

MrmTotalSalesComponent

August 17, 2026

MrmTotalSalesComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-total-sales/mrm-total-sales.component.ts (line 97)

Part of: [Pams](#)

`MrmTotalSalesComponent` is an Angular dashboard widget that retrieves and saves total-sales dashboard data, then calculates totals by year and month. It participates in Angular lifecycle hooks, handles authorization access, initializes widget state, 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>

Method	Signature	Returns
setChaching	setChaching()	void
getChaching	getChaching()	void
switchWithTaxableHandler	switchWithTaxableHandler()	void
switchVendorTypeIdHandler	switchVendorTypeIdHandler()	void
onEnumIdChanged	onEnumIdChanged(e: undefined)	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, 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
sorting	sorting(propertyName: string, propertyType: string, monthIndex: number)	void
syncScroll	syncScroll(e: undefined)	void
openPopup	openPopup(n: string, t: string, i: string, btnTxt: string)	void
savePopup	savePopup()	void
ClosewithDonotSave	ClosewithDonotSave()	void
OnOpenedPopup	OnOpenedPopup()	void

Method	Signature	Returns
closePopup	closePopup()	void

Properties

Property	Type
sharedPopup	SharedPopupComponent
scrollOne	ElementRef
_widget	Widget
_widgetType	string
withoutOptionBar	boolean
editMode	boolean
bookingOrSales	number
withNominal	boolean
withTaxable	boolean
withTaxableBackup	boolean
isPamsData	boolean
enumId	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetEditEnd	any
forecastRules	any[]
calcMrmSalesBy	any
selectedYear	number
viewByDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]

Property	Type
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
allGroupsTotalSales	number
totalSalesAfterCommission	number
totalPercentageOfAchievedSales	number
totalDirectService	number
totalDirectSalesWithoutScrap	number
totalDirectSalesWithScrap	number
totalTotalDirectSalesWithScrap	number
totalCommission	number
totalCumulativeSales	number
popup	any
searchText	string
fixedColumnLeftNumOne	number

Where it refuses work

- `MrmTotalSalesComponent` stops the work with an early return when `a[propertyName] < b[propertyName]` , in 2 places.
- `MrmTotalSalesComponent` stops the work with an early return when `a[propertyName] > b[propertyName]` , in 2 places.
- `MrmTotalSalesComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmTotalSalesComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex].MrmTotalSalesGroups[monthIndex - 1][propertyName]` .
- `MrmTotalSalesComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .
- `MrmTotalSalesComponent` stops the work with an early return when `this.allowedtoEnableEditing` .

Diagram

```
graph LR
    A[Angular Lifecycle] --> B[ngOnInit]
    A --> C[ngOnChanges]
    A --> D[ngOnDestroy]

    B --> E[GetAuthorizationAccess]
    B --> F[initializingData]
    F --> G[getDashboardData]

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

    K[Dashboard Configuration] --> L[saveDashboardData]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { MrmTotalSalesComponent } from './mrm-total-sales.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-mrm-total-sales #totalSalesWidget></app-mrm-total-sales>
    <button type="button" (click)="refreshSalesData()">
      Refresh sales data
    </button>
  `
})
```

```

})
export class DashboardPageComponent {
  @ViewChild('totalSalesWidget')
  totalSalesWidget!: MrmTotalSalesComponent;

  refreshSalesData(): void {
    this.totalSalesWidget.getDashboardData();
  }

  saveWidgetSettings(): void {
    this.totalSalesWidget.saveDashboardData();
  }
}

```

AI Coding Instructions

- Keep initialization logic inside `ngOnInit()` and react to updated bound values through `ngOnChanges()`.
- Preserve the existing `GetAuthorizationAccess()` flow when adding dashboard data requests or save actions.
- Use `initializingData()` for resetting or preparing widget state before processing dashboard results.
- Keep year and month aggregation logic in `calculateSumByYear()` and `calculateSumByMonth()` rather than duplicating calculations in templates.
- Clean up subscriptions, timers, or other retained resources in `ngOnDestroy()`.

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`