emsTradepoint Limited

Webhooks User Guide

June 2025

EMS TRADEPOINT

EMSTRADEPOINT WEBHOOK USER GUIDE

Webhooks provide real-time notifications for trading activity on emsTradepoint's trading platform, eliminating the need for continuous polling of the API. When configured, webhooks automatically push notifications to your system whenever new orders are placed, existing orders are modified or withdrawn, or when trades are executed or voided.

This guide explains how to set up and use webhooks to receive real-time trading notifications from emsTradepoint (ETP).

WHAT ARE WEBHOOKS?

Webhooks are HTTP callbacks that ETP sends to your system when specific events occur on the trading platform. Instead of your system repeatedly checking for updates (polling), webhooks push notifications directly to your specified endpoint as events happen.

BENEFITS OF USING WEBHOOKS

- **Real-time notifications**: Receive immediate updates when trading events occur
- **Reduced API calls**: No need to continuously poll the platform for changes
- **Efficient resource usage**: Lower bandwidth and processing requirements
- **Improved responsiveness**: React instantly to market changes

Remember to test thoroughly in the UAT environment before deploying to production, and always implement proper error handling and monitoring for a robust integration.

WEBHOOK TYPES

ETP's platform supports two types of webhook notifications:

New/Changed Order Notifications

Triggered when:

- A new bid or offer is placed on the exchange
- An existing order is modified
- An order is withdrawn

New/Voided Trade Notifications

Triggered when:

- A new trade is executed
- An existing trade is voided

Important: Webhooks notify you of ALL orders and trades on the platform, not just those from your company.

SETTING UP WEBHOOKS

Webhooks are configured per company in ETP's platform. Contact the ETP support team supportdesk@emstradepoint.co.nz to set up webhooks for your organisation.

Kia ora emsTradepoint Supportdesk,

I'd like to set up webhook notifications for [Your Company Name].

The configuration details are:

- *Type: New/Changed Order AND New/Voided Trade*
- *HTTP Method: POST*
- *URL: https://mycompany.com/api/webhook*
- *Authentication Type: None*

My endpoint is ready to receive notifications. Please let me know when this is configured so I can test it.

REQUIRED CONFIGURATION FIELDS

When requesting webhook setup, provide the following information to the ETP team:

Field	Description	Required	Example
Type	Notification type	Yes	"New/Changed Order" OR "New/Voided Trade"
HTTP Method	HTTP method for webhook calls	Yes	POST (recommended)
URL	Your endpoint to receive notifications	Yes	https://my.server.co.nz/endpoint
Azure Subscription Key	Optional Azure API key	No	your-azure-key
Authentication Type	Security method	Yes	"None" OR "OAuth"

AUTHENTICATION OPTIONS

No Authentication (Recommended)

Since ETP webhooks only contain non-sensitive IDs, No Authentication is recommended:

- **Authentication Type**: None

No additional information is required

OAuth Authentication

For enhanced security, OAuth can be configured:

- **Authentication Type**: OAuth
- **Authentication URL**: Your OAuth token endpoint
- **App Key**: Your application key
- **App Secret**: Your application secret
- **Resource**: Target resource identifier

WEBHOOK PAYLOAD EXAMPLES

New/Changed Order Notification

When an order event occurs, you'll receive a JSON payload containing the order type and unique ID:

```
{
    "bid_id": 10472
}
```

or

```
{
    "offer_id": 15234
}
```

New/Voided Trade Notification

When a trade event occurs, you'll receive a JSON payload with the trade ID:

```
{
    "id": 5848
}
```

PROCESSING WEBHOOK NOTIFICATIONS

RECEIVE THE WEBHOOK

Set up an HTTP endpoint on your server to receive POST requests from ETP. Your endpoint should:

- Accept JSON payloads
- Return HTTP 200 status for successful receipt
- Process notifications quickly to avoid timeouts

EXTRACT THE ID

Parse the JSON payload to extract the relevant ID:

- For order notifications: bid_id or offer_id
- For trade notifications: id

FETCH FULL DETAILS

Use the relevant API to retrieve complete information about the order or trade:

For Orders

- GET /v1/bids/[id]
- GET /v1/offers/[id]

For Trades

- GET /v1/trades[id]

IMPLEMENTATION EXAMPLE

The following is a basic example of processing webhook notifications using python. It is not intended to be used by developers:

WEBHOOK RECEIVER ENDPOINT

```
from flask import Flask, request, jsonify
import requests

app = Flask(__name__)

@app.route('/webhook', methods=['POST'])
def handle_webhook():
    # Get the JSON payload
    data = request.get_json()

    # Process order notification
    if 'bid_id' in data:
        process_bid_update(data['bid_id'])
        elif 'offer_id' in data:
            process_offer_update(data['offer_id'])
        elif 'id' in data:
            process_trade_update(data['id'])

    # Return success response
    return jsonify({'status': 'received'}), 200
```

```
def process_bid_update(bid_id):
     # Fetch full bid details from API
     url = f"https://exchange.emstradepoint.co.nz/api/v1/bids/{bid_id}"
     headers = {'Authorization': f'Bearer {your_access_token}'}

     response = requests.get(url, headers=headers)
     if response.status_code == 200:
            bid_details = response.json()
            # Process the bid information
            print(f"New/Updated bid: {bid_details}")

def process_offer_update(offer_id):
     # Similar implementation for offers
     pass

def process_trade_update(trade_id):
     # Fetch trade details and process
     pass
```

API AUTHENTICATION FOR FOLLOW-UP CALLS

When your webhook endpoint receives a notification, you'll need to authenticate with the API to fetch full details.

OAuth 2.0 Authentication

Before making API calls, obtain an access token:

```
import requests

def get_access_token():
     url = "https://exchange.emstradepoint.co.nz/oauth/token"
     payload = {
         "client_id": "your_client_id",
         "client_secret": "your_client_secret",
         "grant_type": "client_credentials"
     }
     headers = {"Content-Type": "application/json"}

     response = requests.post(url, json=payload, headers=headers)
     if response.status_code == 200:
     return response.json()["access_token"]
         else:
         raise Exception(f"Failed to get token: {response.status_code}")
```

Using the Access Token

Include the token in API requests:

```
def get_bid_details(bid_id, access_token):
    url = f"https://exchange.emstradepoint.co.nz/api/v1/bids/{bid_id}"
    headers = {"Authorization": f"Bearer {access_token}"}

    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        return response.json()
    else:
        raise Exception(f"Failed to get bid details:
{response.status_code}")
```

ERROR HANDLING AND BEST PRACTICES

WEBHOOK ENDPOINT BEST PRACTICES

- **Respond quickly**: Return HTTP 200 within a few seconds
- **Handle duplicates**: Be prepared for potential duplicate notifications
- **Implement retry logic**: For API calls to fetch full details
- **Log all notifications**: For debugging and audit purposes
- **Validate payloads**: Ensure received data matches expected format

ERROR SCENARIOS

- **Webhook delivery failure**: ETP may retry failed deliveries
- **API rate limits**: Follow rate limiting guidelines when fetching details
- **Authentication expiry**: Refresh OAuth tokens as needed

RATE LIMITING

When making follow-up API calls, be aware of ETP's rate limits:

- Monitor for HTTP 429 responses
- Implement backoff strategies for failed requests
- Cache frequently accessed data when appropriate

TESTING YOUR WEBHOOK INTEGRATION

Using the UAT Environment

Test your webhook integration using ETP's UAT environment:

- **UAT Domain**: https://uat-exchange.emstradepoint.co.nz
- **API Documentation**: https://uat-exchange.emstradepoint.co.nz/api/docs

Testing Checklist

- Webhook endpoint receives notifications correctly
- JSON payloads are parsed properly
- API authentication works for follow-up calls
- Error handling functions as expected
- System handles notification volume appropriately

SUPPORT AND TROUBLESHOOTING

COMMON ISSUES

- **Webhook not receiving notifications**: Check endpoint URL and accessibility
- **Authentication failures**: Verify OAuth credentials and token refresh logic
- **API rate limit exceeded**: Implement proper rate limiting and retry logic
- **Missing trade/order details**: Ensure API calls use correct endpoints and filters

GETTING HELP

For webhook setup, configuration issues, or general support:

- **Email**: supportdesk@emstradepoint.co.nz
- **Phone**: (04) 590-6692

API DOCUMENTATION

Live API documentation and testing interface:

- **Production**: Contact ETP for access
- **UAT**: https://uat-exchange.emstradepoint.co.nz/api/docs