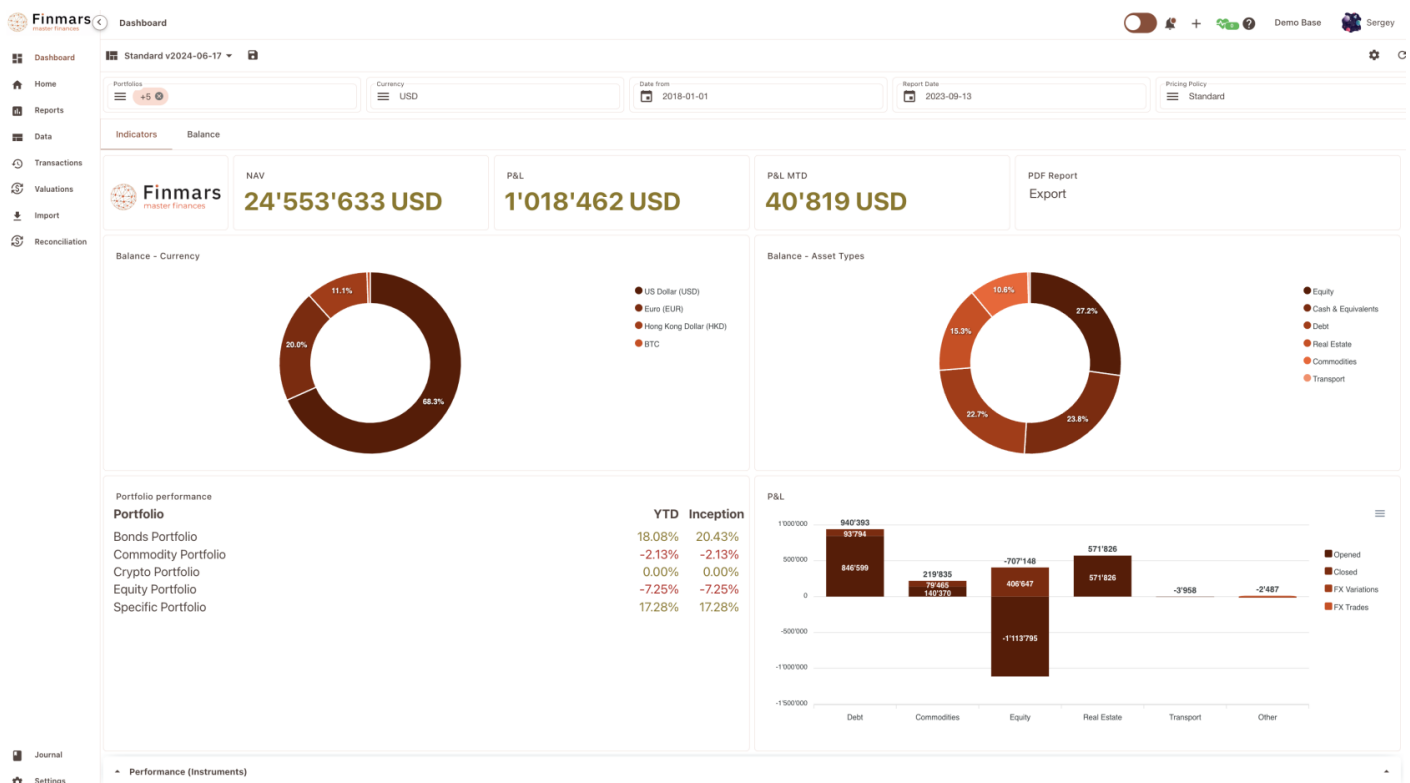


Add Export Button to your Dashboard

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](#)

+ 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

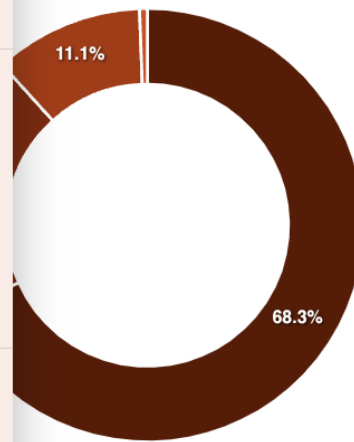
Currency

☰ USD

P&L

3'633 USD

1'018'46



So here a Dashboard Constructor, press [Add New Component](#)

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

Logo NAV (Apex Chart) P&L (Apex Chart) P&L MTD (Apex Chart) PDF Report (Apex Chart)

Add New Component

Drag cards from here

YTD Return: Apex Chart

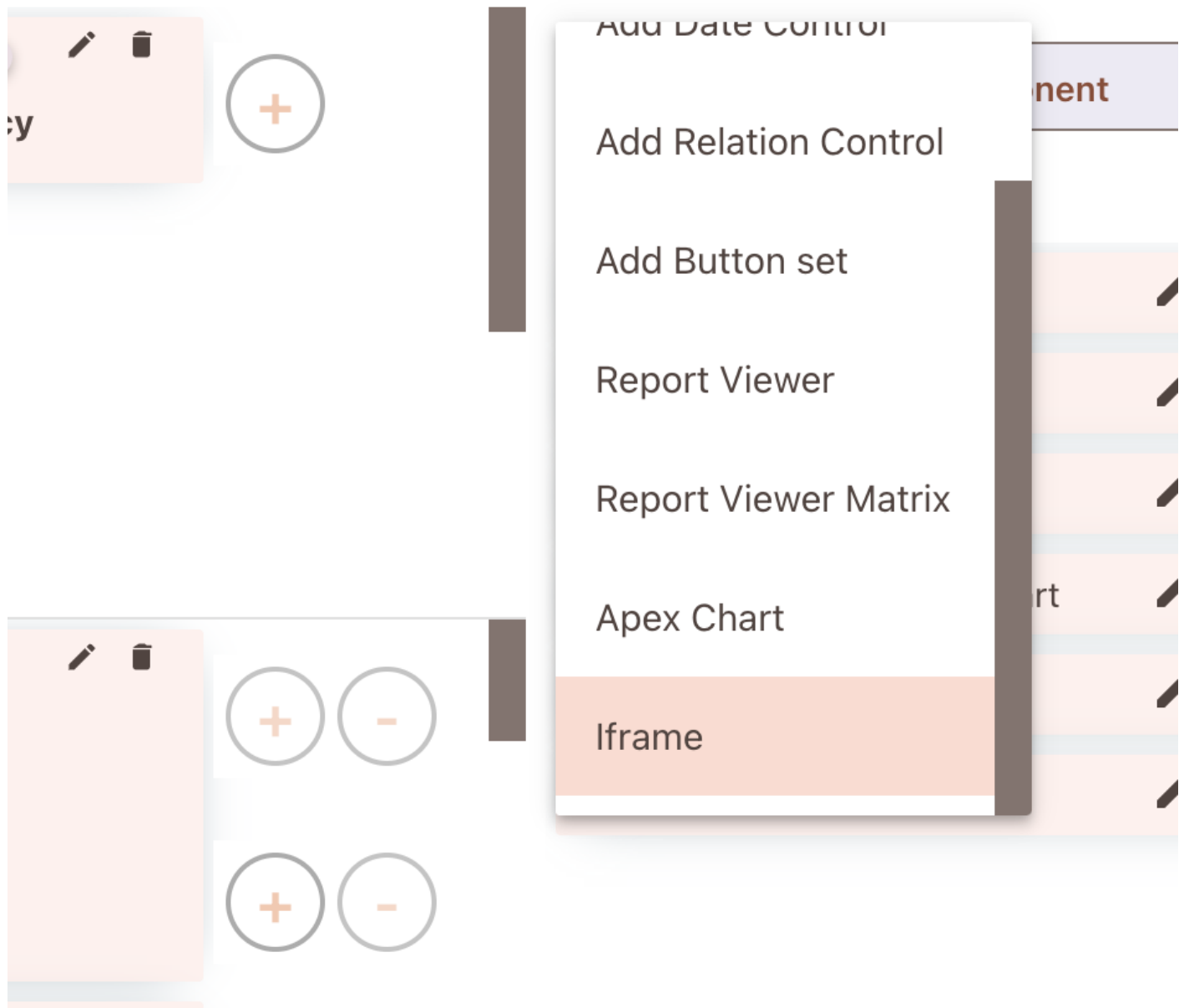
P&L YTD: Apex Chart

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

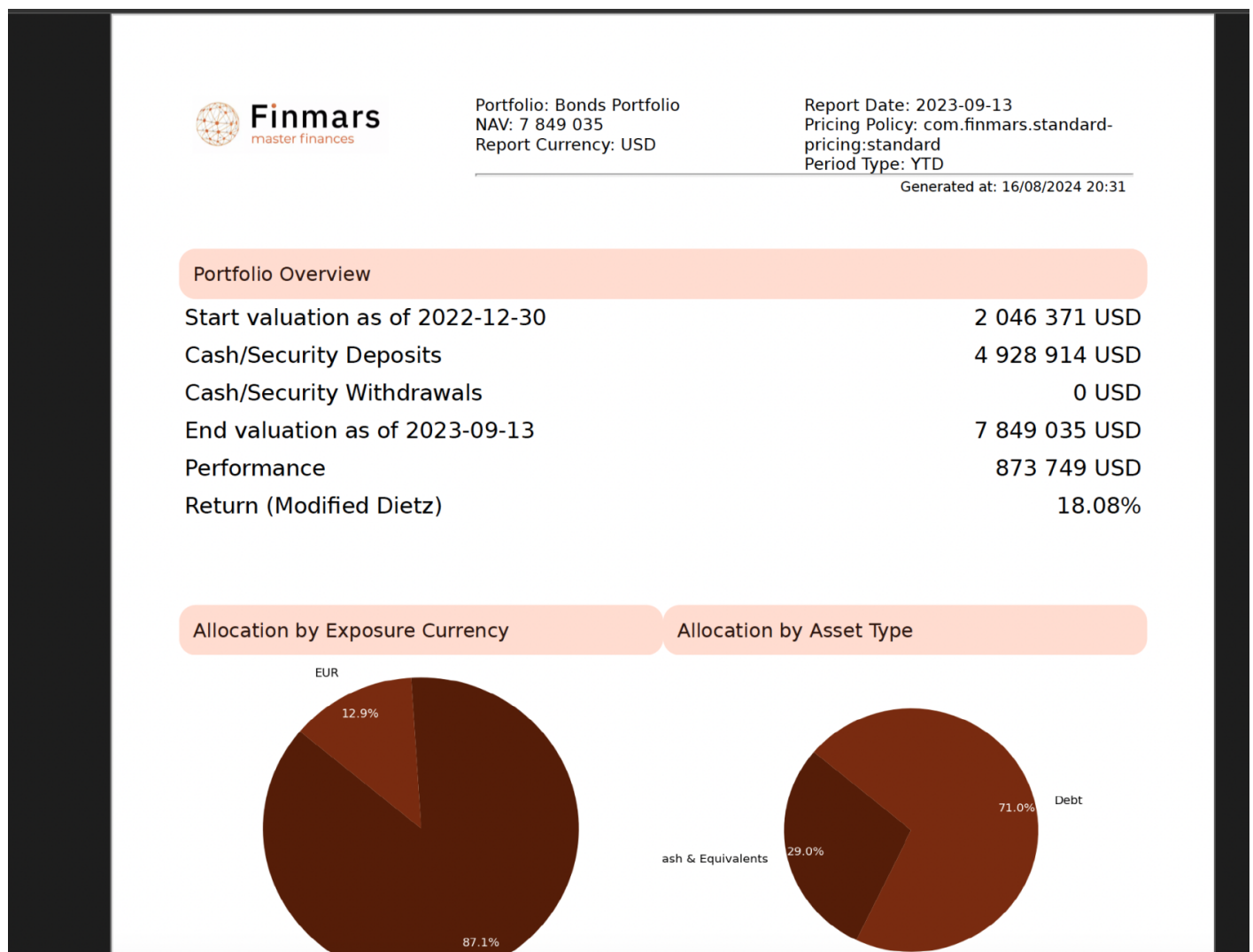
let targetUrl = base_url + url_path
```

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



Revision #3

Created 16 August 2024 20:32:56 by Sergei Zhitenev

Updated 16 August 2024 20:41:58 by Sergei Zhitenev