

MrmWarehouseScrapComponent

August 17, 2026

MrmWarehouseScrapComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-warehouse-scrap/mrm-warehouse-scrap.component.ts (line 90)

Part of: Frontend

MrmWarehouseScrapComponent is an Angular dashboard widget that displays warehouse scrap data. It loads and initializes dashboard state, checks feature subscription access, calculates monthly totals, formats prices and month labels, and saves dashboard configuration changes.

Extends: BaseWidgetComponent

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
ngOnChanges	ngOnChanges()	void
getDashboardData	getDashboardData()	void
initializingData	initializingData(value: BranchData[])	void
saveDashboardData	saveDashboardData()	void
calculateSumByMonth	calculateSumByMonth(monthIndex: number)	void
translateMonthName	translateMonthName(month: MonthData)	string
getpriceFormatted	getpriceFormatted(x: undefined)	string
onResize	onResize()	void
setChaching	setChaching()	void

Method	Signature	Returns
getChaching	getChaching()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, bIndex: number, index: number, propertyName: string)	string
comparingBookingWithPrevSumm	comparingBookingWithPrevSumm(month: MonthsSum, index: number, propertyName: string)	string
isCurrentMonth	isCurrentMonth(month: MonthData)	boolean
getpriceForecastAmount	getpriceForecastAmount(month: MonthData)	number
expectedCurrencyChange	expectedCurrencyChange(e: undefined)	void
selectText	selectText(e: undefined)	void
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthData, supplierId: number)	void
copyValueFromPrevMonth	copyValueFromPrevMonth(month: MonthData, monthIndex: number)	void
sorting	sorting(propertyName: string, propertyType: string, monthIndex: number)	void
openPopup	openPopup(n: string, t: string, i: string, btnTxt: string)	void
savePopup	savePopup()	void
ClosewithDonotSave	ClosewithDonotSave()	void
OnOpenedPopup	OnOpenedPopup()	void
closePopup	closePopup()	void

Properties

Property	Type
sharedPopup	SharedPopupComponent
_widget	Widget
_widgetType	string

Property	Type
yearId	number
periodId	number
countOfEditablePrevMonths	number
withoutOptionBar	boolean
editMode	boolean
bookingOrSales	number
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetEditEnd	any
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]
originalData	BranchData[]
selectedData	BranchData[]
branchCurrency	any
basedOnTendency	boolean
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number

Property	Type
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
popup	any
searchText	string
isModeOfInsertFromPrevMonthActive	boolean
pastDiff	number
urrenFeature	number
sortingDirection	number

Where it refuses work

- `MrmWarehouseScrapComponent` stops the work with an early return when `a[propertyName] < b[propertyName]` , in 2 places.
- `MrmWarehouseScrapComponent` stops the work with an early return when `a[propertyName] > b[propertyName]` , in 2 places.
- `MrmWarehouseScrapComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `MrmWarehouseScrapComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmWarehouseScrapComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex].MRMScrap[index - 1][propertyName]` .
- `MrmWarehouseScrapComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .

Diagram

```
graph LR
  A[Angular lifecycle] --> B[ngOnInit]
  A --> C[ngOnChanges]
  B --> D[checkFeatureSubscription]
  D --> E[getDashboardData]
  E --> F[initializingData]
  F --> G[calculateSumByMonth]
  G --> H[translateMonthName]
  G --> I[getpriceFormatted]
```

```
C --> J[saveDashboardData]
K[Window resize] --> L[onResize]
```

Usage

```
import { Component, ViewChild, AfterViewInit } from '@angular/core';
import { MrmWarehouseScrapComponent } from './mrm-warehouse-scrap.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-mrm-warehouse-scrap
      #warehouseScrapWidget>
    </app-mrm-warehouse-scrap>
  `,
})
export class DashboardPageComponent implements AfterViewInit {
  @ViewChild('warehouseScrapWidget')
  warehouseScrapWidget!: MrmWarehouseScrapComponent;

  ngAfterViewInit(): void {
    if (this.warehouseScrapWidget.checkFeatureSubscription()) {
      this.warehouseScrapWidget.getDashboardData();
    }
  }
}
```

AI Coding Instructions

- Keep data loading inside `getDashboardData()` and initialize derived widget state through `initializingData()`.
- Check `checkFeatureSubscription()` before requesting or rendering subscription-controlled scrap data.
- Recalculate monthly values with `calculateSumByMonth()` after dashboard data changes.
- Use `translateMonthName()` and `getpriceFormatted()` for display formatting instead of duplicating formatting logic in templates.
- Preserve lifecycle behavior in `ngOnInit()`, `ngOnChanges()`, and `onResize()` when changing widget inputs or layout handling.

Used by

3 references from 3 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (3)

- `DashboardTabComponent` — `Frontend/src/app/components/dashboard/dashboard-library/dashboard-tab/dashboard-tab.component.ts :1`
- `DashboardWidgetsModule` — `Frontend/src/app/components/dashboard/dashboard-widgets.module.ts :1`
- `ReportsTabComponent` — `Frontend/src/app/components/reports/reports-library/reports-tab/reports-tab.component.ts :1`