

MrmBookingCityComponent

August 19, 2026

MrmBookingCityComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-booking-city/mrm-booking-city.component.ts (line 118)

Part of: [Frontend](#)

`MrmBookingCityComponent` is an Angular dashboard widget for presenting booking data by city. It checks feature subscription access, loads and initializes dashboard state, calculates monthly and yearly totals, translates month labels, formats prices, and saves dashboard data. The component responds to Angular lifecycle events through `ngOnInit` and `ngOnChanges`.

Extends: `BaseWidgetComponent`

Methods

Method	Signature	Returns
<code>checkFeatureSubscription</code>	<code>checkFeatureSubscription()</code>	<code>boolean</code>
<code>ngOnInit</code>	<code>ngOnInit()</code>	<code>void</code>
<code>ngOnChanges</code>	<code>ngOnChanges()</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>saveDashboardData</code>	<code>saveDashboardData()</code>	<code>void</code>
<code>calculateSumByYear</code>	<code>calculateSumByYear()</code>	<code>void</code>
<code>calculateSumByMonth</code>	<code>calculateSumByMonth(monthIndex: number)</code>	<code>void</code>
<code>translateMonthName</code>	<code>translateMonthName(month: MonthData)</code>	<code>string</code>
<code>getpriceFormatted</code>	<code>getpriceFormatted(x: undefined)</code>	<code>string</code>

Method	Signature	Returns
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: BranchData, bIndex: number, index: number, propertyName: string)	string
comparingMonthsWithPrevSumm	comparingMonthsWithPrevSumm(month: TotalBookingMonths, 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
mouseoverRowSpan	mouseoverRowSpan(e: undefined, bIndex: number)	void
mouseoutRowSpan	mouseoutRowSpan(e: undefined, bIndex: number)	void
syncScroll	syncScroll(e: undefined)	void
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthData, supplierId: number)	void
sorting	sorting(propertyName: string, propertyType: string, monthIndex: 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
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
data	BranchData[]
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetEditEnd	any
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
originalData	BranchData[]
selectedData	BranchData[]
singleMonths	MonthData

Property	Type
branchCurrency	any
basedOnTendency	boolean
SerializeRange	any[]
echartsIntance	any
totalBookingMonths	TotalBookingMonths[]
tempTotalMonth	TotalBookingMonths
popup	any
searchText	string
urrenFeature	number
sortingDirection	number

Where it refuses work

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

Diagram

```
graph LR
  A[ngOnInit / ngOnChanges] --> B[checkFeatureSubscription]
  B -->|Allowed| C[getDashboardData]
  C --> D[initializingData]
  D --> E[calculateSumByYear]
  D --> F[calculateSumByMonth]
  F --> G[translateMonthName]
```

```
E --> H[getpriceFormatted]
F --> H
D --> I[saveDashboardData]
```

Usage

```
import { MrmBookingCityComponent } from './mrm-booking-city.component';

function refreshBookingCityWidget(
  widget: MrmBookingCityComponent,
): void {
  if (!widget.checkFeatureSubscription()) {
    return;
  }

  widget.getDashboardData();
  widget.calculateSumByYear();
  widget.calculateSumByMonth();

  const monthLabel = widget.translateMonthName();
  const formattedPrice = widget.getpriceFormatted();

  console.log({ monthLabel, formattedPrice });

  widget.saveDashboardData();
}
```

AI Coding Instructions

- Let Angular invoke `ngOnInit` and `ngOnChanges` ; keep lifecycle setup inside those hooks rather than calling them from application code.
- Check `checkFeatureSubscription()` before loading or displaying subscription-gated booking data.
- Keep data loading, initialization, calculation, formatting, and persistence in their existing method flow.
- When changing monthly aggregation, update `calculateSumByMonth()` and `translateMonthName()` together so labels match calculated data.
- Treat `getpriceFormatted()` as display formatting and keep numeric calculation logic in the sum methods.

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`