTs": "https://devafp.ddcap.com/api/bcdts/"
}
```

Commodities

Example request

GET <https://devafp.ddcap.com/api/commodities/>

Returns the list of available commodities for the associated account(s) of the user along with the pricing for that commodity. Note this is the full list of commodities which a user's account has access to and does not show the availability of the commodity.

Response body guide

```
{
  "comdty_name": "The full name/description of the commodity",
  "uri": "URI of this commodity",
}
```

```
"comdty_type": "The group this commodity falls under e.g. "BASE" for non-ferrous metals, "PGM" for platinum group metals, "SOFT" for food consumables such as Cocoa, Rice",  
"comdty_weight_name": "the unit measurement name of the commodity",  
"comdty_price": "The daily price fixing in USD for a single unit of the commodity",  
"comdty_code": "The commodity code – this should be used in the body of a trade request",  
"comdty_price_date": "The last date which this commodity price was updated in the ETHOS system"  
}
```

Response example – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
[  
  {  
    "comdty_name": "Fermented and Dried Cocoa Beans",  
    "uri": "https://devafp.ddcap.com/api/commodities/CO",  
    "comdty_type": "SOFT",  
    "comdty_weight_name": "Metric Tonnes",  
    "comdty_price": 2912.1599999999999,  
    "comdty_code": "CO",  
    "comdty_price_date": "17 Oct 2018"  
  },  
  {  
    "comdty_name": "LME Aluminium Alloy",  
    "uri": "https://devafp.ddcap.com/api/commodities/AA",  
    "comdty_type": "BASE",  
    "comdty_weight_name": "Metric Tonnes",  
    "comdty_price": 2020,  
    "comdty_code": "AA",  
    "comdty_price_date": "17 Oct 2018"  
  },  
  {  
    "comdty_name": "LME HG Primary Aluminium",
```

```
"uri": "https://devafp.ddcap.com/api/commodities/AHO",
"comdtty_type": "BASE",
"comdtty_weight_name": "Metric Tonnes",
"comdtty_price": 2020,
"comdtty_code": "AHO",
"comdtty_price_date": "17 Oct 2018"
},
{
  "comdtty_name": "LME HG Primary Aluminium",
  "uri": "https://devafp.ddcap.com/api/commodities/AL",
  "comdtty_type": "BASE",
  "comdtty_weight_name": "Metric Tonnes",
  "comdtty_price": 2020,
  "comdtty_code": "AL",
  "comdtty_price_date": "17 Oct 2018"
},
{
  "comdtty_name": "LME Copper Grade A",
  "uri": "https://devafp.ddcap.com/api/commodities/CU",
  "comdtty_type": "BASE",
  "comdtty_weight_name": "Metric Tonnes",
  "comdtty_price": 6202,
  "comdtty_code": "CU",
  "comdtty_price_date": "17 Oct 2018"
},
{
  "comdtty_name": "LME North American Special Aluminium Alloy Contract (NASAAC)",
  "uri": "https://devafp.ddcap.com/api/commodities/NA",
  "comdtty_type": "BASE",
  "comdtty_weight_name": "Metric Tonnes",
  "comdtty_price": 1480,
  "comdtty_code": "NA",
```



```
"comdty_price_date": "17 Oct 2018"
},
{
  "comdty_name": "LME Primary Nickel",
  "uri": "https://devafp.ddcap.com/api/commodities/Nl",
  "comdty_type": "BASE",
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 12450,
  "comdty_code": "NI",
  "comdty_price_date": "17 Oct 2018"
},
{
  "comdty_name": "LME Standard Lead",
  "uri": "https://devafp.ddcap.com/api/commodities/PB",
  "comdty_type": "BASE",
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2079,
  "comdty_code": "PB",
  "comdty_price_date": "17 Oct 2018"
},
{
  "comdty_name": "Palladium",
  "uri": "https://devafp.ddcap.com/api/commodities/PDZ",
  "comdty_type": "PGMZ",
  "comdty_weight_name": "Troy Ounces",
  "comdty_price": 1084,
  "comdty_code": "PDZ",
  "comdty_price_date": "17 Oct 2018"
},
{
  "comdty_name": "Platinum",
  "uri": "https://devafp.ddcap.com/api/commodities/PTZ",
```

```
[{"comdty_type": "PGMZ",  
  "comdty_weight_name": "Troy Ounces",  
  "comdty_price": 842,  
  "comdty_code": "PTZ",  
  "comdty_price_date": "17 Oct 2018"  
},  
{  
  "comdty_name": "LME Refined Tin (99.85%)",  
  "uri": "https://devafp.ddcap.com/api/commodities/SN",  
  "comdty_type": "BASE",  
  "comdty_weight_name": "Metric Tonnes",  
  "comdty_price": 19175,  
  "comdty_code": "SN",  
  "comdty_price_date": "17 Oct 2018"  
},  
{  
  "comdty_name": "LME Special High Grade Zinc",  
  "uri": "https://devafp.ddcap.com/api/commodities/ZN",  
  "comdty_type": "BASE",  
  "comdty_weight_name": "Metric Tonnes",  
  "comdty_price": 2630,  
  "comdty_code": "ZN",  
  "comdty_price_date": "17 Oct 2018"  
}  
]
```

Commodities – Version 2

Example request

GET <https://devafp.ddcap.com/api/v2/commodities/>

Returns the list of available commodities for the associated account(s) of the user along with the pricing for that commodity. Note this is the full list of commodities which a user's account has access to and does not show the availability of the commodity.

Response body guide

```
{
  "comdty_type": "The group this commodity falls under e.g. "BASE" for non-ferrous metals, "PGM" for platinum group metals, "SOFT" for food consumables such as Cocoa, Rice",
  "exchange_rate": "the exchange rate vs USD to the price shown in the field comdty_price_in_trade_ccy",
  "comdty_weight_name": "the unit measurement name of the commodity",
  "comdty_price": "The daily price fixing in USD for a single unit of the commodity",
  "comdty_price_in_trade_ccy": "This field will show the price of the commodity in the currency of the request. For requests made to /api/v2/commodities/ this field will always show USD prices, however we adjust the currency returned by including our desired currency code in the request, e.g. /api/v2/commodities/?tran_currency=SAR",
  "comdty_code": "The commodity code – this should be used in the body of a trade request",
  "comdty_price_date": "The last date which this commodity price was updated in the ETHOS system",
  "comdty_name": "The full name/description of the commodity",
  "uri": "URI of this commodity",
}
```

Response example – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
[
  {
    "comdty_type": "BASE",
    "exchange_rate": 1.0,
    "comdty_weight_name": "Metric Tonnes",
    "comdty_price": 2675.0,
    "comdty_price_in_trade_ccy": 2675.0,
    "comdty_code": "AA",
```

```
"comdty_price_date": "1 Sep 2021",
"comdty_name": "LME Aluminium Alloy",
"uri": "https://devafp.ddcap.com/api/commodities/AA"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2675.0,
  "comdty_price_in_trade_ccy": 2675.0,
  "comdty_code": "AHO",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME HG Primary Aluminium",
  "uri": "https://devafp.ddcap.com/api/commodities/AHO"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2675.0,
  "comdty_price_in_trade_ccy": 2675.0,
  "comdty_code": "AL",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME HG Primary Aluminium",
  "uri": "https://devafp.ddcap.com/api/commodities/AL"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 9424.0,
  "comdty_price_in_trade_ccy": 9424.0,
```

```
"comdty_code": "CU",
"comdty_price_date": "1 Sep 2021",
"comdty_name": "LME Copper Grade A",
"uri": "https://devafp.ddcap.com/api/commodities/CU"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2300.0,
  "comdty_price_in_trade_ccy": 2300.0,
  "comdty_code": "NA",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME North American Special Aluminium Alloy Contract (NASAAC)",
  "uri": "https://devafp.ddcap.com/api/commodities/NA"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 19039.0,
  "comdty_price_in_trade_ccy": 19039.0,
  "comdty_code": "NI",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME Primary Nickel",
  "uri": "https://devafp.ddcap.com/api/commodities/NI"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2445.0,
```

```
"comdty_price_in_trade_ccy": 2445.0,
"comdty_code": "PB",
"comdty_price_date": "1 Sep 2021",
"comdty_name": "LME Standard Lead",
"uri": "https://devafp.ddcap.com/api/commodities/PB"
},
{
  "comdty_type": "PGMZ",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Troy Ounces",
  "comdty_price": 2492.0,
  "comdty_price_in_trade_ccy": 2492.0,
  "comdty_code": "PDZ",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "Palladium",
  "uri": "https://devafp.ddcap.com/api/commodities/PDZ"
},
{
  "comdty_type": "PGMZ",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Troy Ounces",
  "comdty_price": 1001.0,
  "comdty_price_in_trade_ccy": 1001.0,
  "comdty_code": "PTZ",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "Platinum",
  "uri": "https://devafp.ddcap.com/api/commodities/PTZ"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 1.0,
  "comdty_weight_name": "Metric Tonnes",
```

```
[
  {
    "comdty_price": 34205.0,
    "comdty_price_in_trade_ccy": 34205.0,
    "comdty_code": "SN",
    "comdty_price_date": "1 Sep 2021",
    "comdty_name": "LME Refined Tin (99.85%)",
    "uri": "https://devafp.ddcap.com/api/commodities/SN"
  },
  {
    "comdty_type": "BASE",
    "exchange_rate": 1.0,
    "comdty_weight_name": "Metric Tonnes",
    "comdty_price": 2991.0,
    "comdty_price_in_trade_ccy": 2991.0,
    "comdty_code": "ZN",
    "comdty_price_date": "1 Sep 2021",
    "comdty_name": "LME Special High Grade Zinc",
    "uri": "https://devafp.ddcap.com/api/commodities/ZN"
  }
]
```

Commodity prices in a specific currency

GET https://devafp.ddcap.com/api/v2/commodities/?tran_currency=SAR

We have the platform return commodity prices in currency of our choice by stating the desired currency code in the request. In this example we are requesting prices to be returned in "SAR" (Saudi Riyals).

Available currency codes can be found using the /api/currencies/ or api/v2/currencies/ request.

Response example – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
[
```

```
{
  "comdty_type": "BASE",
  "exchange_rate": 3.75,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2675.0,
  "comdty_price_in_trade_ccy": 10031.25,
  "comdty_code": "AA",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME Aluminium Alloy",
  "uri": "https://devafp.ddcap.com/api/commodities/AA"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 3.75,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2675.0,
  "comdty_price_in_trade_ccy": 10031.25,
  "comdty_code": "AHO",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME HG Primary Aluminium",
  "uri": "https://devafp.ddcap.com/api/commodities/AHO"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 3.75,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2675.0,
  "comdty_price_in_trade_ccy": 10031.25,
  "comdty_code": "AL",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME HG Primary Aluminium",
  "uri": "https://devafp.ddcap.com/api/commodities/AL"
```



```
},
{
  "comdtty_type": "BASE",
  "exchange_rate": 3.75,
  "comdtty_weight_name": "Metric Tonnes",
  "comdtty_price": 9424.0,
  "comdtty_price_in_trade_ccy": 35340.0,
  "comdtty_code": "CU",
  "comdtty_price_date": "1 Sep 2021",
  "comdtty_name": "LME Copper Grade A",
  "uri": "https://devafp.ddcap.com/api/commodities/CU"
},
{
  "comdtty_type": "BASE",
  "exchange_rate": 3.75,
  "comdtty_weight_name": "Metric Tonnes",
  "comdtty_price": 2300.0,
  "comdtty_price_in_trade_ccy": 8625.0,
  "comdtty_code": "NA",
  "comdtty_price_date": "1 Sep 2021",
  "comdtty_name": "LME North American Special Aluminium Alloy Contract (NASAAC)",
  "uri": "https://devafp.ddcap.com/api/commodities/NA"
},
{
  "comdtty_type": "BASE",
  "exchange_rate": 3.75,
  "comdtty_weight_name": "Metric Tonnes",
  "comdtty_price": 19039.0,
  "comdtty_price_in_trade_ccy": 71396.25,
  "comdtty_code": "NI",
  "comdtty_price_date": "1 Sep 2021",
  "comdtty_name": "LME Primary Nickel",
```

```
"uri": "https://devafp.ddcap.com/api/commodities/NL",
{
  "comdty_type": "BASE",
  "exchange_rate": 3.75,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2445.0,
  "comdty_price_in_trade_ccy": 9168.75,
  "comdty_code": "PB",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME Standard Lead",
  "uri": "https://devafp.ddcap.com/api/commodities/PB",
},
{
  "comdty_type": "PGMZ",
  "exchange_rate": 3.75,
  "comdty_weight_name": "Troy Ounces",
  "comdty_price": 2492.0,
  "comdty_price_in_trade_ccy": 9345.0,
  "comdty_code": "PDZ",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "Palladium",
  "uri": "https://devafp.ddcap.com/api/commodities/PDZ",
},
{
  "comdty_type": "PGMZ",
  "exchange_rate": 3.75,
  "comdty_weight_name": "Troy Ounces",
  "comdty_price": 1001.0,
  "comdty_price_in_trade_ccy": 3753.75,
  "comdty_code": "PTZ",
  "comdty_price_date": "1 Sep 2021",
```

```
"comdty_name": "Platinum",
"uri": "https://devafp.ddcap.com/api/commodities/PTZ"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 3.75,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 34205.0,
  "comdty_price_in_trade_ccy": 128268.75,
  "comdty_code": "SN",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME Refined Tin (99.85%)",
  "uri": "https://devafp.ddcap.com/api/commodities/SN"
},
{
  "comdty_type": "BASE",
  "exchange_rate": 3.75,
  "comdty_weight_name": "Metric Tonnes",
  "comdty_price": 2991.0,
  "comdty_price_in_trade_ccy": 11216.25,
  "comdty_code": "ZN",
  "comdty_price_date": "1 Sep 2021",
  "comdty_name": "LME Special High Grade Zinc",
  "uri": "https://devafp.ddcap.com/api/commodities/ZN"
}
]
```

Account structures - BCDT's (BANK, CLIENT, DEAL TYPE)

You will see this abbreviation, BDCT, used in several places in the ETHOS system. It stands for BANK, CLIENT, DEAL TYPE. The BCDT is where much of the trade specific configuration is handled in the system.

Note that although named Bank, Client and Deal type these are actually simply three repositories of data and so you may see examples in the system where the data stored under one of more of them may not appear to be under correct heading e.g. Client repository sometimes houses Product type etc.

The BANK code will be the unique code of the account which you have been allocated by ETHOS personnel and is static.

CLIENT and DEAL TYPE codes will be advised to you by your ETHOS support liaison and may vary depending upon the types of trades you are booking. These codes are static.

Example Request

GET <https://devafp.ddcap.com/api/bcdts/>

Returns all the available BCDT for a BANK with the below headers.

Response body guide – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
{
  "bcdt_allowed_currencies": "The currencies permitted under this BCDT".
  "bcdt_bank_name": "The full name of the bank/entity as would be displayed on documentation".
  "bcdt_allowed_commodity_types": "The group this commodity falls under e.g. "BASE" for non-ferrous metals, "PGM" for platinum group metals,
  "SOFT" for food consumables such as Cocoa, Rice",
  "bcdt_code": "System generated Bank code+Client code+Deal_type_code (BCDT)",
  "structure_uri": "The URI for the structure"
  "bcdt_bank": "The code for the bank the user is logged in under".
  "bcdt_deal_type": "The trade type e.g. FIN, DEP, TAW etc"
  "bcdt_client_name_and_deal_type_name": "Full name of the client and deal type. This can be used to populate the banks system should they
  need to display the list for a user to select from"
  "bcdt_structure_code": "The structure code can be used to request the stage names which are need to complete a trade on the system
  "bcdt_deal_type_name": "Full name of the trade type e.g. Financing, Deposit, Tawarruq etc"
  "bcdt_name": "The full name of the BCDT – a free format field completed by the admin user when creating the structure in the Ethos system".
  "uri": "The direct link for the BCDT you are viewing".
  "bcdt_client_name": "The full client name as available under this Bank+Client+Deal_type (BCDT)".
}
```

```
"bcdt_trade_fields": "Lists all the body fields which must be present in the body of a POST request to api/deals/ in order to enter a trade into the system"
"bcdt_client": "The client code as available under this Bank+Client+Deal_type (BCDT)".
}
```

Example response (for a “TEST BANK” user – a demonstration account) – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
[
{
  "bcdt_allowed_currencies": [
    "AED",
    "AUD",
    "BHD",
    "CAD",
    "CHF",
    "DKR",
    "EGP",
    "EUR",
    "GBP",
    "HKD",
    "INR",
    "JOD",
    "JPY",
    "KWD",
    "LBP",
    "LEB",
    "MYR",
    "NOK",
    "NZD",
    "OMR",
    "PLN",
    "QAR",
```

```
"SAR",
"SEK",
"SGD",
"SYP",
"TRY",
"USD",
"YEN",
"ZAR"
],
"bcdt_bank_name": "Test Bank",
"bcdt_allowed_commodity_types": [
  "PGM",
  "BASE",
  "PGMZ"
],
"bcdt_code": "TEST_BANK_P_TAW",
"structure_uri": "https://devafp.ddcap.com/api/structures/TEST_TAW_STRUCTURE",
"bcdt_bank": "TEST BANK",
"bcdt_deal_type": "TAW",
"bcdt_client_name_and_deal_type_name": "Principal / Weekly Tawarruq",
"bcdt_structure_code": "TEST_TAW_STRUCTURE",
"bcdt_deal_type_name": "Weekly Tawarruq",
"bcdt_name": "TEST_BANK_P_TAW_DEMO_1",
"uri": "https://devafp.ddcap.com/api/bcdts/TEST%20BANK_P_TAW",
"bcdt_client_name": "Principal",
"bcdt_trade_fields": [
  "tran_bank_code",
  "tran_client_code",
  "tran_deal_type",
  "tran_commodity",
  "tran_currency",
  "tran_principal",
```

```
"tran_settlement_date",
"tran_trade_date",
"tran_class",
"tran_original_deal_ref",
"tran_maturity_date",
"tran_your_reference",
"tran_top_up"
],
"bcdt_client": "P"
},
{
  "bcdt_allowed_currencies": [
    "AED",
    "AUD",
    "BHD",
    "CAD",
    "CHF",
    "DKR",
    "EGP",
    "EUR",
    "GBP",
    "HKD",
    "INR",
    "JOD",
    "JPY",
    "KWD",
    "LBP",
    "LEB",
    "MYR",
    "NOK",
    "NZD",
    "OMR",
```

```
"PLN",
"QAR",
"SAR",
"SEK",
"SGD",
"SYP",
"TRY",
"USD",
"YEN",
"ZAR"
],
"bcdt_bank_name": "Test Bank",
"bcdt_allowed_commodity_types": [
  "PGM",
  "BASE"
],
"bcdt_code": "TEST_BANK_P_TAWD",
"structure_uri": "https://devafp.ddcap.com/api/structures/TEST_TAW_STRUCTURE",
"bcdt_bank": "TEST BANK",
"bcdt_deal_type": "TAWD",
"bcdt_client_name_and_deal_type_name": "Principal / Daily Tawarruq",
"bcdt_structure_code": "TEST_TAW_STRUCTURE",
"bcdt_deal_type_name": "Daily Tawarruq",
"bcdt_name": "TEST_BANK_P_TAW_DEMO_2",
"uri": "https://devafp.ddcap.com/api/bcdts/TEST%20BANK_P_TAWD",
"bcdt_client_name": "Principal",
"bcdt_trade_fields": [
  "tran_bank_code",
  "tran_client_code",
  "tran_deal_type",
  "tran_commodity",
  "tran_currency",
```



```
[
  "tran_principal",
  "tran_settlement_date",
  "tran_trade_date",
  "tran_class",
  "tran_maturity_date",
  "tran_your_reference"
],
"bcdt_client": "P"
}
```

Available commodities

Requesting the available commodity for specific trade details

Used to find out which commodities are available to support a trade of a specific size and date.

GET https://devafp.ddcap.com/api/commodities/?tran_bank_code={bank code}&tran_client_code={client code}&tran_deal_type={deal type code}&tran_principal={principal value amount}&tran_currency={currency}&tran_settlement_date={dd mmm yyyy}

Returns the list of available commodities available commodities for a specific; Bank + Client + Deal type + Amount + Currency + Value Date

Example request

GET https://devafp.ddcap.com/api/commodities/?tran_bank_code=TEST_BANK&tran_client_code=P&tran_deal_type=TAW&tran_currency=SAR&tran_principal=100000000&tran_settlement_date=17 Oct 2018

Example response: – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
[
  {
    "comdty_name": "Platinum",
    "uri": "https://devafp.ddcap.com/api/commodities/PTZ",
    "comdty_type": "PGMZ",
    "comdty_weight_name": "Troy Ounces",
    "comdty_price": 842,
    "comdty_code": "PTZ",
    "comdty_price_date": "17 Oct 2018"
  },
]
```

A successful call returns a JSON-formatted response body

Depending on the details of the trade multiple commodities may be shown as available

Note that the **"Comdty_name"** field value e.g. **"Platinum"** is the value which should be displayed to the end user.

The **"Comdty_code"** value e.g. **"PTZ"** is field to be used when sending a POST request to request create a trade. The **"Comdty_code"** value should be inserted in the field **"tran_commodity"**: e.g. **"tran_commodity": "PTZ"** within the body of the POST request to **/api/deals/**

Note that it is not strictly necessary to use the above request, or version 2 request below, to find available commodities, you are able to send your POST request to **/api/deals/** or **/api/v2/deals/** without a value in the **"tran_commodity"**: field the e.g. **"tran_commodity": ""** in this scenario the application will automatically assign you a commodity.

Available commodities – Version 2

Version 2 returns the extra fields "comdty_price_in_trade_ccy" (which matches the native currency in the request, if stated) and "exchange_rate" (which always displays the native currency of the request vs. USD)

Requesting the available commodity for specific trade details

Used to find out which commodities are available to support a trade of a specific size and date.

GET https://devafp.ddcap.com/api/v2/commodities/?tran_bank_code={bank code}&tran_client_code={client code}&tran_deal_type={deal type code}&tran_principal={principal value amount}&tran_currency={currency}&tran_settlement_date={dd mmm yyyy}

Returns the list of available commodities available commodities for a specific; Bank + Client + Deal type + Amount + Currency + Value Date

Example request

GET https://devafp.ddcap.com/api/v2/commodities/?tran_bank_code=TEST_BANK&tran_client_code=P&tran_deal_type=TAW&tran_currency=SAR&tran_principal=1000000000&tran_settlement_date=14 Oct 2021

Example response: – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
[
  {
    "comdty_type": "PGMZ",
    "exchange_rate": 3.75,
    "comdty_weight_name": "Troy Ounces",
    "comdty_price": 2492.0,
    "comdty_price_in_trade_ccy": 9345.0,
    "comdty_code": "PDZ",
    "comdty_price_date": "1 Sep 2021",
    "comdty_name": "Palladium",
    "uri": "https://devafp.ddcap.com/api/commodities/PDZ"
  },
  {
    "comdty_type": "PGMZ",
    "exchange_rate": 3.75,
    "comdty_weight_name": "Troy Ounces",
    "comdty_price": 1001.0,
    "comdty_price_in_trade_ccy": 3753.75,
    "comdty_code": "PTZ",
```

```
"comdty_price_date": "1 Sep 2021",  
"comdty_name": "Platinum",  
"uri": "https://devafp.ddcap.com/api/commodities/PTZ"  
}  
]
```

A successful call returns a JSON-formatted response body

Currencies

Example request

GET <https://devafp.ddcap.com/api/currencies> or <https://devafp.ddcap.com/api/v2/currencies>

Returns full list of currencies and exchange rates, against USD, available to the users account.

Response body guide – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
{  
  "currency_name": "the full name of the currency",  
  "currency_exrate_date": "the date of the currency record on view",  
  "currency_exrate": "the numeric value of the exchange rate of the currency against USD",  
  "currency_code": "the currency code",  
  "uri": "the direct link to view this currency record"  
}
```

Response example (this is a cut down example of the number of currencies that would be seen) – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
[  
  {  
    "currency_name": "United Arab Emirates Dirham",
```

```
[{"currency_exrate_date": "17 Oct 2018",  
  "currency_exrate": 3.673,  
  "currency_code": "AED",  
  "uri": "https://devafp.ddcap.com/api/currencies/AED"},  
{"currency_name": "Saudi Riyal",  
  "currency_exrate_date": "17 Oct 2018",  
  "currency_exrate": 3.75,  
  "currency_code": "SAR",  
  "uri": "https://devafp.ddcap.com/api/currencies/SAR"}]
```

Trade Structures

GET <https://devafp.ddcap.com/api/structures/<structure code>>

Used to view the proposed flow of stages for purchasing and selling a commodity. Any documentation generated by the system will be noted here by name and it will be noted at what stage the document is generated, and the company's system (i.e. seller or purchaser) where the stage is located.

The names of the documentation can be used to recall the documentation, once generated, using the API

The list of available document names can also be found under each trade using the following;

URI /api/deals/<transaction_key>/documents/

Example request

GET https://devafp.ddcap.com/api/structures/TEST_TAW_STRUCTURE

Returns the index of available URL

Response example – HTTP Status: 200 OK*A successful call returns a JSON-formatted response body*

```
{
  "structure_name": "Demo Tawarruq structure",
  "structure_stage": [
    {
      "stage_location": "DDC",
      "stage_code": "TEST_TAW_STAGE_1",
      "stage_documents": [
        "GENERATE AS SELLERS OFFER [tran_your_reference].pdf"
      ],
      "stage commodity sold to bank stage": false
    },
    {
      "stage_location": "DDC",
      "stage_code": "TEST_TAW_STAGE_2",
      "stage_documents": [
        "Purchasers Acceptance.pdf"
      ],
      "stage commodity sold to bank stage": false
    },
    {
      "stage_location": "CTL",
      "stage_code": "TEST_TAW_STAGE_3",
      "stage_documents": [
        "Sale Terms Contract.pdf",
        "Sale Terms Contract Accepted.pdf"
      ],
      "stage commodity sold to bank stage": false
    },
    {
      "stage_location": "CTL",
```

```

    "stage_code": "TEST_TAW_STAGE_4",
    "stage_documents": [
      "Form of Sale Confirmation.pdf",
      "Form of Purchase Confirmation.pdf",
      "Advice of Metals Sold.pdf"
    ],
    "stage_commodity_sold_to_bank_stage": true
  },
  "uri": "https://devafp.ddcap.com/api/structures/TEST_TAW_STRUCTURE",
  "structure_code": "TEST_TAW_STRUCTURE"
}

```

A successful call returns a JSON-formatted response body

Composing a deal request

Example request

POST <https://devafp.ddcap.com/api/deals/> or to received version 2 response formatting send request to <https://devafp.ddcap.com/api/v2/deals/>

POST to /api/deals/ is one of the few requests to the ETHOS REST API which requires the data in the body of the request.

The body fields which must be completed in the body of the POST request can be found on the response to /api/bcdts/ under "bcdt_trade_fields", example as highlighted below

```

"bcdt_trade_fields": [
  "tran_bank_code",
  "tran_client_code",
  "tran_deal_type",
  "tran_commodity",
  "tran_currency",
  "tran_principal",
  "tran_settlement_date",

```

```
"tran_trade_date",
"tran_class",
"tran_maturity_date",
"tran_your_reference"
],
```

A list of all the body fields and the field validations is shown below. Note that you will likely only be required to use a small subset of the fields as specified by /api/bcdts/ under "bcdt_trade_fields"

Available body request fields for /api/deals/

| Field name | Mandatory? | Field value | Validation | Description |
|---------------------------------|------------|-------------|-------------------|--|
| "tran_bank_code", | Yes | | should match BCDT | |
| "tran_client_code", | Yes | | should match BCDT | |
| "tran_free_format_client_name", | | | | |
| "tran_deal_type", | Yes | | should match BCDT | |
| "tran_branch", | | | should match BCDT | |
| "tran_bank_dept", | | | should match BCDT | |
| "tran_trade_date", | NO | | dd MMM yyyy | Leave the trade date field value blank [""] if you expect to be processing trades 24/7. This will allow our servers to assign the current trade date as per UK time. You should take the trade date value as per the POST response to /api/deals/ |

| | | | | |
|-----------------------------|------|---------------------|----------------------|--|
| "tran_settlement_date", | Yes | | dd MMM yyyy | |
| "tran_maturity_date", | Yes* | | dd MMM yyyy | |
| "tran_currency", | Yes | | | |
| "tran_principal", | Yes | | | |
| "tran_commodity", | No | | | If you do not specify the tran_commodity value the platform will auto-assign a best fit of commodity to you |
| "tran_user_price", | | | | |
| "commodity_price_currency", | | | | |
| "tran_your_reference", | Yes* | | Free format field | Free format field |
| "tran_agreed_profit_rate", | | | | Free format field |
| "tran_user_ref_1", | | | | Free format field |
| "tran_user_ref_2", | | | | Free format field |
| "tran_class", | Yes | "STANDARD" | | |
| "tran_top_up", | | "YES", "NO", "SALE" | | |
| "tran_original_deal_ref", | | | Numeric | If "tran_top_up" is "YES" then this field should show the original trade ticket number |
| "tran_cycle_flag", | | NEW EXISTING | | Set to "NEW" if unsure. |
| "tran_cycle_identifier", | | | | Identifies this new/existing cycle. Used to fix prices / commodities over a period of time. |
| "tran_as_messenger", | | "YES", "NO" | | |
| "fee_group", | | | | |
| "aggregate_marker", | | | | Case & Whitespace Sensitive. Combines the Principals of otherwise identical deals and replaces them all with one single deal. |
| "tran_deferred_amount", | | | | |

| | | | | |
|--------------------------------|------|--|----------------------|--|
| "tran_agreement_date", | | | dd MMM yyyy | |
| "tran_agreement_date_2", | | | dd MMM yyyy | |
| "tran_reference_period_start", | | | dd MMM yyyy | |
| "tran_reference_period", | | | Free format field | |
| "transaction_start_date", | | | dd MMM yyyy | |
| "tran_opics_mxg_deal_number", | | | Free format field | |
| "tran_bank_comments", | | | Free format field | |
| "tran_trader_name", | Yes* | | | |
| "tenor", | | | Numeric | |
| "tran_profit_amount" | | | Numeric | |

* if required by BCDT settings.

Body response fields from /api/deals/ or (version 2) /api/v2/deals/

Below is a summary of the output values

| Row # | Output field name | Example (taken from the below response) | Data Type | Notes/Description |
|-------|-------------------|---|-----------|---|
| 1 | "tran_currency" | "USD", | String | The currency code of the trade. A full list is available currencies using a GET request to /api/currencies/ |
| 2 | "tran_bcdt_code" | "TEST BANK_P_TAW", | String | The unique structure code for the trade (this is where we apply the configuration settings). |

| | | | | |
|-------------|-------------------------------|---|--------|--|
| 3 | "tran_settlement_date" | "17 Oct 2018 13:52:02", | String | The value/settlement date of this trade. |
| 4 | "tran_original_maturity_date" | "20 Dec 2018", | String | The maturity date of the original deal, if applicable. |
| 4.2 V2 only | "tran_invoice_key" | "000123" | String | References the invoice number on which the trade appears |
| 5 | "tran_deferred_price" | 0, | Float | The deferred price, if applicable. |
| 6 | "tran_extra_reference2" | "" | String | The extra reference field 2 value, if applicable. |
| 7 | "tran_input_timestamp" | "18 Dec 2018 13:28:33" | String | The timestamp of when this deal was input |
| 8 | "tran_original_key" | "TEST_200247_000001", | String | The unique deal key of the original deal in the series. |
| 9 | "tran_next_stage" | "COMPLETED", | String | The next stage of this deal. If the value is "COMPLETED" there is no next stage. |
| 10 | "tran_extra_reference1" | "" | String | The extra reference field 1 value, if applicable. |
| 11 | "documents_uri" | "https://devafp.com/api/deals/TEST_200247_000002/documents/", | String | List all the uri (used in a GET request) for available documents for the deal. The document names and timestamps are also listed here. |
| 12 | "tran_original_trade_date" | "18 Dec 2018", | String | The date the original deal in this series was input |
| 13 | "tran_fee_usd" | "13.33", | String | The fee for this deal in USD |
| 14 | "tran_deferred_amount" | 0, | Float | The deferred amount, if applicable. |
| 15 | "tran_principal" | 100000, | Float | The value amount of this deal |
| 16 | "tran_structure_code" | "TEST", | String | The structure code can be used with a GET request to /api/structures/TEST to show the stage name and documents that will be generated at those stages. |
| 17 | "tran_dept" | "" | String | The department of the user, if applicable. |
| 18 | "tran_quantity_unit" | "Troy Ounces", | String | The commodity unit of measurement for the trade. |

| | | | | |
|----|---------------------------|-------------------------|--------|--|
| 19 | "tran_end_buyer" | "Condor Trade Limited", | String | The proposed end buyer of the commodity. |
| 20 | "transaction_no" | "200247", | String | The transaction number of the deal (this is taken from the original deal) and is used in the POST body request when making sales to the end buyer. Also used in the POST body request for top-up trades. |
| 21 | "tran_client_code" | "P", | String | The client code (these are configured and managed by DDCAP). |
| 22 | "tran_agreed_profit_rate" | "" | String | The profit rate of the deal, if applicable. |
| 23 | "tran_commodity" | "PDZ", | String | The commodity code on this deal. Can be used with a GET request to /api/commodities/PDZ to provide more information on the commodity. |
| 24 | "tran_cycle_identifier" | "" | String | Cycle identifiers are used by the system to maintain specific commodity prices across set time periods. Usually NA unless you have been advised to use them. |
| 25 | "tran_price" | 4166.66666667, | Float | The unit price of the commodity |
| 26 | "tran_class" | "TEST", | String | In the production environments live deals will have read as "STANDARD". Test deals will read as "TEST". |
| 27 | "transaction_key" | "TEST_200247_000002", | String | The unique system key for this deal. |
| 28 | "tran_client_name" | "Principal", | String | The full client name to display to end users (these are configured and managed by DDCAP). |
| 29 | "tran_trade_date" | "18 Dec 2018", | String | The transaction date of this trade. |
| 30 | "tenor" | "2", | String | The tenor of the original deal (measured from settlement date to maturity date). |
| 31 | "sub_transaction_no" | "2", | String | Original deals are always sub transaction no 1. Any other linked deal (top-up deals or sales) are then given the next available sub transaction number. |

| | | | | |
|----|---------------------------------|--|--------|---|
| 32 | "tran_stage_code" | "TEST_Stage3", | String | The current stage of this deal. In this instance stage 3 is the final stage so the deal is completed are there are no stages to follow. |
| 33 | "tran_fee_exchange_rate" | "3.75000000", | String | The exchange rate vs USD which has been used to calculate the fee for this trade. |
| 34 | "tran_maturity_date" | "20 Dec 2018", | String | The maturity date of the original deal. Sales can be made with value up the end of the maturity date. |
| 35 | "bcdt_uri" | "https://devafp.co ndortrade.com/a pi/bcdts/TEST_P_ W", | String | The uri (used in a GET request) to display the configuration settings of this trade. For example the required body fields for POST requests, permitted commodities, permitted currencies. |
| 36 | "tran_input_user" | "GarethTEST (Gareth TEST REST testing)", | String | The user code and user full name of the account used to input this trade. |
| 37 | "tran_exchange_rate_inverse" | "" | String | The inverse exchange rate, if applicable. |
| 38 | "tran_bank_code" | "TEST", | String | The code of the bank in Ethos. |
| 39 | "tran_initial_seller" | "DD&Co Limited", | String | The initial seller of the commodity (to the bank). |
| 40 | "tran_total_principal" | 10000000, | Float | The total of the principal amount purchased on the original deal and any top-up deals. |
| 41 | "tran_your_reference" | "test sale 1 ", | String | The user entered or auto reference of this deal |
| 42 | "tran_branch" | "" | String | The branch of the user, if applicable. |
| 43 | "tran_exchange_rate" | "3.75000000 SAR = 1 USD ", | String | The exchange rate vs USD which has been used to calculate the commodity price on this trade. |
| 44 | "tran_fee_currency" | "SAR", | String | The currency of this trade |
| 45 | "tran_original_settlement_date" | "18 Dec 2018", | String | The value/settlement date of the original deal |
| 46 | "tran_quantity" | 24, | Float | The quantity of commodity for this deal |
| 47 | "tran_outstanding_quantity" | 2361, | Float | The quantity of commodity still to be sold to the end buyer. This is the quantity of commodity available to be sold to your customers for your underlying trades. This balance will reduce in line with the quantity of commodity you sell. |

| | | | | |
|---------------------|------------------------------|--|--------|--|
| 47.2 V2 only | "stage_status_text" | "Please accept Sellers Offer", | String | Returns dynamic text taken from our platform UI showing the action required on the trade. This value can be used to populate customer facing buttons on your UI in order to enable users to complete all stages of the commodity purchase and sale. |
| 48 | "principal_plus_comm" | "0.00", | String | The principal amount plus broker fees, if applicable. |
| 49 | "original_deal_uri" | "https://devafp.co ndortrade.com/a pi/deals/TEST_200 247_000001", | String | The uri (used in a GET request) to display the details of the original deal |
| 50 | "uri" | "https://devafp.co ndortrade.com/a pi/deals/TEST_200 247_000002", | String | The uri (used in a GET request) to display the details of this deal |
| 51 | "tran_outstanding_principal" | 9900000, | Float | The value amount of commodity still to be sold to the end buyer. This is the value amount (in the deal currency) of commodity available to be sold to your customers for your underlying trades. This balance will reduce in line with the amount of commodity you sell. |
| 52 | "tran_comdty_alloc_number" | "401447 200247 PDZ", | String | The commodity allocation number for this trade. |
| 53 | "tran_fee" | "50.00", | String | The fee in the currency of the deal, if applicable. |
| 54 | "location_rule" | "In vault, Zurich", | String | The location of the commodity. |
| 55 | "tran_tre_type" | "Sell-Off", | String | The trade type as seen by the Ethos system. Types are currently: Original Deal, Sell-Off, Top-Up. |
| 56 | "tran_deal_type" | "TAW", | String | The code of the deal type e.g. DEPOSIT, FINANCING or something less generic. |
| 57 | "tran_total_quantity" | 2385 | Float | The total quantity of commodity purchased, including any Top-Up deals (once accepted). |
| | | | | |

Below is an example of a trade request and response.

POST <https://devafp.ddcap.com/api/deals/>

Example Request: body JSON (application/json)

```
{
  "tran_bank_code": "TEST BANK",
  "tran_client_code": "P",
  "tran_deal_type": "TAW",
  "tran_commodity": "PDZ",
  "tran_currency": "USD",
  "tran_principal": "1000000",
  "tran_trade_date": "",
  "tran_settlement_date": "17-Oct-2018",
  "tran_trader_name": "",
  "tran_maturity_date": "21 Oct18",
  "tran_your_reference": "test 2",
  "tran_class": "STANDARD"
}
```

Response example – HTTP Status: 201 CREATED

A successful call returns a JSON-formatted response body

```
{
  "tran_currency": "USD",
  "tran_bcdt_code": "TEST BANK_P_TAW",
  "tran_settlement_date": "17 Oct 2018",
  "tran_deferred_price": 0,
  "tran_extra_reference2": "",
  "tran_input_timestamp": "17 Oct 2018 13:52:02",
  "tran_next_stage": "TEST_TAW_STAGE_2",
}
```

```
"tran_extra_reference1": "",
"documents_uri": "https://devafp.ddcap.com/api/deals/TEST%20BANK_410808_000001/documents/",
"tran_fee_usd": "12.04",
"tran_deferred_amount": 0,
"tran_principal": 1000000,
"tran_structure_code": "TEST_TAW_STRUCTURE",
"tran_dept": "",
"tran_quantity_unit": "Troy Ounces",
"tran_end_buyer": "Condor Trade Limited",
"transaction_no": "410808",
"tran_client_code": "P",
"tran_agreed_profit_rate": "",
"tran_commodity": "PDZ",
"tran_cycle_identifier": "",
"tran_price": 1083.42361863,
"tran_class": "STANDARD",
"transaction_key ": "TEST BANK_410808_000001",
"tran_client_name": "Principal",
"tran_trade_date": "17 Oct 2018",
"tenor": "4",
"sub_transaction_no": "",
"tran_stage_code": "TEST_TAW_STAGE_1",
"tran_fee_exchange_rate": "4.15320000",
"tran_maturity_date": "21 Oct 2018",
"bcdt_uri": "https://devafp.ddcap.com/api/bcdts/TEST%20BANK_P_TAW",
"tran_input_user": "TESTBANK (TEST BANK USER)",
"tran_exchange_rate_inverse": "",
"tran_bank_code": "TEST BANK",
"tran_initial_seller": "DD&Co Limited",
"tran_your_reference": "test 2",
"tran_branch": "",
"tran_exchange_rate": "1.00000000 USD = 1 USD ",
```



```
"tran_fee_currency": "MYR",  
"tran_quantity": 923,  
"principal_plus_comm": "0.00",  
"uri": "https://devafp.ddcap.com/api/deals/TEST%20BANK_410808_000001",  
"tran_comdty_alloc_number": "401385 410808 PDZ",  
"tran_fee": "50.00",  
"location_rule": "In vault, Zurich",  
"tran_tre_type": "Original Deal",  
"tran_deal_type": "TAW"  
}
```

GET, POST and PATCH Requests sent to Version 2 (i.e. /api/v2/deals/) will be processed in the same manner as requests to the original version (i.e. as requests sent to /api/deals/), with the difference being the response from v2 will contain the additional fields as listed on the Body response fields table above.

Completing Purchase and Sale stages of a trade

The unique identifier for the trade in the ETHOS system is the "transaction_key" e.g. "TEST BANK_410808_000001" as shown above.

You will require the "transaction_key" to compose all other requests related to this trade.

In order to complete the other stages of the trade you will be required to send PATCH requests to "tran_next_stage" e.g. "TEST_TAW_STAGE_2" as shown on the body response above. (With the exception proportional sales where you will need to POST the sale details in the body of the final request(s) – your DDCAP contact will advise you if this is required.)

The trade will be complete once the "tran_next_stage" shows as "COMPLETED" ["tran_next_stage": "COMPLETED"].

PATCH /api/deals/<transaction_key>

Request body JSON (application/json)

A successful call returns a JSON-formatted response body

```
{  
  "tran_stage_code": " tran_next_stage"  
}
```

Note the tran_next_stage to be completed will be found in the body of the response to /api/deals/<deal key>

E.g. "tran_next_code": "TEST_TAW_STAGE_2"

Example request

PATCH https://devafp.ddcap.com/api/deals/TEST BANK_123456_000001/

Example Request body JSON (application/json)

```
{  
  "tran_stage_code": "TEST_TAW_STAGE_2"  
}
```

Response example – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
{  
  "tran_currency": "USD",  
  "tran_bcdt_code": "TEST BANK_P_TAW",  
  "tran_settlement_date": "16 Oct 2018",  
  "tran_deferred_price": 0,  
  "tran_extra_reference2": "",  
  "tran_input_timestamp": "16 Oct 2018 11:34:23",  
}
```

```
"tran_next_stage": "TEST_TAW_STAGE_3",
"tran_extra_reference1": "",
"documents_uri": "https://devafp.ddcap.com/api/deals/TEST%20BANK_410794_000001/documents/",
"tran_fee_usd": "10.00",
"tran_deferred_amount": 0,
"tran_principal": 1000000,
"tran_structure_code": "TEST_TAW_STRUCTURE",
"tran_dept": "",
"tran_quantity_unit": "Troy Ounces",
"tran_end_buyer": "Condor Trade Limited",
"transaction_no": "410794",
"tran_client_code": "P",
"tran_agreed_profit_rate": "",
"tran_commodity": "PDZ",
"tran_cycle_identifier": "",
"tran_price": 1083.42361863,
"tran_class": "STANDARD",
"transaction_key": "TEST BANK_410794_000001",
"tran_client_name": "Principal",
"tran_trade_date": "16 Oct 2018",
"tenor": "5",
"sub_transaction_no": "",
"tran_stage_code": "TEST_TAW_STAGE_2",
"tran_fee_exchange_rate": "4.15550000",
"tran_maturity_date": "21 Oct 2018",
"bcdt_uri": "https://devafp.ddcap.com/api/bcdts/TEST%20BANK_P_TAW",
"tran_input_user": "TESTBANK (TEST BANK USER)",
"tran_exchange_rate_inverse": "",
"tran_bank_code": "TEST BANK",
"tran_initial_seller": "DD&Co Limited",
"tran_your_reference": "test 1",
"tran_branch": "",
```

```
"tran_exchange_rate": "1.00000000 USD = 1 USD ",  
"tran_fee_currency": "MYR",  
"tran_quantity": 923,  
"principal_plus_comm": "0.00",  
"uri": "https://devafp.ddcap.com/api/deals/TEST%20BANK_410794_000001",  
"tran_comdty_alloc_number": "401384 410794 PDZ",  
"tran_fee": "41.56",  
"location_rule": "In vault, Zurich",  
"tran_tre_type": "Original Deal",  
"tran_deal_type": "TAW"  
}
```

You can see above the "tran_next_stage" and current stage "tran_stage_code" have been updated.

Searching for trades

If you have the "transaction_key" you can request the details for that specific trade by sending a GET request to /api/deals/"transaction_key" e.g. /api/deals/BANKA_123456_0000001

GET <https://devafp.ddcap.com/api/deals/>

Returns a list of all trades input into the system associated with the users account

GET https://devafp.ddcap.com/api/deals/?trade_date=dd mmm yyyy

Returns a list of all trades input into the system with a specific trade date. Date format should be dd mmm yyyy
You are also able to search date ranges, details of which are now laid out below;

When making a GET request to /api/deals/, the following new query parameters are now supported:

- from_trade_date
- to_trade_date

- from_settlement_date
- to_settlement_date
- settlement_date

For example:

```
/api/deals/?from_trade_date=8 Jun 2020&to_trade_date=10 jun 2020
```

This request would find all trades with a trade date between 8/6/2020 and 10/6/2020.

Where used in combination, the trade date and settlement date parameters will exclude all trades that don't meet both criteria. So this request would return only trades with a trade date of 8/6/2020 and a settlement date of 10/6/2020:

```
/api/deals/?trade_date=8 Jun 2020&settlement_date=10 jun 2020
```

In the case below, the plain trade date parameter will be treated as if it were the from_trade_date parameter:

```
/api/deals/?trade_date=8 Jun 2020&to_trade_date=10 jun 2020
```

Similarly, if the trade_date parameter was included with a from_trade_date parameter and no to_trade_date, it will be treated as a to_trade_date value. In the below query, the trade_date parameter will just be ignored:

```
/api/deals/?from_trade_date=8 Jun 2020&to_trade_date=10 jun 2020&trade_date=13 Jun 2020
```

The same holds true for the settlement date parameters.

Other useful trade related requests

GET [https://devafp.ddcap.com/api/deals/?status={"tran_stage_code"}](https://devafp.ddcap.com/api/deals/?status={)

Returns a list of all trades input into the system which are at a specific stage e.g. "COMPLETED"

Example request

GET <https://devafp.ddcap.com/api/deals/?status=COMPLETED>

Response example – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body which full details for all matching trades.

```
{
  "tran_currency": "EUR",
  "tran_bcdt_code": "TESTBANK1_CLIENTTest_DEPOSIT",
  "tran_settlement_date": "18 Mar 2020",
  "tran_original_maturity_date": "16 Jun 2020",
  "tran_deferred_price": 0.0,
  "tran_extra_reference2": "",
  "tran_input_timestamp": "18 Mar 2020 14:29:33",
  "tran_original_key": "TESTBANK1_1000166_000001",
  "tran_next_stage": "COMPLETED",
  "tran_extra_reference1": "",
  "documents_uri": "https://devafp.ddcap.com/api/deals/TESTBANK1_1000166_000001/documents/",
  "tran_original_trade_date": "18 Mar 2020",
  "tran_fee_usd": "0.00",
  "tran_deferred_amount": 0.0,
  "tran_principal": 10000000.0,
  "tran_structure_code": "STRUCTURE_ONE",
  "tran_dept": "",
  "tran_quantity_unit": "Troy Ounces",
  "tran_end_buyer": "Condor Trade Limited",
  "transaction_no": "1000166",
  "tran_client_code": "CLIENTTest",
  "tran_agreed_profit_rate": "",
  "tran_commodity": "PDZ",
  "tran_cycle_identifier": "",
  "tran_price": 1740.34110686,
  "tran_class": "STANDARD",
  "transaction_key": "TESTBANK1_1000166_000001",
  "tran_client_name": "Client Test",
}
```

```
"tran_trade_date": "18 Mar 2020",
"tenor": "90",
"sub_transaction_no": "1",
"tran_stage_code": "STRUCTURE_ONE_3",
"tran_fee_exchange_rate": "0.00000000",
"tran_maturity_date": "16 Jun 2020",
"bcdt_uri": "https://devafp.ddcap.com/api/bcdts/TESTBANK1_CLIENTTest_DEPOSIT",
"tran_input_user": "REST_test (RESTful API testing account)",
"tran_exchange_rate_inverse": "",
"tran_bank_code": "TESTBANK1",
"tran_initial_seller": "DD&Co Limited",
"tran_total_principal": 10000000.0,
"tran_your_reference": "123",
"tran_branch": "",
"tran_exchange_rate": "1.1124 USD = 1 EUR",
"tran_fee_currency": "",
"tran_original_settlement_date": "18 Mar 2020",
"tran_quantity": 5746.0,
"tran_outstanding_quantity": 5746.0,
"principal_plus_comm": "0.00",
"original_deal_uri": "https://devafp.ddcap.com/api/deals/TESTBANK1_1000166_000001",
"uri": "https://devafp.ddcap.com/api/deals/TESTBANK1_1000166_000001",
"tran_outstanding_principal": 10000000.0,
"tran_comdty_alloc_number": "401903 1000166 PDZ",
"tran_fee": "0.00",
"location_rule": "In vault, Zurich",
"tran_tre_type": "Original Deal",
"tran_deal_type": "DEPOSIT",
"tran_total_quantity": 5746.0
}
```

Trade Documents

The process by which RESTful API clients can discover documents for themselves is:

/api/ links to /api/deals/

/api/deals/ lists deals, where each deal includes a link to its /api/deals/<transaction key>/documents/ URL

/api/deals/<deal key>/documents/ lists the documents, including the relevant URLs to download them

/api/deals/<deal key>/documents/<filename> is the URI for the individual document file

GET <https://devafp.ddcap.com/api/deals/<transaction key>/documents/>

Returns the documents file which are available to download.

Example request

GET https://devafp.ddcap.com/api/deals/TEST BANK_123456_000001/documents/

Response guide

```
[
  {
    "uploaded": "The time and date the document was created",
    "download uri": "The direct link to download the individual document",
    "filename": "The name of the generated document"
  }
]
```

Response example

A successful call returns a JSON-formatted response body

```
[
  {
    "uploaded": "Wed, 17 Oct 2018 14:51:19 -0000",
    "download uri": "https://devafp.ddcap.com/api/deals/TEST%20BANK_123456_000001/documents/Purchasers%20Acceptance.pdf",
    "filename": "Purchasers Acceptance.pdf"
  }
]
```


Requesting a trade cancellation

Send a DELETE request to `/api/deals/"transaction_key"`

e.g.

DELETE https://devafp.ddcap.com/api/deals/TEST BANK_123456_000001

- Successful response status: 204 **NO CONTENT**
- Unsuccessful response status: 404 **NOT FOUND** and *JSON-formatted* body below:

```
{  
  "message": "Deal TESTBANK_201664_000001 not found."  
}
```

- Unsuccessful response status: **400 BAD REQUEST** and *JSON-formatted* body below:

```
{  
  "message": "Unable to cancel deal: Cannot cancel deal, deal already accepted"  
}
```

Explanation for the above 400 BAD REQUEST response – once the Offer to sell the commodity to the bank has been accepted i.e. usually as soon as the first PATCH stage has been completed then you will no longer be able to cancel the trade using the API, cancelling must be requested manually via email. The flag to stop cancelling trades once accepted/PATCHED is configurable by DDCAP but please note that COMPLETED trades will not usually be able to be cancelled using the APIs, unless there is a specific arrangement in place to permit.

Please note that once a trade has successfully been cancelled you will no longer be able to obtain its details from requests sent to `/api/deals/`

Editing a trade

PATCH /api/deals/{transaction_key}

The request body fields required to edit a trade are the same as those used to input a trade using the initial POST request.

ALL OF THE FIELDS SENT MUST BE THE SAME AS THOSE SENT IN THE INITIAL POST REQUEST, EXCEPT FOR THE FIELDS YOU WISH TO CHANGE, ANY DIFFERENCES, INCLUDING OMISSIONS, WILL BE TREATED AS AMENDMENTS.

Example

We send a POST request

POST <https://devafp.ddcap.com/api/deals/>

With request body

Example Request body JSON (application/json)

```
{
  "tran_bank_code": "TESTBANK",
  "tran_client_code": "P",
  "tran_deal_type": "DEPOSIT",
  "tran_commodity": "",
  "tran_currency": "EUR",
  "tran_principal": "1000000.12",
  "tran_settlement_date": "6 Oct 2020",
  "tran_trade_date": "6 Oct 2020",
  "tran_trader_name": "",
  "tran_maturity_date": "",
  "tran_your_reference": "testing {{current_timestamp}}",
  "tran_class": "STANDARD"
```

```
}
```

Response example – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
{
  "tran_currency": "EUR",
  "tran_bcdt_code": "TESTBANK_P_DEPOSIT",
  "tran_settlement_date": "6 Oct 2020",
  "tran_original_maturity_date": "",
  "tran_deferred_price": 0.0,
  "tran_extra_reference2": "",
  "tran_input_timestamp": "6 Oct 2020 15:10:31",
  "tran_original_key": "TESTBANK_201667_000001",
  "tran_next_stage": "TESTBANK_Stage2",
  "tran_extra_referencel": "",
  "documents_uri": "https://devafp.ddcap.com/api/deals/TESTBANK_201667_000001/documents/",
  "tran_original_trade_date": "6 Oct 2020",
  "tran_fee_usd": "58.36",
  "tran_deferred_amount": 0.0,
  "tran_principal": 1000000.12,
  "tran_structure_code": "TESTBANK_STRUCTURE",
  "tran_dept": "",
  "tran_quantity_unit": "Metric Tonnes",
  "tran_end_buyer": "Condor Trade Limited",
  "transaction_no": "201667",
  "tran_client_code": "P",
  "tran_agreed_profit_rate": "",
  "tran_commodity": "CU",
  "tran_cycle_identifier": "",
  "tran_price": 5714.2864,
  "tran_class": "TEST",
  "transaction_key": "TESTBANK_201667_000001",
  "tran_client_name": "Principal",
  "tran_trade_date": "6 Oct 2020",
  "tenor": ""
}
```

```
"sub_transaction_no": "1",
"tran_stage_code": "TESTBANK_Stage1",
"tran_fee_exchange_rate": "0.85675120",
"tran_maturity_date": "",
"bcdt_uri": "https://devafp.ddcap.com/api/bcdts/TESTBANK_P_DEPOSIT",
"tran_input_user": "GarethTEST (Gareth Test User)",
"tran_exchange_rate_inverse": "",
"tran_bank_code": "TESTBANK",
"tran_initial_seller": "DD&Co Limited",
"tran_total_principal": 0.0,
"tran_your_reference": "testing 2020-10-06T14:09:38.815Z",
"tran_branch": "",
"tran_exchange_rate": "1.1672 USD = 1 EUR",
"tran_fee_currency": "EUR",
"tran_original_settlement_date": "6 Oct 2020",
"tran_quantity": 175.0,
"tran_outstanding_quantity": 0.0,
"principal_plus_comm": "0.00",
"original_deal_uri": "https://devafp.ddcap.com/api/deals/TESTBANK_201667_000001",
"uri": "https://devafp.ddcap.com/api/deals/TESTBANK_201667_000001",
"tran_outstanding_principal": 0.0,
"tran_comdty_alloc_number": "402105 201667 CU",
"tran_fee": "50.00",
"location_rule": "LME Non-UK Bonded Warehouse",
"tran_tre_type": "Original Deal",
"tran_deal_type": "DEPOSIT",
"tran_total_quantity": 0.0
}
```

The below example shows how we make a change to the currency from EUR to SAR.

To do this we will send a PATCH request as shown below

We send a PATCH request to /api/deals/{\"transaction_key\"}

Example

PATCH https://devafp.ddcap.com/api/deals/TESTBANK_201667_000001

With request body

Example Request body JSON (application/json)

```
{
  "tran_bank_code": "TESTBANK",
  "tran_client_code": "P",
  "tran_deal_type": "DEPOSIT",
  "tran_commodity": "",
  "tran_currency": "SAR",
  "tran_principal": "1000000.12",
  "tran_settlement_date": "6 Oct 2020",
  "tran_trade_date": "6 Oct 2020",
  "tran_trader_name": "",
  "tran_maturity_date": "",
  "tran_your_reference": "testing {{current_timestamp}}",
  "tran_class": "STANDARD"
}
```

Note above we have only changed the “tran_currency” field from “EUR” to “SAR” but we include all other details as they were sent originally. Any change to any field will be seen as an edit.

Response example – HTTP Status: 200 OK

A successful call returns a JSON-formatted response body

```
{
  "tran_currency": "SAR",
  "tran_bcdt_code": "TESTBANK_P_DEPOSIT",
  "tran_settlement_date": "6 Oct 2020",
  "tran_original_maturity_date": "",
  "tran_deferred_price": 0.0,
```

```

"tran_extra_reference2": "",
"tran_input_timestamp": "6 Oct 2020 15:10:31",
"tran_original_key": "TESTBANK_201667_000001",
"tran_next_stage": "TESTBANK_Stage2",
"tran_extra_reference1": "",
"documents_uri": "https://devafp.ddcap.com/api/deals/TESTBANK_201667_000001/documents/",
"tran_original_trade_date": "6 Oct 2020",
"tran_fee_usd": "13.33",
"tran_deferred_amount": 0.0,
"tran_principal": 1000000.12,
"tran_structure_code": "TESTBANK_STRUCUTRE",
"tran_dept": "",
"tran_quantity_unit": "Metric Tonnes",
"tran_end_buyer": "Condor Trade Limited",
"transaction_no": "201667",
"tran_client_code": "P",
"tran_agreed_profit_rate": "",
"tran_commodity": "CU",
"tran_cycle_identifier": "",
"tran_price": 20000.0024,
"tran_class": "TEST",
"transaction_key": "TESTBANK_201667_000001",
"tran_client_name": "Principal",
"tran_trade_date": "6 Oct 2020",
"tenor": "",
"sub_transaction_no": "1",
"tran_stage_code": "TESTBANK_Stage1",
"tran_fee_exchange_rate": "3.75000000",
"tran_maturity_date": "",
"bcdt_uri": "https://devafp.ddcap.com/api/bcdts/TESTBANK_P_DEPOSIT",
"tran_input_user": "GarethTEST (Gareth Test User)",
"tran_exchange_rate_inverse": "",
"tran_bank_code": "TESTBANK",
"tran_initial_seller": "DD&Co Limited",
"tran_total_principal": 0.0,

```

```
{
  "tran_your_reference": "testing 2020-10-06T14:58:24.117Z",
  "tran_branch": "",
  "tran_exchange_rate": "3.75000000 SAR = 1 USD ",
  "tran_fee_currency": "SAR",
  "tran_original_settlement_date": "6 Oct 2020",
  "tran_quantity": 50.0,
  "tran_outstanding_quantity": 0.0,
  "principal_plus_comm": "0.00",
  "original_deal_uri": "https://devafp.ddcap.com/api/deals/TESTBANK_201667_000001",
  "uri": "https://devafp.ddcap.com/api/deals/TESTBANK_201667_000001",
  "tran_outstanding_principal": 0.0,
  "tran_comdty_alloc_number": "402105 201667 CU",
  "tran_fee": "50.00",
  "location_rule": "LME Non-UK Bonded Warehouse",
  "tran_tre_type": "Original Deal",
  "tran_deal_type": "DEPOSIT",
  "tran_total_quantity": 0.0
}
```

Any field we include in the initial POST request body, apart from “tran_bank_code” and “tran_trade_date” can be changed.