MrmPmBacklogComponent

August 18, 2026

MrmPmBacklogComponent

Kind: Class

Source: `Frontend/src/app/components/dashboard/dashboard-libary/widgets/mrm-pm-backlog/mrm-pm-backlog.component.ts` (line 44)

Part of: Frontend

`MrmPmBacklogComponent` is an Angular dashboard widget that retrieves and prepares MRM PM backlog data for display. It checks feature subscription access, reacts to lifecycle changes and resizing, calculates performance values, formats price and month labels, and stores cached state.

Extends: `BaseWidgetComponent`

Methods

Method	Signature	Returns
`checkFeatureSubscription`	`checkFeatureSubscription()`	`boolean`
`ngOnInit`	`ngOnInit()`	`void`
`ngOnChanges`	`ngOnChanges()`	`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`

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`

Properties

Property	Type
`_widget`	`Widget`
`_widgetType`	`string`
`yearId`	`number`
`periodId`	`number`
`countOfEditablePrevMonths`	`number`
`withoutOptionBar`	`boolean`
`bookingOrSales`	`number`
`withNominal`	`boolean`
`selectedTeam`	`number`

Property	Type
`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`
`pastDiff`	`number`
`urrenFeature`	`number`

Where it refuses work

- `MrmPmBacklogComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)`.
- `MrmPmBacklogComponent` stops the work with an early return when `month.MonthId == 12`.

- `MrmPmBacklogComponent` stops the work with an early return when `month.TotalValue < this.data[bIndex].MrmBooking[index - 1].TotalValue`.
- `MrmPmBacklogComponent` stops the work with an early return when `month.TotalBookingMonth < this.totalBookingMonths[index - 1].TotalBookingMonth`.
- `MrmPmBacklogComponent` stops the work with an early return when `month.TotalForecastMonth < this.totalBookingMonths[index - 1].TotalForecastMonth`.
- `MrmPmBacklogComponent` stops the work with an early return when `month.MonthId >= new Date().getMonth() + 1`.

Diagram

```
graph LR
  Host[Dashboard Host] --> Component[MrmPmBacklogComponent]
  Component --> Subscription[checkFeatureSubscription]
  Component --> Lifecycle[ngOnInit / ngOnChanges]
  Lifecycle --> Data[getDashboardData]
  Data --> Initialize[initalizingData]
  Initialize --> Performance[calculatePerformancePer]
  Initialize --> Labels[translateMonthName]
  Initialize --> Price[getpriceFormatted]
  Component --> Resize[onResize]
  Data --> Cache[setChaching]
```

Usage

```
import { AfterViewInit, Component, ViewChild } from '@angular/core';
import { MrmPmBacklogComponent } from './mrm-pm-backlog.component';

@Component({
  selector: 'app-dashboard-page',
  template: `<app-mrm-pm-backlog></app-mrm-pm-backlog>`,
})
export class DashboardPageComponent implements AfterViewInit {
  @ViewChild(MrmPmBacklogComponent)
  backlogWidget!: MrmPmBacklogComponent;

  ngAfterViewInit(): void {
    if (!this.backlogWidget.checkFeatureSubscription()) {
      return;
    }

    this.backlogWidget.getDashboardData();

    const performance = this.backlogWidget.calculatePerformancePer();
    const formattedPrice = this.backlogWidget.getpriceFormatted();
```

```
    console.log({ performance, formattedPrice });
  }
}
```

AI Coding Instructions

- Keep subscription checks in place before requesting or displaying backlog data.
- Let Angular call lifecycle hooks; call data-loading methods directly only for explicit refresh behavior.
- Keep display transformations in `calculatePerformancePer`, `translateMonthName`, and `getpriceFormatted` rather than duplicating formatting in templates.
- Update cached state through `setChaching` when changing the data-loading flow.
- Preserve resize handling through `onResize` when modifying chart, layout, or responsive display logic.

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