

MrmCashFlowForecastedComponent

August 18, 2026

MrmCashFlowForecastedComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-cash-flow-forecasted/mrm-cash-flow-forecasted.component.ts (line 105)

Part of: [Pams](#)

`MrmCashFlowForecastedComponent` is an Angular dashboard widget that loads, prepares, and displays forecasted cash-flow data. It responds to lifecycle changes, calculates totals by year and month, reads the currency rate from dashboard headers, and saves dashboard state when needed.

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>fillMonthsSummery</code>	<code>fillMonthsSummery()</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>calculateSumByYear</code>	<code>calculateSumByYear()</code>	<code>void</code>
<code>calculateSumByMonth</code>	<code>calculateSumByMonth(monthIndex: number)</code>	<code>void</code>
<code>getCurrencyRateFromHeader</code>	<code>getCurrencyRateFromHeader(currencyCode: string)</code>	<code>number</code>
<code>setCurrencyTotalToHeader</code>	<code>setCurrencyTotalToHeader(cashFlowObjIndex: number, currencyCode: string, amount:</code>	<code>void</code>

Method	Signature	Returns
	number)	
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
isForecastedMonth	isForecastedMonth(month: CashFlowItemMonthlyDetail)	boolean
isCurrentMonth	isCurrentMonth(month: CashFlowItemMonthlyDetail)	boolean
expectedCurrencyChange	expectedCurrencyChange(e: undefined)	void
selectText	selectText(e: undefined)	void
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthsSum, supplierId: number)	void

Method	Signature	Returns
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
withoutOptionBar	boolean
editMode	boolean
cashFlowDisplay	number
bookingOrSales	number
byGroups	boolean
isPamsData	boolean
selectedTeam	number
forecastRuleId	number
data	BranchData
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any

Property	Type
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
subscription	Subscription
SeriseRange	any[]
echartsIntance	any
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
totalProfitAndLossAmount	number
totalOthers	number
totalTotalSGAndA	number
totalFORX	number
totalTotal	number
totalCumulative	number
totalPercentageUsedBudget	number

Property	Type
popup	any
searchText	string

Where it refuses work

- `MrmCashFlowForecastedComponent` stops the work with an early return when `findMonthIndex > -1` .
- `MrmCashFlowForecastedComponent` stops the work with an early return when `MonthId == 12` .
- `MrmCashFlowForecastedComponent` stops the work with an early return when `month[propertyName] < this.data.BanksSummary[monthIndex - 1][propertyName]` .
- `MrmCashFlowForecastedComponent` stops the work with an early return when `month[propertyName] < this.data.ReceivablesSummary[monthIndex - 1][propertyName]` .
- `MrmCashFlowForecastedComponent` stops the work with an early return when `month[propertyName] < this.data.PayablesSummary[monthIndex - 1][propertyName]` .
- `MrmCashFlowForecastedComponent` stops the work with an early return when `month[propertyName] < this.data.LGsSummary[monthIndex - 1][propertyName]` .

Diagram

```
graph LR
  Init[ngOnInit] --> Load[getDashboardData]
  Changes[ngOnChanges] --> Load
  Load --> Initialize[initializingData]
  Initialize --> Months[fillMonthsSummery]
  Months --> ByYear[calculateSumByYear]
  Months --> ByMonth[calculateSumByMonth]
  Currency[getCurrencyRateFromHeader] --> ByYear
  Currency --> ByMonth
  Initialize --> Save[saveDashboardData]
  Destroy[ngOnDestroy] --> Cleanup[Release subscriptions and resources]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { MrmCashFlowForecastedComponent } from './mrm-cash-flow-forecasted.component';

@Component({
  selector: 'app-dashboard-page',
```

```

template: `
  <app-mrm-cash-flow-forecasted></app-mrm-cash-flow-forecasted>

  <button type="button" (click)="refreshCashFlowForecast()">
    Refresh forecast
  </button>
`
})
export class DashboardPageComponent {
  @ViewChild(MrmCashFlowForecastedComponent)
  private cashFlowForecast?: MrmCashFlowForecastedComponent;

  refreshCashFlowForecast(): void {
    this.cashFlowForecast?.getDashboardData();
  }
}

```

AI Coding Instructions

- Keep data-loading work in `getDashboardData()` and allow Angular lifecycle hooks to trigger refreshes during initialization and input changes.
- Preserve the existing calculation flow: initialize data, populate month summaries, then calculate totals by year and month.
- Use `getCurrencyRateFromHeader()` when calculations depend on the dashboard currency context instead of hard-coding conversion values.
- Clean up subscriptions, timers, or other allocated resources in `ngOnDestroy()`.
- Retain the existing method names, including `fillMonthsSummary()` and `initializingData()`, unless all callers are updated together.

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)

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