Alaric Trading API Documentation

Table of Contents

1. Overview and Specifications

This document describes Alaric Trading WebSocket API which allows clients to submit orders and access other system functions using custom-written software.

1.1 WebSocket API

The WebSocket API is a technology that allows bidirectional interactive communication between the clients and the server. With this API, you can send messages to the Alaric Trading server and receive event-driven responses without having to poll the server for a reply or for an update.

1.2 WebSocket URL and Connection

The connection should use secure SSL(TLS 1.2 or higher) WebSocket and API should allow the clients who provide a valid "Access Token".

- Production URL - **`wss://tapi.alaricsecurities.net/trading`**
- UAT URL - **`wss://tapi.alaricsecurities.net/trading-uat`**

1.3 Message Encoding

All messages are encoded in JSON format. Prices and amounts are presented as strings. All text encoding must use ASCII encoding.

1.4 Date & Time

All dates and times should be as UTC in ISO 8601 Format as following:

```
YYYY-MM-DDThh:mm:ss.fff
```

2. Authentication

All Alaric Trading WebSocket API calls require authentication. A valid login request and access token must be sent with a login request before any messages can be accepted for processing.

2.1 Auth Service

Alaric Authentication and Authorization Service (https://auth-dev.alaricsecurities.net) which implements OpenId Connect (https://openid.net/connect/) and OAuth2 (https://oauth.net/2/) standards.

The clients are responsible to authenticate users on their side by requesting an access token from the Alaric Auth service. The incoming token must be validated by checking the;

- Issuer
- Audience
- Signing Key
- Other relevant fields, optionally.

The signing key can be verified using the public key hash, or the X5C field of the key, which can be fetched from the ***<auth-server>/.well-known/openid-configuration/jwks*** end-point.To reduce the round-trip count, this configuration can be cached periodically for offline checking.

Note: *We have different auth servers for different environments general URL format is following: https://auth[-{ENVIRONMENT}].alaricsecurities.net Environment could be dev, test, uat or null (for production). When you need to access trading api's dev environment you must get a token from dev auth server.*

2.2 Access Token

Base64 encoded (Bearer Token) string of JWT (https://jwt.io/) signed with a certificate (public/private) which can be validated using its public key (***x5c*** field of https://auth-dev.alaricsecurities.net/.well-known/openid-configuration/jwks).

The most simple and fast approach is validating tokens offline without making an additional request to Auth Service, sample code will be provided for implementation details.

Online verification can be implemented by following OAuth2 standards:

Sample

Base64	eyJhbGciOiJSUzI1NiIsImtpZCI6IjAwMDc5NUNFQjZERDlFMUQ2Q z FFMzkxNDAyRDY0QTM4QzhERkNEMEUiLCJ0eXAiOiJhdCtqd3QiLCJ4 NXQiOiJBQWVWenJiZG5oMXNIamtVQXRaS09NamZ6UTQifQ.eyJuYmY iOjE1NzYxNDk1NzYsImV4cCI6MTU3NjE1MzE3NiwiaXNzIjoiaHR0c HM6Ly9hdXRoLWRldi5hbGFyaWNzZWN1cml0aWVzLm5ldCIsImF1ZCI 6WyJEZW1vUHJvdGVjdGVkQVBJIiwiVHJhZGVSZXBvcnRpbmdBUEkiX SwiY2xpZW50X2lkIjoiRGVtb0NsaWVudCIsInN1YiI6IjEiLCJhdXR oX3RpbWUiOjE1NzYxNDk1NzYsImlkcCI6ImxvY2FsIiwibmFtZSI6I iAiLCJlbWFpbCI6ImRlbW91c2VyQGFsYXJpY3NlY3VyaXRpZXMuY29 tIiwicm9sZSI6WyJBZG1pbiIsInVzZXIiXSwicGxhdGZvcm0iOiIxM E4iLCJwZXJtaXNzaW9uIjoiY29tcGxpYW5jZToqOioiLCJlbWFpbF9 2ZXJpZmllZCI6ZmFsc2UsInNjb3BlIjpbImVtYWlsIiwib3BlbmlkI iwicGxhdGZvcm0iLCJwcm9maWxlIiwicm9sZXMiLCJEZW1vUHJvdGV jdGVkQVBJIiwiVHJhZGVSZXBvcnRpbmdBUEkiLCJvZmZsaW5lX2FjY 2VzcyJdLCJhbXIiOlsicHdkIl19.q0p74hzFsKF_5peLp2rZrYiEuF 8dz82ZQ8VvklK19rJKrzPtyBEmsYwPDOFYMp9W5AIzRQ_prll7xhhC ubGv19cxgK72DatBn-aPdsnmSjL1xslSiUy5hlrFlxq8OcIL9Kr0A4 faLyp7bmOMwMzPm8RIxuUUUW8NZFjC2g7__V21mRzUqLSbSSuBFKAP SusuNGLAtEsn9_ZS7TlGe7H1o5ogtD9fGrrOpGu2R0c03OYnLrGqFi P1Fx-rxJxvPnw0j0WCrUHY2cXmDlgZq0QzB9m875anfbVx970GbHjw oOXwMITDP1MInBdOgbEWAwPd3ImNIsgFLiv5UWUVU1PjFA

Header	{
	"alg": "RS256",
	"kid":
	"000795CEB6DD9E1D6C1E391402D64A38C8DFCD0E",
	"typ": "at+jwt",
	"x5t": "AAeVzrbdnh1sHjkUAtZKOMjfzQ4"
	}

Payload	{ "nbf": 1576149576, "exp": 1576153176, "iss": "https://auth- dev.alaricsecurities.net", "aud": ["DemoProtectedAPI", "TradeReportingAPI"], "client_id": "DemoClient", "sub": "1", "auth_time": 1576149576, "idp": "local", "name": " ", "email": "demouser@alaricsecurities.com", "role": ["Admin", "user"], "platform": "10N",

	"permission": "compliance:*:*", "email_verified": false, "scope": ["email", "openid", "platform", "profile", "roles", "DemoProtectedAPI", "TradeReportingAPI", "offline_access"], "amr": ["pwd"] }
Signature	q0p74hzFsKF_5peLp2rZrYiEuF8dz82ZQ8VvklK19rJKrzPtyBEmsY wPDOFYMp9W5AIzRQ_prll7xhhCubGv19cxgK72DatBn-aPdsnmSjL1xslSiUy5hlrFlxq8OcIL9Kr0A4faLyp7bmOMwMzPm8RIxuUUUW8NZFjC2g7__V21mRzUqLSbSSuBFKAPSusuNGLAtEsn9_ZS7TlGe7H1o5ogtD9fGrrOpGu2R0c03OYnLrGqFiP1Fx-rxJxvPnw0j0WCrUHY2cXmDlgZq0QzB9m875anfbVx970GbHjwoOXwMITDP1MInBdOgbEWAwPd3ImNIsgFLiv5UWUVU1PjFA

For Payload field descriptions please refer to https://tools.ietf.org/html/rfc7519#section-4.1

Please check language-specific libraries from https://jwt.io/

Sample C# .Net Core Application:
https://bitbucket.org/alaricdev/alaric-auth-public/src/master/

3. Order Entry

3.1 New Order Request Parameters

Note: *Required column values - Yes (Y), No (N), Conditional (C).*

Parameter Name	Type	Req'd	Description
messageType	string	Y	"neworder"
reqId	string	Y	Request ID
account	string	Y	Trading account
clearingAccount	string	C	May be required for some deployments
clOrdId	string	Y	Client order ID - required to be unique per client
symbol	string	Y	Security ticker (NASDAQ symbology for US equities, e.g. BRK.B or BAC-A)
securityId	string	N	Alternative security identifier
securityIdSource	string	C	Required if securityId is specified.
side	string	Y	Order side. Valid values: "BUY" "1" — Buy order "SELL" "2" — Sell order "SELL_SHORT" "SELSHORT" "5" — Sell short orders (equities only)
orderQty	string	Y	Order quantity, formatted as a string. Expected to be an integer for all asset classes except mutual funds and cryptocurrencies.

ordType	string	Y	Order type. Valid values: "MARKET" "1" — Market order "LIMIT" "2" — Limit order (price is required) "STOP" "3" — Stop order (stop price is required) "STOP LIMIT" "STOP_LIMIT" "4" — Stop limit order (stop price and limit price are required) "PEGGED" "P" — Pegged order
price	string	C	Order limit price (required for limit, stop limit and pegged orders)
stopPrice	string	C	Required for stop and stop limit orders
orderCapacity	string	N	Order capacity. Valid values: "A" Agency "P" Principal "R" Riskless principal If not specified, defaults to Agency.
exDestination	string	Y	Order destination

subDestination	string	N	Order sub-destination, if available (for example: exDestination=INCA, subDestination=BATS)
timeInForce	string	N	Order time in force. Valid values: "DAY" "0" — Day order (default) "AT_THE_OPENING" "AT THE OPENING" "ON OPEN" "ON_OPEN" "2" — On Open "IOC" "IMMEDIATE OR CANCEL" "3" — Immediate or cancel "EXTENDED DAY" "EXTENDED_DAY" "5" — Extended day "AT_THE_CLOSE" "AT THE CLOSE" "ON CLOSE" "ON_CLOSE" — On Close If not specified, defaults to Day.

execInst	string	N	Supported values: "R" Primary Peg (for Pegged orders, pegs Buy to NBBO Bid, Sell to NBBO Offer) "P" Market Peg (for Pegged orders, pegs Buy to NBBO Offer, Sell to NBBO Bid) "M" Midpoint Peg (for Pegged orders, pegs to NBBO Midpoint) "F" ISO (Intermarket Sweep Order)
minQty	string	N	Minimum quantity of an order to be executed
maxFloor	string	N	The disclosed order quantity for hidden and reserve orders
pegDifference	string	N	Signed amount added to the price of the peg for a pegged order.
discretionOffset	string	N	Discretion amount – implicitly added to bid prices and subtracted from offer prices.

locateReqd	string	N	Sell Short orders only. Defaults to Y (Alaric Trading API will perform the ETB/HTB check.) If set to N, locateBroker=MPID is required.
locateBroker	string	C	MPID of the broker which loaned the stock.

clientInfo	string	N	Any free-form string containing client-specific information associated with the order. Not used by Alaric Trading API, but will be echoed back in Execution Reports if specified. Maximum length is 64 characters.
customFix	array	N	Custom FIX tags in the format [{"tag": 123, "value": "ABC"}, {"tag": 456, "value": "XYZ"}]

3.1.1 New Order Request Example

```
{
 "messageType": "neworder",
 "reqId": "02",
 "account": "APITEST",
 "clearingAccount": "APIGROUP",
 "clOrdId": "0001",
 "symbol": "AAPL",
 "side": "BUY",
 "orderQty": "100",
 "ordType": "LIMIT",
 "price": "10.00",
 "exDestination": "INET",
 "timeInForce": "DAY"
}
```

3.1.2 New Order Response Example (Success)

```
{
 "messageType": "neworder",
 "reqId": "02",
 "rc": "202",
 "status": "submitted",
 "account": "APITEST",
 "clearingAccount": "APIGROUP",
 "clOrdId": "0001",
 "symbol": "AAPL",
 "side": "BUY",
 "orderQty": "100",
 "ordType": "LIMIT",
 "price": "10.00",
 "orderCapacity": "A",
 "exDestination": "INET",
 "timeInForce": "DAY"
}

{
 "messageType": "neworder",
 "reqId": "02",
 "rc": "200",
 "status": "accepted",
 "account": "APITEST",
 "clearingAccount": "APIGROUP",
 "clOrdId": "0001",
 "symbol": "AAPL",
 "side": "BUY",
 "orderQty": "100",
 "ordType": "LIMIT",
 "price": "10.00",
 "orderCapacity": "A",
 "exDestination": "INET"
}
```

3.1.3 New Order Response Example (Error)

```
{
 "messageType": "neworder",
 "reqId": "02",
 "rc": "400",
 "error": "INET:Test stocks only",
 "status": "rejected",
 "account": "APITEST",
 "clearingAccount": "APIGROUP",
 "clOrdId": "0001",
 "symbol": "AAPL",
 "side": "BUY",
 "orderQty": "100",
 "ordType": "LIMIT",
 "price": "210.00",
 "exDestination": "INET",
 "timeInForce": "DAY",
}
```

3.2 Cancel Order Request Parameters

Parameter Name	**Type**	**Req'd**	**Description**
messageType	string	Y	"cancelorder"
reqId	string	Y	Request ID
account	string	Y	Trading account
clOrdId	string	Y	Client order ID of the order to be cancelled.

3.2.1 Cancel Order Request Example


```
{
 "messageType": "cancelorder",
 "reqId": "03",
 "account": "APITEST",
 "clOrdId": "0002"
}
```


3.2.2 Cancel Order Response Example (Success)


```
{
 "messageType": "cancelorder",
 "reqId": "03",
 "rc": "202",
 "status": "submitted",
 "account": "APITEST",
 "clOrdId": "0002",
}

{
 "messageType": "cancelorder",
 "reqId": "03",
 "rc": "200",
 "status": "cancelled",
 "account": "APITEST",
 "clOrdId": "0002",
}
```

3.2.3 Cancel Response Example (Error)


```
{
 "messageType": "cancelorder",
 "reqId": "03",
 "rc": "200",
 "error": "INET: Order done",
 "status": "rejected",
 "account": "APITEST",
 "clOrdId": "0002",
}
```


3.3 Modify Order Request Parameters

Parameter Name	Type	Req'd	Description
messageType	string	Y	"modifyorder"
reqId	string	Y	Request ID
account	string	Y	Trading account
clOrdId	string	Y	Client order ID of the modify request - required to be day-unique per client
orderQty	string	N	New order quantity, formatted as a string. Expected to be an integer for all asset classes except mutual funds and cryptocurrencies.
price	string	N	New order limit price (for limit, stop limit and pegged orders)

3.3.1 Modify Order Request Example


```
{
 "messageType": "modifyorder",
 "reqId": "04",
 "account": "APITEST",
 "clOrdId": "0003",
 "orderQty": "200",
 "price": "9.6"
}
```


3.3.2 Modify Order Response Example (Success)

```
{
 "messageType": "modifyorder",
 "reqId": "04",
 "rc": "202",
 "status": "submitted",
 "account": "APITEST",
 "clOrdId": "0003",
 "orderQty": "200",
 "price": "9.6"
}

{
 "messageType": "modifyorder",
 "reqId": "04",
 "rc": "202",
 "status": "modified",
 "account": "APITEST",
 "clOrdId": "0003",
 "orderQty": "200",
 "price": "9.6"
}
```

3.3.3 Response Example (Error)

```
{
 "messageType": "modifyorder",
 "reqId": "04",
 "rc": "200",
 "error": "INET: Order done",
 "status": "rejected",
 "account": "APITEST",
 "clOrdId": "0003",
 "orderQty": "200",
 "price": "9.6",
}
```

4. Get Positions Request Parameters

Requests all positions for an account.

Parameter Name	Type	Req'd	Description
messageType	string	Y	"getpositions"
reqId	string	Y	Request ID
account	string	Y	Trading account

4.1 Get Positions Request Example

```
{
 "reqId": "05",
 "messageType": "getpositions",
 "account": "APITEST"
}
```

4.2 Positions Response or Update - server to client

Returns positions for an account, in response to a GETPOSITIONS request. Can also be sent as an update any time a position changes as a result of order execution. If so, no **reqId** is required and only the changed positions can be sent.

Note: if **qty** is negative, that denotes a short position

```
{
 "reqId": "05",
 "messageType": "positions",
 "account": "APITEST",
 "positions":
 [
  {
 "symbol": "AAPL",
 "qty": 150,
```

```
 "averagePrice": 105.2312
 },
 {
 "symbol": "MSFT",
 "qty": -120,
 "averagePrice": 125.123
 },
 {
 "symbol": "FB",
 "qty": 50,
 "averagePrice": 195.76
 }
]
}
```


5. Get Balances Request Parameters

Requests all balances for an account.

Parameter Name	Type	Req'd	Description
messageType	string	Y	"getbalances"
reqId	string	Y	Request ID
account	string	Y	Trading account

5.1 Get Balances Request Example - client to server


```
{
 "reqId": "06",
 "messageType": "getbalances",
 "account": "APITEST"
}
```


5.2 Balances Response or Update - server to client

Returns all balances for an account, in response to a GETBALANCES request. Can also be sent as an update regularly in order to reflect change in balances. If so, no reqId is required.


```
{
 "reqId": "06",
 "messageType": "balances",
 "account": "APITEST",
 "orderExp": "1000000",
 "posExp": "250000",
 "combinedExp": "750000",
 "cash": "40000",
 "long": "125000",
 "short": "125000"
}
```


6. Get Orders Request Parameters

Requests all orders outstanding for an account.

Parameter Name	Type	Req'd	Description
messageType	string	Y	"getorders"
reqId	string	Y	Request ID
account	string	Y	Trading account

6.1 Get Orders Request Example - client to server


```
{
 "reqId": "07",
```

```
 "messageType": "getorders",
 "account": "APITEST"
}
```

6.2 Orders Response or Update - server to client

Returns orders for an account, in response to a GETORDERS request. Can also be sent as an update any time an order’s status changes, or a new order is posted, or a transaction is executed etc. If so, no **reqId** is required and only some orders can be sent.

```
{
 "reqId": "07",
 "messageType": "getorders",
 "account": "APITEST",
 "orders": "[
      {
      "orderID": "18276322-aa-12",
      "clOrdId": "12345",
      "symbol": "AAPL",
      "side": "buy",
      "orderQty": 150,
      "ordType": "limit",
      "price": 105.23,
      "timeInForce": "DAY",
      "status": "Pending New",
              "createdAt": "2019-01-16T14:18:47.321Z",
      "updatedAt": "2019-01-19T15:23:54.876Z"  …
      },
      {
      "orderID": "18276322-aa-14",
      "clOrdId": "45678",
      "symbol": "FB",
      "side": "buy",
      "orderQty": 150,
      "ordType": "limit",
      "price": 10005.23,
      "timeInForce": "DAY",
      "status": "Rejected",
      "rejectReason": "Price too far out of the
range",
              "createdAt": "2019-01-16T14:18:47.321Z",
           "updatedAt": "2019-01-19T15:23:54.876Z"
```

```
        …
        },
        {
        "orderID": "18276322-aa-24",
        "clOrdId": "xyz234",
        "symbol": "MSFT",
        "side": "sellshort",
        "orderQty": 25,
        "ordType": "stop",
        "stopPrice": 127.15,
        "timeInForce": "Day",
        "status": "PartialFill",
                "createdAt": "2019-01-16T16:12:14.123Z",
        "updatedAt": "2019-01-19T17:12:23.423Z"  …
             "filledSoFar": 12,
             "avgPriceSoFar": 127.14,

        "fills":
        [
        {
             "execId": "1879236772",
             "fillQty": 12,
             "fillPrice": 127.14,
             "timeStamp": "2019-01-19T17:12:23.423Z" }
        ]
        }
       ]
  "
  }
```

7. Peek Locate Request Parameters

User sends a request to find out if there are shares available to short, and how many, for a particular symbol for this user (that is for the user's clearing house). If there are no restrictions – the stock is easy to borrow – then -1 is sent.

Parameter Name	Type	Req'd	Description
messageType	string	Y	"peeklocate"
reqId	string	Y	Request ID
account	string	Y	Trading account
symbol	string	Y	Ticker symbol

7.1 Peek Locate Request Example - client to server


```
{
 "reqId": "08",
 "messageType": "peeklocate",
 "account": "APITEST",
 "symbol": "TLRY"
}
```


7.2 Peek Locate Response - server to client

Returns the amount of shares available for borrowing.


```
{
 "reqId": "08",
 "messageType": "peeklocate",
 "account": "APITEST",
 "symbol": "TLRY",
 "available": "8500"
}
```


8. Reserve Request Parameters

Sends a request to reserve a block of shares for the user to short. Once reserved, those shares stay reserved until the user either executes a short on the whole block or a portion of it (in which case the reserved block is reduced by that amount) or un-reserves the whole block or a portion of it using the UNRESERVE request.

Parameter Name	Type	Req'd	Description
messageType	string	Y	"reserve"
reqId	string	Y	Request ID
account	string	Y	Trading account
symbol	string	Y	Ticker symbol
qty	numeric	Y	Amount to be reserved

8.1 Reserve Request Example - client to server


```
{
 "reqId": "09",
 "messageType": "reserve",
 "account": "APITEST",
 "symbol": "TLRY",
 "qty": "1000"
}
```


8.2 Reserve Response - success


```
{
 "reqId": "09",
 "messageType": "reserve",
 "account": "APITEST",
```

```
 "symbol": "TLRY",
 "qty": "1000",
 "result": "success"
}
```

8.3 Reserve Response - fail

The failure may happen because the user requested more shares than are available for the stock. Even in combination with the **PEEKLOCATE**, the **RESERVE** request may fail if between the two calls another user reserved some shares, reducing the available amount.

```
{
 "reqId": "09",
 "messageType": "reserve",
 "account": "APITEST",
 "symbol": "TLRY",
 "qty": "1000",
 "result": "fail",
 "available": "800"
}
```

The "available" returned field is the same as in the **PEEKLOCATE** response, and shows how many shares are currently available for borrowing for the stock.

9. Peek Reserve Request Parameters

Sends a request to find out how many shares the user has currently reserved for a particular symbol for this user (that is for the user's clearing house).

Parameter Name	Type	Req'd	Description
messageType	string	Y	"reserve"
reqId	string	Y	Request ID
account	string	Y	Trading account

symbol	string	Y	Ticker symbol

9.1 Peek Reserve Request Example - client to server


```
{
 "reqId": "10",
 "messageType": "peekreserve",
 "account": "APITEST",
 "symbol": "TLRY"
}
```


9.2 Peek Reserve Response - server to client


```
{
 "reqId": "10",
 "messageType": "peekreserve",
 "account": "APITEST",
 "symbol": "TLRY",
 "totalAvailable": "900",
 "inOpenOrders": "200"
}
```


The "totalAvailable" returned can be:

-1 = easy to borrow, no restriction on shorting
0 = no shares reserved for this user
>1 = the amount of shares reserved and currently available to short

"inOpenOrders" are the shares that were reserved and are currently locked up in open short orders.

The "totalAvailable" amount is the block of shares that the user reserved earlier, less the amount the user used up in executed orders, less the amount the user has in currently open short orders.

10. Message Update Parameters

Server can send this message any time to notify the user of the client application for any reason. Message can be a particular action for a client application to take or can be just a text message to notify the user for some reason.

Parameter Name	Type	Description
messageType	string	"reserve"
account	string	Trading account
code	numeric	Message code
title	string	Short message summary
message	string	Message description

10.1 Message Update Example - server to client


```
{
 "messageType": "message",
 "account": "APITEST",
 "code": "101",
 "title": "Cancel/Close",
 "message": "Your account is now in cancel/close mode.
You can only cancel orders or close positions."
}
```