

WidgetBookingVsTargetVsForecastByMonthsComponent

August 17, 2026

WidgetBookingVsTargetVsForecastByMonthsComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/widget-booking-vs-target-vs-forecast-by-months/widget-booking-vs-target-vs-forecast-by-months.component.ts (line 45)

Part of: Pams

WidgetBookingVsTargetVsForecastByMonthsComponent is an Angular dashboard widget component for presenting booking, target, and forecast information by month. It loads dashboard and team lookup data, prepares series ranges, calculates performance percentages, and translates month names for display.

Implements: OnInit , OnChanges

Methods

Method	Signature	Returns
ngOnInit	ngOnInit()	void
ngDoCheck	ngDoCheck()	void
ngOnChanges	ngOnChanges()	void
ngOnDestroy	ngOnDestroy()	void
getDashboardData	getDashboardData()	void
initalizingData	initalizingData(value: BranchData[])	void
getTeamLookup	getTeamLookup()	void
setSeriseRange	setSeriseRange()	any[]
calculatePerformancePer	calculatePerformancePer(month: MonthData)	number
translateMonthName	translateMonthName(month: MonthData)	string
getpriceFormatted	getpriceFormatted(x: undefined)	string
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void
onValueChanged	onValueChanged(e: undefined)	void
onChartInit	onChartInit(e: any, index: number)	void

Method	Signature	Returns
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
setValue	setValue(value: number)	void
setForecastValue	setForecastValue(value: number)	void
setMarginAvarage	setMarginAvarage(value: number)	void
setMarginAvarageColor	setMarginAvarageColor(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
originalWidget	Widget
_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[]
originalData	BranchData[]
singleMonths	MonthData
singleBranch	BranchData
branchCurrency	any

Property	Type
basedOnTendency	boolean
subscription	Subscription
ForecastedDeviationPercentage	number
SeriseRange	any[]
initOpts	any
series	any[]
headerData	any
echartsIntance	any
options	any
TargetBooking	`string
AchievedBooking	`string
teamsLookup	any[]

Where it refuses work

- `WidgetBookingVsTargetVsForecastByMonthsComponent` stops the work with an early return when `a < b` .
- `WidgetBookingVsTargetVsForecastByMonthsComponent` stops the work with an early return when `a > b` .
- `WidgetBookingVsTargetVsForecastByMonthsComponent` stops the work with an early return when `month.MonthId == 12` .
- `WidgetBookingVsTargetVsForecastByMonthsComponent` stops the work with an early return when `month.MonthId >= new Date().getMonth() + 1` .
- `WidgetBookingVsTargetVsForecastByMonthsComponent` stops the work with an early return when `month.MonthId <= new Date().getMonth() + 1` .
- `WidgetBookingVsTargetVsForecastByMonthsComponent` stops the work with an early return when `month.MonthId < new Date().getMonth() + 1` .

Diagram

```
graph LR
    Lifecycle[Lifecycle[Angular lifecycle hooks]]
    Lifecycle --> Init[Init[ngOnInit]]
    Lifecycle --> Changes[Changes[ngOnChanges]]
    Lifecycle --> Check[Check[ngDoCheck]]
    Lifecycle --> Destroy[Destroy[ngOnDestroy]]

    Init --> Load[Load[getDashboardData]]
    Changes --> Load
    Load --> Initialize[Initialize[initializingData]]
    Initialize --> Teams[Teams[getTeamLookup]]
    Initialize --> Series[Series[setSeriseRange]]
    Initialize --> Performance[Performance[calculatePerformancePer]]
    Initialize --> Months[Months[translateMonthName]]
```

```
Series --> Widget[Monthly booking vs target vs forecast view]
Performance --> Widget
Months --> Widget
```

Usage

```
import { WidgetBookingVsTargetVsForecastByMonthsComponent } from './widget-booking-vs-target-vs-forecast-by-month

function refreshMonthlyWidget(
  component: WidgetBookingVsTargetVsForecastByMonthsComponent
): void {
  component.getDashboardData();

  const seriesRanges: any[] = component.setSeriseRange();
  const performancePercentage: number = component.calculatePerformancePer();
  const monthLabel: string = component.translateMonthName();

  console.log({
    seriesRanges,
    performancePercentage,
    monthLabel,
  });
}
```

AI Coding Instructions

- Let Angular call `ngOnInit`, `ngOnChanges`, `ngDoCheck`, and `ngOnDestroy`; keep lifecycle-specific setup and cleanup in those methods.
- Keep dashboard retrieval in `getDashboardData()` and initialize derived widget state through `initalizingData()`.
- Preserve the existing `setSeriseRange()` method name when updating callers, even though its spelling differs from “series.”
- Update team-dependent filtering through `getTeamLookup()` before building monthly display data.
- Keep `calculatePerformancePer()` and `translateMonthName()` focused on presentation-ready values used by the widget.

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`