

Match-Trader Platform API



match-trade technologies

- Login (rest)
- Login with one time token (rest)
- Register (rest)
- Refresh token (rest)
- Market Watch (rest)
- Open position (rest)
- Close positions (rest)
- Edit position (rest)
- Get balance (rest)
- Get opened positions (rest)
- Get active orders (rest)
- Create pending order (rest)
- Edit pending-order (rest)
- Cancel pending order (rest)

1. Login (rest)

a. Request:

1. Address: `managerUrl/co-login`
2. Method: POST
3. Headers:

Headers

```
Accept: application/json  
Content-Type: application/json
```

4. Request body:

Request body

```
{  
  "email": "string",  
  "password": "string",  
  "brokerId": "string" // identifier of a broker  
}
```

b. Response:

- i. Correct response code: 200

ii. Response body:

Response

```
{
  "email": "string",
  "token": "string",
  "accounts": [
    {
      "tradingAccountId": "string",
      "offer": {
        "uuid": "string",
        "partnerId": "string",
        "created": "2020-08-19T07:54:11.936Z",
        "name": "string",
        "description": "string",
        "initialDeposit": 0,
        "demo": true,
        "branch": {
          "uuid": "string"
        },
        "system": {
          "demo": true,
          "name": "string",
          "uuid": "string",
          "active": true
        },
        "moneyManager": "string"
      },
      "tradingApiToken": "string",
      "branchUuid": "string",
      "created": "2020-08-19T07:54:11.936Z",
      "uuid": "string"
    }
  ]
}
```

iii. Cookies set:

Set-Cookie

```
co-auth=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9; Max-Age=3600;
Expires=Wed, 19-Aug-2020 08:50:08 GMT; Path=/
//authentication token used for future requests
rt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9; Max-Age=31536000;
Expires=Thu, 19-Aug-2021 07:50:08 GMT; Path=/manager/refresh-
token; Secure; HttpOnly
//authentication token to be used in refresh-token request
```

2. Login with one time token (rest)

a. Request:

1. Address: managerUrl /login/co/with-token
2. Method: POST
3. Headers:

Headers

```
Accept: application/json
Content-Type: application/json
```

4. Request body:

Request body

```
{
  "token": "string"
}
```

b. Response:

- i. Correct response code: 200
- ii. Response body:

Response

```
{
  "email": "string",
  "token": "string",
  "accounts": [
    {
      "tradingAccountId": "string",
      "offer": {
        "uuid": "string",
        "partnerId": "string",
        "created": "2020-08-19T07:54:11.936Z",
        "name": "string",
        "description": "string",
        "initialDeposit": 0,
        "demo": true,
        "branch": {
          "uuid": "string"
        },
        "system": {
          "demo": true,
          "name": "string",
          "uuid": "string",
          "active": true
        },
        "moneyManager": "string"
      },
      "tradingApiToken": "string",
      "branchUuid": "string",
      "created": "2020-08-19T07:54:11.936Z",
      "uuid": "string"
    }
  ]
}
```

iii. Cookies set:

Set-Cookie

```
co-auth=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9; Max-Age=3600;
Expires=Wed, 19-Aug-2020 08:50:08 GMT; Path=/
//authentication token used for future requests
rt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9; Max-Age=31536000;
Expires=Thu, 19-Aug-2021 07:50:08 GMT; Path=/manager/refresh-
token; Secure; HttpOnly
//authentication token to be used in refresh-token request
```

3. Register (rest)

a. Request:

- i. Address: `managerUrl/user`
- ii. Method: POST
- iii. Headers:

```
Accept: application/json
Content-Type: application/json
```

iv. Request body:

```
{
  "offerId": "string", // cannot be null
  "email": "string", // cannot be null
  "password": "string", // cannot be null, minimum 8 characters
  "partnerId": "string", // cannot be null
  "name": "string",
  "country": "string",
  "parentTradingAccountUuid": "string", // optional id of parent
  account
  "surname": "string",
  "dateOfBirth": "date", // YYYY-MM-DD format
  "phone": "string",
  "city": "string",
  "postCode": "string",
  "address": "string",
  "state": "string"
}
```

b. Response:

- i. Correct response code: 200,
- ii. Response body:

```
{
  "token": "string"
}
```

iii. Error response example:

1. Error code: 400
2. Error response Body:

```
MISSING_PARAMETER: email
```

4. Refresh token (rest)

a. Request:

- i. Address: managerUrl/**refresh-token**
- ii. Method: POST
- iii. Headers:

Headers

```
Accept: application/json
Content-Type: application/json
```

iv. Request body: no request body

v. Required cookies:

Required cookie

```
rt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
//token to be refreshed
```

b. Response:

- i. Correct response code: 200
- ii. Response body: no response body
- iii. Cookies set:

Cookies

```
co-auth=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9; Max-Age=3600;
Expires=Wed, 19-Aug-2020 08:50:08 GMT; Path=/
//authentication token used for future requests
rt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9; Max-Age=31536000;
Expires=Thu, 19-Aug-2021 07:50:08 GMT; Path=/manager/refresh-
token; Secure; HttpOnly
//authentication token to be used in refresh-token request
```

5. Market Watch (rest)

a. Request:

- i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/quotations
- ii. Method: GET
- iii. Headers:

```
Auth-trading-api: "string" // trading API token from
tradingAccounts
co-auth: "string" // authentication token
```

iv. Request parameters

```
?symbols=EURUSD,BTCUSD
```

b. Response:

- i. Response body:

```
{
  "body": [
    {
      "symbol": "AUDUSD",
      "bid": "0.74286",
      "ask": "0.74288",
      "change": "0.19",
      "high": "0.74366",
      "low": "0.73980",
      "timestampSec": 1606986916
    },
    {
      "symbol": "BTCUSD",
      "bid": "19190.07",
      "ask": "19203.65",
      "change": "-0.15",
      "high": "19455.99",
      "low": "18877.25",
      "timestampSec": 1606986915
    }
  ]
}
```

6. Open position (rest)

a. Request:

- i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/**position/open**
- ii. Method: POST
- iii. Headers:

Request Headers

```
Accept: application/json
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOiJIUzI1NiIs... // co-auth token
```

iv. Body:

Request body

```
{
  "instrument": "AUDUSD", // shortcut name of instrument
  "orderSide": "BUY", // side of trade: BUY or SELL
  "volume": 0.01, // amount of trade
  "slPrice": 0, // stop-loss price: 0 if not set
  "tpPrice": 0, // take-profit price: 0 if not set
  "isMobile": true // request source: true if mobile, false
if desktop
}
```

b. Response:

i. Response body:

```
{
  "status": "status", // possible status: OK, REJECTED,
PARTIAL_SUCCESS
  "nativeCode": null,
  "errorMessage": null
}
```

ii. Example error:

1. Status code: 401 Unauthorized

7. Close positions (rest)

a. Request:

- i. Destination: `WL_DOMAIN/mtr-api/SYSTEM_UUID/positions/close`
- ii. Method: POST
- iii. Headers:

```
Accept: application/json
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOiJIUzI1NiIs... // co-auth token
```

iv. Body:

```
[ //we can close multiple positions with one request
  {
    "positionId":"M15978271593", // id of opened
position
    "instrument":"AUDUSD", // shortcut name of
instrument
    "orderSide":"BUY", // side of trade: BUY or SELL
    "volume":"0.01" // amount of trade
  }
]
```

b. Response:

i. Response body:

```
{
  "status":"status", // possible status: OK, REJECTED,
PARTIAL_SUCCESS
  "nativeCode":null,
  "errorMessage":null
}
```

ii. Example error:

1. Status code: 401 Unauthorized

8. Edit position (rest)

a. Request:

- i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/**position/edit**
- ii. Method: POST
- iii. Headers:

```
Accept: application/json
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOiJIUzI1NiIs...
```

iv. Body:

```
{
    "instrument": "AUDUSD", // shortcut name of instrument
    "id": "W15978271592", // id of opened position
    "orderSide": "SELL", // side of trade: BUY or SELL
    "volume": 0.01, // amount of trade. Cannot be different
    // than given during open
    "slPrice": 0, // stop-loss price: 0 if not set
    "tpPrice": 0.72689, // take-profit price: 0 if not set
    "isMobile": false // request source: true if mobile,
    // false if desktop
}
```

b. Response:

i. Response body:

```
{
    "status": "status", // possible status: OK, REJECTED,
    // PARTIAL_SUCCESS
    "nativeCode": null,
    "errorMessage": null
}
```

ii. Example error:

1. Status code: 401 Unauthorized

9. Get balance (rest)

a. Request:

- i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/**balance**
- ii. Method: GET
- iii. Headers:

```
Accept: application/json
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOiJIUzI1NiIs...
```

iv. Body: no body

b. Response:

i. Example response body:

```
{
  "balance": "1010.00",
  "equity": "1010.00",
  "margin": "0.00",
  "freeMargin": "1010.00",
  "marginLevel": "0",
  "profit": "0",
  "netProfit": "0",
  "credit": "0.00",
  "currency": "USD"
}
```

ii. Example error:

1. Status code: 401 Unauthorized

10. **Get opened positions (rest)**

a. Request:

- i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/open-positions
- ii. Method: GET
- iii. Headers:

```
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOi...
```

b. Response:

- i. Example response body:

```

{
  "positions": [{
    "id": "M159904237816",
    "symbol": "AUDUSD",
    "volume": "0.01",
    "side": "BUY",
    "openTime": "2020-09-02T13:00:19",
    "openPrice": "0.73335",
    "stopLoss": "0.00000", // 0 => no stopLoss defined
    "takeProfit": "0.00000", // 0 => no takeProfit
    "swap": "-0.00001894",
    "profit": "-0.00121271",
    "netProfit": "-0.00123293",
    "currentPrice": "0.72068",
    "stopLossInMainWallet": "0.00000000",
    "takeProfitInMainWallet": "0.00000000",
    "commission": "-0.00000128",
    "bidPrice": 0.72158,
    "askPrice": 0.72159
  ]
}

```

11. Get active orders (rest)

a. Request:

- i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/**active-orders**
- ii. Method: GET
- iii. Headers:

```

Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOi... // co-auth token

```

b. Response:

- i. Example response body:

```

{
  "orders": [{
    "id": "W1600679776121",
    "symbol": "ETHUSD",
    "volume": "5",
    "type": "LIMIT",
    "creationTime": "2020-09-22T12:31:55",
    "currentPrice": "0.00",
    "activationPrice": "330.346",
    "side": "BUY", // BUY or SELL
    "stopLoss": "0.000", // 0 => no stopLoss defined
    "takeProfit": "0.000", // 0 => no takeProfit
    defined
    "stopLossInMainWallet": "0.00",
    "takeProfitInMainWallet": "0.00",
    "bidPrice": 340.04,
    "askPrice": 342.742
  }]
}

```

12. Create pending order (rest)

a. Request:

- i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/pending-order/create
- ii. Method: POST
- iii. Headers:

```

Accept: application/json
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOiJIUzI1NiIs... // co-auth token

```

iv. Body:

```

{
  "instrument": "BTCUSD",
  "orderSide": "SELL",
  "volume": 0.04,
  "price": 11406.62,
  "type": "LIMIT", // "LIMIT" or "STOP"
  "slPrice": 0, // No stop-loss set
  "tpPrice": 11406.63, //Take Profit price set
  "isMobile": false // Whether we using mobile device or not
}

```

b. Response:

i. Response body:

```
{
    "status": "status", // possible status: OK, REJECTED,
    PARTIAL_SUCCESS
    "nativeCode": null,
    "errorMessage": null
}
```

ii. Example error:

1. Status code: 401 Unauthorized

13. Edit pending-order (rest)

a. Request:

i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/pending-order/edit

ii. Method: POST

iii. Headers:

```
Accept: application/json
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOiJIUzI1NiIs... // co-auth token
```

iv. Body:

```
{
    "instrument": "BTCUSD",
    "id": "W16026573113",
    "orderSide": "BUY",
    "type": "LIMIT", // "LIMIT" or "STOP"
    "volume": 0.01,
    "slPrice": 0, // No stop-loss set
    "tpPrice": 11705.21, //Take Profit price set
    "priceOrder": 11306.43,
    "isMobile": false // Whether we using mobile device or not
}
```

b. Response:

i. Response body:

```
{
    "status": "status", // possible status: OK, REJECTED,
    PARTIAL_SUCCESS
    "nativeCode": null,
    "errorMessage": null
}
```

ii. Example error:

1. Status code: 401 Unauthorized

14. Cancel pending order (rest)

a. Request:

i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/pending-order/cancel

ii. Method: POST

iii. Headers:

```
Accept: application/json
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOiJIUzI1NiIs... // co-auth token
```

iv. Request body:

```
{
  "instrument": "BTCUSD",
  "id": "W16026573113",
  "orderSide": "BUY",
  "type": "LIMIT", // "LIMIT" or "STOP"
  "isMobile": false // Whether we using mobile device or not
}
```

b. Response:

i. Response body:

```
{
  "status": "status", // possible status: OK, REJECTED,
  PARTIAL_SUCCESS
  "nativeCode": null,
  "errorMessage": null
}
```

15. Get closed positions (ws)

a. Request:

i. Destination: : ws/app/snapshot/closed-positions

ii. Method: WS Send

iii. Headers:

```
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOi... // co-auth token
```

b. Body:


```
{
  "symbols":["AUDUSD","BTCUSD"],
  "from":"2020-09-16T06:04:53.071Z",
  "to":"2020-09-23T06:04:53.071Z"
}
```

c. Response:

- i. Destination: /user/queue/snapshot/closed-positions
- ii. Example response body:

```
{
  "operations":[{"
    "openTime":"2020-06-04T06:07:43.827",
    "openPrice":"0.89475",
    "symbol":"EURGBP",
    "id":"W1590737186242",
    "volume":"4.98",
    "stopLoss":"0.00000", // 0 => no stopLoss defined
    "takeProfit":"0.00000", // 0 => no takeProfit
    defined
    "time":"2020-07-16T07:09:17.26",
    "closePrice":"0.90888",
    "commission":"-17.02",
    "swap":"-1128.38",
    "profit":"8821.00",
    "side":"BUY",
    "netProfit":"7675.60"
  }]}
}
```

16. Get ledgers (rest)

a. Request:

- i. Destination: WL_DOMAIN/mtr-api/SYSTEM_UUID/finance
- ii. Method: POST
- iii. Headers:

```
Auth-trading-api: string // trading API token
Cookie: co-auth=eyJhbGciOi...
```

iv. Request body:

```
{
    "types":[0,1,4,2,3], // 0 => DEPOSIT, 1 => WITHDRAWAL, 2
=> CREDIT_IN, 3 => CREDIT_OUT, 4 => AGENT_COMMISSION, 5 => OTHER
    "from":"2020-09-16T06:04:53.071Z",
    "to":"2020-09-23T06:04:53.071Z",
    "limit":200 // maximum number of ledgers
}
```

b. Response:

i. Example response body:

```
{
    "operations":[
        {
            "id":"1600079704179242601",
            "type":"DEPOSIT",
            "time":"2020-09-18T07:40:48.211",
            "profit":"1.00"}, // amount of deposit (positive
for inflows)
        {
            "id":"1600079704344412995",
            "type":"WITHDRAWAL",
            "time":"2020-09-23T05:44:43.475",
            "profit":"-36.78" // amount of withdrawal
(negative for outflows)
        }
    ]
}
```