

MrmCurrencyRateComponent

August 18, 2026

MrmCurrencyRateComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-currency-rate/mrm-currency-rate.component.ts (line 92)

Part of: Frontend

MrmCurrencyRateComponent is a dashboard widget component for loading, initializing, calculating, and saving currency-rate dashboard data. It checks feature subscription access, responds to Angular lifecycle changes, and formats price and month values 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

Method	Signature	Returns
setChaching	setChaching()	void
getChaching	getChaching()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: BranchData, 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
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthData, supplierId: number)	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
_widget	Widget
_widgetType	string
yearId	number
periodId	number

Property	Type
countOfEditablePrevMonths	number
withoutOptionBar	boolean
editMode	boolean
bookingOrSales	number
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetEditEnd	any
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]
originalData	BranchData[]
selectedData	BranchData[]
singleMonths	MonthData
monthsSummery	MonthsSum[]
branchCurrency	any
basedOnTendency	boolean
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number

Property	Type
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
totalPercentageUsedBudget	number
popup	any
searchText	string
pastDiff	number
urrenFeature	number

Where it refuses work

- `MrmCurrencyRateComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `MrmCurrencyRateComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmCurrencyRateComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex].Details[monthIndex - 1][propertyName]` .
- `MrmCurrencyRateComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .
- `MrmCurrencyRateComponent` stops the work with an early return when `AppSettings.calculateMonthsBeforeToday(month.YearId, month.MonthId - 1) == 0 || AppSetting...` .

Diagram

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

Usage

```
import { Component, ViewChild } from '@angular/core';
import { MrmCurrencyRateComponent } from '../mrm-currency-rate/mrm-currency-rate.component';

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

    <button (click)="refreshCurrencyRates()">
      Refresh currency rates
    </button>
  `,
})
export class DashboardPageComponent {
  @ViewChild(MrmCurrencyRateComponent)
  currencyRateComponent?: MrmCurrencyRateComponent;

  refreshCurrencyRates(): void {
    const widget = this.currencyRateComponent;

    if (!widget?.checkFeatureSubscription()) {
      return;
    }

    widget.getDashboardData();

    const formattedPrice = widget.getPriceFormatted();
    console.log('Formatted currency rate:', formattedPrice);
  }
}
```

AI Coding Instructions

- Check `checkFeatureSubscription()` before loading or updating currency-rate dashboard data.
- Keep initialization work in `ngOnInit()` and react to Angular input updates through `ngOnChanges()`.
- Use `initializingData()` as the setup path before calculations or persistence when the component state requires initialization.
- Keep year and month aggregation logic in `calculateSumByYear()` and `calculateSumByMonth()` rather than duplicating calculations in parent dashboard components.

- Use `translateMonthName()` and `getpriceFormatted()` for display values so formatting remains consistent with the widget.

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`