

MrmPmBacklogComponent

August 17, 2026

MrmPmBacklogComponent

Kind: Class

Source: `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-pm-backlog/mrm-pm-backlog.component.ts` (line 44)

Part of: [Pams](#)

`MrmPmBacklogComponent` is an Angular dashboard widget that loads and prepares PM backlog data for display. It responds to lifecycle changes and window resizing, calculates performance values, formats price and month labels, and manages cached widget state.

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>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

Properties

Property	Type
_widget	Widget
_widgetType	string
withoutOptionBar	boolean
bookingOrSales	number
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
forecastRules	any[]

Property	Type
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

Where it refuses work

- `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` .

- `MrmPmBacklogComponent` stops the work with an early return when `month.MonthId <= new Date().getMonth() + 1`.

Diagram

```
graph LR
    Angular[Angular lifecycle] --> Init[ngOnInit]
    Angular --> Changes[ngOnChanges]
    Init --> Data[getDashboardData]
    Changes --> Data
    Data --> Initialize[initializingData]
    Initialize --> Performance[calculatePerformancePer]
    Initialize --> Month[translateMonthName]
    Initialize --> Price[getpriceFormatted]
    Initialize --> Cache[setChaching]
    Resize[onResize] --> Initialize
    Destroy[ngOnDestroy] --> Cleanup[Release component resources]
```

Usage

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

@Component({
  selector: 'app-dashboard-host',
  template: `
    <!-- Render MrmPmBacklogComponent in this dashboard view. -->
    <button type="button" (click)="refreshBacklog()">Refresh backlog</button>
  `
})
export class DashboardHostComponent {
  @ViewChild(MrmPmBacklogComponent)
  backlogComponent?: MrmPmBacklogComponent;

  refreshBacklog(): void {
    this.backlogComponent?.getDashboardData();
  }

  handleResize(): void {
    this.backlogComponent?.onResize();
  }
}
```

AI Coding Instructions

- Keep data loading within `getDashboardData()` and pass returned data through `initializingData()` before updating display state.
- Preserve Angular lifecycle behavior: initialize data in `ngOnInit()`, react to input updates in `ngOnChanges()`, and release subscriptions or listeners in `ngOnDestroy()`.
- Keep the existing method names, including `initializingData()` and `setChaching()`, unless all callers are updated together.
- Recalculate displayed values through `calculatePerformancePer()`, `translateMonthName()`, and `getpriceFormatted()` rather than duplicating formatting logic in the template.
- Route resize-driven layout or data updates through `onResize()` and retain cache updates through `setChaching()`.

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `TranslateHandler` — `Pams/Web/Pams.Web/Client/app/app.module.ts :1`
- `DashboardTabComponent` — `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/dashboard-tab/dashboard-tab.component.ts :1`