MrmBookingTeamsComponent

August 18, 2026

MrmBookingTeamsComponent

Kind: Class

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

Part of: Pams

`MrmBookingTeamsComponent` is an Angular dashboard widget for loading and presenting MRM booking team data. It manages lifecycle updates, data initialization, performance calculation, month and price formatting, resize handling, and cached state.

Methods

Method	Signature	Returns
`ngOnInit`	`ngOnInit()`	`void`
`ngOnChanges`	`ngOnChanges()`	`void`
`ngOnDestroy`	`ngOnDestroy()`	`void`
`getDashboardData`	`getDashboardData()`	`void`
`initalizingData`	`initalizingData(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`
`onValueChanged`	`onValueChanged(e: undefined)`	`void`
`onChartInit`	`onChartInit(e: any, index: number)`	`void`

Method	Signature	Returns
`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`
`withoutOptionBar`	`boolean`
`bookingOrSales`	`number`
`withNominal`	`boolean`
`selectedTeam`	`number`
`forecastRuleId`	`number`
`allowedtoEdit`	`boolean`
`allowedtoEnableEdting`	`boolean`

Property	Type
`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`
`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 `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`.
- `MrmBookingTeamsComponent` 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[initalizingData]
    Initialize --> Performance[calculatePerformancePer]
    Initialize --> Month[translateMonthName]
    Initialize --> Price[getpriceFormatted]
    Initialize --> Cache[setChaching]

    Resize[onResize] --> Initialize
```

Usage

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

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

    <button type="button" (click)="refreshBookingTeams()">
      Refresh booking teams
    </button>
  `,
})
export class DashboardPageComponent {
  @ViewChild(MrmBookingTeamsComponent)
  bookingTeamsComponent?: MrmBookingTeamsComponent;

  refreshBookingTeams(): void {
    this.bookingTeamsComponent?.getDashboardData();
```

```
  }
}
```

AI Coding Instructions

- Let Angular call `ngOnInit`, `ngOnChanges`, and `ngOnDestroy`; keep lifecycle-specific setup and cleanup in those methods.
- Call `getDashboardData()` when the widget needs refreshed dashboard data, then keep derived display values initialized through `initalizingData()`.
- Keep formatting logic in `translateMonthName()` and `getpriceFormatted()` so templates do not duplicate display transformations.
- Preserve `setChaching()` calls when changing the data-loading flow so widget state remains available after dashboard updates.
- Re-run layout-related initialization from `onResize()` when adding chart, grid, or responsive display behavior.

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