

MrmBookingTeamsComponent

August 18, 2026

MrmBookingTeamsComponent

Kind: Class

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

Part of: Frontend

MrmBookingTeamsComponent is an Angular dashboard widget that loads and presents booking-team data. It checks feature subscription access, initializes widget state, calculates performance values, formats display data, and updates cached state when inputs or viewport conditions change.

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
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
sorting	sorting(propertyName: string, propertyType: string, monthIndex: number)	void

Properties

Property	Type
_widget	Widget
_widgetType	string
yearId	number
periodId	number
countOfEditablePrevMonths	number
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
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	TotalBookingMonths[]
tempTotalMonth	TotalBookingMonths
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
urrenFeature	number
sortingDirection	number

Where it refuses work

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

Diagram

```
graph LR
  Dashboard[Dashboard] --> Component[MrmBookingTeamsComponent]
  Component --> Subscription[checkFeatureSubscription]
  Component --> Lifecycle[lifecycle[ngOnInit / ngOnChanges]]
  Lifecycle --> Data[getDashboardData]
  Data --> Initialize[initializingData]
  Initialize --> Performance[calculatePerformancePer]
  Initialize --> Formatting[translateMonthName / getpriceFormatted]
  Component --> Resize[onResize]
  Component --> Cache[setChaching]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { MrmBookingTeamsComponent } from './mrm-booking-teams.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-mrm-booking-teams #bookingTeams></app-mrm-booking-teams>
    <button type="button" (click)="refreshBookingTeams()">
      Refresh
    </button>
  `,
})
export class DashboardPageComponent {
  @ViewChild('bookingTeams')
  bookingTeams?: MrmBookingTeamsComponent;
```

```
refreshBookingTeams(): void {
  if (this.bookingTeams?.checkFeatureSubscription()) {
    this.bookingTeams.getDashboardData();
  }
}
```

AI Coding Instructions

- Keep data loading behind `checkFeatureSubscription()` so unavailable features do not request dashboard data.
- Preserve Angular lifecycle handling in `ngOnInit()` and `ngOnChanges()` when adding input-driven behavior.
- Keep display calculations in `calculatePerformancePer()`, `translateMonthName()`, and `getpriceFormatted()` rather than duplicating formatting in templates.
- Update cache-related behavior through `setChaching()` after changing data initialization or refresh flows.
- Test resize-dependent changes through `onResize()` to avoid stale widget layout or display state.

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`