

MrmWarehouseSupplierComponent

August 17, 2026

MrmWarehouseSupplierComponent

Kind: Class

Source: `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-warehouse-supplier/mrm-warehouse-supplier.component.ts` (line 84)

Part of: [Pams](#)

`MrmWarehouseSupplierComponent` manages warehouse-supplier dashboard data within the dashboard widget library. It handles authorization checks, lifecycle updates, data initialization and persistence, monthly totals, month-name translation, and synchronized scrolling.

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>syncScroll</code>	<code>syncScroll(e: undefined)</code>	<code>void</code>
<code>getpriceFormatted</code>	<code>getpriceFormatted(x: undefined)</code>	<code>string</code>
<code>onResize</code>	<code>onResize()</code>	<code>void</code>

Method	Signature	Returns
setChaching	setChaching()	void
getChaching	getChaching()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, bIndex: number, index: number)	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
scrollOne	ElementRef

Property	Type
scrollTwo	ElementRef
scrollThree	ElementRef
_widget	Widget
_widgetType	string
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[]
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean
subscription	Subscription
SeriseRange	any[]
echartsIntance	any

Property	Type
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
popup	any
searchText	string
isModeOfInsertFromPrevMonthActive	boolean
sortingDirection	number

Where it refuses work

- `MrmWarehouseSupplierComponent` stops the work with an early return when `a[propertyName] < b[propertyName]` , in 2 places.
- `MrmWarehouseSupplierComponent` stops the work with an early return when `a[propertyName] > b[propertyName]` , in 2 places.
- `MrmWarehouseSupplierComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmWarehouseSupplierComponent` stops the work with an early return when `month.GoodsTotal < this.data[bIndex].MrmStock[index - 1].GoodsTotal` .
- `MrmWarehouseSupplierComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .
- `MrmWarehouseSupplierComponent` stops the work with an early return when `this.allowedtoEnableEditing` .

Diagram

```
graph LR
  A[Angular lifecycle] --> B[GetAuthorizationAccess]
  A --> C[initializingData]
  C --> D[getDashboardData]
  D --> E[calculateSumByMonth]
  E --> F[translateMonthName]
  F --> G[Dashboard widget view]
```

```
G --> H[syncScroll]
G --> I[saveDashboardData]
A --> J[ngOnDestroy]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { MrmWarehouseSupplierComponent } from './mrm-warehouse-supplier.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-mrm-warehouse-supplier></app-mrm-warehouse-supplier>

    <button type="button" (click)="refreshWidget()">
      Refresh
    </button>
  `,
})
export class DashboardPageComponent {
  @ViewChild(MrmWarehouseSupplierComponent)
  warehouseSupplierWidget!: MrmWarehouseSupplierComponent;

  refreshWidget(): void {
    this.warehouseSupplierWidget.getDashboardData();
    this.warehouseSupplierWidget.calculateSumByMonth();
    this.warehouseSupplierWidget.syncScroll();
  }

  saveWidgetState(): void {
    this.warehouseSupplierWidget.saveDashboardData();
  }
}
```

AI Coding Instructions

- Keep authorization checks in `GetAuthorizationAccess()` before loading or saving dashboard data.
- Follow Angular lifecycle flow: initialize data in `ngOnInit()`, respond to updated bindings in `ngOnChanges()`, and release subscriptions or listeners in `ngOnDestroy()`.
- Preserve the existing `initializingData()` method name when calling or modifying it, even though its spelling differs from `initializingData`.
- Recalculate monthly totals with `calculateSumByMonth()` after dashboard data changes, then update month labels through `translateMonthName()`.

- Keep linked scroll containers synchronized through `syncScroll()` without creating recursive scroll event updates.

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`