

MrmFinanceLgComponent

August 17, 2026

MrmFinanceLgComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-finance-lg/mrm-finance-lg.component.ts (line 72)

Part of: Frontend

MrmFinanceLgComponent is an Angular dashboard widget that loads and presents finance-related data for subscribed users. It manages lifecycle updates, calculates performance values, formats price and month display values, responds to resize events, and stores cached widget state.

Extends: BaseWidgetComponent

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
ngOnChanges	ngOnChanges()	void
getDashboardData	getDashboardData()	void
initializingData	initializingData(value: BranchData[])	void
calculatePerformancePer	calculatePerformancePer(month: MonthData)	number
translateMonthName	translateMonthName(month: MonthData)	string
getpriceFormatted	getpriceFormatted(x: undefined)	string
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void

Method	Signature	Returns
onValueChanged	onValueChanged(e: undefined)	void
onChartInit	onChartInit(e: any, index: number)	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
mouseoverRowSpan	mouseoverRowSpan(e: undefined, bIndex: number)	void
mouseoutRowSpan	mouseoutRowSpan(e: undefined, bIndex: number)	void
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, bIndex: number, monthIndex: number, propertyName: string)	string
comparingBookingWithPrevSumm	comparingBookingWithPrevSumm(month: TotalBookingMonths, index: number)	string
comparingMonthsWithPrevSumm	comparingMonthsWithPrevSumm(month: TotalBookingMonths, index: number, propertyName: string)	string
setValue	setValue(value: number)	void
setForecastValue	setForecastValue(value: number)	void
setMarginAvarage	setMarginAvarage(value: number)	void
isForecast	isForecast(month: MonthData)	boolean
isAchievedOrCurrent	isAchievedOrCurrent(month: MonthData)	boolean
getpriceForecastAmount	getpriceForecastAmount(month: MonthData)	number
expectedCurrencyChange	expectedCurrencyChange(e: undefined)	void

Properties

Property	Type
_widget	Widget
_widgetType	string
yearId	number
periodId	number

Property	Type
countOfEditablePrevMonths	number
withoutOptionBar	boolean
bookingOrSales	number
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]
originalData	BranchData[]
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	TotalBookingMonths[]
tempTotalMonth	TotalBookingMonths
pastDiff	number
urrenFeature	number

Where it refuses work

- `MrmFinanceLgComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .

- `MrmFinanceLgComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmFinanceLgComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex].MrmlGs[monthIndex - 1][propertyName]` .
- `MrmFinanceLgComponent` stops the work with an early return when `month.TotalAmountMonth < this.totalBookingMonths[index - 1].TotalAmountMonth` .
- `MrmFinanceLgComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .
- `MrmFinanceLgComponent` stops the work with an early return when `month.MonthId >= new Date().getMonth() + 1` .

Diagram

```
graph LR
  A[Angular lifecycle] --> B[MrmFinanceLgComponent]
  B --> C[checkFeatureSubscription]
  C --> D[getDashboardData]
  D --> E[initializingData]
  E --> F[calculatePerformancePer]
  E --> G[translateMonthName]
  E --> H[getpriceFormatted]
  B --> I[onResize]
  B --> J[setChaching]
```

Usage

```
import { Component, ViewChild, ViewContainerRef } from '@angular/core';
import { MrmFinanceLgComponent } from './mrm-finance-lg.component';

@Component({
  selector: 'app-dashboard-host',
  template: '<ng-container #widgetHost></ng-container>',
})
export class DashboardHostComponent {
  @ViewChild('widgetHost', { read: ViewContainerRef })
  widgetHost!: ViewContainerRef;

  loadFinanceWidget(): void {
    const widgetRef = this.widgetHost.createComponent(MrmFinanceLgComponent);

    if (widgetRef.instance.checkFeatureSubscription()) {
      widgetRef.instance.getDashboardData();
    }
  }
}
```

AI Coding Instructions

- Check `checkFeatureSubscription()` before requesting or displaying finance data.
- Let Angular call `ngOnInit()` and `ngOnChanges()` through the component lifecycle; do not call lifecycle hooks directly from application code.
- Keep data preparation in `initalizingData()` so calculated, translated, and formatted values stay consistent.
- Use `calculatePerformancePer()`, `translateMonthName()`, and `getpriceFormatted()` for display values instead of duplicating formatting logic in templates.
- Preserve the existing `setChaching()` method name when updating related callers, even though its spelling differs from “caching.”

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`