

MrmProfitLossComponent

August 17, 2026

MrmProfitLossComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-profit-loss/mrm-profit-loss.component.ts (line 79)

Part of: [Frontend](#)

`MrmProfitLossComponent` is an Angular dashboard widget that prepares and displays profit-and-loss data for the MRM dashboard. It checks feature access, loads or initializes dashboard data, calculates year and month totals, formats prices, and saves dashboard state.

Extends: `BaseWidgetComponent`

Methods

Method	Signature	Returns
<code>checkFeatureSubscription</code>	<code>checkFeatureSubscription()</code>	<code>boolean</code>
<code>ngOnInit</code>	<code>ngOnInit()</code>	<code>void</code>
<code>ngOnChanges</code>	<code>ngOnChanges()</code>	<code>void</code>
<code>getDashboardData</code>	<code>getDashboardData()</code>	<code>void</code>
<code>initializingData</code>	<code>initializingData(value: BranchData[])</code>	<code>void</code>
<code>saveDashboardData</code>	<code>saveDashboardData()</code>	<code>void</code>
<code>calculateSumByYear</code>	<code>calculateSumByYear()</code>	<code>void</code>
<code>calculateSumByMonth</code>	<code>calculateSumByMonth(monthIndex: number)</code>	<code>void</code>
<code>translateMonthName</code>	<code>translateMonthName(month: BranchData)</code>	<code>string</code>
<code>getpriceFormatted</code>	<code>getpriceFormatted(x: undefined)</code>	<code>string</code>
<code>onResize</code>	<code>onResize()</code>	<code>void</code>

Method	Signature	Returns
setChaching	setChaching()	void
getChaching	getChaching()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: BranchData, bIndex: number, propertyName: string)	string
comparingBookingWithPrevSumm	comparingBookingWithPrevSumm(month: MonthsSum, index: number, propertyName: string)	string
isCurrentMonth	isCurrentMonth(month: MonthData)	boolean
getpriceForecastAmount	getpriceForecastAmount(month: MonthData)	number
expectedCurrencyChange	expectedCurrencyChange(e: undefined)	void
selectText	selectText(e: undefined)	void
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthData, supplierId: number)	void
openPopup	openPopup(n: string, t: string, i: string, btnTxt: string)	void
savePopup	savePopup()	void
ClosewithDonotSave	ClosewithDonotSave()	void
OnOpenedPopup	OnOpenedPopup()	void
closePopup	closePopup()	void

Properties

Property	Type
sharedPopup	SharedPopupComponent
_widget	Widget
_widgetType	string
yearId	number
periodId	number
countOfEditablePrevMonths	number

Property	Type
withoutOptionBar	boolean
editMode	boolean
bookingOrSales	number
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetEditEnd	any
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]
originalData	BranchData[]
selectedData	BranchData[]
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number
yearBooking	number
totalForecasts	number

Property	Type
tempTotalTarget	TotalTargets
totalProfitAndLossAmount	number
totalOthers	number
totalTotalSGAndA	number
totalFORX	number
totalTotal	number
totalCumulative	number
totalPercentageUsedBudget	number
popup	any
searchText	string
pastDiff	number
urrenFeature	number

Where it refuses work

- `MrmProfitLossComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `MrmProfitLossComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmProfitLossComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex - 1][propertyName]` .
- `MrmProfitLossComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .
- `MrmProfitLossComponent` stops the work with an early return when `AppSettings.calculateMonthsBeforeToday(month.YearId, month.MonthId - 1) == 0 || AppSetting...` .

Diagram

```
graph LR
  A[Angular lifecycle] --> B[checkFeatureSubscription]
  A --> C[getDashboardData]
  C --> D[initializingData]
  D --> E[calculateSumByYear]
```

```
D --> F[calculateSumByMonth]
E --> G[translateMonthName]
F --> H[getpriceFormatted]
D --> I[saveDashboardData]
```

Usage

```
import { MrmProfitLossComponent } from './mrm-profit-loss.component';

const component = new MrmProfitLossComponent();

// Angular normally calls lifecycle hooks.
component.ngOnInit();
component.ngOnChanges();

if (component.checkFeatureSubscription()) {
  component.getDashboardData();
  component.calculateSumByYear();
  component.calculateSumByMonth();

  const monthName = component.translateMonthName();
  const formattedPrice = component.getpriceFormatted();

  component.saveDashboardData();
}
```

AI Coding Instructions

- Keep dashboard loading and persistence inside `getDashboardData`, `initializingData`, and `saveDashboardData` rather than duplicating data setup in lifecycle hooks.
- Check `checkFeatureSubscription()` before requesting or rendering feature-specific profit-and-loss data.
- Recalculate totals through `calculateSumByYear()` and `calculateSumByMonth()` after changing dashboard data.
- Preserve the existing `getpriceFormatted()` method name when calling or updating formatting logic.
- Handle input-driven updates in `ngOnChanges()` and initial setup in `ngOnInit()`.

Used by

3 references from 3 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (3)

- DashboardTabComponent — Frontend/src/app/components/dashboard/dashboard-library/dashboard-tab/dashboard-tab.component.ts :1
- DashboardWidgetsModule — Frontend/src/app/components/dashboard/dashboard-widgets.module.ts :1
- ReportsTabComponent — Frontend/src/app/components/reports/reports-library/reports-tab/reports-tab.component.ts :1