

MrmPmShipmentComponent

August 18, 2026

MrmPmShipmentComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-pm-shipment/mrm-pm-shipment.component.ts (line 43)

Part of: Frontend

MrmPmShipmentComponent is an Angular dashboard widget for presenting MRM PM shipment data. It checks feature subscription access, loads and initializes dashboard data, calculates performance percentages, formats display values, handles resize events, and manages cached state.

Extends: BaseWidgetComponent

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
ngOnChanges	ngOnChanges()	void
getDashboardData	getDashboardData()	void
initializingData	initializingData(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

- `MrmPmShipmentComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `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` .

Diagram

```
graph LR
    Dashboard[Dashboard Container] --> Component[MrmPmShipmentComponent]
    Component --> Subscription[checkFeatureSubscription]
    Component --> Lifecycle[ngOnInit / ngOnChanges]
    Lifecycle --> Data[getDashboardData]
    Data --> Initialize[initializingData]
    Initialize --> Performance[calculatePerformancePer]
    Initialize --> Formatting[translateMonthName / getPriceFormatted]
    Component --> Resize[onResize]
    Component --> Cache[setChaching]
```

Usage

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

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

  refreshShipmentData(): void {
    const widget = this.shipmentWidget;

    if (!widget?.checkFeatureSubscription()) {
      return;
    }

    widget.getDashboardData();
  }
}
```

```

const performancePercentage = widget.calculatePerformancePer();
const formattedPrice = widget.getpriceFormatted();

console.log({ performancePercentage, formattedPrice });
}
}

```

AI Coding Instructions

- Keep data loading behind `checkFeatureSubscription()` so shipment data is not requested when the feature is unavailable.
- Let Angular invoke `ngOnInit()` and `ngOnChanges()` ; call `getDashboardData()` only for explicit refresh behavior.
- Update initialization logic through `initializingData()` after changing shipment-data inputs or response mapping.
- Preserve formatting through `translateMonthName()` and `getpriceFormatted()` instead of formatting month or price values in templates.
- Review `setChaching()` when changing dashboard refresh behavior so cached shipment data remains consistent.

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`