

MrmFinanceLgComponent

August 18, 2026

MrmFinanceLgComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-finance-lg/mrm-finance-lg.component.ts (line 72)

Part of: [Pams](#)

`MrmFinanceLgComponent` is an Angular dashboard widget that retrieves and prepares finance data for display. It responds to component lifecycle and resize events, calculates performance values, translates month names, formats prices, and manages cached state.

Methods

Method	Signature	Returns
<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>calculatePerformancePer</code>	<code>calculatePerformancePer(month: MonthData)</code>	<code>number</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>
<code>getChaching</code>	<code>getChaching()</code>	<code>void</code>
<code>onValueChanged</code>	<code>onValueChanged(e: undefined)</code>	<code>void</code>

Method	Signature	Returns
onChartInit	onChartInit(e: any, index: number)	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
mouseoverRowSpan	mouseoverRowSpan(e: undefined, bIndex: number)	void
mouseoutRowSpan	mouseoutRowSpan(e: undefined, bIndex: number)	void
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, bIndex: number, monthIndex: number, propertyName: string)	string
comparingBookingWithPrevSumm	comparingBookingWithPrevSumm(month: TotalBookingMonths, index: number)	string
comparingMonthsWithPrevSumm	comparingMonthsWithPrevSumm(month: TotalBookingMonths, index: number, propertyName: string)	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
withoutOptionBar	boolean
bookingOrSales	number
withNominal	boolean

Property	Type
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEditing	boolean
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]
originalData	BranchData[]
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean
subscription	Subscription
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	TotalBookingMonths[]
tempTotalMonth	TotalBookingMonths

Where it refuses work

- `MrmFinanceLgComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmFinanceLgComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex].MrmLGs[monthIndex - 1][propertyName]` .
- `MrmFinanceLgComponent` stops the work with an early return when `month.TotalAmountMonth < this.totalBookingMonths[index - 1].TotalAmountMonth` .
- `MrmFinanceLgComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .

- `MrmFinanceLgComponent` stops the work with an early return when `month.MonthId >= new Date().getMonth() + 1` .
- `MrmFinanceLgComponent` stops the work with an early return when `month.MonthId <= new Date().getMonth() + 1` .

Diagram

```
graph LR
    Lifecycle[Angular lifecycle] --> Init[ngOnInit]
    Lifecycle --> Changes[ngOnChanges]
    Lifecycle --> Destroy[ngOnDestroy]

    Init --> Fetch[getDashboardData]
    Changes --> Fetch
    Fetch --> Initialize[initializingData]

    Initialize --> Performance[calculatePerformancePer]
    Initialize --> Month[translateMonthName]
    Initialize --> Price[getpriceFormatted]

    Resize[onResize] --> Cache[setChaching]
    Destroy --> Cache
```

Usage

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { MrmFinanceLgComponent } from './mrm-finance-lg.component';

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

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

    fixture = TestBed.createComponent(MrmFinanceLgComponent);
    component = fixture.componentInstance;

    fixture.detectChanges();
  });

  it('loads and formats dashboard data', () => {
    component.getDashboardData();

    const performance = component.calculatePerformancePer();
```

```

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

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

```

AI Coding Instructions

- Keep data retrieval inside `getDashboardData()` and prepare widget state through `initializingData()`.
- Recalculate display-dependent values after input changes and resize events when the rendered layout depends on them.
- Preserve the existing lifecycle cleanup in `ngOnDestroy()` and keep cache handling in `setChaching()`.
- Use `calculatePerformancePer()`, `translateMonthName()`, and `getpriceFormatted()` for display values instead of duplicating formatting logic in templates.

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)

- `TranslateHandler` — `Pams/Web/Pams.Web/Client/app/app.module.ts`:1