

MrmNetProfitComponent

August 18, 2026

MrmNetProfitComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-net-profit/mrm-net-profit.component.ts (line 93)

Part of: [Pams](#)

`MrmNetProfitComponent` is a dashboard widget component for loading, initializing, saving, and summarizing MRM net-profit data. It checks authorization access, responds to Angular lifecycle events, calculates totals by year or month, and translates month names for display.

Methods

Method	Signature	Returns
<code>GetAuthorizationAccess</code>	<code>GetAuthorizationAccess(url: string)</code>	<code>void</code>
<code>ngOnInit</code>	<code>ngOnInit()</code>	<code>void</code>
<code>ngOnChanges</code>	<code>ngOnChanges()</code>	<code>void</code>
<code>ngOnDestroy</code>	<code>ngOnDestroy()</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
withoutOptionBar	boolean
editMode	boolean
bookingOrSales	number

Property	Type
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEditing	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
subscription	Subscription
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
totalOthers	number

Property	Type
totalDepreciationAmount	number
totalTotalCogsAmount	number
totalTotalCogsAchieved	number
totalTotalNetProfitAmount	number
totalTotalNetProfitAchieved	number
totalTotalDepreciationAmount	number
totalTotalDepreciationAchieved	number
totalTotalFORXAmount	number
totalTotalFORXAchieved	number
totalTotalEBITAmount	number
totalIncomeTAXAmount	number
totalTotalEBITDAAmount	number
totalTotalEBITDAAchieved	number
totalTotalEBITAmount	number
totalTotalEBITDAAchieved	number
totalTotalSalesAmount	number
totalTotalSalesAchieved	number

Where it refuses work

- `MrmNetProfitComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmNetProfitComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex - 1][propertyName]` .
- `MrmNetProfitComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .
- `MrmNetProfitComponent` stops the work with an early return when `this.allowedtoEnableEdting` .
- `MrmNetProfitComponent` stops the work with an early return when `this.allowedtoEdit` .

Diagram

```
graph LR
  Angular[Angular lifecycle] --> Init[ngOnInit]
```

```

Angular --> Changes[ngOnChanges]
Angular --> Destroy[ngOnDestroy]

Init --> Access[GetAuthorizationAccess]
Access --> Load[getDashboardData]
Load --> Initialize[initializingData]

Initialize --> YearTotal[calculateSumByYear]
Initialize --> MonthTotal[calculateSumByMonth]
Initialize --> MonthLabel[translateMonthName]

Widget[Dashboard interaction] --> Save[saveDashboardData]

```

Usage

```

import { Component, ViewChild } from '@angular/core';
import { MrmNetProfitComponent } from './mrm-net-profit.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-mrm-net-profit #netProfitWidget></app-mrm-net-profit>

    <button type="button" (click)="refreshNetProfit()">
      Refresh net profit
    </button>

    <button type="button" (click)="saveNetProfit()">
      Save dashboard settings
    </button>
  `
})
export class DashboardPageComponent {
  @ViewChild('netProfitWidget')
  netProfitWidget!: MrmNetProfitComponent;

  refreshNetProfit(): void {
    this.netProfitWidget.getDashboardData();
  }

  saveNetProfit(): void {
    this.netProfitWidget.saveDashboardData();
  }
}

```

AI Coding Instructions

- Let Angular invoke `ngOnInit` , `ngOnChanges` , and `ngOnDestroy` ; do not call lifecycle hooks from feature code.
- Check `GetAuthorizationAccess()` before adding data-loading behavior that exposes dashboard values.
- Keep data preparation in `initializingData()` after `getDashboardData()` so calculated values use loaded data.
- Use `calculateSumByYear()` and `calculateSumByMonth()` for dashboard totals rather than duplicating aggregation logic in templates.
- Keep month display formatting in `translateMonthName()` so month labels remain consistent across the widget.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `TranslateHandler` — `Pams/Web/Pams.Web/Client/app/app.module.ts` :1