

MrmBookingSupplierComponent

August 18, 2026

MrmBookingSupplierComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-booking-supplier/mrm-booking-supplier.component.ts (line 45)

Part of: [Pams](#)

`MrmBookingSupplierComponent` is an Angular dashboard widget for displaying MRM booking supplier data. It loads and initializes dashboard state, calculates performance values, formats month and price output, and coordinates scroll and resize behavior.

Methods

Method	Signature	Returns
<code>ngOnInit</code>	<code>ngOnInit()</code>	<code>void</code>
<code>ngOnChanges</code>	<code>ngOnChanges()</code>	<code>void</code>
<code>ngOnDestroy</code>	<code>ngOnDestroy()</code>	<code>void</code>
<code>getDashboardData</code>	<code>getDashboardData()</code>	<code>void</code>
<code>initializingData</code>	<code>initializingData(value: BranchData[])</code>	<code>void</code>
<code>calculatePerformancePer</code>	<code>calculatePerformancePer(month: MonthData)</code>	<code>number</code>
<code>translateMonthName</code>	<code>translateMonthName(month: MonthData)</code>	<code>string</code>
<code>getpriceFormatted</code>	<code>getpriceFormatted(x: undefined)</code>	<code>string</code>
<code>syncScroll</code>	<code>syncScroll(e: undefined)</code>	<code>void</code>
<code>onResize</code>	<code>onResize()</code>	<code>void</code>
<code>setChaching</code>	<code>setChaching()</code>	<code>void</code>
<code>getChaching</code>	<code>getChaching()</code>	<code>void</code>
<code>onValueChanged</code>	<code>onValueChanged(e: undefined)</code>	<code>void</code>

Method	Signature	Returns
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
withoutOptionBar	boolean
bookingOrSales	number

Property	Type
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
subscription	Subscription
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	TotalBookingMonths[]
tempTotalMonth	TotalBookingMonths
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
searchText	string
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 `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` .
- `MrmBookingSupplierComponent` stops the work with an early return when `month.TotalForecastMonth < this.totalBookingMonths[index - 1].TotalForecastMonth` .

Diagram

```
graph LR
    Angular[Angular lifecycle] --> Init[ngOnInit]
    Angular --> Changes[ngOnChanges]
    Angular --> Destroy[ngOnDestroy]

    Init --> Load[getDashboardData]
    Changes --> Load
    Load --> Initialize[initializingData]

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

    Scroll[syncScroll] --> Widget[MrmBookingSupplierComponent]
    Resize[onResize] --> Widget
```

Usage

```
import { TestBed } from '@angular/core/testing';
import { MrmBookingSupplierComponent } from './mrm-booking-supplier.component';

const fixture = TestBed.createComponent(MrmBookingSupplierComponent);
fixture.detectChanges();

const widget = fixture.componentInstance;

// Refresh dashboard data when required by the host view.
widget.getDashboardData();
```

```
// Keep related scroll areas aligned.  
widget.syncScroll();
```

AI Coding Instructions

- Keep dashboard data retrieval inside `getDashboardData()` and state preparation inside `initializingData()`.
- Preserve Angular lifecycle behavior in `ngOnInit()`, `ngOnChanges()`, and `ngOnDestroy()` when adding subscriptions or event listeners.
- Keep display calculations in `calculatePerformancePer()`, `translateMonthName()`, and `getpriceFormatted()` rather than placing formatting logic in templates.
- Update `syncScroll()` and `onResize()` together when changing the widget layout or scrollable DOM elements.

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)

- `TranslateHandler` — `Pams/Web/Pams.Web/Client/app/app.module.ts :1`