

MrmCashPositionComponent

August 17, 2026

MrmCashPositionComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/mrm-cash-position/mrm-cash-position.component.ts (line 89)

Part of: [Pams](#)

`MrmCashPositionComponent` is an Angular dashboard widget that retrieves, initializes, and saves cash-position dashboard data. It checks authorization, obtains currency rates, aggregates monthly values, and calculates a performance percentage for the dashboard view.

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>saveDashboardData</code>	<code>saveDashboardData()</code>	<code>void</code>
<code>initializingData</code>	<code>initializingData(value: BranchData[])</code>	<code>void</code>
<code>calculateSumByMonth</code>	<code>calculateSumByMonth(monthIndex: number)</code>	<code>void</code>
<code>getCurrencyRates</code>	<code>getCurrencyRates()</code>	<code>void</code>
<code>calculatePerformancePer</code>	<code>calculatePerformancePer(month: MonthData)</code>	<code>number</code>
<code>translateMonthName</code>	<code>translateMonthName(month: MonthData)</code>	<code>string</code>
<code>getpriceFormatted</code>	<code>getpriceFormatted(x: undefined)</code>	<code>string</code>

Method	Signature	Returns
syncScroll	syncScroll(e: undefined)	void
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void
onValueChanged	onValueChanged(e: undefined)	void
onChartInit	onChartInit(e: any, index: number)	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
isCurrentMonth	isCurrentMonth(month: MonthData)	boolean
comparingWithPrevMonth	comparingWithPrevMonth(month: MonthData, bIndex: number, index: number, propertyName: string)	string
comparingMonthsWithPrevSumm	comparingMonthsWithPrevSumm(month: TotalBookingMonths, index: number, propertyName: string)	string
setValue	setValue(value: number)	void
setForecastValue	setForecastValue(value: number)	void
setMarginAvarage	setMarginAvarage(value: number)	void
isForecast	isForecast(month: MonthData)	boolean
isAchievedOrCurrent	isAchievedOrCurrent(month: MonthData)	boolean
getpriceForecastAmount	getpriceForecastAmount(month: MonthData)	number
expectedCurrencyChange	expectedCurrencyChange(e: undefined)	void
selectText	selectText(e: undefined)	void
mouseoverRowSpan	mouseoverRowSpan(e: undefined, bIndex: number)	void
mouseoutRowSpan	mouseoutRowSpan(e: undefined, bIndex: number)	void
editSelectedMonthPrepare	editSelectedMonthPrepare(month: MonthData)	void
sorting	sorting(propertyName: string, propertyType: string, monthIndex: number)	void

Properties

Property	Type
scrollOne	ElementRef
scrollTwo	ElementRef
scrollThree	ElementRef
_widget	Widget
_widgetType	string
withoutOptionBar	boolean
editMode	boolean
bookingOrSales	number
withNominal	boolean
selectedTeam	number
forecastRuleId	number
allowedtoEdit	boolean
allowedtoEnableEditing	boolean
widgetEditRequest	any
widgetEditEnd	any
selectedData	BranchData[]
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]
originalData	BranchData[]
currencyRateData	currencyRatesData[]
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean

Property	Type
subscription	Subscription
SerializeRange	any[]
echartsIntance	any
totalBookingMonths	TotalBookingMonths[]
tempTotalMonth	TotalBookingMonths
sortingDirection	number

Where it refuses work

- `MrmCashPositionComponent` stops the work with an early return when `a[propertyName] < b[propertyName]` , in 2 places.
- `MrmCashPositionComponent` stops the work with an early return when `a[propertyName] > b[propertyName]` , in 2 places.
- `MrmCashPositionComponent` stops the work with an early return when `month.MonthId == 12` .
- `MrmCashPositionComponent` stops the work with an early return when `this.allowedtoEnableEdting` .
- `MrmCashPositionComponent` stops the work with an early return when `this.allowedtoEdit` .
- `MrmCashPositionComponent` stops the work with an early return when `month[propertyName] < this.data[bIndex].MrmCashPositions[index - 1][propertyName]` .

Diagram

```
graph LR
  View[Dashboard View] --> Component[MrmCashPositionComponent]
  Component --> Auth[GetAuthorizationAccess]
  Component --> Init[initializingData]
  Component --> Load[getDashboardData]
  Component --> Rates[getCurrencyRates]
  Rates --> Monthly[calculateSumByMonth]
  Monthly --> Performance[calculatePerformancePer]
  Component --> Save[saveDashboardData]
  Component --> Destroy[ngOnDestroy]
```

Usage

```
import { TestBed } from '@angular/core/testing';
import { MrmCashPositionComponent } from './mrm-cash-position.component';

const fixture = TestBed.createComponent(MrmCashPositionComponent);
const cashPositionWidget = fixture.componentInstance;

// Runs Angular lifecycle hooks and loads initial widget state.
fixture.detectChanges();

// Refresh dashboard data when required by the host dashboard.
cashPositionWidget.getDashboardData();

fixture.detectChanges();

const performance = cashPositionWidget.calculatePerformancePer();
console.log('Cash position performance:', performance);
```

AI Coding Instructions

- Keep data loading inside `getDashboardData()` and initialization logic inside `initializingData()` so lifecycle methods remain focused on orchestration.
- Check authorization through `GetAuthorizationAccess()` before exposing or loading cash-position data.
- Refresh currency-rate-dependent calculations through `getCurrencyRates()` before calling `calculateSumByMonth()` or `calculatePerformancePer()`.
- Preserve the existing `initializingData()` method name when calling or refactoring it, including its current spelling.
- Release subscriptions, timers, or other resources in `ngOnDestroy()` when adding asynchronous dashboard behavior.

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`

