

MrmCurrencyRateComponent

August 18, 2026

MrmCurrencyRateComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-currency-rate/mrm-currency-rate.component.ts (line 77)

Part of: [Pams](#)

`MrmCurrencyRateComponent` is an Angular dashboard widget that retrieves, initializes, and saves currency-rate dashboard data. It responds to lifecycle changes, calculates totals by year and month, translates month names, and cleans up when the component is destroyed.

Methods

Method	Signature	Returns
<code>GetAuthorizationAccess</code>	<code>GetAuthorizationAccess(url: string)</code>	<code>void</code>
<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>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>translateMonthName</code>	<code>translateMonthName(month: BranchData)</code>	<code>string</code>
<code>getpriceFormatted</code>	<code>getpriceFormatted(x: undefined)</code>	<code>string</code>
<code>onResize</code>	<code>onResize()</code>	<code>void</code>

Method	Signature	Returns
setChaching	setChaching()	void
getChaching	getChaching()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
comparingWithPrevMonth	comparingWithPrevMonth(month: BranchData, bIndex: number, propertyName: string)	string
comparingBookingWithPrevSumm	comparingBookingWithPrevSumm(month: MonthsSum, index: 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
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
bookingOrSales	number

Property	Type
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEditing	boolean
widgetEditRequest	any
widgetEditEnd	any
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]
originalData	BranchData[]
selectedData	BranchData[]
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean
subscription	Subscription
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	MonthsSum[]
tempMonthSum	MonthsSum
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets
totalPercentageUsedBudget	number

Property	Type
popup	any
searchText	string

Where it refuses work

- `MrmCurrencyRateComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmCurrencyRateComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex - 1][propertyName]` .
- `MrmCurrencyRateComponent` stops the work with an early return when `month[propertyName] < this.totalBookingMonths[index - 1][propertyName]` .
- `MrmCurrencyRateComponent` stops the work with an early return when `this.allowedtoEnableEdting` .
- `MrmCurrencyRateComponent` stops the work with an early return when `this.allowedtoEdit` .

Diagram

```
graph LR
    A[Angular Lifecycle] --> B[ngOnInit]
    A --> C[ngOnChanges]
    A --> D[ngOnDestroy]

    B --> E[GetAuthorizationAccess]
    E --> F[getDashboardData]
    F --> G[initializingData]

    G --> H[calculateSumByYear]
    G --> I[calculateSumByMonth]
    G --> J[translateMonthName]

    G --> K[saveDashboardData]
```

Usage

```
import { MrmCurrencyRateComponent } from './mrm-currency-rate.component';

export function refreshCurrencyRateWidget(
  widget: MrmCurrencyRateComponent,
) {
```

```

widget.GetAuthorizationAccess();
widget.getDashboardData();

const yearlyTotal = widget.calculateSumByYear();
const monthlyTotal = widget.calculateSumByMonth();
const monthName = widget.translateMonthName();

widget.saveDashboardData();

return {
  yearlyTotal,
  monthlyTotal,
  monthName,
};
}

```

AI Coding Instructions

- Let Angular call `ngOnInit`, `ngOnChanges`, and `ngOnDestroy`; avoid calling lifecycle hooks from application code.
- Keep the existing `initializingData()` method name when changing related code, including its current spelling.
- Fetch and initialize dashboard data before calling `calculateSumByYear()` or `calculateSumByMonth()`.
- Keep authorization checks in `GetAuthorizationAccess()` aligned with dashboard data access behavior.
- Add cleanup for subscriptions, timers, or other resources in `ngOnDestroy()` when extending data-loading logic.

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)

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

