MIAX Options Exchange
MIAX Pearl Options Exchange
MIAX Emerald Options Exchange
MIAX Sapphire Options Exchange

Member Firm Portal API Specifications

Modified November 2025

© 2025 Miami International Holdings, Inc. and its subsidiaries. All Rights Reserved.

7 Roszel Rd | Princeton, NJ 08540

Table of Contents

1.0 MFP API Format & Parameters

1.1 General Information

The Member Firm Portal (MFP) API allows member firms' systems to retrieve data and invoke various transactional services. The MFP API is provided as an internet accessible HTTPS/REST protocol. The API exposes functionality that allows member firms to automate the following tasks:

- Authentication
- Trade Search
- Trade Clearing Updates
- Market Maker Assignments
- Risk Protection Configuration and Management

1.2 Access and Certification Testing

In order to use the MFP API, member firms must:

- Request MFP API access in the Firm Test Bed (FTB) environment from MIAX Trading Operations
- Certify the MFP API functionality in the Firm Test Bed (FTB) environment

MIAX Trading Operations will provide certification test scripts and facilitate the certification testing process with the member firm. Upon completion of certification, MIAX Trading Operations will provision access in the production environment and provide the member firm with application credentials required to use the MFP API.

1.3 Authentication

In order to use MFP APIs, the member firm's client application must first invoke the Authentication Service and authenticate using the username and password of the member firm's client application. If authentication is successful, the Authentication Service returns Authentication and Refresh tokens along with the respective token expiration times. All MFP API requests to view and update data require the Access token to be included in the request. If the Access token is omitted, expired or incorrect, the MFP API request will be rejected.

If the Access token has expired, but the Refresh token is valid, then the member firm can request a new Access token by providing the Refresh token to the Authentication Service. Successful submission of a valid, unexpired Refresh token to the Authentication Service will result in issuance of new Authentication and Refresh tokens.

If the Refresh token has expired, then the customer must reauthenticate using the process described above.

1.4 Exchange Codes and API End-Point URLs

This specification document uses {api_url} to indicate the base URL to be used for accessing the API end-points. The {api_url} and end-point URLs for the Member Firm Portal (MFP) API are environment specific. Member firms should use the following {api_url} values when using the MFP API:

- Firm Test Bed 1 Environment
    - **`https://ftb1.api.miaxportal.com/mfp-api`**
- Firm Test Bed 2 Environment
    - **`https://ftb2.api.miaxportal.com/mfp-api`**
- Production Environment
    - **`https://api.miaxportal.com/mfp-api`**

The exchange codes to be used in service end-point URLs are:

- **M** – MIAX Options
- **P** – Pearl Options
- **D** – Emerald Options
- **S** – Sapphire Options

1.5 Request & Response Model

Request

The MFP API supports two types of requests:

HTTP GET – used to retrieve data
HTTP POST – used to submit a data modification request

MFP API requests are submitted as a set of key/value pairs in JSON format. The service definitons detailed in this document define the specific request parameter requirements that are applicable to each service. Every request, whether GET or POST, must include a valid Access token in order to be processed. The Access token should be included in the HTTP Header field Authorization with value "Bearer <token value>". For example:

```
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
```

In addition, all requests must include the following HTTP Header fields:

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

The following HTTP header is recommended to be included on all requests:

```
Connection: keep-alive
```

All Member Firm Portal API end-points, except for the Authentication and Refresh services, are unique per exchange. All Member Firm Portal API end-points support a versioning strategy that allows multiple versions of the API to be available to member firms at the same time. Different API end-points may have different active versions. The version for each API end-point is included in its URL. The timing and functionality of upcoming changes, updates or deprecations of API versions are announced by MIAX Trading Operations via Member Firm Alerts.

To access a specific exchange API end point, the exchange code and version are included in the URL of the service call, as follows:

```
{api_url}/{service}/{version}/{endpoint-path}/{exchange_code}
```

For example, the following URL would invoke the IsTradeCorrectionWindowOpen service for the MIAX Options exchange:

```
{api_url}/trades/v1/isTradeCorrectionWindowOpen/M
```

Example Request

```
GET /trades/v1/search/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Connection: keep-alive
{
 "tradeDate": "2020-08-19",
 "securitySymbol": "SPY",
 "startTime": "09:30:00",
 "endTime": "16:15:00"
}
```

Response

Responses to MFP API requests are returned as a set of key/value pairs in JSON format. The service definitions define the fields included in the response along with an indication of whether they are required (always present in the response).

As new features and functionality are added to MFP API, it may be necessary to add new fields to the response layout. Member firms are encouraged to design their applications to support the addition of new fields and to not rely on specific field positons, as these may change.

The following table defines the structure of an MFP API response:

Field	Format	Required	Value	Description
status	number	Yes	100 through 700	HTTP status code 200 = Success 700 = Request Completed but with Errors For more details, see "error"
error	string	No	"Bad request"	If status is not 200, contains the HTTP error code description.
timestamp	ISO datetime format	Yes	"2022-08-01T10:00:00.000-04:00"	The time when request was received.
path	string	Yes	"/trades/v1/isTradeCorrectionWindowOpen/M"	The requested endpoint.
message	string	No	"No exchange with exchange code 'Q' was found."	If status is 600 through 700, will contain a detailed error message.
result	string	Yes	JSON	Response payload in JSON format as per the service definition.
deprecated	boolean	No	true	A value of "true" indicates this endpoint has been deprecated
deprecated_action	string	Conditional	"Moved"	Possible Values: "Moved" "Removed" "Upgrade Required" If the value of the "deprecated" field is true, then deprecated_action is required. Indicates the action (if any) that is expected for the deprecated end point
deprecated_message	string	Conditional	"This feature will not be supported after version v2"	If the value of the "deprecated" field is true, then deprecated_message is required. Provides a message indicating details of the action (if any) that is

				required by member firms with regards to the deprecated endpoint
sunset	boolean	No	true	A value of “true” indicates this endpoint has been sunset
sunset_action	string	Conditional	“Removed”	Possible Values: “Moved” “Removed” “Upgrade Required” If the value of the “sunset” field is true, then sunset_action is required. Indicates the action (if any) that is expected for the end point that is sunset
sunset_message	string	Conditional	“This endpoint has moved to a different service path”	If the value of the “sunset” field is true, then sunset_message is required. Provides a message indicating details of the action (if any) that is required by member firms with regards to the endpoint that is sunset

Example Response

```
HTTP/1.1 200 OK
{
 "status": 200,
 "timestamp": "2022-08-01T15:00:00.000-04:00",
 "path": " /trades/v1/isTradeCorrectionWindowOpen/M",
 "result": {
  "correctionWindowOpen": true
 },
}
```

1.6 Examples for Request & Response per Service

This document includes examples of Request and Response messages for all API end points.

See Appendix B: Examples

1.7 Current API End Point Versions

Service Name	Supported MIAX Options Exchange(s)	Current API Path	Current Version
2.1 Authentication	All	`/auth/v1/openid-connect/token`	v1
2.2 Refresh	All	`/auth/v1/openid-connect/refresh`	v1
3.1 Get Current Business Date	All	`/exchange/v1/getCurrentBusinessDate/{exchange_code}`	v1
3.2 Get Next Business Date	All	`/exchange/v1/getNextBusinessDate/{exchange_code}`	v1
4.1 Get Firm Codes	All	`/firms/v1/getFirmCodes/{exchange_code}`	v1
5.1 Is Trade Correction Window Open	All	`/trades/v1/isTradeCorrectionWindowOpen/{exchange_code}`	v1
5.2 Trade Search Sevice	All	`/trades/v1/search/{exchange_code}`	v1
5.3 Trade History Sevice	MIAX Sapphire Options	`/trades/v1/history/{exchange_code}?<query-parameters>`	v1
5.4 Trade Restore Sevice	MIAX Sapphire Options	`/trades/v1/restore/{exchange_code}`	v1
5.5 Trade Clearing Update Service	All	`/trades/v1/correctTrades/{exchange_code}`	v1
6.2 Get Assignments for Business Date	All	`/assignments/v1/getAssignmentsForBusinessDate/{exchange_code}`	v1
6.3 MPID Assignments	All	`/assignments/v1/addAssignments/{exchange_code}`	v1
6.3 MPID Assignments	All	`/assignments/v1/updateAssignments/{exchange_code}`	v1
6.3 MPID Assignments	All	`/assignments/v1/deleteAssignments/{exchange_code}`	v1

2.0 Authentication Services

2.1 Authentication

Member firms can use this API to authenticate with the MFP API. The credentials required by the Authentication Service are provided by the MIAX Trading Operations group as part of the firm access provisioning process. Member firms will be assigned one username/password by the MIAX Trading Operations group as part of the initial certification process. Firms can request multiple username/passwords as needed per the member firm's system segmentation requirements.

If authentication is successful, this service returns Authentication and Refresh tokens. The Authentication token must be included on all requests to any MFP API services. The Refresh token may be used to acquire a new Authentication token.

Authentication tokens expire after 30 minutes.
Refresh tokens expire after 8 hours.

Request

- URL
  - `{api_url}/auth/v1/openid-connect/token`
- Method
  - POST
- Header
  - Content Type: application/json
  - Accepts: application/json
- Body

Field	Required	Data Type	Example	Description
username	Yes	string	"test"	Username for the specific member firm's application accessing the MFP API
password	Yes	string	"test"	Password assigned to the member firm's application accessing the MFP API

Response

Field	Required	Data Type	Example	Description
Standard response fields as per Response Model				
result { /* Provided if status is 200 */				
access_token	Yes	string	JWT token	Access token that must be included in all business endpoint requests. An invalid, expired or missing token will cause the request to fail.
expires_in	Yes	number	14400	Time in seconds after which the access token will expire.
refresh_token	Yes	string	JWT token	Refresh token that can be used to request a new access token.
refresh_expires_in	Yes	number	43200	Time in seconds after which the refresh token will expire.
token_type	Yes	string	"Bearer"	Type of access token (always Bearer).
not_before_policy	Yes	number	0	Time before which the token will not be accepted for processing (always 0).
}				

Examples

See Appendix B: Example: 2.1 Authentication

2.2 Refresh

When an Authentication token expires, users can re-acquire an Authentication token by using the Refresh API. This API retrieves a fresh Authentication token for the specified firm API user using a submitted refresh token. If authentication is successful, returns Authentication and Refresh tokens.

Request

- URL
  - `{api_url}/auth/v1/openid-connect/refresh`
- Method
  - POST
- Header
  - Content Type: application/json
  - Accepts: application/json
- Body

Field	Required	Data Type	Example	Description
refresh_token	Yes	string	“1234567890ABCDEFGHIJK…”	Refresh token that was obtained during authentication.

Response

Field	Required	Data Type	Example	Description
Standard response fields as per Response Model				
result { /* Provided if status is 200 */				
access_token	Yes	string	JWT token	Access token that must be included in all business endpoint requests. An invalid, expired or missing token will cause the request to fail.
expires_in	Yes	number	14400	Time in seconds after which the access token will expire.

refresh_token	Yes	string	JWT token	Refresh token that can be used to request a new access token.
refresh_expires_in	Yes	number	43200	Time in seconds after which the refresh token will expire.
token_type	Yes	string	“Bearer”	Type of access token (always Bearer).
not_before_policy	Yes	number	0	Time before which the token will not be accepted for processing (always 0).
}				

Examples

See Appendix B: Example: 2.2 Refresh

3.0 Exchange Services

3.1 Get Current Business Date

The Get Current Business Date service returns the current business date for the specified exchange.

Request

- URL
    - `{api_url}/exchange/{version}/getCurrentBusinessDate/{exchange_code}`
- Method
    - GET
- Header
    - Accepts: application/json
    - Authorization: Bearer {access_token}
- Body
    - This service does not require any submission paramaters

Response

Field	Required	Data Type	Example	Description
Standard response fields as per Response Model				
result { /* Provided if status is 200 */}				
currentBusinessDate	Yes	string in ISO date format	“2020-06-22”	Date corresponding to Current Business Date for the requested exchange

Examples

See Appendix B: Example: 3.1 Get Current Business Date

3.2 Get Next Business Date

The Get Next Business Date service returns the next business date for the specified exchange.

Request

- URL
    - `{api_url}/exchange/{version}/getNextBusinessDate/{exchange_code}`
- Method
    - GET
- Header
    - Accepts: application/json
    - Authorization: Bearer {access_token}
- Body
    - This service does not require any submission paramaters

Response

Field	Required	Data Type	Example	Description
Standard response fields as per Response Model				
result { /* Provided if status is 200 */				
nextBusinessDate	Yes	string in ISO date format	"2020-06-22"	Date corresponding to Next Business Date for the requested exchange
}				

Examples

See Appendix B: Example: 3.2 Get Next Business Date

4.0 Firm Services

4.1 Get Firm Codes

The Get Firm Codes service provides MIAX specific Firm Code/s for which the member firm is certified to access and modify configurations (such as Market Maker Assignments) at the specified exchange

Request

- URL
  - `{api_url}/firms/{version}/getFirmCodes/{exchange_code}`
- Method
  - GET
- Header
  - Accepts: application/json
  - Authorization: Bearer {access_token}
- Body
  - This service does not require any submission paramaters

Response

Field	Required	Data Type	Example	Description
Standard response fields as per Response Model				
result { /* Provided if status is 200 */				
firmCodes	Yes	string array	“ABCD”, “WXYZ”	Comma separated list of Firm Codes which the member firm is entitled to access at the specified exchange
}				

Examples

See Appendix B: Example: 4.1 Get Firm Codes

5.0 Trade Services

5.1 Is Trade Correction Window Open

The IsTradeCorrectionWindowOpen service identifies whether the correction window is open for a specified exchange. Trade corrections can only be submitted while the correction window is open. Generally, the correction window closes approximately 60 minutes after the end of the trading session.

Request

- URL
  - `{api_url}/trades/{version}/isTradeCorrectionWindowOpen/{exchange_code}`
- Method
  - GET
- Header
  - Accepts: application/json
  - Authorization: Bearer {access_token}
- Body
  - This service does not require any submission paramaters

Response

Field	Required	Data Type	Example	Description
Standard response fields as per Response Model				
result { /* Provided if status is 200 */				
correctionWindowOpen	Yes	boolean	true / false	Flag indicating whether the correction window is open for the specified exchange.
}				

Examples

See Appendix B: Example: 5.1 Is Trade Correction Window Open

5.2 Trade Search Service

Member firms may use the Trade Search service to retrieve options trades based on specified search criteria. The response will include the Trade ID, Version and other fields that are required on submissions to the Correct Trade service. The response to the Search Trade service is limited to 1000 records.

All trade search requests must include at minimum the following fields:

Field	Required	Data Type	Example	Description
tradeDate	Yes	string on ISO date format	"2020-06-22"	Date for which to search trades
executingMpid	Conditional	string	"ABC1"	MPID which executed the trade. In addition to tradeDate, one of the following MUST be provided: • executingMpid • securitySymbol • symbol • tradeId • underlying • transactionId Member firms may elect to provide more than one of the above search fields to further narrow the scope of the search results
securitySymbol	Conditional	string	"SPY"	Symbol of the security being traded. In addition to tradeDate, one of the following MUST be provided • executingMpid • securitySymbol • symbol • tradeId • underlying • transactionId Member firms may elect to provide more than one of the above search fields to further narrow the scope of the search results
tradeId	Conditional	number	123456789	Trade ID of the trade being searched. In addition to tradeDate, one of the following MUST be provided • executingMpid • securitySymbol

				• symbol • tradeId • underlying • transactionId Member firms may elect to provide more than one of the above search fields to further narrow the scope of the search results
underlying	Conditional	string	“SPY”	Symbol of an underlying security. In addition to tradeDate, one of the following MUST be provided • executingMpid • securitySymbol • symbol • tradeId • underlying • transactionId Member firms may elect to provide more than one of the above search fields to further narrow the scope of the search results
transactionId	Conditional	number	123456789	Transaction Id of the trade being searched. All clearing trades generated from a single matching engine transaction will have the same transaction Id. All trades that are part of a complex execution will have the same transaction Id In addition to tradeDate, one of the following MUST be provided • executingMpid • securitySymbol • symbol • tradeId • underlying • transactionId Member firms may elect to provide more than one of the above search fields to further narrow the scope of the search results

Request

- URL
  - `{api_url}/trades/v1/search/{exchange_code}`
- Method
  - POST
- Header
  - Content-Type: application/json
  - Accepts: application/json
  - Authorization: Bearer {access_token}
- Body
  - Search criteria that can be specified in the request body as JSON fields:

Field	Required	Data Type	Example	Description
tradeDate	Yes	string in ISO date format	"2020-06-22"	Date for which to search trades.
executingMpid	Conditional	string	"ABC1"	MPID which executed the trade. This is a conditionally required field, see details under Search Trades - Options
securitySymbol	Conditional	string	"SPY"	Symbol of the security being traded. This is a conditionally required field, see details under Search Trades - Options
tradeId	Contitional	number	123456789	Trade ID of the trade being searched. This is a conditionally required field, see details under Search Trades - Options
underlying	Conditional	string	"SPY"	Symbol of an underlying security. This is a conditionally required field, see details under Search Trades - Options
transactionId	Conditional	number	123456789	Transaction Id of the trade (or group of trades) being searched. This is a conditionally required field, see details under Search Trades - Options

securityScope	Conditional	string	"P"	Only applicable to exchangeCodes M, D, S Identifies type of instrument. P – Options U – Equity
clearingMpid	No	string	"ABC1"	MPID which cleared the trade
clientOrderId	No	string	"12345667890ABCD"	Client order ID.
cmta	No	number	123	CMTA account.
floorTradesOnly	No	Boolean	true	This field is applicatble to exchangeCode = S. If included for exchangeCodes M, P, D, this field will be ignored • If this field is included in the request for exchangeCode other than S, it will be ignored • If set to true, Trade results will be limited to Sapphire Floor trades in addition to any other applicable filters provided in the request *Default value is null which will return all floor and electronic trades*
customerAccount	No	string	"ABCxd"	Customer Account as submitted on the order.
endTime	No	string in ISO time format	"16:00:00"	Time up to which to search trades for the specified tradeDate. *Default value is 17:00:00*

expirationDate	No	string in ISO Date Format	“2022-08-26”	Expiration date of the option.
executionId	No	number	123456789012345678	Exchange assigned unique Execution Id
liquidityIndicator	No	string	“A”	For exchangeCode M, DA = Add R = Remove ‘ ‘ (space) = N/A (Not Applicable) For exchangeCode P, S M = Maker T = Taker ‘ ‘ (space) = N/A (Not Applicable)
occNumbers	No	string	“00001”	OCC number on the trade as string.
openCloseIndicator	No	string	“O”	Position Indicator For exchangeCode M, D *Option leg:* O = Opening position C = Closing position *Stock leg:* ‘ ‘ (space) = N/A (Not Applicable) For exchangeCode P, S O = Opening position C = Closing position
optionType	No	string	“C”	Type of option: C = Call P = Put
orderId	No	string	“12345678901234567890”	Exchange assigned Order ID on the trade.
orderText	No	string	“TEXT”	Order Text as submitted on the order.
origin	No	string	“B”	Origin type of the trade C = Priority Customer N = Non-Priority customer F = Firm

				B = Broker/Dealer M = MIAX member Market Maker A = Away Exchange Market Maker
price	No	string	"2.0700"	The price of the trade as a string with precision of 4 decimal places.
quantity	No	number	100	Quantity of the trade.
side	No	string	"B"	Side of the trade: B = Buy side S = Sell side
startTime	No	string in ISO time format	"09:30:00"	the trade search start time provided in the format hh:mm:ss *Default value 08:00:00*
strikePrice	No	string	"150.00"	Strike price of the option as a string with precision of 2 decimal places.

Response

Field		**Data Type**	**Example**	**Description**
Standard response fields as per Response Model				
result : [/* Provided if status is 200 */				
{				
tradeId		number	1234567890	Exchange assigned Trade ID
tradeDate		string in U.S. date format	"08/19/2022"	Date of the trade
tradeTime		string in U.S. time format	"09:48:29.123"	Time of the trade

transactionId		number	1234567890	All clearing trades generated from a single Engine transaction will have the same transaction ID
transactionTime		string in U.S. time format	"09:48:29.123"	Time of the transaction
tradeAction		string	"C"	Current state of the trade N – New trade was created as a result of a correction C – Clearing Change X – Trade was Cancelled by MIAX Trading Operations"" (empty string) – Trade was never corrected Corrections to Price or Size of the Trade can only be made by the MIAX Trading Operations Group
tradeVersion		number	1	Current version of the trade. The trade version is distinct from the correctionNumber of the trade
correctionNumber		number	1	Count of how many times a trade is corrected. Increments by 1 for each subsequent correction. Cancelling a trade does NOT increment its correctionNumber
executionId		number	123456789012345678	Exchange assigned unique Execution Id
strategyId		number	123456789	Only applicable to exchangeCode M, D, S Exchange assigned unique Complex Strategy Id for the day, Only applicable for complex order executions, otherwise set to zero
tradeAsOfDate		string in U.S. date format	"08/25/22"	As-of Date for As-of Trades
underlyingSymbol		string	"IBM"	Underlying security
securitySymbol		string	"IBM"	Symbol of the security

expirationDate		string in U.S. date format	"08/26/22"	Expiration date of the option
strikePrice		string	"110.00"	Strike price of the option
optionKind		string	"C"	Type of option: C = Call P = Put
quantity		number	100	Quantity
price		string	"4.5000"	Price of the trade with precision of 4 decimal places
side		string	"S"	Side of the trade: B = Buy S = Sell
sideFromFloor		boolean	true	This field is applicatble to exchangeCode = S. If set to true – indicates this trade side originated on the Sapphire Floor (Not included in the trade search response for exchangeCodes M, P, D)
introducingBadgeId		string	"ABC1"	This field is applicable for floor trades on exchangeCode = S. Sapphire assigned alphanumeric, 4 character Badge ID of the firm's Floor Broker who announced the deal on the floor. (Not included in the trade search response for exchangeCodes M, P, D)
billingTradeType		string	"D"	This field is applicable for floor trades on exchangeCode = S. Member firm assigned optional billing data. (Not included in the trade search response for exchangeCodes M, P, D)
executionExchangeCode		string	"M"	Market where a customer option order was traded, potentially after routing A = NYSE Amex

				B = BOX C = CBOE D = MIAX Emerald E = EDGX H = ISE Gemini I = ISE J = ISE Mercury M = MIAX Options N = NYSE Arca P = MIAX Pearl Q = NASDAQ OMX S = MIAX Sapphire T = NASDAQ BX U = MEMX V = IEX W = C2 (from CBOE) X = NASDAQ OMX PHLX Z = BATS
originalTradeId		number	1234567890	This field is applicatble to exchangeCode = S. For trades that are a result of an execution being split or trades that are restored using the Trade Restore Service, this field contains the tradeId of the original trade that was acted upon. For all other trades, this field will be blank. (Not included in the trade search response for exchangeCodes M, P, D)
productId		number	1234567890	Exchange-specific product ID
quoteId		string	“1234567890”	Quote ID
origin		string	“B”	Origin type of the trade C = Priority Customer N = Non-Priority customer F = Firm B = Broker/Dealer M = MIAX member Market Maker A = Away Exchange Market Maker
exMpid		string	“ABC1”	Executing MPID

clearingNumber		string	"00001"	OCC Clearing Number used to the clear the trade
clMpid		string	"ABC1"	Clearing MPID
cmta		number	123	CMTA in case of a give-up trade
multiAccountCode		string	"QA5"	Sub or multi account ID specified in FIX order
openCloseIndicator		string	"O"	Open/Clode indicator O = Open C = Close
clOrderId		string	"AB123456_123456"	Client order ID
directedFirmCode		string	"ABCD"	Only applicable to exchangeCode M, D Directed firm code for directed orders
customerAccount		string	"123456789"	Customer Account ID
orderText		string	"Order text"	Order text, optional data submitted on FIX order
orderDate		string in U.S. date format	"08/26/2022"	Order date
}				
{ } /* Multiple trade sides included if multiple trades match the search criteria. Limited to a max of 50 trade sides. */				
]				

Examples

See Appendix B: Example: 5.2 Search Trades Service

5.3 Trade Clearing Update Service

The Trade Clearing Update Service is available for the following exchanges:

- MIAX Options
- Pearl Options
- Emerald Options
- Sapphire Options

Member firms can use the the Trade Clearing Update Service API to submit clearing corrections for executions. The API accepts clearing update requests for a minimum of 1 execution and up to a maximum of 50 executions as part of a single request. All tradeIds included in a bulk clearing edit request must be unique. If an execution requires more than one clearing change, then each change must be submitted as a unique clearing edit request for that tradeId. For each tradeId, only fields that have changed as part of the clearing correction/s should to be submitted as part of the clearing update request.

Clearing updates submitted via the Trade Clearing Update Service API are added to a queue for processing. Member firms receive an immediate response regarding acceptance or rejection of their requested submission to the clearing edit queue.

Members can verify that the requested clearing edits are successful in one of two ways:

- Via the real-time Clearing Trade Drop interface
- Via the MFP Trade Search API

Successful clearing changes visible via the member firm's Clearing Trade Drop or via the MFP Trade Search Service API confirm the clearing edits have also been successfully submitted to the OCC.

The stock leg of an Options trade can be corrected via the Trade Clearing Update Service API on the current trade date. Only the following fields are allowed to be corrected on stock leg trades: executing MPID and Stock Clearing Accounts.

Note: T+1 Post Trade Adjustments

T+1 Trade Corrections are supported on the MIAX Sapphire Exchange. For additional information on when this feature will be available in the MIAX Options, Pearl Options and Emerald Options exchanges, please contact MIAX Trading Operations

- Stock Trades (SecurityScope = "U") on the MIAX Sapphire Exchange – Post Trade Adjustments to Stock Trades on T+1 is not supported. All corrections for Stock Trades must be made on the current business date.
- Options Trades (SecurityScope = "P") on the MIAX Sapphire Exchange – Post Trade Adjustments for non material changes are supported for Current and up to 2 past business dates. If trades are more than 2 business dates in the past, the trade correction request will be rejected

Post Trade Adjustments made to Options Trades on T+1 on the MIAX Sapphire Exchange will result in the following:

A new As-Of trade with original clearing values on the contra side of the trade and the corrected clearing values on the originator side of the trade. The contra side of the As-Of trade is not editable. The As-Of date is the trade date of the original past day trade. Each trade can be corrected only once on T+1. All subsequent non-material changes must be made to the resultant As-Of trade on the current business date.

All Trade correction requests must contain, at a minimum, the following fields

Field	Required	Data Type	Example	Description
tradeId	Yes	number	1234567890	Exchange assigned Trade ID
side	Yes	string	"S"	Side of the trade to be corrected: B = Buy S = Sell
correctionNumber	Yes	number	1	The most recent correction number for the trade that is to be corrected. CorrectionNumber for trades is available via the Clearing Trade Drop and via the MFP Trade Search Service API. If the correctionNumber provided is not the most recent, the trade correction request will be rejected.
tradeDate	Yes	string in ISO date format	"2020-06-22"	The date on which the trade occurred. For exchangeCode M, P, D: Only Current business date trade corrections are supported. For exchangeCode S: See Note: T+1 Post Trade Adjustments

Request

- URL
  - `{api_url}/trades/v1/correctTrades/{exchange_code}`
- Method
  - POST
- Header
  - Content-Type: application/json
  - Accepts: application/json
  - Authorization: Bearer {access_token}
- Body

Field	Required	Data Type	Example	Description
{				
tradeCorrections : [				
{				
tradeId	Yes	number	12345678	Exchange assigned Trade ID. This is a required field, see details under Trade Clearing Update Service
side	Yes	string	“S”	The side of the trade to be corrected: B = Buy S = Sell This is a required field, see details under Trade Clearing Update Service
correctionNumber	Yes	number	1	The last updated correction number for the trade that is to be corrected. This is 0 for new trades. The correction number for any trade is available via Clearing Trade Drop and via the MFP Trade Search Service API.

				This is a required field, see details under Trade Clearing Update Service
tradeDate	Yes	string in ISO date format	“2022-09-29”	The date on which trade occurred. This is a required field, see details under Trade Clearing Update Service
clearingEdits : [				
{				
executingMpid	Conditional	string	“ABC1”	Applicable to securityScope = “U”, “P” MPID which executed the trade
quantity	Conditional	number	100	Applicable to securityScope = “U”, P” The quantity of all corrected parts of a trade must match the quantity of all original parts of the trade.
cmta	Conditional	number	123	Only applicable to securityScope = “P”. Ignored if securityScope = “U” The clearing member trade agreement ID under which the options trade clears.
customerAccount	Conditional	string	“123456789”	Only applicable to securityScope = “P”. Ignored if securityScope = “U” Customer Account ID
multiAccountCode	Conditional	string	“QA5”	Only applicable to securityScope = “P”. Ignored if securityScope = “U” Sub or multi account ID specified in FIX order. For exchangeCode =

				S only – If the trade is from the Sapphire Floor and the multiAccountCode field is populated on a post trade correction that changes the origin to Market Maker, then the multiAccountCode value will be removed.
openCloseIndicator	Conditional	string	“O”	Only applicable to securityScope = “P”. Ignored if securityScope = “U” Open/Clode indicator O = Open C = Close
orderText	Conditional	string	“Order text”	Only applicable to securityScope = “P”. Ignored if securityScope = “U” Order text, optional data submitted on FIX order
origin	Conditional	string	“C”	Only applicable to securityScope = “P”. Ignored if securityScope = “U” For exchangeCode = S, if the trade is from the Sapphire Floor and the origin on a post trade correction is changed to Market Maker, the multiAccountCode value (if populated) will be removed Trades consummated on any of the MIAX electronic exchanges do not support changes to the origin value when the origin of the trade is a Market Maker Origin type of the trade C = Priority Customer N = Non-Priority customer

				F = Firm B = Broker/Dealer M = MIAX member Market Maker A = Away Exchange Market Maker

introducingBadgeId	Conditional	string	“ABC1”	Only applicable to securityScope = “P”. Ignored if securityScope = “U”
				
				This field is applicatble to exchangeCode = S. Ignored if included in the request for exchangeCodes M, P, D.
				
				• For Sapphire Options trades, if a trade correction request attempts to correct this field on a trade that
				  ○ originated on the Floor and has an origin = Market Maker or
				  ○ originated on the Electronic Market
				then this request will be rejected
				
				• For Sapphire Options Floor trades where the origin of the trade is not Marker Marker:
				  ○ Introducing Badge Ids and Executing Mpids for any trade must be associated with the same Member Firm
				
				Introducing Badge Ids are alphanumeric with 4 char length. For a list of Badge Ids associated with a specific Member Firm,

				please contact MIAX Trading Operations
billingTradeType	Conditional	string	“D”	Only applicable to securityScope = “P”. Ignored if securityScope = “U” This field is applicatble to exchangeCode = S. Ignored if included in the request for exchangeCodes M, P, D). Member firm provided one byte optional billing data. Valid values published via Technical Circular. Invalid values are ignored
stockClearingAccount	Conditional	string	“ABCD”	Only applicable to securityScope = “U”. Ignored if securityScope = “P” Allows member firms to change the Stock Clearing Account on the stock leg of an options Trade
}				
{ /*Additional clearing edits for the same tradeId + side + correctionNumber, such as in case of splitting an execution***/ }**				
]				
}				
{ /* Additional executions that require clearing edits included as part of the same request, up to a maximum of 50 ***/ }**				
]				
}				

Response

Field	Format	Value	Description
Standard response fields as per Response Model			
result: { /* Provided if the status is 200 */			
tradeCorrections: [			
{			
tradeId	Number	12345678	Exchange-specific trade ID.
side	String	"B"	The side of the trade: B = Buy S = Sell
submissionStatus	String	"success"	Indicates whether the correction was successfully submitted to the MFP API clearing edit queue. success failure
info	String	"Trade correction successfully submitted"	Please see Appendix A for a list of possible info messages that can be sent for "success" or "failure" status
}			
{ /* Additional tradeIds – a response is provided for every tradeId included in the request ***/ }**			
]			
}			

Examples

See Appendix B: Example: 5.3 Trade Clearing Update Service

5.4 Trade History Service

Member firms may use the Trade History service to retrieve the audit trail of changes made to options trades based on specified search criteria. The response includes the incremental changes (as obsolete corrected versions of the trade) as well as any trade(s) created as a result of trade splitting. The response to the Trade History service is limited to 1000 records.

This service is currently offered only for the MIAX Sapphire Exchange. Please contact MIAX Trading Operations for further information on when this service will be available on MIAX Options, Pearl Options and Emerald Options exchanges.

Request

- URL
  - `{api_url}/trades/v1/history/{exchange_code}?tradeDate=<tradeDate>&tradeId=<tradeId>&side=<tradeSide>`
- Method
  - GET
- Header
  - Content-Type: application/json
  - Accepts: application/json
  - Authorization: Bearer {access_token}
- Body
  - This service does not require any submission paramaters\
- Query Parameters

Field	Required	Data Type	Example	Description
tradeDate	Yes	string in ISO date format	“2020-06-22”	Date of the trade
tradeId	Yes	number	1234567890	Exchange assigned Trade ID
side	Yes	String	“S”	Side of the trade to be corrected: B = Buy S = Sell

Response

Field		Data Type	Example	Description
Standard response fields as per Response Model				
result : [*/* Provided if status is 200 */*				
{				
tradeId		number	1234567890	Exchange assigned Trade ID
tradeDate		string in U.S. date format	"08/19/2022"	Date of the trade
tradeTime		string in U.S. time format	"09:48:29.123"	Time of the trade
transactionId		number	1234567890	All clearing trades generated from a single Engine transaction will have the same transaction ID
transactionTime		string in U.S. time format	"09:48:29.123"	Time of the transaction
tradeAction		string	"C"	State of the trade N – New trade was created as a result of a correction C – Clearing Change X – Trade was Cancelled by MIAX Trading Operations"" (empty string) – Trade was never corrected Corrections to Price or Size of the Trade can only be made by the MIAX Trading Operations Group
tradeVersion		number	1	Version of the trade. The trade version is distinct from the correctionNumber of the trade

correctionNumber		number	1	Count of how many times a trade is corrected. Increments by 1 for each subsequent correction. Cancelling a trade does NOT increment its correctionNumber
executionId		number	123456789012345678	Exchange assigned unique Execution Id
strategyId		number	123456789	Only applicable to exchangeCode M, D, S Exchange assigned unique Complex Strategy Id for the day, Only applicable for complex order executions, otherwise set to zero
tradeAsOfDate		string in U.S. date format	"08/25/22"	As-of Date for As-of Trades
underlyingSymbol		string	"IBM"	Underlying security
securitySymbol		string	"IBM"	Symbol of the security
expirationDate		string in U.S. date format	"08/26/22"	Expiration date of the option
strikePrice		string	"110.00"	Strike price of the option
optionKind		string	"C"	Type of option: C = Call P = Put
quantity		number	100	Quantity
price		string	"4.5000"	Price of the trade with precision of 4 decimal places
side		string	"S"	Side of the trade: B = Buy S = Sell
sideFromFloor		boolean	true	This field is applicatble to exchangeCode = S. Not included in the trade history response for exchangeCodes M, P, D.

				If set to true – indicates this trade side originated on the Sapphire Floor
introducingBadgeId		string	"ABC1"	This field is applicable for floor trades on exchangeCode = S. Sapphire assigned alphanumeric, 4 character Badge ID of the firm's Floor Broker who announced the deal on the floor. (Not included in the trade search response for exchangeCodes M, P, D)
billingTradeType		string	"D"	This field is applicable for floor trades on exchangeCode = S. Member firm assigned optional billing data. (Not included in the trade search response for exchangeCodes M, P, D)
executionExchangeCode		string	"M"	Market where a customer option order was traded, potentially after routing A = NYSE Amex B = BOX C = CBOE D = MIAX Emerald E = EDGX H = ISE Gemini I = ISE J = ISE Mercury M = MIAX Options N = NYSE Arca P = MIAX Pearl Q = NASDAQ OMX S = MIAX Sapphire T = NASDAQ BX U = MEMX W = C2 (from CBOE) X = NASDAQ OMX PHLX Z = BATS
originalTradeId		number	1234567890	Currently available for exchange code = S. This field will not be in the response for exchange codes M, P, D. A future enhancement may add this field to exchange codes M, P, D. In this case,

				a new version of the API spec will be published. For trades that are a result of an execution being split or trades that are restored using the Trade Restore Service, this field contains the tradeId of the original trade that was acted upon. For all other trades, this field will be blank.
productId		number	1234567890	Exchange-specific product ID
quoteId		string	"1234567890"	Quote ID
origin		string	"B"	Origin type of the trade C = Priority Customer N = Non-Priority customer F = Firm B = Broker/Dealer M = MIAX member Market Maker A = Away Exchange Market Maker
exMpid		string	"ABC1"	Executing MPID
clearingNumber		string	"00001"	OCC Clearing Number used to the clear the trade
clMpid		string	"ABC1"	Clearing MPID
cmta		number	123	CMTA in case of a give-up trade
multiAccountCode		string	"QA5"	Sub or multi account ID specified in FIX order
openCloseIndicator		string	"O"	Open/Clode indicator O = Open C = Close
clOrderId		string	"AB123456_123456"	Client order ID

directedFirmCode		string	"ABCD"	Only applicable to exchangeCode M, D Directed firm code for directed orders
customerAccount		string	"123456789"	Customer Account ID
orderText		string	"Order text"	Order text, optional data submitted on FIX order
orderDate		string in U.S. date format	"08/26/2022"	Order date
}				
{ } /* *Multiple trade sides included if the trade has been through multiple corrections. All trades (in the form of trade sides) generated as a result of corrections to the specified trade id will be included in this result. Limited to a max of 1000 trade sides.* */				
]				

Examples

See Appendix B: Example: 5.4 Trade History Service

5.5 Trade Restore Service

Member firms may use the Trade Restore Service to undo post trade corrections made to a trade. Post trade adjustments for clearing changes (including trade splits) can be reversed by Member Firms using this service. The Trade Restore Service cannot act upon a trade that has no post trade corrections and cannot be used to undo a material change performed by MIAX Trading Operations.

Member Firms must coordinate post trade adjustments to Price and/or Quantity (without splitting the Trade) with MIAX Trading Operations as required.

The Trade Restore Service is subject to the following conditions:

- **This service is currently offered for only the MIAX Sapphire Exchange for both Sapphire Electronic and Sapphire Floor Trades**. Please contact MIAX Trading Operations for further information on when this service will be available on MIAX Options, Pearl Options and Emerald Options exchanges.
- **The trade restore service can only be used for trades for the current business date**. For changes to trades from prior business date(s), please contact MIAX Trading Operations.
- Restoring the trade to a prior version will cause all subsequent clearing changes for the trade, its splits (if any) and any clearing adjustments for the respective splits (if any) to be canceled. Successfully restoring the trade to a prior version will be communicated via the member firm's Clearing Trade Drop as a new trade with a new trade id.

See example below:

Order of Operations	Description of Action	Clarification/Note
Step 1	TradeId 1 at $2.25 for quantity 100 is matched on MIAX Sapphire Exchange	TradeId 1, Correction 0 created
Step 2	Member Firm performs a post trade adjustment updating clearing information for TradeId 1	TradeId 1, Correction 1 updated
Step 3	Member Firm uses the Trade Restore Service to restore TradeId 1 to its original state (correction 0)	TradeId 1, Correction 1 is canceled
Step 4	TradeId 2 at $2.25 for quantity 100 is created	TradeId 2, Correction 0 created as a result of the restore request

Request

- URL
  - `{api_url}/trades/v1/restore/{exchange_code}`
- Method
  - POST
- Header
  - Content-Type: application/json
  - Accepts: application/json
  - Authorization: Bearer {access_token}
- Body
  - API Criteria that can be specified in the request body as JSON fields:

Field	Required	Data Type	Example	Description
tradeDate	Yes	string on ISO date format	"2020-06-22"	Date for which to search trades
tradeId	Yes	number	123456789	Trade ID of the trade being searched. Member firms may elect to provide more than one of the above search fields to further narrow the scope of the search results
correctionNumber	Yes	number	2	Correction number of the trade version that the trade must be restored to.

Response

Field	Format	Value	Description
Standard response fields as per Response Model			
result: { /* Provided if the status is 200 */			
tradeId	number	12345678	Exchange-specific trade ID.

correctionNumber	number	12345678	Correction number of the trade version that the trade must be restored to.
submissionStatus	String	“success”	Indicates whether the correction was successfully submitted to the MFP API clearing edit queue. • success • failure
info	String	“Trade restore request successfully submitted”	Please see Appendix A for a list of possible info messages that can be sent for “success” or “failure” status
}			

Examples

See Appendix B: Example: 5.5 Trade State Restore Service

6.0 Market Maker Assignment Services

6.1 Effective Dates for Assignments

For all MIAX Options Exchanges (MIAX Options, Pearl Options, Emerald Options, Sapphire Options)
Generally, the system date changes to next business date approximately 60 minutes after the end of the trading session. Any changes made to MPID Underlying Assignments prior to the system date changing are effective for the next trading session which is generally the next business date at the specified exchange. MPID Underlying Assignments added or updated after the system date has changed to next business date are generally effective two business days later at the specified exchange.

All MIAX Options exhanges provide an API for member firms to verify the current and next business date at the respective exchange.

6.2 Get Assignments for Business Date

The GetAssignmentsForBusinessDate service provides a list of assignments for all Market Maker MPIDs of the Member Firm in all the Underlyings that the MPID/s are assigned to at the specified exchange.

Request

- URL
  - **`{api_url}/assignments/{version}/getAssignmentsForBusinessDate/{exchange_code}`**
- Method
  - POST
- Header
  - Accepts: application/json
  - Authorization: Bearer {access_token}
- Body

Field	Required	Data Type	Example	Description
businessDate	Yes	string in ISO date format	“2020-06-22”	The date for which MPID assignments are requested. Only requests for the next business date and current Business Date are supported.
firmCode	Yes	string	“ABCD”	MIAX specific Firm Code for which the Member Firm is requesting assignments for the next business date
mpid	No	string	“ABC1”	If provided, only assignments for the specified MPID in all assigned Underlyings are returned. Member firms may elect to provide one or more of the following optional fields to further refine the list of assignments returned • underlying (for exchangeCode M,P, D, S) • symbol (for exchangeCode H) • role
underlying	No	string array	“ABC”,”DEF”,”XYZ”	Only applicable for exchangeCode M, P, D, S. If this field is provided in a request for exchangeCode H, it will be ignored If provided, only assignments for the specified Underlying/s in all assigned MPIDs for the member firm are returned Member firms may elect to provide one or more of the following optional fields to further refine the list of assignments returned • mpid • role
symbol	No	string array	“ABC”,”DEF”,”XYZ”	Only applicable for exchangeCode H. If this field is provided in a request for exchangeCode M,P,D, S, it will be ignored If provided, only assignments for the specified Symbol/s in all assigned MPIDs for the member firm are returned

				Member firms may elect to provide one or more of the following optional fields to further refine the list of assignments returned • mpid • role

role	No	string	“LMM”	For exchangeCode M, D:
				“LMM” – Lead Market Maker
				“RMM” – Registered Market Maker
				
				For exchangeCode P, S:
				“RMM” – Registered Market Maker
				
				For exchangeCode H:
				“P” – Primary
				“S” – Supplementary
				
				If provided, only assignments for the specified roles in all Underlyings assigned to MPIDs for the member firm are returned Member firms may elect to provide one or more of the following optional fields to further refine the list of assignments returned
				• mpid
				  ○ underlying (for exchangeCode M,P, D, S)
				• symbol
				  ○ (for exchangeCode H)

Response

Field	Format	Value	Description
Standard response fields as per Response Model			
result: { /* Provided if the status is 200 */			
businessDate	string in ISO date format	“2020-06-22”	Business Date for which MPID assignments are requested.
assignments: [			
{			
underlying	string	“ABCD”	Only applicable for exchangeCode M, P, D, S Underlying with the mpid assignment. If the member firm request included a conditional underlying field in the request, then the response is limited to the underlyings included in the request If this field is provided in a request for exchangeCode H it will be ignored
symbol	string	“ABCD”	Only applicable for exchangeCode H. Symbol with the mpid assignment. If the member firm request included a conditional symbol field in the request, then the response is limited to the symbols included in the request If this field is provided in a request for exchangeCode M,P,D, S, it will be ignored
mpid	string	“ABC1”	MPID with the underlying assignment. If the member firm request included a conditional mpid field in the request, then the response is

			limited to the mpid included in the request
role	string	"LMM"	For exchangeCode M, D: "LMM" – Lead Market Maker "RMM" – Registered Market Maker "PLMM" – Primary Lead Market Maker For exchangeCode P, S: "RMM" – Registered Market Maker For exchangeCode H: "P" – Primary "S" – Supplementary
}			
{ /* Additional assignments for the same Firm Code – All assignments that meet the API request criteria are returned***/ }**			
]			
}			

Examples

See Appendix B: Example: 6.2 Get Assignments for Business Date

6.3 MPID Assignments

The MPID Assignments service provides an API for member firms to add, update or delete assignments for Market Maker MPIDs in specified Underlyings at the specified exchange in respective exchange supported roles. For the date on which the assignment is effective at the respective exchange, see Effective Dates for Assignments. All MPID Assignments included in an Add/Update/Delete Request must have a unique Underlying, MPID and Role combination. If an MPID or Underlying requires more than one assignment change, then each change must be submitted as a unique MPID Assignment request.

For all MIAX Options Exchanges (MIAX Options, Pearl Options, Emerald Options) – PLMM Assignments cannot be added, updated or deleted via the MFP MPID Assignments API. PLMM assignment changes can only be performed by the MIAX Trading Operations group.

The MPID Assignments service supports three operations: Add, Update, Delete. To access an API end point for a specific operation, it is included in the URL of the service call, as follows:

`{api_url}/assignments/{version}/{operation}/{exchange_code}`

The following URLs invoke the Add, Update and Delete operations for the MPID Assignment service for the exchange specified by the exchange code:

Add:
`{api_url}/assignments/{version}/addAssignments/{exchange_code}`

Update:
`{api_url}/assignments/{version}/updateAssignments/{exchange_code}`

Delete:
`{api_url}/assignments/{version}/deleteAssignments/{exchange_code}`

For example, the following URL would invoke the add MPID assignments service for the MIAX Options exchange:

`{api_url}/assignments/v1/addAssignments/M`

Request

- URL
  - `{api_url}/assignments/{version}/{operation}/{exchange_code}`
- Method
  - POST
- Header
  - Content-Type: application/json
  - Accepts: application/json
  - Authorization: Bearer {access_token}
- Body

Field	Required	Data Type	Example	Description
{				
nextBusinessDate	Yes	string in ISO date format	"2020-06-22"	Next business day at the specified exchange. This is the date on which the MPID assignments that are part of this request will be effective for the specified operation at the specified exchange. Changes can only be requested for next business date
firmCode	Yes	string	"ABCD"	MIAX exchange specific Firm Code for which assignments are being added. All Mpids included in the request must belong to this Firm Code. Each request supports one (and only one) Firm Code
mpidAssignments : [				
{				
underlying	Yes	string	"ABC"	Only applicable for exchangeCode M, P, D, S: Underlying specified in the request

mpid	Yes	String	“ABC1”	The MPID to be assigned to the specified Underlying. This MPID must be assigned to the Firm Code in the request
role	Conditional	String	“LMM”	This field is required if the operation is Add or Update. Optional if the operation is Delete For exchangeCode M, D: “LMM” – Lead Market Maker “RMM” – Registered Market Maker For exchangeCode P, S: “RMM” – Registered Market Maker
}				
{ /* Additional MPID assignments that the member firm wishes to change – one assignment per mpid+underlying+role ***/ }**				
]				
}				

Response

Field	**Format**	**Value**	**Description**
Standard response fields as per Response Model			
result: { /* Provided if the status is 200 */			
effectiveDate	string in ISO date format	“2020-06-22”	Date on which the assignments were requested at the specified exchange
firmCode	string	“ABCD”	Firm Code for which assignments are added at the specified exchange.
mpidAssignments: [			

{			
underlying	string	"ABC"	Only applicable for exchangeCode M, P, D, S: Underlying for which assignment is being added/updated/deleted
mpid	string	"ABC1"	MPID which is assigned to the specified underlying/symbol
role	string	"LMM"	Role in which the MPID is assigned to the specified underlying/symbol
status	string	"Next Day Assigned"	This is a conditional field. Only provided if the specified operation (add/update/delete) is successful.
info	string	"Updated Successfully"	This is a conditional field. Only provided if the specified operation (add/update/delete) is successful. Provides feedback on each individual assignment action
}			
{ /* Additional assignment status – a response is provided for every assignment change included in the request ***/ }**			
]			
}			

Examples

See Appendix B: Example: 6.3 MPID Assignments

Appendix A

Response Error Codes and Messages

Status	Error	Sample Message
HTTP Header Code 200		
200		Success
HTTP Header Code 207		
700	Request completed but with errors	Please review info field(s) for errors. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
701	Request received outside accepted time	Please review info field(s) for errors. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
HTTP Header Code 4XX or 5XX		
600	Application Error	There was a problem processing this Request. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
601	Exchange Service Error	There was a problem processing this Request. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
602	Firm Service Error	There was a problem processing this Request. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
603	Trade Correction Service Error	There was a problem processing this Request. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
604	Symbol Assignment Service Error	There was a problem processing this Request. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
605	Risk Protection Service Error	There was a problem processing this Request. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
606 – 649	Reserved for future use	

HTTP Header Code 403		
650	Authorization Error	There was a problem processing this Request. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
651	User Exchange Entitlement Error	Not authorized to access Exchange {Exchange Name}
652	User Application Entitlement Error	Not authorized to access Application {Application}
653	User Firm Entitlement Error	Not authorized to access Firm {Firm Code}
654	User MPID Entitlement Error	Not authorized to access MPID {MPID}
655	User OCC Number Entitlement Error	Not authorized to access OCC number {OCC Number}
656	User Access Privilege Error	Not authorized for edit privileges to application {application}
657 – 674	Reserved for future use	
HTTP Header Code 400		
675	Validation Error	There was a problem processing this Request. Please contact MIAX Trading Operations at 609-897-7302 for assistance.
676	Required Field Missing	Field {field name} is required
677	Invalid Field Value	Field value is invalid for field {field name}
678	Missing Dependent Field	Missing dependent field {field name} in the request
679	Service Not Available	Not available for the selected exchange {exchange code}
680 – 699	Reserved for future use	

Appendix B: Examples

Example: 2.1 Authentication

Sample Request

```
POST /auth/v1/openid-connect/token HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: application/json
Content-Length: 45
 {
 "username": "test",
 "password": "test"
 }
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 417
Connection: Keep-Alive
{
 "result": {
  "access_token":
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890",
  "expires_in": 14400,
  "refresh_expires_in": 43200,
  "refresh_token":
"1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz",
  "token_type": "Bearer",
  "not-before-policy": 0
 },
 "status": 200,
 "timestamp": "2022-08-01T09:30:00.000-04:00",
 "path": "/auth/v1/openid-connect/token"
}
```

Example: 2.2 Refresh

Sample Request

```
POST /auth/v1/openid-connect/refresh HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: application/json
Content-Length: 89
 {
 "refresh_token":
"1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
 }
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 399
Connection: Keep-Alive
{
"result": {
  "access_token":
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890",
  "expires_in": 14400,
  "refresh_expires_in": 43200,
  "refresh_token":
"1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz",
  "token_type": "Bearer",
  "not-before-policy": 0
},
"status": 200,
"timestamp": "2022-08-01T10:00:00.000-04:00",
"path": "/auth/v1/openid-connect/refresh"
}
```

Example: 3.1 Get Current Business Date

Sample Request

```
GET /exchange/v1/getCurrentBusinessDate/M HTTP/1.1
Accept: application/json
Authorization: Bearer
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 195
Connection: Keep-Alive
{
 "result": {
  "currentBusinessDate": "2020-06-22"
 },
 "status": 200,
 "timestamp": "2022-08-01T15:00:00.000-04:00",
 "path": "/exchange/v1/getCurrentBusinessDate/M"
}
```

Example: 3.2 Get Next Business Date

Sample Request

```
GET /exchange/v1/getNextBusinessDate/M HTTP/1.1
Accept: application/json
Authorization: Bearer
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Sample Response

```
 HTTP/1.1 200 OK
content-type: application/json
content-length: 195
Connection: Keep-Alive
{
 "result": {
  "nextBusinessDate": "2020-06-22"
 },
 "status": 200,
 "timestamp": "2022-08-01T15:00:00.000-04:00",
 "path": "/exchange/v1/getNextBusinessDate/M"
}
```

Example: 4.1 Get Firm Codes

Sample Request

```
GET /firms/v1/getFirmCodes/M HTTP/1.1
Accept: application/json
Authorization: Bearer
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 195
Connection: Keep-Alive
{
 "result": {
  "firmCodes":[
   "ABCD",
   "WXYZ"
 ]
 },
 "status": 200,
 "timestamp": "2022-08-01T15:00:00.000-04:00",
 "path": "/firms/v1/getFirmCodes/M"
}
```

Example: 5.1 Is Trade Correction Window Open

Sample Request

```
GET /trades/v1/isTradeCorrectionWindowOpen/M HTTP/1.1
Accept: application/json
Authorization: Bearer
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 195
Connection: Keep-Alive
{
 "result": {
  "correctionWindowOpen": true
 },
 "status": 200,
 "timestamp": "2022-08-01T15:00:00.000-04:00",
 "path": "/trades/v1/isTradeCorrectionWindowOpen/M"
}
```

Example: 5.2 Search Trades Service

Sample Request

```
GET /trades/v1/search/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 123
{
 "tradeDate": "2022-09-22",
 "securitySymbol": "SPY",
 "startTime": "11:30:00",
 "endTime": "15:15:00"
}
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 1835
Connection: Keep-Alive
{
 "result":[
  {
   "tradeId": 12345678,
   "tradeDate": "08/19/20",
   "tradeTime": "14:30:33.000.000000",
   "transactionId": 12345678,
   "transactionTime": "14:30:36.703.480045",
   "tradeAction":"N",
   "tradeVersion":1,
   "correctionNumber":1,
   "executionId":123456789012345678,
   "strategyId":12345678,
   "tradeAsOfDate": "",
   "underlyingSymbol": "SPY",
```

```
  "securitySymbol": "SPY",
  "expirationDate": "01/19/23",
  "strikePrice": "50.00",
  "optionKind":"C",
  "quantity":1,
  "price": "6.16",
  "side":"B",   "executionExchangeCode": "M",
  "productId":12345678,
  "quoteId": "1234567890l2345678",
  "exMpid":"AB12",
  "clearingNumber":000,
  "clMpid":"AB12",
  "cmta":"",
  "multiAccountCode":"",
  "openCloseIndicator":"O",
  "clOrderId":"ABC12",
  "directedFirmCode":"",
  "customerAccount":"",
  "orderText":"",
  "orderDate":"08/19/20"
 },
 {
  "tradeId": 87654321,
  "tradeDate": "08/19/20",
  "tradeTime": "11:30:33.000.000000",
  "transactionId": 87654321,
  "transactionTime": "11:30:36.703.480045",
  "tradeAction": "",
  "tradeVersion": 1,
  "correctionNumber": 0,
  "executionId": 12345678987654321,
  "strategyId": 87654321,
  "tradeAsOfDate": "",
  "underlyingSymbol": "SPY",
  "securitySymbol": "SPY",
  "expirationDate": "01/20/23",
  "strikePrice": "205.00",
  "optionKind": "C",
  "quantity": 10,
  "price": "1.10",
  "side": "B",   "executionExchangeCode": "M",
```

```
    "productId": 12345678,
    "quoteId": "1234567890123456789",
    "exMpid": "AB12",
    "clearingNumber": 000,
    "clMpid": "AB12",
    "cmta": "",
    "multiAccountCode": "",
    "openCloseIndicator": "O",
    "clOrderId": "ABC12",
    "directedFirmCode": "",
    "customerAccount": "",
    "orderText": "",
    "orderDate": "08/19/20"
  }
  ],
 "status": 200,
 "timestamp": "2022-09-22T15:00:00.000-04:00",
 "path": "/trades/v1/search/M"
}
```

Example: 5.3 Trade Clearing Update Service

Example 1 – Single Execution Clearing Change

Sample Request

```
POST /trades/v1/correctTrades/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 311
{
"tradeCorrections": [
  {
   "tradeId":12345678,
   "side": "S",
   "correctionNumber": 1,
   "tradeDate": "2020-08-19",
   "clearingEdits": [
    {
     "executingMpid": "ABC1",
     "cmta": 12345,
     "customerAccount": "123456789",
     "multiAccountCode": "ABC",
     "openCloseIndicator": "O"
    }
   ]
  }
 ]
}
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 288
Connection: Keep-Alive
{
"result": {
 "tradeCorrections": [
   {
   "tradeId": 12345678,
   "side": "S",
   "submissionStatus": "success",
   "info": "Trade correction successfully submitted"
   }
 ]
},
"status": 200,
"timestamp": "2022-08-01T15:00:00.000-04:00",
"path": "/trades/v1/correctTrades/M"
}
```

Example 2 – Spitting an Execution

Sample Request

```
POST /trades/v1/correctTrades/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 429
{
"tradeCorrections": [
  {
   "tradeId": 87654321,
   "side": "B",
   "correctionNumber": 0,
```

```
   "tradeDate": "2020-08-19",
   "clearingEdits": [
    {
     "executingMpid": "XY12",
     "quantity": 8,
     "customerAccount": "ABC1234XYZ",
     "multiAccountCode": "AB1",
     "openCloseIndicator": "C"
    },
    {
     "executingMpid": "AB12",
     "quantity": 2,
     "cmta": 54321,
     "customerAccount": "ABC1234XYZ",
     "orderText": "SplitExecution"
    }
   ]
  }
 ]
}
```


Sample Response


```
HTTP/1.1 200 OK
content-type: application/json
content-length: 287
Connection: Keep-Alive
{
"result": {
 "tradeCorrections": [
  {
   "tradeId": 87654321,
   "side": "B",
   "submissionStatus": "success",
   "info": "Trade correction successfully submitted"
  }
 ]
},
"status": 200,
"timestamp": "2022-08-01T15:00:00.000-04:00",
"path": "/trades/v1/correctTrades/M"
```

```
}
```

Example 3 – Bulk Correction – same field values corrected in multiple, separate executions

Sample Request

```
POST /trades/v1/correctTrades/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 733
{
"tradeCorrections": [
  {
   "tradeId":12233445,
   "side": "B",
   "correctionNumber": 1,
   "tradeDate": "2020-08-19",
   "clearingEdits": [
    {
     "executingMpid": "XY12",
     "customerAccount": "123456789",
     "openCloseIndicator": "C"
    }
   ]
  },
  {
   "tradeId": 12345678,
   "side": "S",
   "correctionNumber": 0,
   "tradeDate": "2020-08-19",
   "clearingEdits": [
    {
     "executingMpid":"XY12",
     "customerAccount":"123456789",
     "openCloseIndicator":"C"
    }
```

```
    ]
   },
   {
    "tradeId":87654321,
    "side": "B",
    "correctionNumber": 2,
    "tradeDate": "2020-08-19",
    "clearingEdits": [
     {
      "executingMpid": "XY12",
      "customerAccount": "123456789",
      "openCloseIndicator": "C"
     }
    ]
   }
  ]
}
```


Sample Response


```
content-type: application/json
content-length: 561
Connection: Keep-Alive
{
"result": {
 "tradeCorrections": [
   {
    "tradeId": 12233445,
    "side": "B",
    "submissionStatus": "success",
    "info": "Trade correction successfully submitted"
   },
   {
    "tradeId": 12345678,
    "side": "S",
    "submissionStatus": "failure",
    "info": "Trade has already been canceled"
   },
   {
    "tradeId": 87654321,
    "side": "B",
```

```
      "submissionStatus": "success",
      "info": "Trade correction successfully submitted"
     }
    ]
  },
  "status": 700,
  "error": "Request completed, but with errors",
  "message": "Please review info field(s) for errors. Please contact
MIAX Trading Operations at 609-897-7302 for assistance",
  "timestamp": "2022-08-01T15:00:00.000-04:00",
  "path": "/trades/v1/correctTrades/M"
}
```

Example 4 – Bulk Correction – different field values corrected in multiple, separate executions

Sample Request

```
POST /trades/v1/correctTrades/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 755
{
"tradeCorrections": [
   {
    "tradeId": 12233445,
    "side": "B",
    "correctionNumber": 1,
    "tradeDate": "2020-08-19",
    "clearingEdits": [
     {
      "executingMpid": "XY12",
      "customerAccount": "123456789",
      "openCloseIndicator": "C"
     }
    ]
   },
```

```
  {
   "tradeId": 12345678,
   "side": "S",
   "correctionNumber": 0,
   "tradeDate": "2020-08-19",
   "clearingEdits": [
    {
     "cmta": 12345,
     "multiAccountCode": "ABC",
     "openCloseIndicator": "O",
     "orderText": "SampleOrderText"
    }
   ]
  },
  {
   "tradeId": 87654321,
   "side": "B",
   "correctionNumber": 2,
   "tradeDate": "2020-08-19",
   "clearingEdits": [
    {
     "executingMpid": "ABC1",
     "cmta": 54321,
     "customerAccount": "123456789",
     "multiAccountCode": "ABC"
    }
   ]
  }
 ]
}
```


Sample Response


```
HTTP/1.1 207 OK
content-type: application/json
content-length: 560
Connection: Keep-Alive
{
"result": {
  "tradeCorrections": [
  {
```

```
    "tradeId": 12233445,
    "side": "B",
    "submissionStatus": "success",
    "info": "Trade correction successfully submitted"
   },
   {
    "tradeId": 12345678,
    "side": "S",
    "submissionStatus": "success",
    "info": "Trade correction successfully submitted"
   },
   {
    "tradeId": 87654321,
    "side": "B",
    "submissionStatus": "failure",
    "info": "This trade has already been canceled"
   }
  ]
},
"status": 700,
"error": "Request completed but with errors",
"message": "Please review info field(s) for errors. Please contact
MIAX Trading Operations at 609-897-7302 for assistance",
"timestamp": "2022-08-01T15:00:00.000-04:00",
"path": "/trades/v1/correctTrades/M"
}
```

Example 5 – Bulk Correction – different field values corrected in multiple, separate executions including a split execution

Sample Request

```
POST /trades/v1/correctTrades/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 1091
{
```

```
"tradeCorrections": [
  {
    "tradeId": 12233445,
    "side": "B",
    "correctionNumber": 1,
    "tradeDate": "2020-08-19",
    "clearingEdits": [
    {
     "executingMpid": "XY12",
     "customerAccount": "123456789",
     "openCloseIndicator": "C"
    }
   ]
  },
  {
    "tradeId": 87654321,
    "side": "B",
    "correctionNumber": 0,
    "tradeDate": "2020-08-19",
    "clearingEdits": [
    {
     "executingMpid": "XY12",
     "quantity": 8,
     "customerAccount": "ABC1234XYZ",
     "multiAccountCode": "AB1",
     "openCloseIndicator": "C"
    },
    {
     "executingMpid": "AB12",
     "quantity": 2,
     "cmta": 12345,
     "customerAccount": "ABC1234XYZ",
     "orderText": "SplitExecution"
    }
   ]
  },
  {
    "tradeId": 88776543,
    "side": "B",
    "correctionNumber": 2,
    "tradeDate": "2020-08-19",
```

```
    "clearingEdits": [
     {
      "executingMpid": "ABC1",
      "cmta": 54321,
      "customerAccount": "123456789",
      "multiAccountCode": "ABC"
     }
    ]
   },
   {
    "tradeId": 12334455,
    "side": "S",
    "correctionNumber": 0,
    "tradeDate": "2020-08-19",
    "clearingEdits": [
     {
      "customerAccount": "987654321",
      "openCloseIndicator": "C"
     }
    ]
   }
  ]
}
```

Sample Response

```
HTTP/1.1 207 OK
content-type: application/json
content-length: 727
Connection: Keep-Alive
{
"result": {
 "tradeCorrections": [
  {
   "tradeId": 12233445,
   "side": "B",
   "submissionStatus": "success",
   "info": "Trade correction successfully submitted"
  },
  {
   "tradeId": 87654321,
```

```
    "side": "B",
    "submissionStatus": "failure",
    "info": "Total split trade quantity does not equal original trade
quantity"
  },
  {
    "tradeId": 88776543,
    "side": "B",
    "submissionStatus": "success",
    "info": "Trade correction successfully submitted"
  },
  {
    "tradeId": 12334455,
    "side": "S",
    "submissionStatus": "success",
    "info": "Trade correction successfully submitted"
  }
 ]
},
      "status": 700,
      "error": "Request completed but with errors",
      "message": "Please review info field(s) for errors. Please
contact MIAX Trading Operations at 609-897-7302 for assistance",
      "timestamp": "2022-08-01T15:00:00.000-04:00",
      "path": "/trades/v1/correctTrades/M"
}
```

Example 6 – Sapphire Floor Execution Clearing Change

Sample Request

```
POST /trades/v1/correctTrades/S HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 311
{
"tradeCorrections": [
```

```
  {
   "tradeId":12345678,
   "side": "S",
   "correctionNumber": 1,
   "tradeDate": "2020-08-19",
   "clearingEdits": [
    {
     "executingMpid": "ABC1",
     "billingTradeType": “D”,
     "introducingBadgeId": "XE12",
     "multiAccountCode": "ABC",
     "openCloseIndicator": "O"
    }
   ]
  }
 ]
}
```


Sample Response


```
HTTP/1.1 200 OK
content-type: application/json
content-length: 288
Connection: Keep-Alive
{
"result": {
 "tradeCorrections": [
   {
   "tradeId": 12345678,
   "side": "S",
   "submissionStatus": "success",
   "info": "Trade correction successfully submitted"
   }
 ]
},
"status": 200,
"timestamp": "2022-08-19T15:00:00.000-04:00",
"path": "/trades/v1/correctTrades/S"
}
```

Example: 5.4 Trade History Service

Sample Request

```
GET /trades/v1/history/S?tradeDate=2022-09-22&tradeId=12345678&side=B
HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 123
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 1835
Connection: Keep-Alive
{
 "result":[
  {
   "tradeId": 12345678,
   "tradeDate": "09/22/22",
   "tradeTime": "14:30:33.000.000000",
   "transactionId": 12345678,
   "transactionTime": "14:30:36.703.480045",
   "tradeAction":"X",
   "tradeVersion":3,
   "correctionNumber":1,
   "executionId":1234567890123456789,
   "strategyId":12345678,
   "tradeAsOfDate": "",
   "underlyingSymbol": "SPY",
   "securitySymbol": "SPY",
   "expirationDate": "01/19/23",
   "strikePrice": "50.00",
   "optionKind":"C",
   "quantity":1,
```

```
  "price": "6.16",
  "side":"B",
  "sideFromFloor":"N",
       "executionExchangeCode": "S",
  "productId":12345678,
  "quoteId": "123456789012345678",
  "exMpid":"AB12",
  "clearingNumber":000,
  "clMpid":"AB12",
  "cmta":"",
  "multiAccountCode":"",
  "openCloseIndicator":"O",
  "clOrderId":"ABC12",
  "directedFirmCode":"",
  "customerAccount":"",
  "orderText":"",
  "orderDate":"09/22/22"
 },
 {
  "tradeId": 12345678,
  "tradeDate": "09/22/22",
  "tradeTime": "14:30:33.000.000000",
  "transactionId": 12345678,
  "transactionTime": "14:30:36.703.480045",
  "tradeAction":"C",
  "tradeVersion":2,
  "correctionNumber":1,
  "executionId":123456789012345678,
  "strategyId":12345678,
  "tradeAsOfDate": "",
  "underlyingSymbol": "SPY",
  "securitySymbol": "SPY",
  "expirationDate": "01/19/23",
  "strikePrice": "50.00",
  "optionKind":"C",
  "quantity":1,
  "price": "6.16",
  "side":"B",
  "sideFromFloor":"N",
       "executionExchangeCode": "S",
  "productId":12345678,
```

```
 "quoteId": "12345678901234 5678",
 "exMpid":"AB12",
 "clearingNumber":000,
 "clMpid":"AB12",
 "cmta":"",
 "multiAccountCode":"",
 "openCloseIndicator":"O",
 "clOrderId":"ABC12",
 "directedFirmCode":"",
 "customerAccount":"",
 "orderText":"",
 "orderDate":"09/22/22"
},
{
 "tradeId": 12345678,
 "tradeDate": "09/22/22",
 "tradeTime": "14:30:33.000.000000",
 "transactionId": 12345678,
 "transactionTime": "14:30:36.703.480045",
 "tradeAction":"N",
 "tradeVersion":1,
 "correctionNumber":0,
 "executionId":1234567890123456 78,
 "strategyId":12345678,
 "tradeAsOfDate": "",
 "underlyingSymbol": "SPY",
 "securitySymbol": "SPY",
 "expirationDate": "01/19/23",
 "strikePrice": "50.00",
 "optionKind":"C",
 "quantity":1,
 "price": "6.16",
 "side":"B",
 "sideFromFloor":"N",
     "executionExchangeCode": "S",
 "productId":12345678,
 "quoteId": "12345678901234 5678",
 "exMpid":"AB12",
 "clearingNumber":000,
 "clMpid":"AB12",
 "cmta":"",
```

```
   "multiAccountCode":"",
   "openCloseIndicator":"O",
   "clOrderId":"ABC12",
   "directedFirmCode":"",
   "customerAccount":"",
   "orderText":"",
   "orderDate":"09/22/22"
  }
  ],

 "status": 200,
 "timestamp": "2022-09-22T15:00:00.000-04:00",
 "path": "/trades/v1/history/S"
}
```

Example: 5.5 Trade Restore Service

Sample Request

```
POST /trades/v1/restore/S HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 311
{
 "tradeId":12345678,
 "correctionNumber": 2,
 "tradeDate": "2022-08-19"
}
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 288
Connection: Keep-Alive
{
"result": {
   "tradeId": 12345678,
   "correctionNumber": 2,
   "submissionStatus": "success",
   "info": "Trade restore request successfully submitted"
},
"status": 200,
"timestamp": "2022-08-19T15:00:00.000-04:00",
"path": "/trades/v1/restore/S"
}
```

Example: 6.2 Get Assignments for Business Date

Sample Request

```
GET /assignments/v1/getAssignmentsForBusinessDate/M HTTP/1.1
Accept: application/json
Authorization: Bearer
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
{
"businessDate": "2020-06-22",
"firmCode": "ABCD",
"mpid": "ABC1",
"underlying": [
  "ABCD",
  "WXYZ",
  "PQRS"
 ],
"role": "LMM"
}
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 360
Connection: Keep-Alive
{
"result": {
"businessDate": "2022-06-22",
 "assignments": [
  {
   "underlying": "ABCD",
   "mpid": "ABC1",
   "role": "LMM"
  },
  {
   "underlying": "WXYZ",
   "mpid": "ABC1",
```

```
      "role": "LMM"
    }
  ]
},
"status": 200,
"timestamp": "2022-08-01T15:00:00.000-04:00",
"path": "/assignments/v1/getAssignmentsForBusinessDate/M"
}
```

Example: 6.3 MPID Assignments

Example 1: Adding an assignment – MIAX Options

Sample Request

```
POST /assignments/v1/addAssignments/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 311
{
"nextBusinessDate": "2020-06-22",
"firmCode": "ABCD",
"mpidAssignments": [
  {
   "underlying": "ABC",
   "mpid": "ABC1",
   "role": "LMM"
  },
  {
   "underlying": "XYZ",
   "mpid": "ABC1",
   "role": "RMM"
  },
  {
   "underlying": "PQR",
   "mpid": "DEF2",
   "role": "LMM"
  }
 ]
}
```

Sample Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 429
Connection: Keep-Alive
{
"result": {
 "effectiveDate": "2020-06-22",
 "firmCode": "ABCD",
 "mpidAssignments": [
  {
   "underlying": "ABC",
   "mpid": "ABC1",
   "role": "LMM",
   "status": "Next Day Assigned",
   "info": "Updated successfully"
  },
  {
   "underlying": "PQR",
   "mpid": "DEF2",
   "role": "LMM",
   "status": "Next Day Assigned",
   "info": "Updated successfully"
  },
 ]
},
"status": 200,
"timestamp": "2022-08-01T15:00:00.000-04:00",
"path": "/assignments/v1/addAssignments/M"
}
```

Example 2: Adding an Assignment – MIAX Options, request has an error

Sample Request

```
POST /assignments/v1/addAssignments/M HTTP/1.1
Authorization: Bearer
12345678ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 311
{
"nextBusinessDate": "2020-06-21",
"firmCode": "ABCD",
"mpidAssignments": [
  {
   "underlying": "ABC",
   "mpid": "ABC1",
   "role": "LMM"
  },
  {
   "underlying": "XYZ",
   "mpid": "ABC1",
   "role": "RMM"
  }
 ]
}
```

Sample Response

```
{
 "error": "Invalid Field Value",
 "message": "Invalid field businessDate value in the request. Expected
2022-06-22, but got 2022-06-21.",
 "path": "/assignments/v1/addAssignments/M",
 "status": 677,
 "timestamp": "2022-08-01T15:00:00.000-04:00",
}
```

Appendix C: Revision History

Revision Date	Version	Description
12/11/2024	2025.05-0	Made API Spec Assent Class specific – Options only

12/30/2024	2025.05-0.V4	• Divided Options and Equities API Specs into separate documents
		• Renamed Search Trades – Options service to Trade Search Service
		• Changes to exisiting requests and responses
		  ○ Removed fields, On Request, From Service : Trade Search
		    ▪ currentTradesOnly, correctedOnly
		  ○ Added fields, On Request, From Service : Trade Search
		    ▪ fromFloor
		  ○ Added fields, On Response, To Service : Trade Search
		    ▪ originalTradeId, sideFromFloor
		• Added ability to fetch trade history using Trade History Service
		• Added ability to undo trade corrections using Trade Restore Service
		• Organized examples into appendix

03/14/2025	2025.05-0.V5	• Minor cleanup and support for sapphire floor trade corrections • Added side query parameter to trade history • Added error code 679 - Service Not Available

4/14/2025	2025.05-0.V6	• Added the following parameters to Clearing Update Service request:
		  ○ Billing Trade Type
		  ○ Introducing Badge Id

5/1/2025	2025-05-0.OPTIONS-.RELEASE	• Created Release API Spec for Sapphire Floor Options Support
9/19/2025	2025-09-0. OPTIONS.RELEASE	• Added additional support for Sapphire Floor post trade adjustments to handle business rules for multiAccountCode and trade origin • Added support to search trades by transaction Id
10/27/2025	2025.10-0. OPTIONS.RELEASE	• T+1 Trade corrections for Stock Trades no longer supported • T+1 Trade Corrections for Options Trades are limited to one correction per trade

miax®

miaxglobal.com