

MrmTotalSalesComponent

August 18, 2026

MrmTotalSalesComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-total-sales/mrm-total-sales.component.ts (line 97)

Part of: Frontend

MrmTotalSalesComponent is an Angular dashboard widget that loads, initializes, and saves total-sales dashboard data. It checks feature subscription access, aggregates sales values by year or month, and formats price and month labels for display.

Extends: BaseWidgetComponent

Implements: OnChanges

Methods

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

Method	Signature	Returns
onResize	onResize()	void
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

Method	Signature	Returns
OnOpenedPopup	OnOpenedPopup()	void
closePopup	closePopup()	void

Properties

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

Property	Type
viewByDatasource	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
tempTotalTarget	TotalTargets
allGroupsTotalSales	number
totalSalesAfterCommission	number
totalPercentageOfAchievedSales	number
totalDirectService	number
totalDirectSalesWithoutScrap	number
totalDirectSalesWithScrap	number
totalTotalDirectSalesWithScrap	number
totalCommission	number
totalCumulativeSales	number
popup	any

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 `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `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]` .

Diagram

```
graph LR
  Lifecycle[Angular lifecycle] --> Init[ngOnInit / ngOnChanges]
  Init --> Subscription[checkFeatureSubscription]
  Subscription --> Data[getDashboardData]
  Data --> Initialize[initializingData]
  Initialize --> Year[calculateSumByYear]
  Initialize --> Month[calculateSumByMonth]
  Year --> Format[getpriceFormatted]
  Month --> Translate[translateMonthName]
  Data --> Save[saveDashboardData]
```

Usage

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

@Component({
  selector: 'app-dashboard-host',
  template: `
    <app-mrm-total-sales></app-mrm-total-sales>

    <button type="button" (click)="refreshSales()">
      Refresh sales data
    </button>
  `,
})
```

```
export class DashboardHostComponent {
  @ViewChild(MrmTotalSalesComponent)
  salesWidget!: MrmTotalSalesComponent;

  refreshSales(): void {
    if (this.salesWidget.checkFeatureSubscription()) {
      this.salesWidget.getDashboardData();
    }
  }
}
```

AI Coding Instructions

- Keep subscription checks ahead of dashboard data requests when adding data-loading paths.
- Preserve the existing `initializingData` method name unless all callers are updated together.
- Recalculate year and month totals after changing the source sales data.
- Use `translateMonthName` and `getpriceFormatted` for display values instead of duplicating formatting logic.
- Keep lifecycle-triggered loading compatible with both `ngOnInit` and `ngOnChanges`.

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`