

MRMCashFlowComponent

August 18, 2026

MRMCashFlowComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-cash-flow/mrm-cash-flow.component.ts (line 147)

Part of: Frontend

MRMCashFlowComponent is an Angular dashboard widget that loads, prepares, and saves MRM cash-flow data. It checks feature subscription access, builds monthly and yearly summaries, and derives currency rates from dashboard header data.

Extends: BaseWidgetComponent

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
ngOnChanges	ngOnChanges()	void
getDashboardData	getDashboardData()	void
fillMonthsSummery	fillMonthsSummery()	void
initalizingData	initalizingData(value: BranchData)	void
saveDashboardData	saveDashboardData()	void
calculateSumByYear	calculateSumByYear()	void
calculateSumByMonth	calculateSumByMonth(monthIndex: number)	void
getCurrencyRateFromHeader	getCurrencyRateFromHeader(currencyCode: string)	number

Method	Signature	Returns
setCurrencyTotalToHeader	setCurrencyTotalToHeader(cashFlowObjIndex: number, currencyCode: string, amount: number)	void
resetCurrencyTotal	resetCurrencyTotal(cashFlowObjIndex: number)	void
calculateEditingSumByMonth	calculateEditingSumByMonth(cashFlowObjIndex: number, itemIndex: number, currencyIndex: number, monthIndex: number)	void
translateMonthName	translateMonthName(MonthId: number)	string
getpriceFormatted	getpriceFormatted(x: undefined)	string
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void
switchVendorTypeIdHandler	switchVendorTypeIdHandler()	void
switchByGroupsHandler	switchByGroupsHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: CashFlowItemMonthlyDetail, cashFlowObjIndex: number, monthIndex: number, propertyName: string)	string
comparingSummeryWithPrevMonth	comparingSummeryWithPrevMonth(month: MonthsSum, monthIndex: number, propertyName: string)	string
comparingEditingWithPrevMonth	comparingEditingWithPrevMonth(month: CashFlowItemMonthlyDetail, cashFlowObjIndex: number, itemIndex: number, currencyIndex: number, monthIndex: number, propertyName: string)	string
isCurrentMonth	isCurrentMonth(month: MonthsSum)	boolean
expectedCurrencyChange	expectedCurrencyChange(e: undefined)	void
selectText	selectText(e: undefined)	void
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthsSum, supplierId: number)	void

Method	Signature	Returns
viewSelectedMonthPrepare	viewSelectedMonthPrepare(month: MonthsSum, supplierId: number)	void
fromNow	fromNow(DateTime: any)	void
mouseoverRowSpan	mouseoverRowSpan(e: undefined, bIndex: number)	void
mouseoutRowSpan	mouseoutRowSpan(e: undefined, bIndex: 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
yearId	number
periodId	number
countOfEditablePrevMonths	number
withoutOptionBar	boolean
editMode	boolean
viewDetailsMode	boolean
cashFlowDisplay	number
bookingOrSales	number
byGroups	boolean
isPamsData	boolean

Property	Type
selectedTeam	number
forecastRuleId	number
data	BranchData
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetViewRequest	any
widgetEditEnd	any
forecastRules	any[]
currencies	CompanyCurrency[]
selectedYear	number
currentMonthIndex	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
originalData	BranchData
selectedData	BranchData
monthsSummery	MonthsSum[]
branchCurrency	any
salesCurrency	any
basedOnTendency	boolean
SeriseRange	any[]
echartsIntance	any
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
totalProfitAndLossAmount	number

Property	Type
totalOthers	number
totalTotalSGAndA	number
totalFORX	number
totalTotal	number
totalCumulative	number
totalPercentageUsedBudget	number
popup	any
searchText	string
DateDisplayFormatForTypeScript	string

Where it refuses work

- MRMCashFlowComponent stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- MRMCashFlowComponent stops the work with an early return when `findMonthIndex > -1` .
- MRMCashFlowComponent stops the work with an early return when `MonthId == 12` .
- MRMCashFlowComponent stops the work with an early return when `month[propertyName] < this.data.BanksSummary[monthIndex - 1][propertyName]` .
- MRMCashFlowComponent stops the work with an early return when `month[propertyName] < this.data.ReceivablesSummary[monthIndex - 1][propertyName]` .
- MRMCashFlowComponent stops the work with an early return when `month[propertyName] < this.data.PayablesSummary[monthIndex - 1][propertyName]` .

Diagram

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

```
E --> J[getCurrencyRateFromHeader]
F --> K[saveDashboardData]
```

Usage

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { MRMCashFlowComponent } from './mrm-cash-flow.component';

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

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [MRMCashFlowComponent],
    });

    fixture = TestBed.createComponent(MRMCashFlowComponent);
    component = fixture.componentInstance;
  });

  it('loads and summarizes dashboard cash-flow data', () => {
    fixture.detectChanges();

    if (component.checkFeatureSubscription()) {
      component.getDashboardData();
      component.fillMonthsSummery();
      component.calculateSumByMonth();
      component.calculateSumByYear();
      component.saveDashboardData();
    }
  });
});
```

AI Coding Instructions

- Check `checkFeatureSubscription()` before requesting or displaying cash-flow data.
- Keep data preparation in `initalizingData()` and `fillMonthsSummery()` so monthly and yearly calculations operate on the same state.
- Call `getDashboardData()` when relevant dashboard inputs change through `ngOnChanges()`.
- Read exchange-rate values through `getCurrencyRateFromHeader()` rather than duplicating header parsing logic.

- Preserve the existing save flow by updating component data before calling `saveDashboardData()` .

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `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`