

MrmWarehouseSupplierComponent

August 18, 2026

MrmWarehouseSupplierComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-warehouse-supplier/mrm-warehouse-supplier.component.ts (line 84)

Part of: Frontend

MrmWarehouseSupplierComponent is an Angular dashboard widget for warehouse supplier data. It checks feature subscription access, loads and saves dashboard data, prepares month-based totals, formats display values, and synchronizes scrolling.

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
syncScroll	syncScroll(e: undefined)	void
getpriceFormatted	getpriceFormatted(x: undefined)	string
onResize	onResize()	void

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
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[]
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean

Property	Type
SerializeRange	any[]
echartsIntance	any
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
popup	any
searchText	string
isModeOfInsertFromPrevMonthActive	boolean
pastDiff	number
urrenFeature	number
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 `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `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]` .

Diagram

```
graph LR
  Lifecycle[Angular lifecycle] --> Component[MrmWarehouseSupplierComponent]
  Component --> Subscription[checkFeatureSubscription]
  Component --> Data[getDashboardData]
  Data --> Initialize[initializingData]
  Initialize --> Totals[calculateSumByMonth]
  Component --> Save[saveDashboardData]
  Totals --> MonthLabel[translateMonthName]
  Totals --> PriceFormat[getpriceFormatted]
  Component --> Scroll[syncScroll]
```

Usage

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { MrmWarehouseSupplierComponent } from './mrm-warehouse-supplier.component';

describe('MrmWarehouseSupplierComponent', () => {
  let fixture: ComponentFixture<MrmWarehouseSupplierComponent>;
  let component: MrmWarehouseSupplierComponent;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [MrmWarehouseSupplierComponent],
    }).compileComponents();

    fixture = TestBed.createComponent(MrmWarehouseSupplierComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('loads dashboard data for subscribed users', () => {
    if (component.checkFeatureSubscription()) {
      component.getDashboardData();
      component.calculateSumByMonth();

      const monthName = component.translateMonthName();
      const formattedPrice = component.getpriceFormatted();

      expect(monthName).toBeDefined();
      expect(formattedPrice).toBeDefined();
    }
  });
});
```

AI Coding Instructions

- Keep subscription checks in place before requesting or displaying warehouse supplier dashboard data.
- Follow Angular lifecycle flow: initialize data through `ngOnInit` and refresh derived state through `ngOnChanges` when inputs change.
- Preserve the existing `initializingData` method name when calling it, even though its spelling differs from `initializing`.
- Recalculate month totals after changing source dashboard data so formatted month and price values stay current.
- Keep `syncScroll` tied to the relevant scroll containers and avoid calling it before the widget view is available.

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`