

WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent

August 19, 2026

WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent

Kind: Class

Source: Frontend/src/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 32)

Part of: Frontend

WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent is an Angular dashboard widget component for presenting booking, target, and forecast data across months and variants. It checks feature subscription access, loads team lookup data, calculates performance, formats display values, translates month names, and manages cached widget state.

Extends: BaseWidgetComponent

Implements: OnInit

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
ngDoCheck	ngDoCheck()	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

Property	Type
FoVsTaWidget	Widget
_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
initOpts	any
series	any[]
headerData	any
echartsIntance	any
TargetBooking	`string
AchievedBooking	`string
teamsLookup	any[]
selectedTeam	number
currenFeature	number

Where it refuses work

- WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent stops the work with an early return when `month.MonthId == 12` .

Diagram

```
graph LR
  A[Angular lifecycle] --> B[ngOnInit]
  A --> C[ngDoCheck]
  B --> D[checkFeatureSubscription]
  B --> E[getTeamLookup]
  C --> F[getChaching]
  C --> G[setChaching]
  E --> H[Booking / Target / Forecast widget data]
  H --> I[calculatePerformancePer]
  H --> J[translateMonthName]
  H --> K[getpriceFormatted]
```

```
L[Window resize] --> M[onResize]
M --> H
```

Usage

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent } from './widget-booking-vs-target-vs-forecast-by-months-l

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

  beforeEach(() => {
    fixture = TestBed.createComponent(
      WidgetBookingVsTargetVsForecastByMonthsByVariantsComponent,
    );
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('calculates and formats widget values', () => {
    const hasAccess = component.checkFeatureSubscription();
    const performance = component.calculatePerformancePer();
    const formattedPrice = component.getPriceFormatted();

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

AI Coding Instructions

- Keep subscription checks in `checkFeatureSubscription()` before loading or displaying restricted widget data.
- Use `getTeamLookup()` as the integration point for team-related lookup data instead of duplicating lookup logic in lifecycle hooks.
- Preserve the existing cache method names, `setChaching()` and `getChaching()`, when updating cache-related behavior.
- Recalculate display state through existing methods such as `calculatePerformancePer()`, `translateMonthName()`, and `getpriceFormatted()` rather than formatting values directly in templates.
- Keep resize-related display updates inside `onResize()` so chart or layout refresh behavior remains in one place.

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`