

MrmAccountPayableComponent

August 18, 2026

MrmAccountPayableComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/mrm-account-payable/mrm-account-payable.component.ts (line 76)

Part of: [Frontend](#)

MrmAccountPayableComponent is a dashboard widget component that loads, initializes, saves, and aggregates account payable data. It also manages available currencies and checks whether the current feature subscription allows the widget to be used.

Extends: BaseWidgetComponent

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
getCurrencies	getCurrencies()	void
ngOnInit	ngOnInit()	void
ngOnChanges	ngOnChanges()	void
getDashboardData	getDashboardData()	void
initializingData	initializingData(value: BranchData)	void
saveDashboardData	saveDashboardData()	void
calculateSumByYear	calculateSumByYear()	void
calculateSumByMonth	calculateSumByMonth(monthIndex: number)	void
AddNewCurrency	AddNewCurrency(account: number, SalesCom: number)	void

Method	Signature	Returns
onCurrencyChanged	onCurrencyChanged(e: undefined, account: number, SalesCom: number, currencyIndex: number)	void
translateMonthName	translateMonthName(MonthId: number)	string
getpriceFormatted	getpriceFormatted(x: undefined)	string
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void
switchByGroupsHandler	switchByGroupsHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, accountIndex: number, salesComIndex: number, currencyIndex: number, monthIndex: number, propertyName: string)	string
comparingSummeryWithPrevMonth	comparingSummeryWithPrevMonth(month: MonthData, monthIndex: 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
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

Method	Signature	Returns
closePopup	closePopup()	void

Properties

Property	Type
sharedPopup	SharedPopupComponent
_widget	Widget
_widgetType	string
yearId	number
periodId	number
countOfEditablePrevMonths	number
withoutOptionBar	boolean
editMode	boolean
bookingOrSales	number
byGroups	boolean
selectedTeam	number
forecastRuleId	number
data	BranchData
allowedtoEdit	boolean
allowedtoEnableEdting	boolean
widgetEditRequest	any
widgetEditEnd	any
forecastRules	any[]
currencies	CompanyCurrency[]
selectedYear	number
currentMonthIndex	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData

Property	Type
originalData	BranchData
selectedData	BranchData
singleMonths	MonthData
monthsSummery	MonthsSum[]
nonGroupAccountReceivables	MonthData[]
branchCurrency	any
basedOnTendency	boolean
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
popup	any
searchText	string
urrenFeature	number

Where it refuses work

- `MrmAccountPayableComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `MrmAccountPayableComponent` stops the work with an early return when `MonthId == 12` .

- `MrmAccountPayableComponent` stops the work with an early return when `month[propertyName] < this.data.Products.Currencies[currencyIndex].Details[monthIndex - 1]...` .
- `MrmAccountPayableComponent` stops the work with an early return when `month[propertyName] < this.data.Scrap.Currencies[currencyIndex].Details[monthIndex - 1][p...` .
- `MrmAccountPayableComponent` stops the work with an early return when `month[propertyName] < this.monthsSummery[monthIndex - 1][propertyName]` .
- `MrmAccountPayableComponent` stops the work with an early return when `AppSettings.calculateMonthsBeforeToday(month.YearId, month.MonthId - 1) == 0 || AppSetting...` .

Diagram

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

Usage

```
import { MrmAccountPayableComponent } from './mrm-account-payable.component';

// Angular creates the component and runs lifecycle hooks.
// Call dashboard actions from event handlers or surrounding dashboard logic.
function refreshPayables(
  component: MrmAccountPayableComponent
): void {
  if (!component.checkFeatureSubscription()) {
    return;
  }

  component.getCurrencies();
  component.getDashboardData();
  component.calculateSumByYear();
  component.calculateSumByMonth();
}
```

```

}

// Example action for adding a currency through the widget.
function addCurrency(component: MrmAccountPayableComponent): void {
  component.AddNewCurrency();
  component.saveDashboardData();
}

```

AI Coding Instructions

- Keep subscription validation through `checkFeatureSubscription()` before loading or editing account payable widget data.
- Use `ngOnInit()` for initial widget setup and `ngOnChanges()` when input-driven dashboard state changes.
- Preserve the existing initialization flow: load dashboard data, run `initializingData()`, then calculate yearly and monthly totals.
- Call `saveDashboardData()` after changes that should persist, including currency updates from `AddNewCurrency()`.
- Keep currency retrieval in `getCurrencies()` so dashboard currency state remains managed by the component.

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`