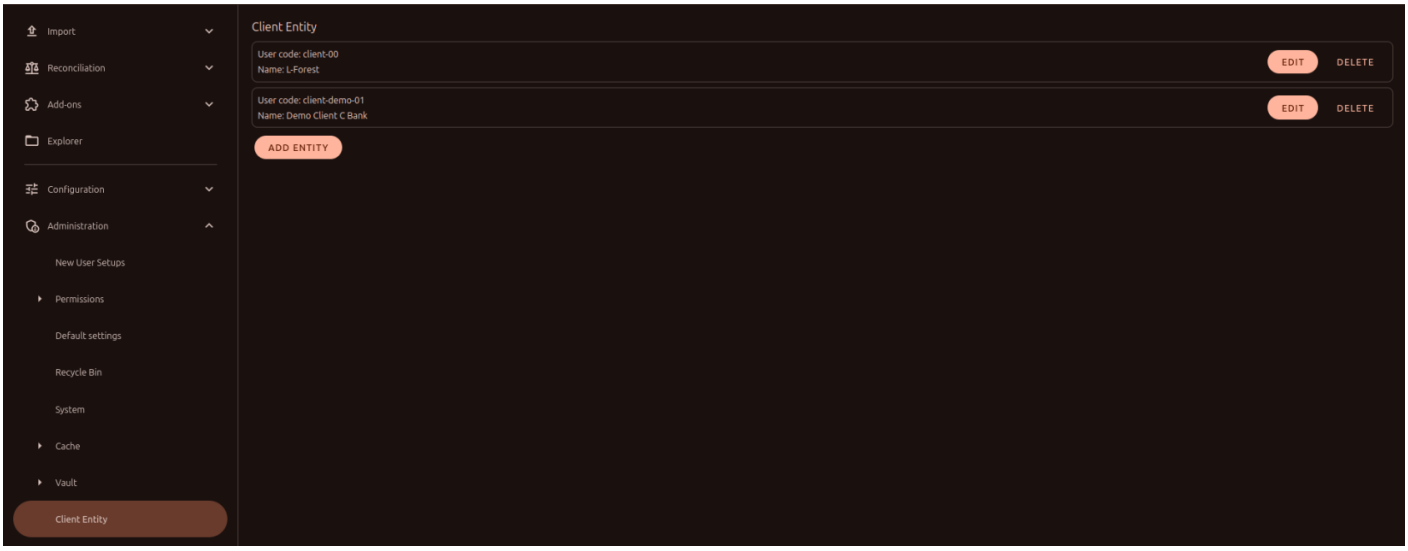
Client Entity

It stores the data of an individual client. It provides a single point of access to user data and **API** keys (third-party services) associated with customer portfolios.

- How to create an Client Entity?
- Description
- Client Secret

How to create an Client Entity?

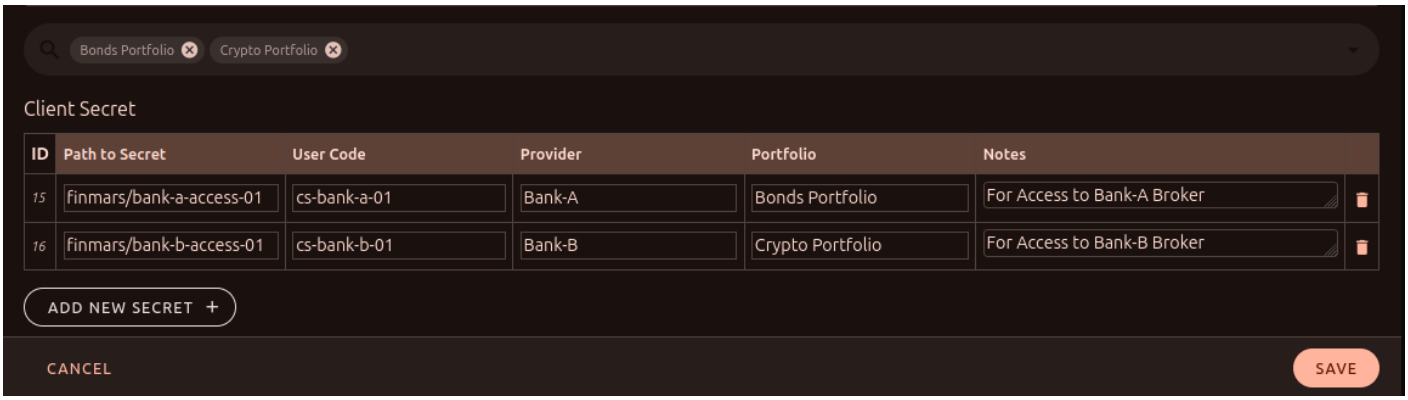
You need to get a new client into the system. To do this, find the item **Client Entity** in the side menu. On the page we can add a new client, or edit an existing one.



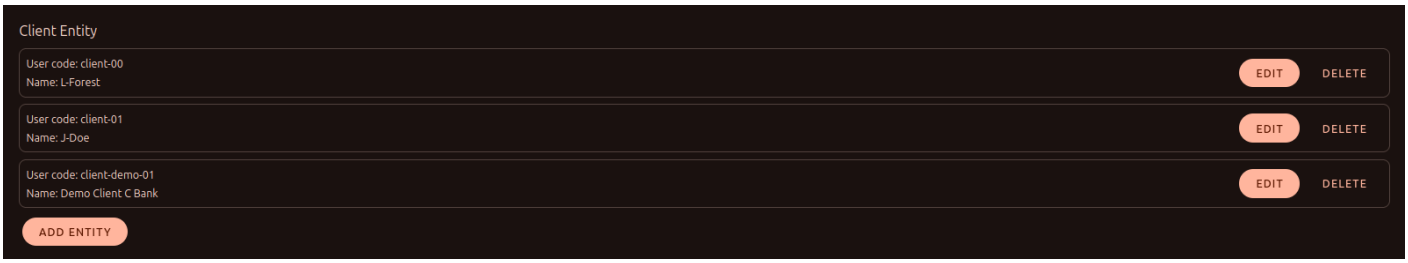
Our job is to add client into the system, so we clicked on the *Add Entity* button and a form appeared in front of us with the client's data filled in. Part of the form is reserved for the client's data, and the second part for his API keys.

Let's fill out the client data. Fill in the client given and select the portfolios that will belong to it.

We can also immediately fill in the data for the API key of a third-party broker. For example, our client has two portfolios. In two different banks.
Path to Secret - path to the vault (with API Key).
Provider - name of broker, source, bank, API.
Portfolio - Portfolio name (e.g. above I chose *Bonds Portfolio* and *Crypto Portfolio* for the client).



Once created, the client will be available for editing. For example, we can add a new API to it or modify an existing one.


Client Entity
User code: client-00
Name: L-Forest
EDIT
DELETE
User code: client-01
Name: J-Doe
EDIT
DELETE
User code: client-demo-01
Name: Demo Client C Bank
EDIT
DELETE
ADD ENTITY

Description

Table of Contents

- Description
- Examples
- Cookbook
- F.A.Q.
- API Documentation

Description

Financial meaning

It stores the data of an individual client. It provides a single point of access to user data and **API** keys (third-party services) associated with customer portfolios.

Platform abstraction

A **Client** in Finmars platform is a container for client information, this entity is referenced by:

- client_secret an entity that stores the API keys to the brokers, one client's connection to many secrets.
- portfolio container for financial assets, one client's connection to many portfolios.

Examples

Name	Short Name	Public Name	First Name	Last Name	User code	Notes	Email	Phone
J-Doe	J-Doe	J-Doe	Josh	Doe	client-00	-	J.Doe@bank-c.com	+8812345

- `User code`: workspace unique identifier of the client
- `Name`: full name
- `Short name`: short name, showed in other relations
- `Public name`: public view name for users without access
- `First name`: public view name for users without access
- `Last name`: public view name for users without access
- `Notes`: custom description
- `Email`: client's email
- `Phone`: client's phone number (5-15 digits long)

Cookbook

CRUD

Operations within platform.

Use Cases

What for it's used.

F.A.Q.

Frequently asked questions.

API documentation

Link to API documentation.

Client Secret

Table of Contents

- Description
- Examples
- Cookbook
- F.A.Q.
- API Documentation

Description

Financial meaning

It stores the data of an API key of broker, bank, API. It provides a single point of access to user data and **API** keys (third-party services) associated with customer portfolios.

Platform abstraction

Client Secret in the Finmars platform is information about the client's API key, which refers to:

- client container for client information.

Examples

User code	Provider	Portfolio	Path to Secret	Client ID	Notes
client-00	Bank A	Bonds Portfolio	finmars/bank-a-access	1	-

- `User code`: workspace unique identifier of the client
- `Provider`: name of broker, source, bank, API
- `Portfolio`: portfolio name
- `Path to Secret`: path to the vault (with API key)
- `Notes`: custom description
- `Client ID`: ID of client

Cookbook

CRUD

Operations within platform.

Use Cases

Now u can use some thing like this in workflow script:

```
payload = kwargs.get("payload")
client = payload.get("client")
secrets = workflow.finmars.get_secrets(client, provider="exante")


credentials = vault.get_secret(secret.path_to_secret)
```

F.A.Q.

Frequently asked questions.

API documentation

Link to API documentation.