MrmPmShipmentComponent

August 18, 2026

MrmPmShipmentComponent

Kind: Class

Source: `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-libary/widgets/mrm-pm-shipment/mrm-pm-shipment.component.ts` (line 43)

Part of: Pams

`MrmPmShipmentComponent` is an Angular dashboard widget for presenting MRM PM shipment data and related performance values. It loads and initializes dashboard data, responds to lifecycle and resize events, formats display values, and stores 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`

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

- `MrmPmShipmentComponent` stops the work with an early return when `month.MonthId == 12`.
- `MrmPmShipmentComponent` stops the work with an early return when `month.TotalValue < this.data[bIndex].MrmBooking[index - 1].TotalValue`.
- `MrmPmShipmentComponent` stops the work with an early return when `month.TotalBookingMonth < this.totalBookingMonths[index - 1].TotalBookingMonth`.
- `MrmPmShipmentComponent` stops the work with an early return when `month.TotalForecastMonth < this.totalBookingMonths[index - 1].TotalForecastMonth`.
- `MrmPmShipmentComponent` stops the work with an early return when `month.MonthId >= new Date().getMonth() + 1`.

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

Diagram

```
graph LR
  A[Angular lifecycle] --> B[MrmPmShipmentComponent]
  B --> C[ngOnInit]
  B --> D[ngOnChanges]
  B --> E[ngOnDestroy]
  C --> F[getDashboardData]
  F --> G[initalizingData]
  G --> H[calculatePerformancePer]
  G --> I[translateMonthName]
  G --> J[getpriceFormatted]
  B --> K[onResize]
  B --> L[setChaching]
```

Usage

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

@Component({
  selector: 'app-dashboard-host',
  template: `
    <!-- Render the shipment widget through its configured Angular selector. -->
  `
})
export class DashboardHostComponent {
  @ViewChild(MrmPmShipmentComponent)
  shipmentWidget?: MrmPmShipmentComponent;

  refreshShipmentData(): void {
    this.shipmentWidget?.getDashboardData();
  }

  handleDashboardResize(): void {
    this.shipmentWidget?.onResize();
  }
}
```

AI Coding Instructions

- Keep data loading inside `getDashboardData()` and use `initalizingData()` to prepare received values for display.

- Do not call `ngOnInit()`, `ngOnChanges()`, or `ngOnDestroy()` directly; Angular invokes lifecycle hooks.
- Update display calculations through `calculatePerformancePer()`, `translateMonthName()`, and `getpriceFormatted()` rather than duplicating formatting logic in templates.
- Call `setChaching()` after updating data that should be retained for later dashboard rendering.

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