

MrmGrossMarginComponent

August 18, 2026

MrmGrossMarginComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-gross-margin/mrm-gross-margin.component.ts (line 98)

Part of: Frontend

MrmGrossMarginComponent is an Angular dashboard widget for loading, initializing, calculating, formatting, and saving MRM gross-margin data. It checks feature-subscription access, responds to Angular lifecycle changes, and prepares monthly and yearly totals for display.

Extends: BaseWidgetComponent

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
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void

Method	Signature	Returns
switchWithTaxableHandler	switchWithTaxableHandler()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, bIndex: number, monthIndex: number, propertyName: string)	string
comparingMonthlySummaryWithPrevMonth	comparingMonthlySummaryWithPrevMonth(month: MonthData, monthIndex: number, propertyName: string)	string
comparingMonthlyComulativeWithPrevMonth	comparingMonthlyComulativeWithPrevMonth(month: MonthData, 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
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
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetEditEnd	any
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
newResponse	NewResponse
data	BranchData[]
originalData	BranchData[]
selectedData	BranchData[]
singleMonths	MonthData
branchCurrency	any

Property	Type
basedOnTendency	boolean
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
allGroupsTotalSales	number
totalSalesAfterCommission	number
totalSalesSummary	number
totalDirectService	number
totalDirectSalesWithoutScrap	number
totalDirectSalesWithScrap	number
totalTotalDirectSalesWithScrap	number
totalCommission	number
totalCumulativeSales	number
popup	any
searchText	string
fixedColumnLeftNumOne	number

Where it refuses work

- `MrmGrossMarginComponent` stops the work with an early return when `a[propertyName] < b[propertyName]` , in 2 places.
- `MrmGrossMarginComponent` stops the work with an early return when `a[propertyName] > b[propertyName]` , in 2 places.
- `MrmGrossMarginComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `MrmGrossMarginComponent` stops the work with an early return when `month.MonthId == 12` .

- `MrmGrossMarginComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex].MrmCogsGroups[monthIndex - 1][propertyName]` .
- `MrmGrossMarginComponent` stops the work with an early return when `month[propertyName] < this.newResponse.MonthlySummary[monthIndex - 1][propertyName]` .

Diagram

```
graph LR
  A[Angular lifecycle] --> B[MrmGrossMarginComponent]
  B --> C[checkFeatureSubscription]
  B --> D[getDashboardData]
  D --> E[initializingData]
  E --> F[calculateSumByYear]
  E --> G[calculateSumByMonth]
  F --> H[getpriceFormatted]
  G --> H
  B --> I[saveDashboardData]
  B --> J[translateMonthName]
```

Usage

```
import { MrmGrossMarginComponent } from './mrm-gross-margin.component';

const component = new MrmGrossMarginComponent();

// Angular calls lifecycle hooks when the component is rendered or updated.
component.ngOnInit();
component.ngOnChanges();

if (component.checkFeatureSubscription()) {
  component.getDashboardData();
  component.initializingData();

  component.calculateSumByYear();
  component.calculateSumByMonth();

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

  component.saveDashboardData();

  console.log({ monthLabel, formattedPrice });
}
```

AI Coding Instructions

- Call `checkFeatureSubscription()` before loading or saving gross-margin dashboard data.
- Keep data-loading logic in `getDashboardData()` and data preparation in `initializingData()` .

- Recalculate monthly and yearly totals after changing the underlying gross-margin data.
- Use `translateMonthName()` for displayed month labels instead of duplicating month-name conversion logic.
- Use `getpriceFormatted()` when rendering gross-margin values in the widget UI.

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)

- `DashboardWidgetsModule` — `Frontend/src/app/components/dashboard/dashboard-widgets.module.ts :1`