

MrmCogsComponent

August 18, 2026

MrmCogsComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-cogs/mrm-cogs.component.ts (line 84)

Part of: [Pams](#)

`MrmCogsComponent` is an Angular dashboard widget for displaying and managing MRM cost-of-goods-sold data. It checks authorization access, loads and saves dashboard data, aggregates values 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: MonthData)</code>	<code>string</code>
<code>getpriceFormatted</code>	<code>getpriceFormatted(x: undefined)</code>	<code>string</code>
<code>onResize</code>	<code>onResize()</code>	<code>void</code>
<code>setChaching</code>	<code>setChaching()</code>	<code>void</code>

Method	Signature	Returns
getChaching	getChaching()	void
switchWithTaxableHandler	switchWithTaxableHandler()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: BranchData, bIndex: number, monthIndex: 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
mouseoverRowSpan	mouseoverRowSpan(e: undefined, bIndex: number)	void
mouseoutRowSpan	mouseoutRowSpan(e: undefined, bIndex: number)	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

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

Property	Type
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
allGroupsTotalSales	number
totalPercentageOfAchievedSales	number
totalCumulativeSales	number
totalDirectSales	number
totalTotalDirectSales	number
totalDirectService	number
totalDirectSalesWithScrap	number
totalTotalDirectSalesWithScrap	number
totalCommission	number
_totalSalesAfterCommission	number
popup	any
searchText	string

Where it refuses work

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

Diagram

```
graph LR
  A[Angular Lifecycle] --> B[GetAuthorizationAccess]
```

```

A --> C[ngOnInit]
A --> D[ngOnChanges]
A --> E[ngOnDestroy]

C --> F[getDashboardData]
C --> G[initializingData]
D --> F

F --> H[calculateSumByYear]
F --> I[calculateSumByMonth]
I --> J[translateMonthName]

G --> K[saveDashboardData]

```

Usage

```

import { Component, ViewChild } from '@angular/core';
import { MrmCogsComponent } from './mrm-cogs.component';

@Component({
  selector: 'app-dashboard-host',
  template: `
    <app-mrm-cogs #mrmCogs></app-mrm-cogs>

    <button type="button" (click)="refreshCogsData()">
      Refresh COGS data
    </button>
  `,
})
export class DashboardHostComponent {
  @ViewChild('mrmCogs')
  mrmCogsComponent?: MrmCogsComponent;

  refreshCogsData(): void {
    this.mrmCogsComponent?.getDashboardData();
  }

  saveCogsData(): void {
    this.mrmCogsComponent?.saveDashboardData();
  }
}

```

AI Coding Instructions

- Keep data loading inside `getDashboardData()` and initialize widget state through `initializingData()` .

- Run authorization checks through `GetAuthorizationAccess()` before exposing or updating dashboard data.
- Preserve Angular lifecycle handling in `ngOnInit()`, `ngOnChanges()`, and `ngOnDestroy()` when adding subscriptions or input-driven refresh behavior.
- Use `calculateSumByYear()` and `calculateSumByMonth()` for aggregation rather than duplicating calculation logic in templates or parent components.
- Route month labels through `translateMonthName()` so displayed month names remain consistent with the component's translation behavior.

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)

- `TranslateHandler` — `Pams/Web/Pams.Web/Client/app/app.module.ts :1`
- `DashboardTabComponent` — `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/dashboard-tab/dashboard-tab.component.ts :1`
- `ReportsTabComponent` — `Pams/Web/Pams.Web/Client/app/components/dashboard/reports-library/reports-tab/reports-tab.component.ts :1`