

MrmDataTableComponent

August 17, 2026

MrmDataTableComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/mrm/mrm-booking/mrm-data-table/mrm-data-table.component.ts (line 45)

Part of: Pams

MrmDataTableComponent manages data displayed in the MRM booking data table. It retrieves dashboard data, initializes table state, calculates performance values, formats month and price output, handles resize events, and stores cached data during its Angular lifecycle.

Implements: OnInit , OnChanges , OnDestroy

Methods

Method	Signature	Returns
ngOnInit	ngOnInit()	void
ngOnChanges	ngOnChanges()	void
ngOnDestroy	ngOnDestroy()	void
getDashboardData	getDashboardData()	void
initializingData	initializingData(value: BranchData[])	void
calculatePerformancePer	calculatePerformancePer(month: MonthData)	number
translateMonthName	translateMonthName(month: MonthData)	string
getpriceFormatted	getpriceFormatted(x: undefined)	string
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void

Method	Signature	Returns
onValueChanged	onValueChanged(e: undefined)	void
onChartInit	onChartInit(e: any, index: number)	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
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

Properties

Property	Type
triangleNeedle	DxCircularGaugeComponent
_widget	Widget
_widgetType	string
withoutOptionBar	boolean
bookingOrSales	number
withNominal	boolean
selectedTeam	number
forecastRuleId	number
forecastRules	any[]
selectedYear	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	BranchData[]

Property	Type
originalData	BranchData[]
singleMonths	MonthData
branchCurrency	any
basedOnTendency	boolean
subscription	Subscription
SeriseRange	any[]
echartsIntance	any
totalBookingMonths	TotalBookingMonths[]
tempTotalMonth	TotalBookingMonths
totalTargets	number
yearBooking	number
totalForecasts	number
tempTotalTarget	TotalTargets

Where it refuses work

- MrmDataTableComponent stops the work with an early return when `month.Month == 12` .
- MrmDataTableComponent stops the work with an early return when `month.Month >= new Date().getMonth() + 1` .
- MrmDataTableComponent stops the work with an early return when `month.Month <= new Date().getMonth() + 1` .

Diagram

```
graph LR
    Angular[Angular lifecycle] --> Init[ngOnInit]
    Angular --> Changes[ngOnChanges]
    Angular --> Destroy[ngOnDestroy]

    Init --> Dashboard[getDashboardData]
    Dashboard --> Initialize[initializingData]
    Initialize --> Performance[calculatePerformancePer]
    Initialize --> Month[translateMonthName]
    Initialize --> Price[getpriceFormatted]
```

```
Resize[onResize] --> Initialize
Initialize --> Cache[setChaching]
```

Usage

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { MrmDataTableComponent } from './mrm-data-table.component';

describe('MrmDataTableComponent', () => {
  let fixture: ComponentFixture<MrmDataTableComponent>;
  let component: MrmDataTableComponent;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [MrmDataTableComponent],
    }).compileComponents();

    fixture = TestBed.createComponent(MrmDataTableComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('loads and formats dashboard table data', () => {
    component.getDashboardData();

    const performance = component.calculatePerformancePer();
    const monthName = component.translateMonthName();
    const formattedPrice = component.getpriceFormatted();

    expect(performance).toBeDefined();
    expect(monthName).toBeDefined();
    expect(formattedPrice).toBeDefined();
  });
});
```

AI Coding Instructions

- Keep data loading and state preparation coordinated through `getDashboardData()` and `initializingData()` .
- Preserve the existing `initializingData` method name when calling or refactoring it, even though its spelling differs from `initializing` .
- Recalculate display values through `calculatePerformancePer()` , `translateMonthName()` , and `getpriceFormatted()` rather than duplicating formatting logic in templates.
- Handle viewport-dependent table updates through `onResize()` and clean up subscriptions or listeners in `ngOnDestroy()` .

- Review cache behavior in `setChaching()` when changing dashboard data sources or table refresh flows.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `TranslateHandler` — `Pams/Web/Pams.Web/Client/app/app.module.ts :1`