

MrmBookingSupplierComponent

August 18, 2026

MrmBookingSupplierComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-booking-supplier/mrm-booking-supplier.component.ts (line 45)

Part of: Frontend

MrmBookingSupplierComponent is an Angular dashboard widget for displaying supplier booking data. It checks feature access, loads and initializes dashboard data, formats display values, and keeps scroll behavior aligned during resize and content updates.

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
syncScroll	syncScroll(e: undefined)	void
onResize	onResize()	void
setChaching	setChaching()	void

Method	Signature	Returns
getChaching	getChaching()	void
onValueChanged	onValueChanged(e: undefined)	void
onChartInit	onChartInit(e: any, index: number)	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, bIndex: number, index: number)	string
comparingBookingWithPrevSumm	comparingBookingWithPrevSumm(month: TotalBookingMonths, index: number)	string
comparingForecastWithPrevSumm	comparingForecastWithPrevSumm(month: TotalBookingMonths, index: number)	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
sorting	sorting(propertyName: string, propertyType: string, monthIndex: number)	void

Properties

Property	Type
scrollOne	ElementRef
scrollTwo	ElementRef
scrollThree	ElementRef
_widget	Widget
_widgetType	string

Property	Type
yearId	number
periodId	number
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
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets

Property	Type
searchText	string
urrenFeature	number
sortingDirection	number

Where it refuses work

- `MrmBookingSupplierComponent` stops the work with an early return when `a[propertyName] < b[propertyName]` , in 2 places.
- `MrmBookingSupplierComponent` stops the work with an early return when `a[propertyName] > b[propertyName]` , in 2 places.
- `MrmBookingSupplierComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `MrmBookingSupplierComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmBookingSupplierComponent` stops the work with an early return when `month.TotalValue < this.data[bIndex].MrmBooking[index - 1].TotalValue` .
- `MrmBookingSupplierComponent` stops the work with an early return when `month.TotalBookingMonth < this.totalBookingMonths[index - 1].TotalBookingMonth` .

Diagram

```
graph LR
    Dashboard[Dashboard] --> Component[MrmBookingSupplierComponent]

    Component --> Subscription[checkFeatureSubscription]
    Component --> Init[ngOnInit]
    Component --> Changes[ngOnChanges]
    Init --> Data[getDashboardData]
    Changes --> Data
    Data --> Initialize[initializingData]

    Initialize --> Performance[calculatePerformancePer]
    Initialize --> Month[translateMonthName]
    Initialize --> Price[getpriceFormatted]

    Component --> Scroll[syncScroll]
    Component --> Resize[onResize]
    Resize --> Scroll
```

Usage

```
import { AfterViewInit, Component, ViewChild } from '@angular/core';
import { MrmBookingSupplierComponent } from './mrm-booking-supplier.component';

@Component({
  selector: 'app-dashboard-host',
  template: `
    <!-- Render the booking supplier widget in this dashboard view. -->
    `
})
export class DashboardHostComponent implements AfterViewInit {
  @ViewChild(MrmBookingSupplierComponent)
  bookingSupplierWidget?: MrmBookingSupplierComponent;

  ngAfterViewInit(): void {
    const widget = this.bookingSupplierWidget;

    if (widget?.checkFeatureSubscription()) {
      widget.getDashboardData();
    }
  }
}
```

AI Coding Instructions

- Let Angular call `ngOnInit` and `ngOnChanges` ; keep data-loading and initialization logic aligned with these lifecycle hooks.
- Check `checkFeatureSubscription()` before adding interactions that require the booking supplier feature.
- Keep display calculations in `calculatePerformancePer` , `translateMonthName` , and `getpriceFormatted` rather than duplicating formatting logic in templates.
- Call `syncScroll()` after changes that affect linked scroll containers, and keep resize handling in `onResize()` .
- Preserve the existing `initializingData()` method name when calling or refactoring it.

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