

# Portfolio

## Table of Contents

- [Description](#)
- [Examples](#)
- [Cookbook](#)
- [F.A.Q.](#)
- [API Documentation](#)

## Description

### *Financial meaning*

**Portfolio** refers to a collection of financial assets. Portfolios are held directly by investors and/or managed by financial professionals. Effective portfolio management involves diversification, asset allocation, and regular performance monitoring to optimize returns and manage risks.

### *Platform abstraction*

**Portfolio** in Finmars platform is a container for financial assets (cash ([currency](#)) & [instruments](#)) allocation.

### **Portfolio also is used in:**

- [performance](#) calculation with its [registers](#) grouped in [bundles](#)
- [reconciliation](#) using special [portfolio type](#) and [portfolio reconcile group](#).

## Examples

User code	Portfolio Type	Name	Short name	Public name	Notes	First Transaction Date	First Cash Flow Date	Accounts	Transaction types	Responsibles	Counterparties
obl01	Manual	Bonds	Bonds	Bonds	-	2022-12-01	2022-12-01				

- `User code`: workspace unique identifier of the portfolio with source prefix as it's multi-source entity

- `Portfolio Type`: configuration-defined criterion for reconciliation
- `Name`: full name
- `Short name`: short name, showed in other relations
- `Public name`: public view name for users without access
- `Notes`: custom description for portfolio
- `First Transaction Date`: date of the first transaction in the portfolio (changes by the logic described below)
- `First Cash Flow Date`: date of the first transaction in the portfolio for base transaction class == `Cash-Inflow` or `Cash-Outflow` (changes by the logic described below)
- `Accounts`: (not strict, informative) selected accounts that can be used for this portfolio
- `Transaction types`: (not strict, informative) selected transaction types that can be used for this portfolio
- `Responsibles`: (not strict, informative) selected responsables that can be used for this portfolio
- `Counterparties`: (not strict, informative) selected counterparties that can be used for this portfolio

Portfolio has 2 attributes `First Transaction Date` and `First Cash Flow Date` which are used for performance calculations with workflow and performance report and the logic for them is defined below:

### Logic for defining `First Transaction Date` & `First Cash Flow Date` based on complex transactions:

- Portfolio entity has 2 properties:
  - `first_transcation_date` (name = `First transaction date`, value = `null`)
  - `first_cash_flow_date` (name = `First cash flow date`, value = `null`)
- For complex transaction CRUD operations For each base transaction** in the complex transaction:
  - if **Book**
    - if base transaction class == `Cash-Inflow` or `Cash-Outflow`
      - if `trade_date` < `first_cash_flow_date`
        - `first_cash_flow_date` = `trade_date`
      - if `trade_date` < `first_transcation_date`
        - `first_transcation_date` = `trade_date`
    - if **Rebook !!!**
      - if base transaction class == `Cash-Inflow` or `Cash-Outflow`
        - if **new** `trade_date` < `first_cash_flow_date`
          - `first_cash_flow_date` = `trade_date`
        - else
          - if **old** `trade_date` == `first_cash_flow_date`
            - `first_cash_flow_date` = find new `first_cash_flow_date`
        - if **new** `trade_date` < `first_transcation_date`
          - `first_transcation_date` = `trade_date`
        - else
          - if **old** `trade_date` == `first_transcation_date`

1. `first_transcation_date` = find new `first_transcation_date`
3. if **Delete**
  1. if base transaction class == `Cash-Inflow` or `Cash-Outflow`
    1. if `trade_date` == `first_cash_flow_date`
      1. if true `first_cash_flow_date` = find new `first_cash_flow_date`
    2. if `trade_date` == `first_transcation_date`
      1. if true `first_transcation_date` = find new `first_transcation_date`

## Cookbook

### CRUD

- **Create:** [How to Create a Portfolio Entity](#)
- **Read:** [How to Read a Portfolio Entity](#)
- **Update:** [How to Update a Portfolio Entity](#)
- **Delete:** [How to Delete a Portfolio Entity](#)

### Use Cases

- Within a portfolio, transactions and instrument positions are grouped using **registers**, which feed into bundles and ultimately affect portfolio-level calculations and reporting.
- Auto-populated fields such as **First Transaction Date** and **First Cash Flow Date** are critical inputs for internal workflows, audit tracking, and performance evaluation logic.
- Leverage the **Portfolio Type** field to define reconciliation logic and reporting behavior (e.g., General, Manual, or Position Only).

## F.A.Q.

Frequently asked questions.

## API documentation

[Link to API documentation.](#)

---

Revision #14

Created 15 July 2024 19:23:01 by Sergei Osipov

Updated 4 May 2025 17:47:11 by Anna Kostiv