

MrmCashFlowForecastedComponent

August 19, 2026

MrmCashFlowForecastedComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-cash-flow-forecasted/mrm-cash-flow-forecasted.component.ts (line 187)

Part of: Frontend

MrmCashFlowForecastedComponent manages forecasted cash-flow data for the dashboard widget. It loads report settings and dashboard data, prepares monthly summaries, prevents edits for previous months, and saves user changes when the feature subscription permits access.

Extends: BaseWidgetComponent

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
ngOnChanges	ngOnChanges()	void
getDashboardDataByMonth	getDashboardDataByMonth()	void
getDashboardData	getDashboardData()	void
getReportSetting	getReportSetting()	void
fillMonthsSummery	fillMonthsSummery()	void
initalizingData	initalizingData(value: BranchData)	void
disableEditingInPrevMonths	disableEditingInPrevMonths()	void
saveDashboardData	saveDashboardData()	void
saveDashboardDataForNotForecast	saveDashboardDataForNotForecast()	void

Method	Signature	Returns
calculateSumByYear	calculateSumByYear()	void
calculateSumByMonth	calculateSumByMonth(monthIndex: number)	void
getCurrencyRateFromHeader	getCurrencyRateFromHeader(currencyCode: string)	number
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
isForecastedMonth	isForecastedMonth(YearId: number, MonthId: number)	boolean
isCurrentMonth	isCurrentMonth(YearId: number, MonthId: number)	boolean
expectedCurrencyChange	expectedCurrencyChange(e: undefined)	void
selectText	selectText(e: undefined)	void

Method	Signature	Returns
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthsSum)	void
viewSelectedMonthPrepare	viewSelectedMonthPrepare(month: MonthsSum, supplierId: number)	void
fromNow	fromNow(DateTime: any)	void
calculateRowSpanCount	calculateRowSpanCount(itemNum: number)	number
mouseoverRowSpan	mouseoverRowSpan(e: undefined, bIndex: number)	void
mouseoutRowSpan	mouseoutRowSpan(e: undefined, bIndex: number)	void
toggolePurchaseInvoicesDropdown	toggolePurchaseInvoicesDropdown()	void
closePurchaseInvoicesDropdown	closePurchaseInvoicesDropdown()	void
toggoleSalesInvoicesDropdown	toggoleSalesInvoicesDropdown()	void
closeSalesInvoicesDropdown	closeSalesInvoicesDropdown()	void
toggoleCommInvoicesDropdown	toggoleCommInvoicesDropdown()	void
closeCommInvoicesDropdown	closeCommInvoicesDropdown()	void
toggoleOverdueDelayDropdown	toggoleOverdueDelayDropdown()	void
closeOverdueDelayDropdown	closeOverdueDelayDropdown()	void
closeDropdown	closeDropdown(e: undefined)	void
traceInvoicesDropdownPeriod	traceInvoicesDropdownPeriod(periodId: number)	string
keyDown	keyDown(e: DxNumberBoxTypes.KeyDownEvent)	void
submitChanges	submitChanges()	void
saveBookingRulesInConfiguration	saveBookingRulesInConfiguration()	void

Properties

Property	Type
sharedPopup	SharedPopupComponent
PurchaseInvoicesDropdown	ElementRef
SalesInvoicesDropdown	ElementRef

Property	Type
CommInvoicesDropdown	ElementRef
OverdueDelayDropdown	ElementRef
_widget	Widget
_widgetType	string
yearId	number
periodId	number
countOfEditablePrevMonths	number
withoutOptionBar	boolean
editMode	boolean
viewModeId	number
viewModeEnum	any
cashFlowDisplay	number
bookingOrSales	number
byGroups	boolean
isPamsData	boolean
selectedTeam	number
CashFlowYearId	number
CashFlowMonthId	number
forecastRuleId	number
data	BranchData
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetViewRequest	any
widgetEditEnd	any
forecastRules	any[]
currencies	CompanyCurrency[]
selectedYear	number

Property	Type
currentMonthIndex	number
diff	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
totalOthers	number

Where it refuses work

- `MrmCashFlowForecastedComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `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]` .

Diagram

```
graph LR
  A[ngOnInit / ngOnChanges] --> B[checkFeatureSubscription]
  B --> C[getReportSetting]
  C --> D[getDashboardData]
  D --> E[getDashboardDataByMonth]
  E --> F[fillMonthsSummery]
  F --> G[initalizingData]
  G --> H[disableEditingInPrevMonths]
  H --> I[saveDashboardData]
```

Usage

```
import { MrmCashFlowForecastedComponent } from './mrm-cash-flow-forecasted.component';

// Example from an Angular test or configured component fixture.
const component = fixture.componentInstance as MrmCashFlowForecastedComponent;

component.ngOnInit();

if (component.checkFeatureSubscription()) {
  component.getDashboardDataByMonth();

  // After updating forecast values through the widget UI:
  component.saveDashboardData();
}
```

AI Coding Instructions

- Keep dashboard loading logic aligned with `ngOnInit` and `ngOnChanges` so the widget refreshes when its inputs change.
- Check `checkFeatureSubscription()` before requesting or saving forecast data that depends on subscription access.
- Call `getReportSetting()` before preparing data that depends on report configuration.
- Preserve the edit restrictions applied by `disableEditingInPrevMonths()` when changing month or forecast input behavior.

- Update monthly summary data through `fillMonthsSummery()` after loading or changing dashboard values.

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`