

WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent

August 17, 2026

WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/widget-booking-vs-target-vs-forecast-by-months-by-variants/widget-booking-vs-target-vs-forecast-by-months-by-variants.component.ts (line 34)

Part of: Pams

WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent manages a dashboard widget that compares booking values with target and forecast data across months and variants. It loads team lookup data, calculates performance percentages, formats month and price values, responds to resize events, and stores widget state through caching methods.

Implements: OnInit

Methods

Method	Signature	Returns
ngOnInit	ngOnInit()	void
ngDoCheck	ngDoCheck()	void
ngOnDestroy	ngOnDestroy()	void
getTeamLookup	getTeamLookup()	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
onValueChanged	onValueChanged(e: undefined)	void
onChartInit	onChartInit(e: any, index: number)	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void
setValue	setValue(value: number)	void

Properties

Property	Type
triangleNeedle	DxCircularGaugeComponent
BoVsTaWidget	Widget
BoVsFoWidget	Widget
FoVsTaWidget	Widget

Property	Type
_widget	Widget
_widgetType	string
forecastRuleId	number
forecastRules	any[]
bookingOrSales	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
data	Data
months	MonthData[]
singleMonths	MonthData
branchCurrency	any
withNominal	boolean
basedOnTendency	boolean
subscription	Subscription
initOpts	any
series	any[]
headerData	any
echartsIntance	any
TargetBooking	`string
AchievedBooking	`string
teamsLookup	any[]
selectedTeam	number

Where it refuses work

- WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent stops the work with an early return when `month.MonthId == 12` .

Diagram

```
graph LR
    Dashboard[Dashboard] --> Widget[WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent]
    Widget --> Lifecycle[Lifecycle[Angular lifecycle]]
    Lifecycle --> Init[Init[ngOnInit]]
    Lifecycle --> Check[Check[ngDoCheck]]
    Lifecycle --> Destroy[Destroy[ngOnDestroy]]
    Widget --> TeamLookup[TeamLookup[getTeamLookup]]
    Widget --> Metrics[Metrics[calculatePerformancePer]]
    Widget --> Formatting[Formatting[translateMonthName]]
    Widget --> FormattingPrice[FormattingPrice[getpriceFormatted]]
    Widget --> Resize[Resize[onResize]]
    Widget --> Cache[Cache[setChaching]]
    Cache --> Restore[Restore[getChaching]]
```

Usage

```
import { WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent } from './widget-booking-vs-target-vs-forecast-by-months-l

function readWidgetValues(
  widget: WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent,
) {
  widget.getTeamLookup();

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

  widget.setChaching();

  return {
    performance,
    monthName,
    formattedPrice,
  };
}
```

AI Coding Instructions

- Let Angular invoke `ngOnInit` , `ngDoCheck` , and `ngOnDestroy` ; keep initialization and cleanup logic within the matching lifecycle method.
- Call `getTeamLookup()` before rendering data that depends on team selection or team-specific values.
- Keep calculations in `calculatePerformancePer()` consistent with the booking, target, and forecast data supplied to the widget.
- Use `translateMonthName()` and `getpriceFormatted()` for display values rather than duplicating formatting logic in the template.
- Preserve the existing cache flow by pairing `setChaching()` with `getChaching()` when adding persisted widget state.

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`