

MrmCashPositionComponent

August 18, 2026

MrmCashPositionComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-cash-position/mrm-cash-position.component.ts (line 89)

Part of: Frontend

MrmCashPositionComponent is an Angular dashboard widget that retrieves and saves cash-position dashboard data, initializes its working state, and calculates monthly totals and performance. It checks feature subscription access, loads currency rates, and translates month names for dashboard display during lifecycle updates.

Extends: BaseWidgetComponent

Methods

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

Method	Signature	Returns
syncScroll	syncScroll(e: undefined)	void
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void
onValueChanged	onValueChanged(e: undefined)	void
onChartInit	onChartInit(e: any, index: number)	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
isCurrentMonth	isCurrentMonth(month: MonthData)	boolean
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, bIndex: number, index: number, propertyName: string)	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
selectText	selectText(e: undefined)	void
mouseoverRowSpan	mouseoverRowSpan(e: undefined, bIndex: number)	void
mouseoutRowSpan	mouseoutRowSpan(e: undefined, bIndex: number)	void
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthData)	void
sorting	sorting(propertyName: string, propertyType: string, monthIndex: number)	void

Properties

Property	Type
scrollOne	ElementRef
scrollTwo	ElementRef
scrollThree	ElementRef
_widget	Widget
_widgetType	string
yearId	number
periodId	number
countOfEditablePrevMonths	number
withoutOptionBar	boolean
editMode	boolean
bookingOrSales	number
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetEditEnd	any
selectedData	BranchData[]
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]
originalData	BranchData[]
currencyRateData	currencyRatesData[]

Property	Type
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean
SerializeRange	any[]
echartsIntance	any
totalBookingMonths	TotalBookingMonths[]
tempTotalMonth	TotalBookingMonths
pastDiff	number
urrenFeature	number
sortingDirection	number

Where it refuses work

- `MrmCashPositionComponent` stops the work with an early return when `a[propertyName] < b[propertyName]` , in 2 places.
- `MrmCashPositionComponent` stops the work with an early return when `a[propertyName] > b[propertyName]` , in 2 places.
- `MrmCashPositionComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `MrmCashPositionComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmCashPositionComponent` stops the work with an early return when `AppSettings.calculateMonthsBeforeToday(month.YearId, month.MonthId - 1) == 0 || AppSetting...` .
- `MrmCashPositionComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex].MrmCashPositions[index - 1][propertyName]` .

Diagram

```
graph LR
  A[Angular Lifecycle] --> B[MrmCashPositionComponent]
  B --> C[checkFeatureSubscription]
  B --> D[ngOnInit]
  B --> E[ngOnChanges]
```

```

D --> F[getDashboardData]
D --> G[initializingData]
G --> H[getCurrencyRates]
G --> I[calculateSumByMonth]
I --> J[calculatePerformancePer]
B --> K[translateMonthName]
B --> L[saveDashboardData]

```

Usage

```

import { Component, ViewChild } from '@angular/core';
import { MrmCashPositionComponent } from './mrm-cash-position.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-mrm-cash-position></app-mrm-cash-position>

    <button type="button" (click)="refreshCashPosition()">
      Refresh cash position
    </button>
  `,
})
export class DashboardPageComponent {
  @ViewChild(MrmCashPositionComponent)
  cashPositionWidget!: MrmCashPositionComponent;

  refreshCashPosition(): void {
    if (!this.cashPositionWidget.checkFeatureSubscription()) {
      return;
    }

    this.cashPositionWidget.getDashboardData();
    this.cashPositionWidget.calculateSumByMonth();

    const performance = this.cashPositionWidget.calculatePerformancePer();
    console.log('Cash position performance:', performance);
  }
}

```

AI Coding Instructions

- Keep subscription checks in place before loading or displaying cash-position data.
- Preserve the Angular lifecycle flow in `ngOnInit` and `ngOnChanges`; avoid duplicating initialization work outside the component lifecycle.

- Load currency rates before calculations that depend on converted monetary values.
- Keep the existing `initializingData` method name when updating call sites, since changing it can break references.
- Save dashboard state through `saveDashboardData` after modifying data that should persist between dashboard updates.

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`