Exante

Standard configuration integrated with the Exante broker

Installation Guide

Initial proxy server setup and configuration

APISIX Setup

Finmars APISIX Setup Guide

Abstract

As result of that instruction you will have proxy to Exante Broker e.g. https://exante-proxy.client.com where client.com your domain

Prerequisites

Before starting make sure you have the following:

1. A machine running Linux with Ubuntu 22.04.
2. Docker installed on your machine.
3. A public IP address.
4. Generated API keys in Exante Account Page
5. You need to convert `API Key` and `Secret Key` to Basic Auth Token String (Use any public base64 encoder, in other words it should be `base64(api_key + ":" + secret_key)`)

Steps to Set Up APISIX

1. **Run the APISIX Docker Container:**

   ```
   docker run -d --name apache-apisix -p 9080:9080 -p 9443:9443 -e
   APISIX_STAND_ALONE=true apache/apisix
   ```

2. **Create the `apisix.yaml` Configuration File:**
   Create a file named `apisix.yaml` with your desired configuration.
3. **Copy the `apisix.yaml` File to the APISIX Container:**

   ```
   docker cp apisix.yaml apache-apisix:/var/tmp/
   ```

4. **Move the `apisix.yaml` File to the APISIX Configuration Directory:**

   ```
   docker exec -i apache-apisix sh -c 'cp /var/tmp/apisix.yaml /usr/local/apisix/conf/'
   ```

5. **Reload the APISIX Configuration:**

```
docker exec -it apache-apisix apisix reload
```

Setting Up HTTPS

If you want your APISIX instance to listen for HTTPS connections, follow these additional steps:

1. **Obtain an SSL Certificate:**
   Get or create an SSL certificate and export it along with its private key to PEM format.
2. **Update `ssls.yaml`:**
   Put the PEM contents into the corresponding sections of `ssls.yaml`.
3. **Include SSL Configuration in `apisix.yaml`:**
   Add the contents of `ssls.yaml` to `apisix.yaml` before the `#END` instruction.

```
apisix:
  node_listen: 9080           # HTTP inbound port
  ssl:
    enable: true
    listen:
      - port: 9443           # HTTPS inbound port
        enable_http3: false

consumers:
  - username: finmars        # client name
    plugins:
      basic-auth:
        username: foo        # inbound credentials, that you will pass to Finmars IT Team
        password: bar


# upstream.yaml

upstreams:
  - id: 1
    nodes:
      api-demo.exante.eu:443: 1
    scheme: https
    pass_host: node
```

```
    type: roundrobin
  - id: 2
    nodes:
      api-live.exante.eu:443: 1
    scheme: https
    pass_host: node
    type: roundrobin


# plugin_config.yaml

plugin_configs:
  - id: 1
    plugins:
      basic-auth: {}
      proxy-rewrite:
        headers:
          set:  # credentials used to connect to api-demo.exante.eu # Basic Auth string from
API key and Secret Key
            Authorization: "Basic
NDQ3YTIxNzEtYWNkMi00N2Q0LWEyOWYtOTk4ZTA5YzQ2Njk0OkJSMUpEQk9NUGdXVmxpN2wyeDBW"
        regex_uri:
          - "^/demo/(.*)"
          - "/$1"
  - id: 2
    plugins:
      basic-auth: {}
      proxy-rewrite:
        headers:
          set:  # credentials used to connect to api-live.exante.eu # Basic Auth string from
API key and Secret Key
            Authorization: "Basic
NDQ3YTIxNzEtYWNkMi00N2Q0LWEyOWYtOTk4ZTA5YzQ2Njk0OkJSMUpEQk9NUGdXVmxpN2wyeDBW"
        regex_uri:
          - "^/live/(.*)"
          - "/$1"


# routes.yaml
```

```
routes:
  - uris:
     - /demo/md/*/accounts
     - /demo/md/*/symbols/*
     - /demo/md/*/summary/*
     - /demo/md/*/ohlc/*
     - /demo/md/*/transactions
     - /demo/trade/*/orders/*
    upstream_id: 1
    plugin_config_id: 1
  - uris:
     - /live/md/*/accounts
     - /live/md/*/symbols/*
     - /live/md/*/summary/*
     - /live/md/*/ohlc/*
     - /live/md/*/transactions
     - /live/trade/*/orders/*
    upstream_id: 2
    plugin_config_id: 2
#END
```

Also part if you dont have nginx proxy

```
# ssls.yaml

ssls:
  - id: 1
    snis:
      - test.com             # hostname where apisix instance is running
    cert: |
      -----BEGIN CERTIFICATE-----
      MIICojCCAYoCCQDoby1LjeWcZzANBgkqhkiG9w0BAQsFADARMQ8wDQYDVQQDDAZS
      T09UQ0EwIBcNMjQwNzEyMjE1MjEzWhgPMjEyNDA2MTgyMTUyMTNaMBMxETAPBgNV
      BAMMCHRlc3QuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzs5Y
      uQvVwF3CFbNlJPk8OO7QlKA8UuapWdSKgT59Vqnkk0/7dfUHfP6pOFG8kxQexZBG
```

```
  uxbID+nCM175pJyjXjd2mZngtj2E1l3vUHOH6hhE0o5xLAgrDbBxgHnMZHyh/4xz
  quXgho0nv1R0i+OcIou4viyDp4BslYBJUONskEihOyUTJaSUS6zR6Q1aABEVEgYi
  OF9Wyr2tJ58jTlrRvFgrsDXeOoyf81diAmWePg4mETanejbaDBhtCQH8PllFfU1b
  TeqqZXRAdEOkbV23o4BfPYPB73Px1bdo0u/wGZTTY2DKQwd9QPg8fg0dT8dvegke
  3dUUx/XozqekMt3w2QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAnBUBhEnk2zjP5
  P9AJ/2cwfrQSBJ/Bti5vW/xAAEAWxOWqpccdsenr9/T8D4n1g5SZJeXhmRWdvSdB
  pYbUrOa4ISjpvotxze4dCOhQo2EO9XupQZPKbioyZTWJczwgxmccdKLH8uJD2llZ
  e/JKVfFH7AxO9Cz9I3+oyfEehPjemb5dJg/HQa+nWCC7ic7tC4mvd72GyGqAUb4e
  omhzWfmSsDxQpmJauA6uW0swfH+Ws+id6YhwArsxgMYcwURQ6L1YeaXKs3kTRnrN
  9XVOTt1NQ0aHPlBo4kGDujJ+aCfH9bReFbwa1CV6nS6wDqXuOSWEkKa69hXI3Wzp
  gv0mZ+28
  -----END CERTIFICATE-----

key: |
  -----BEGIN RSA PRIVATE KEY-----
  MIIEpAIBAAKCAQEAzs5YuQvVwF3CFbNlJPk8OO7QlKA8UuapWdSKgT59Vqnkk0/7
  dfUHfP6pOFG8kxQexZBGuxbID+nCM175pJyjXjd2mZngtj2E1l3vUHOH6hhE0o5x
  LAgrDbBxgHnMZHyh/4xzquXgho0nv1R0i+OcIou4viyDp4BslYBJUONskEihOyUT
  JaSUS6zR6Q1aABEVEgYiOF9Wyr2tJ58jTlrRvFgrsDXeOoyf81diAmWePg4mETan
  ejbaDBhtCQH8PllFfU1bTeqqZXRAdEOkbV23o4BfPYPB73Px1bdo0u/wGZTTY2DK
  Qwd9QPg8fg0dT8dvegke3dUUx/XozqekMt3w2QIDAQABAoIBAHZaPQhZr5CRl7tX
  mXDZeg+TDKeiNCO1ggG40zM4Ef8A56EuytgszIZKL4ndrS/2+c1SzkfPt9rzioJf
  vjrosc3/q84n9CQXfOg5hfXiyEu+a9ScVERAwHLrIWnHSqPPd96KAMAzIpWePrsO
  mExejEJw999OFmJL6th8PHkgTkcbYRKv8K7WnNDIHDUK6iSg0rtyvg2YDgRDDiRv
  GS3ofHdO5WxQSrqxYdle8aThuKV31gjt3nlfDNinWKHGEmvXbSGu7HGCk61NlK+j
  py3lkTZLeoxVaoZ5fIjdA/aOWDgv9HQBxtiCJd4VGC6xAsLT9gBy9c6cJFdb52Nc
  w/jjn6ECgYEA/gx6UrCcp2TuZYU/xkEIMi6mVe7d60w7672pK6FxcNrBiV0T/+F1
  o2CwdOiq0X9IJkBLDIGM2fFTPtpIER60f8P/RnfzlX7GenOX+voR9zpYePuyIKaf
  UW6n4ZB4wlt7ZIoRwTAwo/Whhx+XGL4iVosw2DIf6tFXRWI0+8dCew0CgYEA0GT6
  Xp0yi/D9UjZ1vM/fpojbZh6gZaK6zQjWi5CrcMzC2FyDOK9BTS2BWe28gmgV2Sab
  9/fZJmXnpAn1eiI2Co8b4p+44QAPtPUpCslAiVROS1vaEv/S+0ag682jDFWnOR5C
  VYSg4hyCOf3KAOJb9mgL/B/vFEnq1YBEr3P6af0CgYEAqeFOKt1O4+DqSZjA/KGg
  CW6IbA4+94kSyKEa7sIWSZD+ugwzw3fQYl/Vn1e2YkDqzilQBhzbQyHMuMreRhRy
  Pr2hhk3P5LfRoTCLAJkYSmoJn10v0AWbo6iLOpqRJeZmrORm2viOjhVC3kiRkUoT
  TCvnjap3DV7PLOZu565nFkECgYEAtVC2Wi3RtdqWvboulHoU8HOw2Nga3HNKrmxb
  JxFXaQxvFwrfDSnG2lyWZ+UmGBxxrf8ewxA9OmB9u8cCcyJi/KrpKzOnCvUftWV9
  MSpLYXEdsgmX4uH88q3QA3pmu6umlFbUhk2gITuGvugmosBQHUMH8nTicjeh/+Lb
  YAC7xw0CgYA8rePgM3xCn5U7zZPbvraYa7UZKvmrBAJpnh+oaYVpF4B1SCVJl23V
  Ym/xVKzR3KOQ8eKiUOUTV23CeZ1CSIDeDS5Y3yjVmfBaBxlGmF/Z4kuvfTpk2Lns
```

```
AjX3slshjCQjfNtVfelqXW+9QDyZbDZpNs5Cfl8/awj0+Mwv/isxVA==
-----END RSA PRIVATE KEY-----
```

How To Test

Make a CURL request to APISIX proxy

```
curl -u foo:bar 0.0.0.0:9080/demo/md/3.0/accounts
```

or open `0.0.0.0.9080/demo/md/3.0/accounts` in browser and pass Basic Auth Credentials

Get help

- APISIX Documentation [https://apisix.apache.org/docs/apisix/getting-started/README/]
- Exante Documentation [https://api-live.exante.eu/api-docs/]
- Finmars IT Help: s.zhitenev@finmars.com

Vault Setup

Connecting Vault

Currently performed by a Finmars employee at the client's request.

Initializing Vault

Prerequisites

- A created database with an activated license

Actions

1. Go to `Settings -> Vault`
2. Activate Vault and save the secret keys.
   Do not lose these keys, as they cannot be recovered. If the keys are lost, the data saved in the Vault will be lost.
3. Inform the Finmars employee of the Vault token

Unseal Vault

Prerequisites

- A created database with an activated license
- Vault initialized

Actions

1. Go to `Settings -> Vault`
2. Click the Unseal Vault button
3. Enter the three keys one by one

Adding Access to Proxy Server

Prerequisites

- A created database with an activated license
- Vault initialized
- Vault unsealed
- Proxy Server deployed

Actions

1. Go to `Settings -> Vault`
2. Click the `Add Engine` button
3. Enter the name `finmars`
4. Click the `Add Secret` button
5. Add the following `json` content:

```
{
    "data_url": "{url_to_proxy_server}/md/3.0/",
    "trading_url": "{url_to_proxy_server}/trade/3.0/",
    "api_key": "foo",
    "secret_key": "bar"
}
```

- `data_url` - URL to the proxy server's information endpoint specifying the API version - example `https://demo-exante.finmars.com/demo/md/3.0/`
- `trading_url` - URL to the proxy server's trading endpoint (required for enriching transaction information with orders) specifying the API version - example `https://demo-exante.finmars.com/demo/trade/3.0/`
- `api_key`, `secret_key` - credentials for `basic auth` to the proxy server

7. Click the "Save" button

Configuration Installation

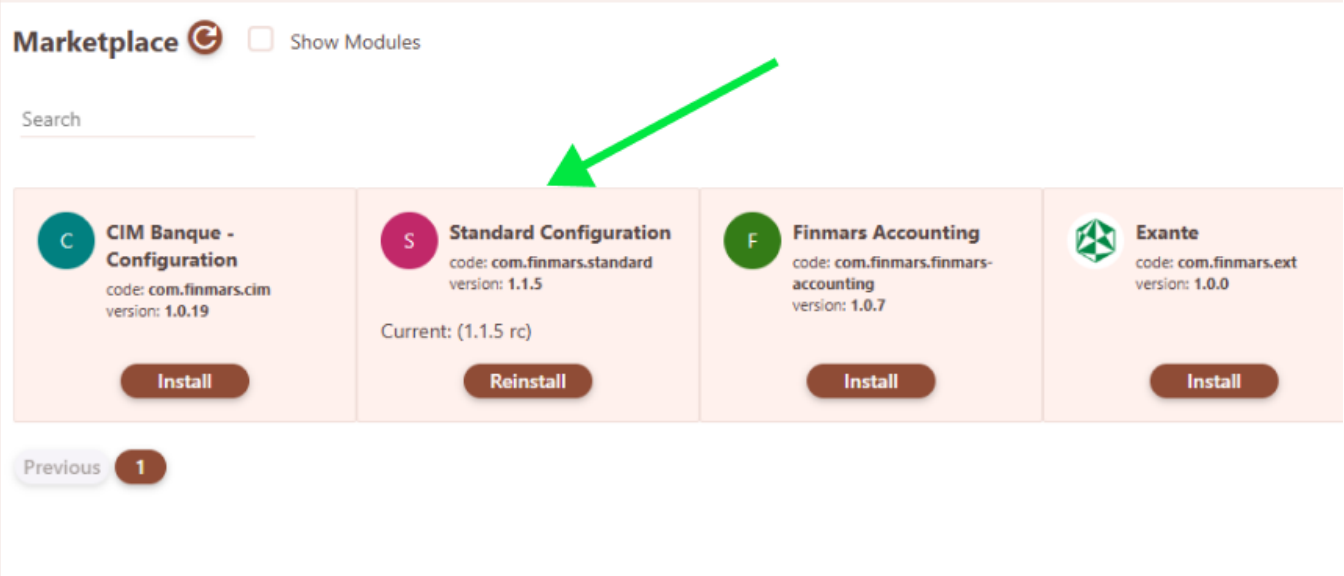
Installing Modules

Prerequisites

- A created database with an activated license

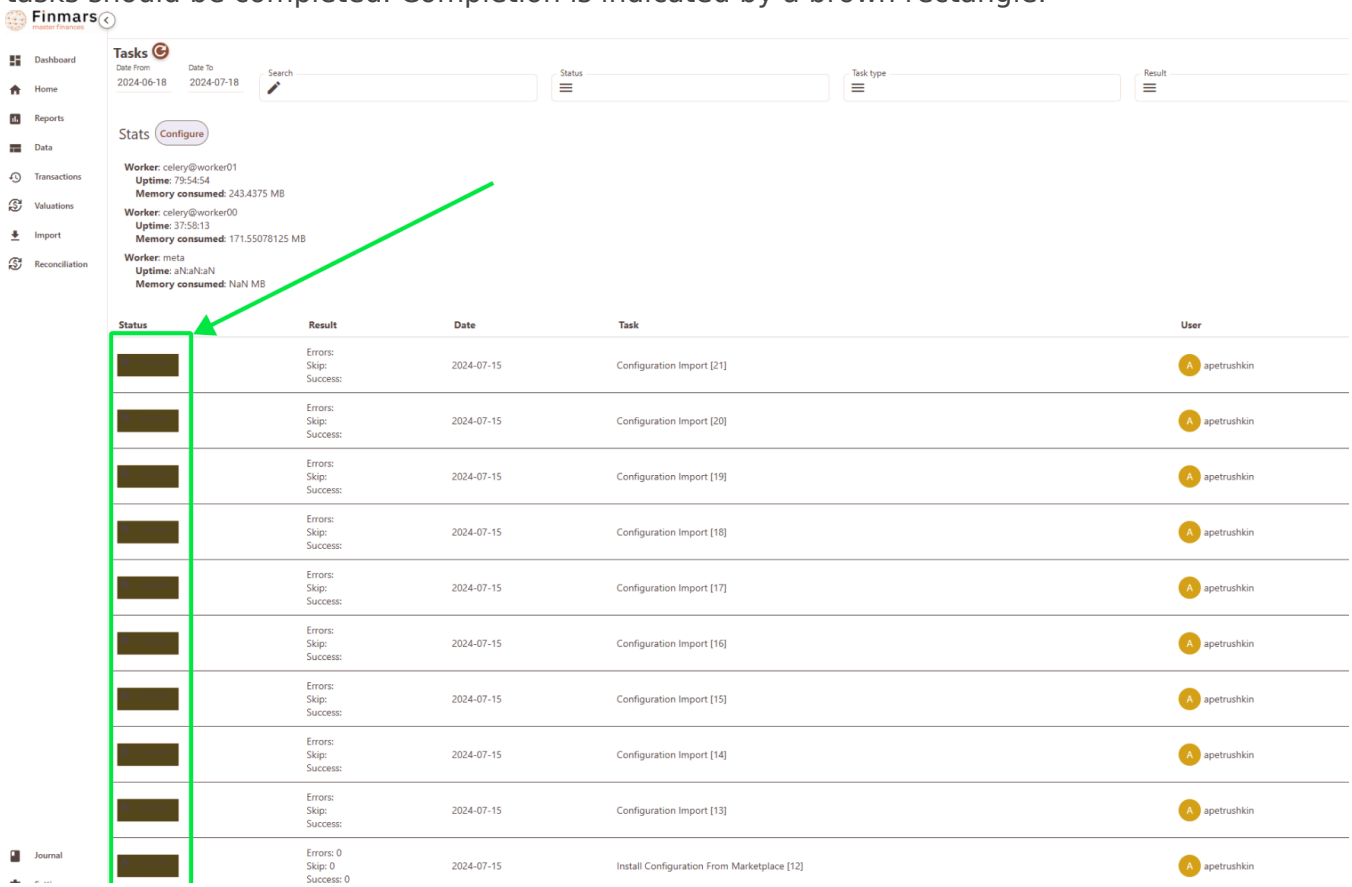
Actions

1. Go to `Settings -> Configuration -> Marketplace`
2. Select the `Standard Configuration` package and install it by clicking the install button



3. Wait for the module installation to complete

4. You can check the success of the module installation in the `Data -> Tasks` section. All tasks should be completed. Completion is indicated by a brown rectangle.



Installation Guide

Setting up AWS EC2 with Ubuntu, Docker, and a Public IP

Follow these steps to quickly set up a basic server environment on Amazon Cloud (AWS).

Step 1: Log in to AWS Console

- Visit https://console.aws.amazon.com
- Log in using your AWS account credentials.

Step 2: Create an EC2 Instance

- From the AWS console, search for and select **EC2**.
- In the EC2 Dashboard, click **Instances** → **Launch instances**.

Step 3: Select Ubuntu Server

- Under **Name and tags**, give your server a recognizable name (e.g., "MyUbuntuServer").
- Under **Application and OS Images (Amazon Machine Image)**, select **Ubuntu Server 22.04 LTS**. (or 24.04 LTS)
- Leave other settings at default unless instructed otherwise.

Step 4: Configure Security Settings

- Under **Network settings**, select **Create security group**.
- Ensure that at least the following ports are allowed:
  - **SSH (22)** for secure access to your server.
  - Add custom rules if needed, for example:
    - HTTP (80)
    - HTTPS (443)
- Click **Launch instance**.

Step 5: Connect to Your Instance

- After your instance launches, click on its name to see details.
- Locate the **Public IPv4 address**; this is your public IP address.

Step 6: Access Your Instance via SSH

- Open your terminal or command prompt.

- Connect using SSH (replace your-key.pem with the path to your downloaded AWS key file, and public-ip-address with your actual public IP):

```
chmod 400 your-key.pem
ssh -i your-key.pem ubuntu@public-ip-address
```

Step 7: Install Docker on Ubuntu

Follow Ubuntu | Docker Docs

You're all set!

You now have:

- Ubuntu Server 22.04 running on AWS
- Docker installed
- A public IP address to access your server remotely

Secure Base64 Encoding via Linux Console

Step 1: Open your Linux terminal.

Step 2: Execute the following command (replace api_key and secret_key with your real keys):

echo -n "api_key:secret_key" | base64

Example:
If your keys are:

- API Key: abc123
- Secret Key: xyz789

Then run:

echo -n "abc123:xyz789" | base64

Output will look like:

YWJjMTIzOnh5ejc4OQ==

How it works securely:

- echo -n prevents adding a newline at the end of the text, which ensures correct encoding.
- base64 is a built-in utility that encodes the data right on your system without internet exposure.

You can then safely use this encoded string as your Basic Auth token:

Authorization: Basic YWJjMTIzOnh5ejc4OQ==

Data Download

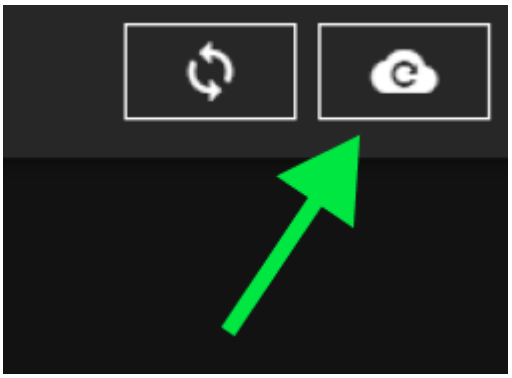
Historical Data Download

Prerequisites

- Standard Configuration Package is installed
- Proxy server deployed
- Vault is configured
- Credentials to Proxy Server is saved to the Vault

Restarting the workflow service

1. Go to the `Data -> Workflows` section
2. Click the "Refresh Storage" button



Configuring download and import parameters

1. Download the Input JSON files for setting up the automatic download of historical data
2. Edit the following fields in the first two objects under `workers` section in the `exante_historical_step_1.json`:

   `"name": "Download Positions"`
   - `date_from` - account opening date (there may be a limit of one year with Exante).
   - `date_to` - enter yesterday's date
   - `portfolios` - add account identifiers for download
   - `secret` - add the name of the secret containing credentials in the vault

`"name": "Download Transactions"`

- `date_from` - account opening date (there may be a limit of one year with Exante).
- `date_to` - enter yesterday's date
- `portfolios` - add account identifiers for download
- `secret` - add the name of the secret containing credentials in the vault

```
{
    "user_code": "workflow-manager",
    "configuration_code": "com.finmars.standard-workflow",
    "name":"Exante Historical - Step 1 (Download): positions, transactions",
    "workers": [
        {
            "order": 1,
            "configuration_code": "com.finmars.standard-workflow",
            "name": "Download Positions",
            "user_code": "download-exante-positions",
            "state_type": "period",
            "download_options":{
                "date_from": "2024-01-01", # <---------------- edit this
                "date_to": "2024-07-14", # <---------------- edit this
                "type":"day",
                "periodicity":"monthly",
                "portfolios": ["IKF1218.001"], # <---------------- edit this
                "secret":"itech-demo" # <---------------- edit this
            },
            "data_options": {
                "global_status": null,
                "source": null,
                "type": null,
                "portfolios": null
            },
            "import_options": {
                "scheme": null,
                "import_type": null,
                "pricing_policy": null
            },
            "calculation_options": {
                "date_from": null,
                "date_to": null,
                "portfolios": null
```

```
        },
        "state_options": {
            "input_path": null
        }
    },
    {
        "order": 2,
        "configuration_code": "com.finmars.standard-workflow",
        "name": "Download Transactions",
        "user_code": "download-exante-transactions",
        "state_type": "period",
        "download_options":{
            "date_from": "2024-01-01", # <---------------- edit this
            "date_to": "2024-07-14", # <---------------- edit this
            "type": "period",
            "periodicity":"monthly",
            "portfolios": ["IKF1218.001"], # <---------------- edit this
            "secret":"itech-demo" # <---------------- edit this
        },
        "data_options": {
            "global_status": null,
            "source": null,
            "type": null,
            "portfolios": null
        },
        "import_options": {
            "scheme": null,
            "import_type": null,
            "pricing_policy": null
        },
        "calculation_options": {
            "date_from": null,
            "date_to": null,
            "portfolios": null
        },
        "state_options": {
            "input_path": null
        }
    },
    {
```

```
            "order": 3,
            "configuration_code": "com.finmars.standard-workflow",
            "name": "Generate State",
            "user_code": "generate-state",
            "state_type": "fixed",
            "download_options":{
                "date_from": null,
                "date_to": null,
                "type": null,
                "periodicity":null,
                "portfolios": null,
                "secret":null
            },
            "data_options": {
                "global_status": null,
                "source": null,
                "type": null,
                "portfolios": null
            },
            "import_options": {
                "scheme": null,
                "import_type": null,
                "pricing_policy": null
            },
            "calculation_options": {
                "date_from": null,
                "date_to": null,
                "portfolios": null
            },
            "state_options": {
                "input_path": "/input-workflows/exante/exante_historical_step_2.json"
            }
        }
    ]
}
```

4. Edit the `secret` in `exante_historical_step_6.json` similarly to the previous step:

```
{
    "user_code": "workflow-manager",
    "configuration_code": "com.finmars.standard-workflow",
    "name":"Exante Historical - Step 6 (Download): market data",
    "schedule": null,
    "workers": [
        {
            "order": 1,
            "configuration_code": "com.finmars.standard-workflow",
            "name": "Download Prices (Exante)",
            "user_code": "download-exante-prices",
            "state_type": "period",
            "download_options": {
                "date_from": "first_transaction_date",
                "date_to": "now",
                "type": "period",
                "periodicity": "monthly",
                "portfolios": ["All"],
                "secret": "itech-demo" # <------------ edit this
            },
            "data_options": {
                "global_status": null,
                "source": null,
                "type": null,
                "portfolios": null
            },
            "import_options": {
                "scheme": null,
                "import_type": null,
                "pricing_policy": null
            },
            "calculation_options": {
                "date_from": null,
                "date_to": null,
                "portfolios": null
            },
            "state_options": {
                "input_path": null
            }
```

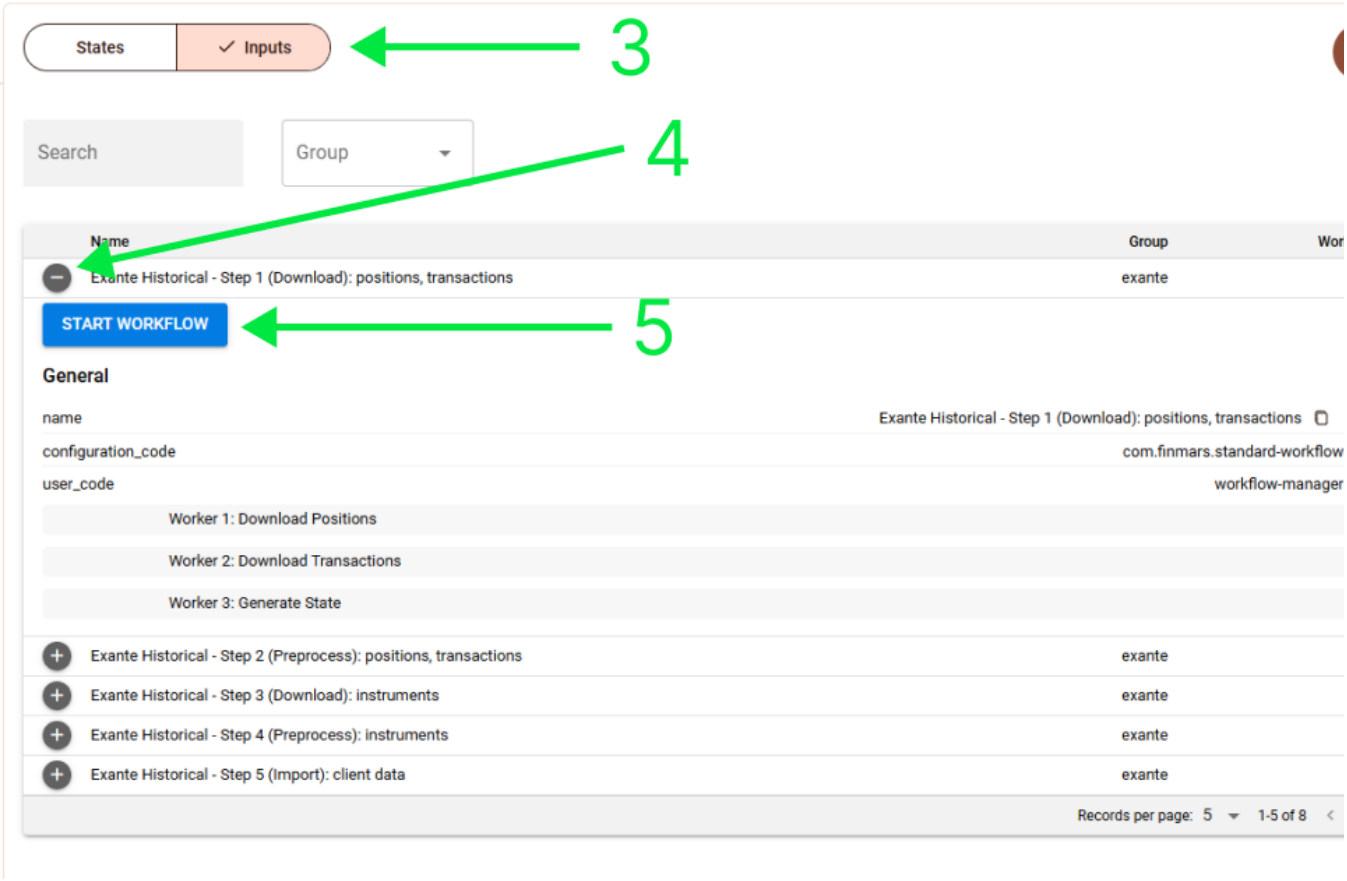
```
    },
    {
        "order": 2,
        "configuration_code": "com.finmars.standard-workflow",
        "name": "Download Fx Rates (Unified DB)",
        "user_code": "download-finmarsdb-fx-rates",
        "state_type": "period",
        "download_options": {
            "date_from": "first_transaction_date",
            "date_to": "now",
            "type": "period",
            "periodicity": "monthly",
            "portfolios": ["All"],
            "secret": null
        },
        "data_options": {
            "global_status": null,
            "source": null,
            "type": null,
            "portfolios": null
        },
        "import_options": {
            "scheme": null,
            "import_type": null,
            "pricing_policy": null
        },
        "calculation_options": {
            "date_from": null,
            "date_to": null,
            "portfolios": null
        },
        "state_options": {
            "input_path": null
        }
    },
    {
        "order": 3,
        "configuration_code": "com.finmars.standard-workflow",
        "name": "Generate State",
```

```
            "user_code": "generate-state",
            "state_type": "fixed",
            "download_options": {
                "date_from": null,
                "date_to": null,
                "type": null,
                "periodicity": null,
                "portfolios": null,
                "secret": null
            },
            "data_options": {
                "global_status": null,
                "source": null,
                "type": null,
                "portfolios": null
            },
            "import_options": {
                "scheme": null,
                "import_type": null,
                "pricing_policy": null
            },
            "calculation_options": {
                "date_from": null,
                "date_to": null,
                "portfolios": null
            },
            "state_options": {
                "input_path": "/input-workflows/exante/exante_historical_step_7.json"
            }
        }
    ]
}
```

Running the historical data download

1. Go to the `Dashboard` section
2. Check which layout is selected. It should be `Workflow`. If another layout is selected, change it to the required one.

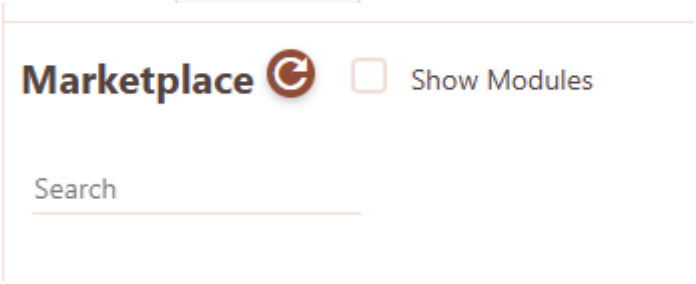
3. Select the `Inputs` section
4. Expand the details for `Exante Historical - Step 1 (Download): positions, transactions
5. Click the `Start Worfklow` button



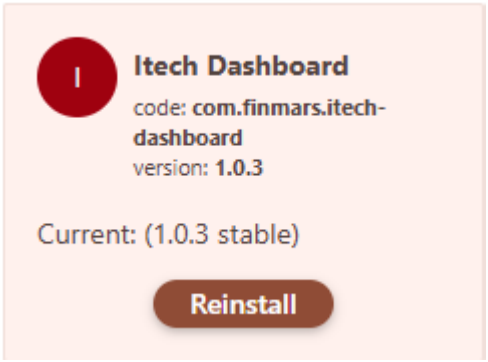
Dashboard Setup

Integration to 3rd party

1. Go to the `Settings -> Configuration -> Marketplace`
2. Click checkbox `Show Modules`



3. Install `Itech dashboard` module



4. Generate a token by visiting the link

```
https://eu-central-2.finmars.com/{realm_code}/{space_code}/api/v1/auth-tokens/personal-access-token/create-token/
```

5. Enter:
   - the `name` and `user_code` of the token (at your discretion),
   - set `access_level` - `admin`,
   - `days to live` - at your discretion
6. Prepare to embed a script with JS code on an external site from the example below:
   - add the token to it
   - change `realm_code` and `space_code` in the `loadSecureModule` function
   - change the values for the variables in `realCode` and `spaceCode`
   - change the identifier for the selector from the HTML page where the dashboard element needs to be inserted

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Itech External Dashboard</title>
    <style type="text/css">
      html {
        font-size: 100%;
      }

      body {
        font-family: helvetica;
      }

    </style>
</head>
<body>

<h1>Itech</h1>

<div style="display: flex;">

<div style="margin-right: 20px;">
  <div><label>External Date Picker</label></div>
  <input class="date-picker" type="date" name="" value="2024-07-16">
</div>

<div>
  <div><label>External Portfolio Picker</label></div>
  <select class="portfolio-picker">
    <option>IKF1218.001</option>
  </select>
</div>

</div>

<div>

  <div class="finmars-holder" style="padding: 20px;  margin: 20px;">

    <div style="margin-bottom: 1rem;">Section for Finmars Dashboard</div>
```

```
    <div id="dashboardContainer"></div>

  </div>
</div>




<script>

    var FINMARS_API_KEY = "YOUR_API_KEY"

    function loadSecureModule(url, apiKey) {
        return fetch(url, {
            headers: {
                'Authorization': `Bearer ${apiKey}`
            }
        })
        .then(response => {
            if (!response.ok) {
                throw new Error(`HTTP error! Status: ${response.status}`);
            }
            return response.text();
        })
        .then(scriptText => {
            const blob = new Blob([scriptText], { type: 'application/javascript' });
            const scriptURL = URL.createObjectURL(blob);
            return import(scriptURL);
        });
    }

    loadSecureModule('https://eu-central-
2.finmars.com/realm0sqjq/space00ql8/api/storage/workflows/com/finmars/itech-
dashboard/dashboard.js', FINMARS_API_KEY)
        .then(async (module) => {

          const FinmarsDashboard = module.default;

          const dashboard = new FinmarsDashboard({
                apiKey: FINMARS_API_KEY,
```

```
                selector: '#dashboardContainer',
                realmCode: "realm00000",
                spaceCode: "space00000",
                domain: "eu-central-2.finmars.com",
            });

          await dashboard.init();

          dashboard.dashboardStore.setDateTo("2024-07-16");
          dashboard.dashboardStore.setPortfolios(["IKF1218.001"]);
          dashboard.renderDashboard();

          document.querySelector('.date-picker').addEventListener("change", function(event) {

            console.log('datePicker.event', event);

            dashboard.dashboardStore.setDateTo(event.target.value);
            dashboard.renderDashboard();


          })

          document.querySelector('.portfolio-picker').addEventListener("change",
function(event) {

            console.log('datePicker.event', event);

            dashboard.dashboardStore.setPortfolios([event.target.value]);
            dashboard.renderDashboard();


          })

        })
        .catch(error => console.error('Error loading the script:', error));



</script>
```

```
</body>
</html>
```

7. Next, you can change the `date-picker` and `portfolio-picker` to your own HTML elements

Exante HTTP API Integration Pipeline (Work In Progress)

work in progress

Overview of the Integration Purpose

The Exante HTTP API Integration Pipeline for Finmars allows external clients to **automatically import portfolio data, transactions, and market data from the Exante trading platform into Finmars**. This integration is designed to keep your Finmars portfolios synchronized with your Exante brokerage accounts, providing up-to-date positions, transaction history, currency conversions, and instrument prices for comprehensive portfolio analysis. By using Exante's HTTP API, the connector fetches account information (portfolios), daily account summaries (positions and balances), transaction records, instrument details, currency lists, foreign exchange (FX) rates, and price data. It then transforms this data into Finmars's standard format for portfolios, accounts, transactions, instruments, currencies, FX rates, and prices, and imports it into your Finmars workspace. The end result is that users can see their Exante account activity and valuations reflected in Finmars with minimal manual effort.

This connector is particularly useful for **daily synchronization** of trading activity and valuations. For example, at the end of each day, Finmars can retrieve the day's trades, account balance updates, and closing prices from Exante, ensuring that performance metrics and positions in Finmars are accurate. It also **supports initial historical import** – allowing you to pull in past transactions and historical position data from a specified date range (e.g., the past year or since account inception) when first setting up the integration.

Credentials and Access Setup

Before using the connector, you must have **Exante API credentials** (API keys) and configure them in Finmars:

- **Obtain Exante API Keys**: Log in to your Exante account and generate API credentials. Exante provides a Client ID, an API Key, and a Secret Key for their HTTP API (refer to Exante's API documentation and support guides for instructions. Ensure API access is enabled for your account on Exante's side.
- **Finmars Secret Storage**: In Finmars, store the Exante credentials securely (for example, in Finmars Vault or secret manager). The connector expects the credentials in JSON format. Below is an **example credentials JSON structure**:

```
{
    "url": "https://api-live.exante.eu",
    "client_id": "<YOUR_EXANTE_CLIENT_ID>",
    "api_key": "<YOUR_EXANTE_API_KEY>",
    "secret_key": "<YOUR_EXANTE_SECRET_KEY>"
}
```

- **Save this JSON under a secret path that Finmars can access**. For instance, you might name the secret path as "`<your>-exante-http-api`" (replacing `<your>` with an identifier for your organization or user). In the connector's configuration (payload), you will reference this path. The `url` should point to the Exante API base (as in the example, the live API endpoint), in case you use proxy to filter out Order execution - type it instead of "*https://api-live.exante.eu*".
- **Permissions**: Ensure the Exante API keys have permission to read account data, transactions, instrument and orders info. The Finmars connector will use these keys to call endpoints.
- **Secret Path and Provider**: In the connector payload configuration, you will specify `secret_path` and `secret_provider`. Typically, `secret_provider` will be "finmars" (indicating use of Finmars's internal secret storage) and `secret_path` will be the name you chose (e.g., `"mycompany-exante-http-api"`). This allows the connector to retrieve the stored credentials at runtime.

Connector Payload Structure

To run the Exante HTTP API import, you will provide a **payload** – a JSON object that instructs the workflow what data to fetch and how. Below are **two payload examples**: one for **an initial historical import** and one for **a daily incremental update**.

Initial Import Payload Example

To import a specific date range of historical data

```
{
    "date_from": "YYYY-MM-DD",
    "date_to": "YYYY-MM-DD",
    "periodicity": "daily",
    "secret_path": "<your>-exante-http-api",
    "secret_provider": "finmars",
    "pricing_policies": ["com.finmars.exante-http-api:prc_pol"],
    "debug_mode": true,
    "exact_portfolios": ["<AccountID1>", "<AccountID2>", ...]
```

```
}
```

- **Explanation**: You provide a start date (`date_from`) and end date (`date_to`) **for the historical period** you want to import. The `periodicity` should be be `"daily"` in this case – indicating that the connector will handle data on a per-day basis within that range. The `secret_path` and `secret_provider` point to your stored Exante API credentials as discussed. The `pricing_policies` field is set to the Exante connector's pricing policy (more on this below) – this ensures that any imported price or FX data is tagged with the correct pricing policy in Finmars. `"debug_mode": true` is optional and will produce more verbose logging (helpful during initial setup or troubleshooting). The `exact_portfolios` field can list specific Exante account IDs to import; if omitted, the connector will attempt to **import all portfolios associated with the credentials**. If you have multiple accounts under the same API credentials, you can use `exact_portfolios` to limit the import to a subset (by Exante account ID). Replace `<AccountID1>`, etc., with the actual Exante account identifiers if needed, or omit this field to import all.

Daily Update Payload Example

For recurring daily sync after initial import

```
{
    "date_from": null,
    "date_to": null,
    "periodicity": "custom",
    "secret_path": "<your>-exante-http-api",
    "secret_provider": "finmars",
    "pricing_policies": ["com.finmars.exante-http-api:prc_pol"],
    "debug_mode": false,
    "exact_portfolios": ["<AccountID1>", "<AccountID2>", ...]
}
```

- **Explanation**: **For daily updates**, you can set `date_from` and `date_to` to null and use a `"custom"` periodicity. In this mode, the connector will determine the appropriate date range automatically (for example, it might fetch the latest data since the last import – e.g., "yesterday to today"). Essentially, leaving **the dates null signals the workflow to run an incremental update**. You still provide the `secret_path`/`provider` and pricing policy as before. `debug_mode` can typically be `false` for regular runs. The same `exact_portfolios` filter applies if you only want certain portfolios updated.

Important: Ensure that the `pricing_policies` list includes `"com.finmars.exante-http-api:prc_pol"` **exactly as shown above**. This string is the identifier of the Exante Pricing Policy within Finmars. It is used during import to tag all price and FX records with the correct policy, and it will also be used for portfolio configuration (pricing policy assignment). If you do not include it in the payload, the

connector’s code will add it automatically in most cases, but it’s good practice to specify it. Without the correct pricing policy, your imported prices and FX rates may not be applied correctly to portfolio valuations.

Initial Setup – Creating the Portfolio in Finmars

When you run the connector for the first time with an initial payload, it will contact Exante and retrieve the list of accounts (portfolios) associated with your API credentials. The connector uses Exante’s Accounts API (endpoint GET `md/3.0/accounts`) to get all account IDs and details. For each Exante account found, the connector will create a corresponding Portfolio in Finmars. The Finmars portfolio’s user_code will typically be set to the Exante accountId (so it matches one-to-one). Portfolio metadata such as account name, type, or currency might also be imported if provided by Exante.

Portfolio Import: The workflow’s first step is downloading portfolios and then importing them. All portfolios are fetched and saved, then processed into a standard format required by Finmars. Finally, an import step creates them in the Finmars system. If multiple accounts are present, multiple portfolios will be created (unless filtered by `exact_portfolios`).

Portfolio “_POS” Entities: During preprocessing, the connector will also create special portfolio entries with a “_POS” suffix. For each portfolio X from Exante, a corresponding portfolio X_POS might be generated. These “_POS” portfolios are used internally to handle position data. You will see them in Finmars as additional portfolios; however, you typically do not need to interact with them. They are primarily for the connector’s internal use (e.g., to store daily position change records or serve as a placeholder for position imports). You can ignore the _POS portfolios in normal usage – they will not contain transactions or balances that require user intervention. (In the current implementation, importing of position transactions into _POS portfolios is optional or a planned feature – see notes below – but the portfolios are created for future compatibility.)

Assigning Pricing Policy: After the portfolio import, it is crucial to ensure that each new portfolio (and its registers) is configured to use the Exante HTTP API pricing policy. Finmars uses pricing policies to determine how instrument prices and FX rates are applied to portfolios. The Exante connector comes with its own pricing policy (com.finmars.exante-http-api:prc_pol) which is tailored to Exante’s data characteristics. For example, Exante treats certain instruments (like CFDs and futures) differently for P&L calculations, and this custom policy accounts for that. If the connector does not automatically set the pricing policy during import (as of this writing, automatic assignment is a TODO item), you should manually set the portfolio’s pricing policy in Finmars to “Exante HTTP API Pricing Policy” (or similarly named) after creation. Typically, you would navigate to the portfolio’s settings (in Finmars UI) and select the Exante pricing policy for the portfolio’s registers. This ensures that all valuations (FX conversions, accruals, CFD/Future mark-to-market calculations, etc.) use the correct logic consistent with Exante’s data. Not doing so may result in

incorrect profit/loss calculations. (For example, Finmars calculates the market value of CFD and FUTURE instrument types as the difference between trade price and current market price under this policy, aligning with how these instruments should be treated.)

Data Import Process – Step by Step

Once portfolios are set up, the connector proceeds to download and import all other relevant data in a logical sequence. The major steps are outlined below:

1. **Download Account Summaries (Positions/Balances)**: The connector retrieves daily account summary data (positions) from Exante for each portfolio. Using the Exante Account Summary API for each account and each date, it gets information about positions and cash balances. Specifically, for each date in the range date_from to date_to, a call is made to GET md/3.0/summary/{accountId}/{date}/{currency} (usually with base currency USD). This returns the account's positions on that date – including all holdings of instruments and currencies. The connector loops through each day and saves the summary data. These daily snapshots are crucial for deriving historical FX rates and instrument prices. Note: Exante's Account Summary API returns data in a nested structure (positions as lists or dictionaries). The connector handles some quirks, for example, Exante returns a single summary per request (not paginated) and the connector wraps it into a list for uniform processing.
    - **Handling of Dates**: It's important to be aware that Exante defines the "end of day" cut-off at 5:00 PM New York time (ET) for positions. In other words, Exante's daily positions roll over at 17:00 America/New_York timezone. The connector is aware of this and adjusts dates accordingly. For example, a position timestamp of 2025-05-13 17:00:00 EDT is considered part of the next day (May 14) in Exante's system. Finmars will convert these timestamps so that positions are correctly attributed to the intended date. You do not need to manually adjust for this; just note that the connector's notion of a "position date" will align with Exante's convention. This means Finmars uses Trade Date only (no separate Value Date) for positions, matching Exante's approach. All transactions and positions are recorded with their trade date, ensuring that the daily balance in Finmars matches Exante's daily balance. (Exante does not provide or use a distinct value date in position calculations, so Finmars follows suit and relies on trade date for daily position records.)
    - **Expired Instruments**: One limitation – Exante's data does not provide details for expired instruments in the account summary. If an instrument expired before or during the period and is no longer held, the summary may not list it, which could lead to a gap in instrument info. Finmars attempts to retrieve any missing instrument data via other means (e.g., from positions of adjacent dates if possible). However, keep in mind that a completely expired instrument might not appear at all; in such cases, ensure you have the instrument details from another source if needed.

2. **Download Transactions**: In parallel with positions, the connector fetches transaction history for each portfolio. It uses the Exante Transactions API (GET md/3.0/transactions) providing the accountId and the date range. Exante returns all transactions within the specified period. The connector will retrieve these in one go per account (not day by day, since the transactions API supports a range). All transactions from date_from up to date_to (inclusive) are pulled and stored. If you run with date_from=null and date_to=null in a daily job, it will likely fetch the latest transactions (the connector might internally compute the last date or use stored state to avoid duplicates). The connector logs the date range it's using for transactions when it runs, so you can verify in the log output that it's selecting the correct period.
    - Only active (non-deleted) Finmars portfolios are considered. The connector will iterate through each imported portfolio (skipping any that are marked deleted or the special _POS ones) and request that account's transactions. Each account's transactions are saved for processing.
3. **Download Instruments Data**: As transactions and positions are gathered, the connector identifies all unique instruments (symbols) that appear in those records. It then calls Exante's Symbols API to get instrument details for each symbolId (and also symbol specifications, such as lot size, instrument type, tick size, etc.). Specifically, it uses GET md/3.0/symbols/{symbolId} and GET md/3.0/symbols/{symbolId}/specification for each needed instrument. These calls are rate-limited to avoid hitting Exante's limits (the connector has built-in rate limit decorators, e.g., max 60 calls per minute for symbols). Instrument data includes fields like instrument name, type (stock, bond, option, etc.), currency, exchange, and other metadata that Finmars uses to create an instrument master record. All instrument info retrieved is saved and later imported into Finmars's instruments list. If an instrument is not found (Exante might not return it or it's de-listed), the connector logs a warning or error. Generally, this step ensures that every security or asset referenced by any transaction or position is defined in Finmars.
    - Supported Instrument Types: Based on testing, the connector handles common instrument types like STOCK, BOND, CFD, FUTURE as provided by Exante. Exante may also have types like Calendar spreads, FX Spot, Index, Option, Fund, etc. However, in Exante's HTTP API data, those might not appear or may not be fully supported. In fact, instrument types such as CALENDAR_SPREAD, CURRENCY (for currency pairs), FX_SPOT, INDEX, OPTION, FUND were noted as not present in the data (or not supported). Finmars will import what it finds; if you hold exotic instrument types not supported by the connector, they could be skipped or require custom handling. (If an instrument type isn't recognized, the connector may still import it generically or log it as a warning.)
    - Missing Instrument Details: Exante's API might not provide certain instrument attributes like factor or pool factor (for bonds/MBS). The connector currently does not populate those either. This mainly affects instruments like bonds with accrual factors – Finmars will not have the factor info, which might slightly impact calculations like accrued interest. In practice, this is a minor limitation; Finmars can still track coupon payments via transactions.
4. **Import Currencies**: After instrument data is gathered, the connector moves on to currencies. It compiles a list of all currencies encountered – this includes base currencies

of accounts, transaction settlement currencies, and instrument currencies. Finmars maintains a list of currency entities and currency exchange rates. The connector will ensure each relevant currency is present as a Currency entity in Finmars (e.g., USD, EUR, HKD, etc.). The import flow has a step for currencies where it parses and creates any new currency entries. Usually, common currencies will already exist in Finmars, but if not, this step adds them. Additionally, the connector sets up account records in Finmars. In Finmars data model, an "Account" often refers to a combination of Portfolio + Currency (representing a cash account or position account in that currency). After currencies are imported, the connector runs an accounts parse & import step. This will create an Account entry for each portfolio-currency combination that appears in your data. For example, if your Exante portfolio has balances in USD and EUR, Finmars will have two account records under that portfolio – one for USD, one for EUR. These are used to track cash and interim balances per currency in the portfolio. The connector automatically creates these based on the data (you do not need to manually create currency accounts; the connector does it upon detecting a currency in use).

5. **Import Instruments**: The previously downloaded instrument details are now imported into Finmars's instrument master list. The connector preprocesses the raw instrument data into Finmars's standard instrument schema, then performs an import step. After this step, you will see all relevant securities/tickers listed in your Finmars Instruments library, each with basic metadata (symbol, name, type, ISIN if available, etc.). Finmars uses these instrument records to link transactions and to record price histories. If any instrument failed to import (due to missing data), it would be logged as an error. In a successful run, every symbol in your Exante transactions/positions should have a corresponding instrument in Finmars. (One note: since Exante doesn't provide info for expired instruments directly, if you have historical transactions in a symbol that you no longer hold, the connector relies on those transactions or positions to still carry the symbol ID so it can fetch details. If some instrument from very far back is completely absent from recent data, you might need to manually import it if it's missing.)
6. **Import FX Rates**: Using the daily account summary data collected, the connector next computes and imports FX rates for each currency for each date in the range. This is necessary for multi-currency portfolios so that Finmars can convert values to the base currency or perform currency conversions on each day. Exante's API itself does not provide a direct historical FX rates endpoint. However, in the Account Summary data, each currency's balance is typically presented in a base currency (like USD). The connector leverages this: for each day's position snapshot, it takes the conversion rate implied by the account summary and uses it as that day's FX rate in Finmars. For example, if on 2025-06-04 the account summary shows an HKD cash balance and its equivalent in USD, the ratio gives the USD/HKD rate for that day. These are imported as currency exchange rate records (Finmars Currency History) under the Exante pricing policy. The workflow step for FX rates parses the position data and generates FX rate entries. After import, each currency (other than your base currency) will have a historical FX rate for each day of the imported period, available in Finmars. This allows Finmars to accurately calculate daily NAV and P&L across currencies.
    - FX Rate Discrepancies: Be aware that there can be small discrepancies or multiple FX rates on the same day. Exante sometimes uses slightly different FX rates for the

same currency on the same date depending on context (cash vs instrument conversion). For instance, on 2025-06-04, a client's data showed the HKD to USD rate as 0.127437943 for cash, but an HKD-denominated stock on that day used 0.127435883. These differences are very minor (often due to rounding or timing). Finmars will import what it finds – typically one rate per currency per day from the summary. If the summary contains multiple rates, Finmars might use one (for example, possibly the cash rate). The key point is that such tiny differences might exist; it's an artifact of Exante's data and generally shouldn't significantly affect portfolio valuation.

- Missing FX Rates: If a currency appears in transactions but not in positions for a given day, there is a potential issue: Finmars might not have an FX rate for that day. For example, if you had a small balance in a currency that wasn't carried overnight (thus not in the end-of-day summary), Exante's summary might not list it, and Finmars wouldn't have a rate. If this happens, that day's valuation in Finmars for that currency will be incomplete (e.g., you'll see no FX rate error or the transaction value might not convert). The documentation recommends addressing such discrepancies with Exante support if they occur – essentially, ensure that any currency used is reflected in the account summary by end of day. In practice, this is rare and usually only if transactions occurred but the position closed before end-of-day cutoff. Keep an eye on the logs; the connector may log a warning if it cannot find an FX rate for a currency on a needed date.

7. **Import Prices**: Next, the connector imports daily instrument prices for each security held in the portfolio. Similar to FX, Exante doesn't provide a separate historical price feed via this API, but the account summary includes valuations from which prices can be derived. Specifically, for each instrument position on each day, Exante provides either the market value or enough info (average price, quantity, etc.) to compute a market price. The connector goes through each position record and pulls out the instrument's closing price for that day. These prices are then imported into Finmars as Price History records for the instruments (again tagged with the Exante pricing policy). After this step, in Finmars you will have daily price points for each instrument for the dates the instrument was in the portfolio. This allows Finmars to calculate daily market values and P&L properly.
    1. **Pricing Details**: Finmars obtains these prices directly from the position summaries. If an instrument appears in the transactions but not in any position (say you bought and sold it intra-day, and by end of day it wasn't holding), then there may be no end-of-day price for it in the imported data. In such cases, Finmars might not have a price history entry for that instrument on that date. This again is a rare scenario (most traded instruments will show up at least in the day's summary if held at EOD). If it does occur, valuations for that instrument on that day may not be fully reflected.
    2. **Accrued Interest Exclusion**: Note that Exante's prices do not include accrued interest. For interest-bearing instruments (like bonds), Exante typically separates the accrued interest from the clean price. Finmars will import the clean prices as given. This means the price history in Finmars is clean price; any accrued interest component is handled via separate interest transactions (e.g., coupon payments or accrual entries in transactions). So if you notice that a bond's price in Finmars doesn't seem to include interest, that is expected – the connector is preserving

Exante’s convention. Finmars will still capture interest through transactions labeled as “Interest” or “Accrued Interest” (if provided), rather than in the price.

8. **Import Transactions**: Finally, the connector processes all the Exante Transactions and imports them into Finmars. This is a complex step, as each transaction from Exante must be translated into the Finmars transaction format, which might involve splitting or merging records and calculating additional fields for proper accounting. The connector groups transactions by type and processes each type with specific transformation logic. It covers all common transaction categories that Exante produces:
   - Trades (BUY/SELL): These are typically identified as type "TRADE" in Exante. The connector will map these to Finmars transactions with selectors “Buy” or “Sell” accordingly. It will carry over details like quantity, instrument, trade date, and use the trade price (if available). Exante quirk: Exante’s transaction feed does not include the trade price or principal directly for trades. To address this, the connector tries to retrieve the price from the Order API for that trade if possible. The connector will call GET trade/3.0/orders/{orderId} to find the executed price. If it succeeds, it uses that price to calculate the principal (trade amount). If it cannot find an order (for example, if the order information isn’t accessible), the trade transaction in Finmars might be imported without a principal amount (cash consideration). In such a case, the resulting P&L calculation could be off because Finmars wouldn’t know the cost basis. The connector logs if it cannot obtain trade price data. This scenario is uncommon, but if you notice a trade in Finmars with zero principal or missing cost, it might be due to this limitation – and you may need to manually adjust it or contact support.
   - Cash Movements (Deposits/Withdrawals): Exante transaction types like "FUNDING/WITHDRAWAL" and "ELECTRONIC TRANSFER" correspond to cash coming in or going out. The connector will import these as Finmars transactions with selectors “Deposit” or “Withdraw” as appropriate. They will be recorded in the portfolio’s cash account for the currency of the transfer. Typically, the amounts and dates come directly from Exante data.
   - Currency Conversions (AUTOCONVERSION): Exante may automatically convert cash between currencies (for example, to cover a negative balance or fees). These show up as type "AUTOCONVERSION". Exante’s API, however, does not clearly link the two sides of the conversion (the outflow from one currency and the inflow to another). Finmars handles this by treating each autoconversion entry separately as a cash transaction. The connector will categorize an autoconversion as a Deposit or Withdraw in the respective currency’s account, depending on the sign (positive amount becomes a Deposit, negative becomes a Withdraw). It essentially splits the currency exchange into two independent transactions – one leaving one currency, one entering another. In Finmars, these will appear as unrelated deposit/withdraw, which is by design since the linkage isn’t provided by Exante. The important thing is that your cash balances remain correct (one currency decreases, the other increases). The connector sets the transaction code for these using whatever reference Exante provides (if any) or a generated key, and notes that it’s an autoconversion. There is no trade price for these (since it’s currency exchange, the “trade price” field is set to 0). Note: Because of this approach, when viewing

transactions in Finmars you won’t see an explicit “FX conversion” pair; you’ll just see a withdrawal in one currency and a deposit in another for the same timestamp and amount. This is normal given Exante’s limitations.

- Dividends (DIVIDEND): Cash dividends from stocks or funds will appear as "DIVIDEND" transactions. The connector will import these likely as “Income (Non-instrument)” with the description of the dividend. The amount will credit the cash account in the dividend currency. These are straightforward. U.S. tax withholding might appear as separate "US TAX" or "TAX" transactions (the connector supports those types as well) – they will import as tax expenses (likely “Expense (Non-instrument)” entries).
- Interest (INTEREST) and Funding: Interest paid or charged on cash balances is given as "INTEREST" (and possibly "ACCRUED INTEREST" for some scenarios). These are imported as Non-instrument income or expense transactions. The connector decides the selector based on the sign – interest received will be categorized as Income (Non-instrument), interest paid (negative) as Expense (Non-instrument). The entire interest amount will be recorded in the cash account. If Exante provides separate fields for tax or VAT on interest, the connector includes those in the transaction details as well. Accrued interest might not come through as a transaction unless realized; typically, interest is realized periodically.
- Fees and Commissions: Transactions like "COMMISSION", "FEE", "CUSTODY FEE", "BANK CHARGE", "MARKET DATA FEE" etc. are all expenses. The connector will import them usually as Expense (Non-instrument) transactions, deducting from the cash account. Commissions might also be attached to trades in Exante’s data; if so, the connector may incorporate commission into trade transactions or as separate entries depending on how Exante reports them. (Many of these fees types were listed among supported types, indicating the connector is prepared to handle them.)
- Transfers of Securities (SECURITY TRANSFER, SUBACCOUNT TRANSFER): These transactions occur when you move securities between accounts (or subaccounts) without a trade. Exante represents them somewhat like internal buys/sells (with no cash exchange). The connector books these as transfers of instrument in or out. In Finmars, they will appear with selectors “Transfer Instrument (In)” or “Transfer Instrument (Out)”, rather than buy/sell. A challenge here is that Exante’s transaction record for a transfer does not include a price at which to value the transferred asset (since no cash is involved). Finmars needs a “principal” (cost basis) for the transfer to keep P&L correct. The connector tackles this by calculating a transfer price using available data. It looks at the instrument’s average price on the day of transfer and the previous day’s average price to infer what cost basis the transferred quantity had. Essentially, it uses the account summary’s averagePrice for that instrument (and quantity before vs after) to compute an implied value. The derived price is then applied as the transaction’s principal (so that a transfer-in has a “buy” value and a transfer-out a “sell” value). This allows P&L calculations to recognize if a gain or loss was realized through the transfer (though generally transfers shouldn’t create P&L – any difference would be due to rounding). If the instrument wasn’t held prior to transfer (so no previous average price), the connector defaults the transfer price to 0, effectively treating it as a zero-cost move. Caution: If multiple transfers of the

same instrument into the same portfolio occur on the same day or in close succession, the above average price method can become unreliable (the “previous” quantity and price might both change multiple times). This can distort P&L for that instrument. Currently, the connector doesn’t have a special mechanism for multiple same-day transfers – it will calculate each in isolation. Finmars recognizes that these situations might lead to incorrect average cost or P&L records. As a result, the documentation notes that such transactions will be flagged as “Unidentified” in Finmars for further review. In practice, this means if the connector detects an inconsistency or cannot confidently compute a transfer price, it may import the transaction in a generic way (undefined) and you might need to manually adjust it. Keep an eye on any Security Transfer transactions in Finmars; if they appear as “Unidentified” or have zero principal, consider reviewing their cost basis. Fortunately, internal security transfers are not very frequent for most users.

- Corporate Actions (STOCK SPLIT, etc.): A stock split in Exante’s transactions appears as two records (one for the reduction of old shares, one for the addition of new shares, for example). The connector expects a pair of records for "STOCK SPLIT" and will combine them to record a stock split event. In Finmars, this may be imported as a split transaction that adjusts quantities without cash movement. Similarly, other corporate actions like mergers might not be explicitly provided in the API – if any appear, the connector would try to handle them or mark them as unidentified. (From the data list, splits are handled; no data was noted for some corp action types beyond splits.)
- Other Transactions: Types such as "LEI ISSUANCE & RENEWAL" or any Exante-specific admin fees would likely come through as generic fees (the connector lists them but if no examples were available, they might be processed as a general FEE or Expense).
- After transformation, the connector imports all transactions into Finmars using the Transaction Import pipeline. Each transaction will now be visible under the respective portfolio (and account) in Finmars, with proper date, type, amount, instrument (if applicable), etc. Finmars’s calculation engine will, after import, recompute portfolio balances and P&L using the imported transactions along with the price and FX data. The connector even triggers a post-import calculation step to update derived records (like cumulative P&L, performance metrics). This means once the workflow is done, your Finmars portfolio should show correct positions and values for the latest date, and you can run reports on historical data.

Error Handling and Logging

The Exante connector is designed to be robust, but errors can happen (network issues, data issues, etc.). Here’s how it handles errors and what you should watch for:

- **Workflow Logs**: Finmars typically provides logs for each workflow run. When you execute the connector, check the log output for any error or warning messages. The

connector logs an "info" message for each major step and will log "error" messages if something fails. For example, if an API call to Exante fails (network timeout or invalid credentials), you'll see an error logged with the reason. The connector will mark that step as failed but will attempt to continue to the next portfolio or next step when possible.

- **Partial Failures**: The workflow processes portfolios sequentially; if one portfolio's data fetch fails, it may continue with the next portfolio (this prevents one bad account from stopping all others). Similarly, within a portfolio, if one day's positions fail to download, it might skip that day and try the next. All such skips are logged. It uses an internal state tracking (with statuses like "success", "error", "skip") to keep track of what's processed. At the end, it reports a summary of how many records were processed successfully or skipped.
- **Data Issues**: If the connector receives data but finds something unexpected (like a transaction type it doesn't recognize), it will still import a record for it under a default classification. It uses an "undefined_records" bucket for transactions that can't be categorized. These will be imported as "Undefined" transactions in Finmars with as much info as possible, so that no data is lost. You should review any transactions marked as undefined – they may require manual classification or could indicate a new transaction type that needs support. In our context, undefined transactions might arise if Exante introduces a new transaction type not in the connector's list.
- **Unidentified Classification in Finmars**: As mentioned earlier, certain transactions might be deliberately classified as "Unidentified" when their effect on P&L is unclear or if they could distort calculations (e.g., multiple same-day security transfers). This is to alert you that manual intervention might be needed. Unidentified transactions are usually excluded from P&L until resolved. If you see any, investigate those scenarios specifically.
- **Retry and Rate Limits**: The connector has built-in rate limiting for Exante API calls to avoid hitting usage caps. If you are importing a large range (e.g., multiple years of daily data or many instruments), the process might be slow but it is throttled intentionally. Do not interrupt the process – allow it to finish to avoid partial imports. If a rate limit is exceeded or Exante rejects a request, the connector may retry after a pause. In the logs, you might notice it waiting or see messages about rate limiting.
- **Idempotency and Re-runs**: If you run the connector for the same date range multiple times, it is generally safe. The connector might skip re-importing transactions that are already processed. For instance, it keeps track if it has processed transactions up to a certain date for a portfolio (to avoid duplicates). The code shows messages like "Transactions already processed for {date_range}, skipping" in such cases. If you want to force re-import, you may need to clear the state or use a different date range. Normally, you won't need to re-run initial loads unless you suspect missing data. For daily updates, scheduling it daily is fine; it will naturally fetch only the new data beyond the last run.
- **Failure Recovery**: If the workflow fails in the middle (e.g., due to a server restart or an unexpected error), you can typically re-run it for the same period. Since the connector writes intermediate data files, it might detect some files exist and either reuse or overwrite them. A clean re-run will not duplicate data in Finmars (duplicate checking is in place via transaction codes and state tracking). It's always good to verify after any failure: check if some data was partially imported. If so, either manually remove the partial import or consult Finmars support on how to safely re-run. In most cases, re-running will just fill in

what was missing.

Use Case Walkthroughs

1. **Initial Historical Import – Example**: Let’s say you are onboarding an Exante account into Finmars for the first time. You have trading data for the past year that you want in Finmars. You would do the following:
   - Credential setup: Obtain your Exante API keys and save them in Finmars Vault under my-exante-http-api (for example).
   - Configure payload: Create a JSON payload specifying the date range, for example "date_from": "2024-01-01", "date_to": "2024-12-31", along with your secret_path: "my-exante-http-api", and include the pricing policy as shown in the initial payload example. Turn on debug_mode: true if you want detailed logs for this first run.
   - Run the connector: You might trigger this via Finmars’s workflow scheduler or a manual run option. Start the “Exante HTTP API Import” task with your payload.
   - Monitor progress: In the Finmars interface, watch the workflow logs. You should see messages indicating portfolios being downloaded and imported, then positions for each date, transactions, etc. If your date range is large, this can take some time (the system may process day by day). Ensure no critical errors appear. Minor warnings can be noted for review but might not stop the process.
   - Verify portfolios: Once complete, check the Finmars “Portfolios” section. You should see a new portfolio corresponding to your Exante account (or multiple if you had several). Also note the presence of a portfolio with “_POS” – that is expected. Check the portfolio’s settings to ensure the pricing policy is set to Exante’s. If not, manually set it.
   - Verify transactions and balances: Open the portfolio in Finmars and look at the transaction ledger or account balances. You should see all deposits, withdrawals, trades, etc., matching what your Exante statements show for the imported period. Compare a couple of sample transactions (e.g., a known trade or dividend) to confirm the amounts and dates match. Check that the portfolio’s ending positions on the final date (date_to) align with Exante’s actual holdings. If everything is correct, Finmars should show the same positions and cash balances as Exante did on that date.
   - Review P&L and calculations: With the pricing policy in place, Finmars will compute profit/loss. You might want to run a Portfolio Performance report or check the P&L for a specific period to ensure it looks reasonable. Because the import included daily prices and FX rates, the valuations should be accurate. If you notice any discrepancies (for example, P&L off by a small amount), consider the caveats above (e.g., slight FX rounding differences or missing price for an intra-day trade). These usually resolve or have minimal impact.
   - Address Unidentified items: If any transactions were marked Unidentified or any instrument shows as “Unknown”, you may need to resolve those. For instance, if an instrument didn’t import and a transaction is unidentified, you might manually

create that instrument in Finmars and link the transaction. Or if a transfer was unidentified, double-check the cost basis for that asset in Exante and manually adjust the transaction in Finmars. Such cases should be rare – most standard data will import cleanly.

- Lock-in Starting Balances: If your date_from was not the very beginning of the account (e.g., you imported 2024 data but the account existed in 2023), be aware that transactions prior to date_from were not imported. In Finmars, the portfolio's starting balances on 2024-01-01 would be zero by default, which isn't accurate. In these scenarios, the connector can create opening balance adjustments using the first day's position snapshot. (If the connector's design includes an initial position import as transactions, those would appear on date_from as opening entries.) If it didn't, you might need to manually enter an opening balance transaction for holdings as of 2023-12-31 so that 2024's starting position is correct. Check the first day of the imported range in Finmars: if positions show up on the first day, it means the connector likely accounted for them; if not, consider adjusting. Going forward, if you always import continuously, this won't be an issue.

2. **Daily Incremental Update – Example**: After the historical load, you'll want to keep data current. Suppose it's now the next day and you want to import yesterday's trading activity. You would:
   - Set up a scheduled task in Finmars to run the Exante connector daily (e.g., every night or early morning after markets close). Use the daily payload (with date_from/date_to null and periodicity: "custom"). Finmars can be configured to feed the current date or simply let the connector handle it.
   - On schedule, the connector runs. It will fetch the account summary for yesterday and today (or just yesterday, depending on configuration) and the transactions for yesterday. Typically, if run on day D+1 with null dates, it might fetch all new transactions since last run (which effectively covers day D). The positions of day D are used for prices and FX.
   - The import steps execute similarly: it should find no new portfolios or instruments unless you started trading a brand new instrument (then it will add that instrument). It will import yesterday's FX rate and closing prices, and any transactions (trades, deposits, etc. from yesterday).
   - Verify in Finmars that the portfolio's data for yesterday is updated. For example, if you made trades on that day, check that they appear. Check that your positions as of end of day are correct.
   - Finmars will now have performance data up through yesterday. Each daily run appends the new data. Because the connector is incremental, it won't duplicate previous days. If a day has no transactions, it may still update FX rates and prices for that day's positions. If there were absolutely no changes, the run might simply log "no data" and not import anything (which is fine).
   - Continue to monitor daily runs occasionally. If a daily run fails (say Exante API was down), Finmars might alert you or you'll see an error in logs. In such case, you can retry that day manually. It's important to capture every day's positions for completeness of prices and FX; missing a day means missing a price point (though the next day's import will still update positions and valuations, it's best to have

continuous data).

Additional Notes & Best Practices

- **Pricing Policy Usage**: Always maintain the Exante pricing policy on all relevant portfolios. This policy not only handles the valuation logic but also segregates Exante's price data. Finmars allows multiple pricing policies so that, for instance, if you had other data sources for prices, they won't conflict. The connector specifically imports FX rates and prices under the "com.finmars.exante-http-api:prc_pol" policy, so your portfolio must reference it to use those numbers. If you clone or rename portfolios, remember to check the pricing policy setting.
- **Data Consistency**: The connector relies on consistency between transactions and positions. If something is off (e.g., a trade missing from transactions or a position appearing without transactions), it usually indicates a data issue on Exante's side. For example, if the transaction history indicates a currency balance but the position summary doesn't, there will be a missing FX rate for that currency. Such inconsistencies should be raised to Exante if encountered. The Finmars connector cannot fabricate missing data; it will import what's available and highlight the gaps.
- **Time Zones**: All dates are handled as described (with the 5pm cutoff). When viewing in Finmars, dates are typically shown as date only (no timestamps for transactions, or in your local timezone for UI). The important point is that transactions are recorded on the date they occurred in Exante's system, and positions are marked by date (with that NY cutoff rule applied). If you are in a different timezone, just note that a trade at, say, 6pm London time might show up as next day if 6pm London is 1pm New York (still same date) – generally no issues here, just trust the connector's assignment.
- **Testing in Debug Mode**: When first running the connector, using debug_mode: true is helpful. It will log more details about each transformation and API call, which can be invaluable for troubleshooting if something doesn't look right. Once you're confident in the integration, you can run in normal mode (debug_mode: false) to reduce log verbosity.
- **Performance Considerations**: Importing a large history (many accounts or many years) can take time. You might consider breaking very large imports into smaller chunks (e.g., one year at a time) if you encounter timeouts. The connector has a mechanism (TODO in code) to split date ranges and potentially to resume partial imports, but as a user, simpler is to just reduce the range and run multiple times if needed. The payload's periodicity field when set to "daily" suggests the connector processes day by day internally. If you find the process slow, know that it's pacing due to API rate limits. Avoid overlapping runs or running multiple instances for the same credentials simultaneously, as that could hit rate limits or cause data conflicts.
- **Support for New Data Types**: Keep an eye on Finmars release notes – the connector may be updated to support new Exante features or transaction types over time. If Exante introduces something like a new transaction type (for instance, a corporate action type), an update to the connector might be needed. Until then, such a transaction would appear as "Undefined" in Finmars (meaning you might manually adjust it). You can reach out to Finmars support in those cases with logs; since the connector logs the raw transaction

that it couldn’t handle, support can analyze it.

By following this guide, you should be able to configure and use the Exante HTTP API connector effectively. It automates the heavy lifting of data transfer and ensures your Finmars environment mirrors your Exante account activity. Always double-check critical financial figures after the initial integration (such as total portfolio value, cash balances, and a few sample trades) to confirm accuracy. Once set up, the ongoing maintenance is minimal – typically just monitoring the daily sync and addressing any rare anomalies that arise.