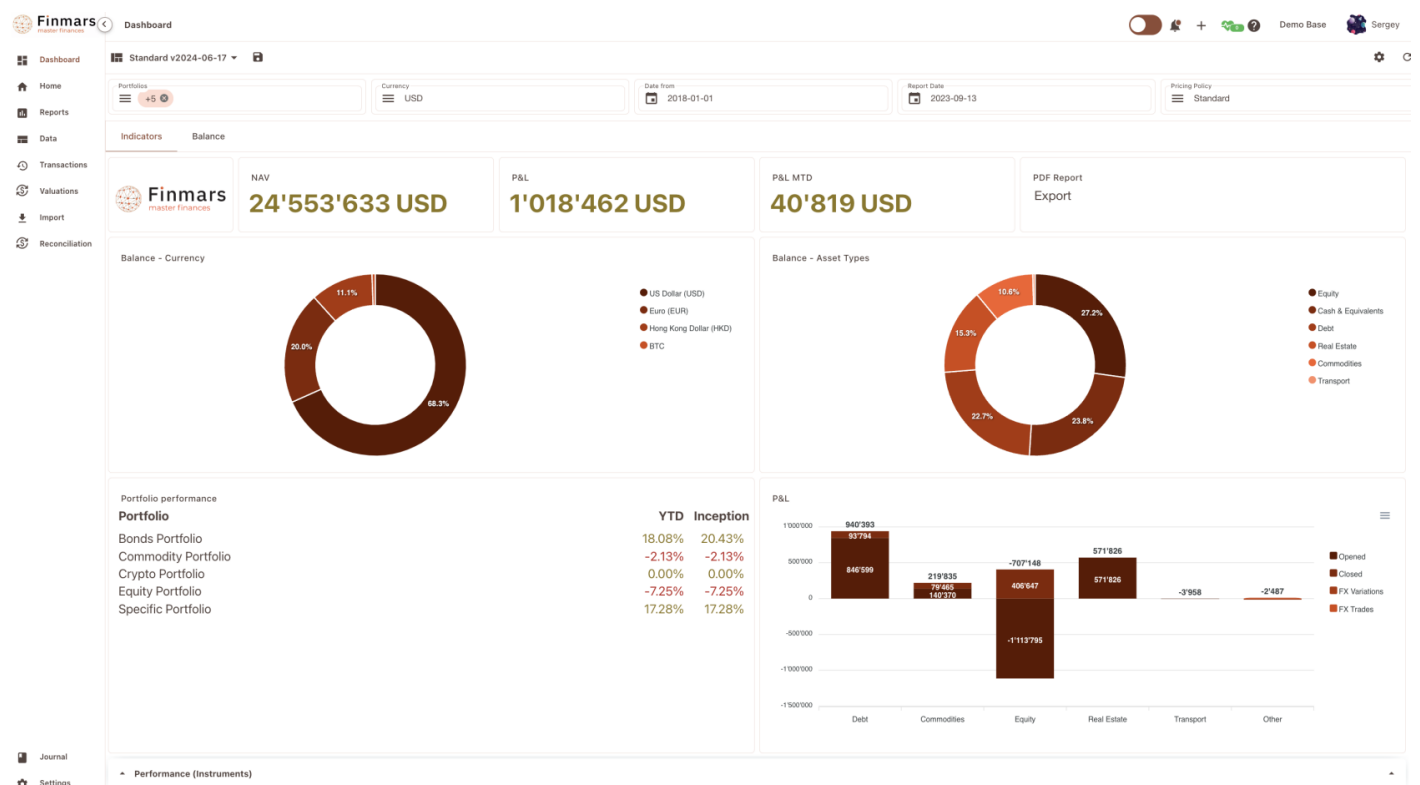


# Add Export Button to your Dashboard

Here, we found out how to add Export Button that generates PDF and saves it locally on User Device when pressed.

Here is result, on right corner we have block with [Export Button](#)



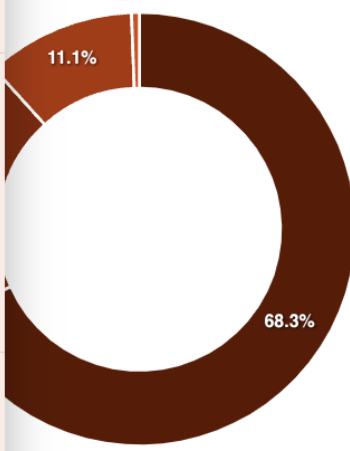
To achieve that, lets edit Dashboard Layout. Proceed to [Edit Dashboard Layout](#)

- Dashboard
- Home
- Reports
- Data
- Transactions
- Valuations
- Import
- Reconciliation

Standard v2024-06-17

Currency  
USD

3'633 USD      P&L  
1'018'46



- New layout
- PDF Builder
- Standard
- Standard (P&L)
- Standard Dashboard
- Standard v2024-06-17
- Standard v2024-06-17\_copy
- Save
- Open
- Make a copy
- Edit Dashboard Layout
- Export
- Select from list
- Invites (0)
- Rename
- Share
- Make default
- Delete

So here a Dashboard Constructor, press **Add New Component**

- Dashboard
- Home
- Reports
- Data
- Transactions
- Valuations
- Import
- Reconciliation

Configuration: com.finmars.standard-layout    User code: standard\_dashboard-2024-0...  
Result: com.finmars.standard-layouts:standard\_dashboard-2024-06-17

Dashboard Layout Name: Standard v2024-06-17    Deactivate Top Panel     Is Default

Portfolios (Control)    Currency (Control)    Date from (Control)    Report Date (Control)    Pricing Policy (Control)

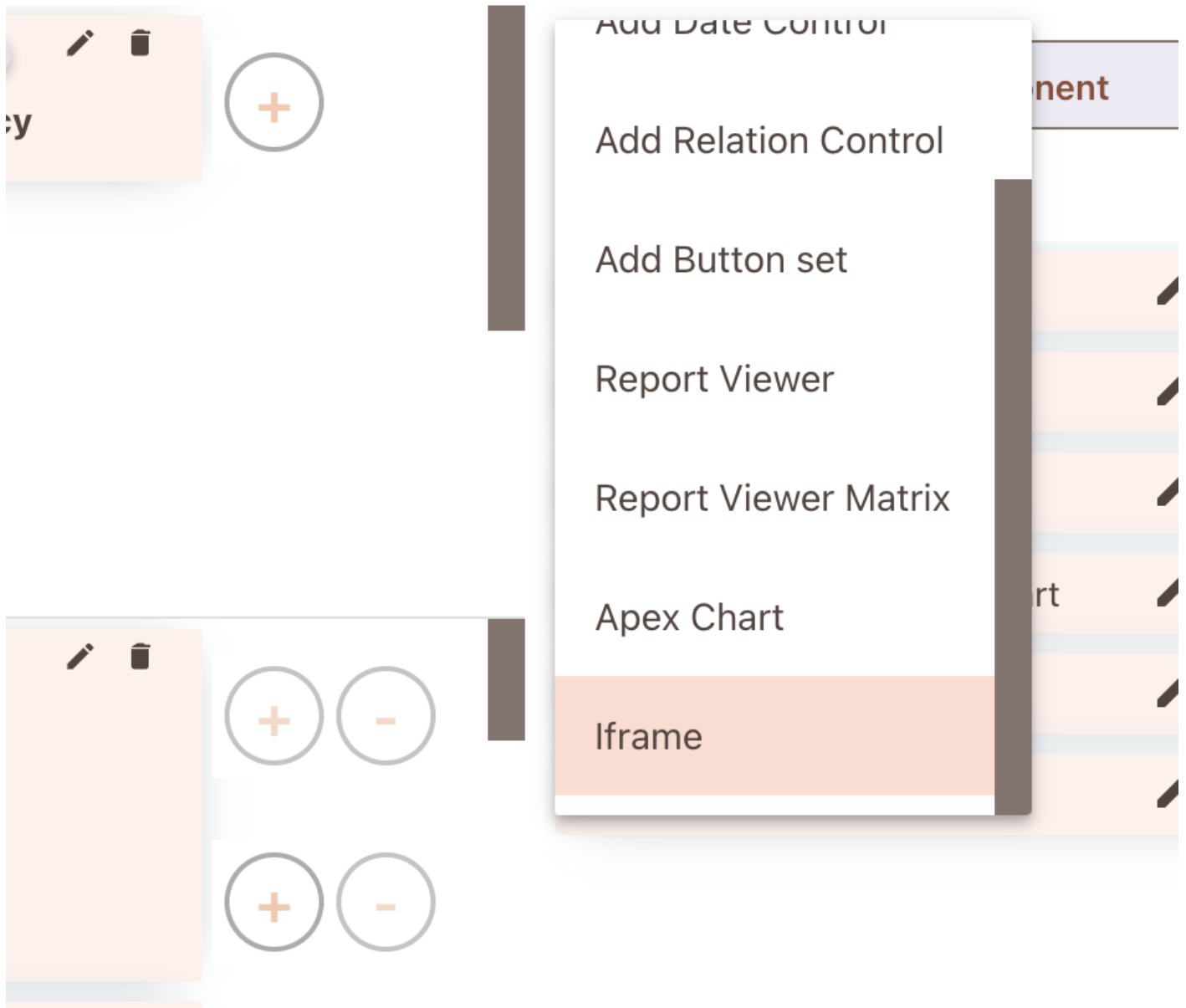
Change order of tabs

Tab name: Indicators    Tab name: Balance

Add New Component

- Drag cards from here
- YTD Return: Apex Chart
  - P&L YTD: Apex Chart
  - Transactions: Apex Chart
  - Portfolio metrics: Apex Chart
  - More: Accordion

Add Apex Chart (soon will be renamed to Custom Code Component)



To make it all work, we borrow code from PDF Builder

```
var selector = '.dashboard-apex-chart-' + scope.vm.componentData.id

var element = document.querySelector(selector);
```

```
element.innerHTML = '<button>Export</button>'

function getCookieValue(name) {
    const cookies = document.cookie.split('; '); // Split cookies into array of
"name=value"
    for (let i = 0; i < cookies.length; i++) {
        const cookie = cookies[i].split('='); // Split each "name=value" into [name,
value]
        if (cookie[0] === name) {
            return decodeURIComponent(cookie[1]); // Return the decoded value if the
name matches
        }
    }
    return null; // Return null if the cookie is not found
}

element.addEventListener('click', function() {

    let currentUrl = window.location.href;

    let base_url = window.location.href.split('/a/')[0]

    var portfolios = filters.outputs.portfolios || [];
    var report_date = filters.outputs.report_date || utils.getYesterdayDate();

    var portoflio = 'Bonds Portfolio'

    if (filters.outputs.portfolios.length){
        portoflio = filters.outputs.portfolios[0]
    }

    let url_path = '/api/storage/workflows/com/finmars/pdf-
builder/example/report.html?report_date='+filters.outputs.report_date+'&portfolio=' +
portoflio

    // Replace '_builder.html' with '.html' in the URL
    let targetUrl = base_url + url_path
```

```
fetch(window.location.origin + '/services/pdf-engine/print/', {
  method: "POST",
  credentials: 'include',
  body: JSON.stringify({
    "url": targetUrl,
    "access_token": getCookieValue('access_token')
  }),
  headers: {
    Accept: 'application/json',
    'Content-type': 'application/json'
  }
}).then(response => {
  if (response.ok) return response.blob();
  throw new Error('Failed to generate PDF.');
```

})

```
.then(blob => {
  // Create a URL for the blob object
  const url = window.URL.createObjectURL(blob);

  // Create a link element
  const a = document.createElement('a');
  a.style.display = 'none';
  a.href = url;

  // Extract filename for the PDF
  let baseName = targetUrl.split('/').pop().replace('.html', '').split('?')[0]; // Strip
the path and extension

  a.download = baseName + '.pdf'; // You can set the default filename here

  // Append the link, trigger the download, and remove the link
  document.body.appendChild(a);
  a.click();
  window.URL.revokeObjectURL(url);
  document.body.removeChild(a);
})
.catch(error => {
  console.error('Error during PDF export:', error);
  alert('Error exporting PDF. Check the console for more details.');
```

});

```
})
```

Idea is simple, we just add <button> and add Event Listener To it, in Event Listener we make an REST API call to PDF Printer Service (puppeteer)

Things to Consider

```
let url_path = '/api/storage/workflows/com/finmars/pdf-  
builder/example/report.html?report_date='+filters.outputs.report_date+'&portfolio=' +  
portfolio  
  
let targetUrl = base_url + url_path
```

We hardcoded `url_path` of our report.html

Place instead of it path to your .html PDF report, also provide extra settings if required like

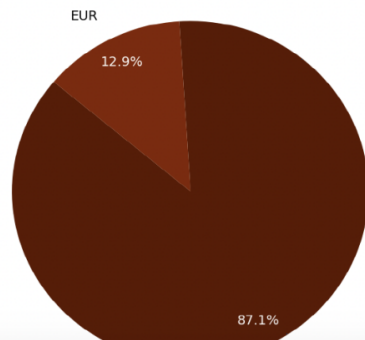
```
?report_date='+filters.outputs.report_date+'&portfolio=' + portfolio
```

Thats it! Now your clients can just press button on dashboard to get .pdf Report

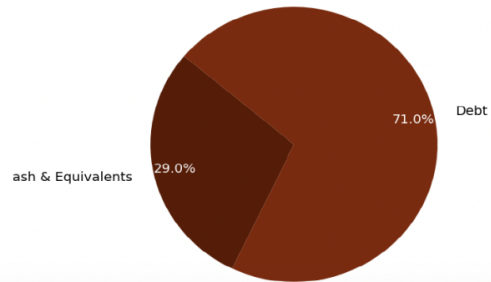
### Portfolio Overview

Start valuation as of 2022-12-30	2 046 371 USD
Cash/Security Deposits	4 928 914 USD
Cash/Security Withdrawals	0 USD
End valuation as of 2023-09-13	7 849 035 USD
Performance	873 749 USD
Return (Modified Dietz)	18.08%

### Allocation by Exposure Currency



### Allocation by Asset Type



Revision #3

Created 2024-08-16 20:32:56 UTC by Sergei Zhitenev

Updated 2024-08-16 20:41:58 UTC by Sergei Zhitenev