

WidgetBookingVsTargetVsForecastByMonthsComponent

August 18, 2026

WidgetBookingVsTargetVsForecastByMonthsComponent

Kind: Class

Source: Frontend/src/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 43)

Part of: Frontend

WidgetBookingVsTargetVsForecastByMonthsComponent manages dashboard data for a month-based comparison of bookings, targets, and forecasts. It checks feature subscription access, loads dashboard and team lookup data, prepares chart series ranges, calculates performance percentages, and translates month names for display.

Extends: BaseWidgetComponent

Implements: OnInit , OnChanges

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
ngDoCheck	ngDoCheck()	void
ngOnChanges	ngOnChanges()	void
getDashboardData	getDashboardData()	void
initializingData	initializingData(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

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
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

Property	Type
singleBranch	BranchData
branchCurrency	any
basedOnTendency	boolean
ForecastedDeviationPercentage	number
SeriseRange	any[]
initOpts	any
series	any[]
headerData	any
echartsIntance	any
options	any
TargetBooking	`string
AchievedBooking	`string
teamsLookup	any[]
currenFeature	number

Where it refuses work

- WidgetBookingVsTargetVsForecastByMonthsComponent stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- 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` .

Diagram

```
graph LR
  A[Angular lifecycle hooks] --> B[checkFeatureSubscription]
  A --> C[getDashboardData]
  C --> D[initalizingData]
  D --> E[getTeamLookup]
  D --> F[setSeriseRange]
  F --> G[Booking vs Target vs Forecast chart]
  D --> H[calculatePerformancePer]
  D --> I[translateMonthName]
```

Usage

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

function refreshWidgetData(
  widget: WidgetBookingVsTargetVsForecastByMonthsComponent
): void {
  if (!widget.checkFeatureSubscription()) {
    return;
  }

  widget.ngOnInit();
  widget.getDashboardData();

  const series = widget.setSeriseRange();
  const performance = widget.calculatePerformancePer();
  const monthLabel = widget.translateMonthName();

  console.log({ series, performance, monthLabel });
}
```

AI Coding Instructions

- Keep subscription checks in place before requesting or presenting dashboard data.
- Preserve the Angular lifecycle flow through `ngOnInit`, `ngOnChanges`, and `ngDoCheck` when changing refresh behavior.
- Keep `initializingData` and `setSeriseRange` aligned with the data shape returned by `getDashboardData`.
- Update `getTeamLookup` when adding team-based filters or lookup dependencies.
- Preserve the existing method spelling `initializingData` and `setSeriseRange` unless all call sites are updated.

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)

- `DashboardWidgetsModule` — `Frontend/src/app/components/dashboard/dashboard-widgets.module.ts :1`