

MrmWarehouseScrapComponent

August 18, 2026

MrmWarehouseScrapComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-warehouse-scrap/mrm-warehouse-scrap.component.ts (line 90)

Part of: [Pams](#)

`MrmWarehouseScrapComponent` is an Angular dashboard widget for loading, preparing, and displaying warehouse scrap data. It checks authorization, responds to lifecycle changes, calculates monthly totals, formats month and price values, and saves dashboard state.

Methods

Method	Signature	Returns
<code>GetAuthorizationAccess</code>	<code>GetAuthorizationAccess(url: string)</code>	<code>void</code>
<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>saveDashboardData</code>	<code>saveDashboardData()</code>	<code>void</code>
<code>calculateSumByMonth</code>	<code>calculateSumByMonth(monthIndex: number)</code>	<code>void</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>

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
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
subscription	Subscription
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number
yearBooking	number
totalForecasts	number

Property	Type
tempTotalTarget	TotalTargets
popup	any
searchText	string
isModeOfInsertFromPrevMonthActive	boolean
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 `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]` .
- `MrmWarehouseScrapComponent` stops the work with an early return when `this.allowedtoEnableEdting` .

Diagram

```
graph LR
  Init[ngOnInit] --> Auth[GetAuthorizationAccess]
  Auth --> Load[getDashboardData]
  Load --> Initialize[initializingData]
  Initialize --> Sum[calculateSumByMonth]
  Sum --> Month[translateMonthName]
  Sum --> Price[getpriceFormatted]
  Changes[ngOnChanges] --> Load
  Initialize --> Save[saveDashboardData]
  Destroy[ngOnDestroy] --> Cleanup[Release subscriptions or state]
```

Usage

```
import { ComponentRef } from '@angular/core';
import { MrmWarehouseScrapComponent } from './mrm-warehouse-scrap.component';

function refreshWarehouseScrapWidget(
  componentRef: ComponentRef<MrmWarehouseScrapComponent>
): void {
  const widget = componentRef.instance;

  // Angular runs ngOnInit during initial change detection.
  componentRef.changeDetectorRef.detectChanges();

  // Reload data after an external dashboard filter or refresh action.
  widget.getDashboardData();

  const monthLabel = widget.translateMonthName();
  const formattedPrice = widget.getPriceFormatted();

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

AI Coding Instructions

- Let Angular invoke `ngOnInit` , `ngOnChanges` , and `ngOnDestroy` ; do not call lifecycle hooks from application code.
- Call `GetAuthorizationAccess()` before exposing or loading dashboard scrap data.
- Keep data preparation in `initializingData()` and monthly aggregation in `calculateSumByMonth()` rather than duplicating calculations in templates.
- Preserve the existing `initializingData` method spelling when calling or modifying it.
- Use `saveDashboardData()` for dashboard-state persistence and `getDashboardData()` when widget data must be refreshed.

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)

- `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`
- `ReportsTabComponent` — `Pams/Web/Pams.Web/Client/app/components/dashboard/reports-library/reports-tab/reports-tab.component.ts :1`

