

WidgetForecastVsTargetThisYearComponent

August 18, 2026

WidgetForecastVsTargetThisYearComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-forecast-vs-target-this-year/widget-forecast-vs-target-this-year.component.ts (line 30)

Part of: Frontend

WidgetForecastVsTargetThisYearComponent manages the dashboard widget that compares this year's forecast against its target. It checks subscription access, loads dashboard data, prepares chart values, formats pricing, and responds to forecast or chart-type switches.

Extends: BaseWidgetComponent

Implements: OnInit

Methods

Method	Signature	Returns
ngOnInit	ngOnInit()	void
checkFeatureSubscription	checkFeatureSubscription()	boolean
getDashboardData	getDashboardData()	void
initializingData	initializingData(value: MonthData[])	void
getpriceFormatted	getpriceFormatted(x: undefined)	string
calculatForecastVsTarget	calculatForecastVsTarget(value: MonthData)	number
switchIsForecastHandler	switchIsForecastHandler()	void
switchChartTypeHandler	switchChartTypeHandler()	void
onChartInit	onChartInit(e: any, index: number)	void
setValue	setValue(value: number)	void

Properties

Property	Type
_widget	Widget
withNominal	boolean
selectedYear	number
todayWidgetsEnum	any
Branches	BranchessTendancy[]
MonthData	MonthData[]
IsForecast	boolean
initOpts	any
series	any[]
headerData	any
echartsIntance	any
options	any
TargetBooking	`string
AchievedBooking	`string
currenFeature	number

Where it refuses work

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

Diagram

```
graph LR
    Dashboard[Dashboard] --> Widget[WidgetForecastVsTargetThisYearComponent]
    Widget --> Subscription[Subscription[checkFeatureSubscription]]
    Widget --> Data[Data[getDashboardData]]
    Data --> Initialize[Initialize[italizingData]]
    Initialize --> Calculation[Calculation[calculatForecastVsTarget]]
    Calculation --> Chart[Chart[onChartInit]]
    Widget --> ForecastSwitch[ForecastSwitch[switchIsForecastHandler]]
    Widget --> ChartSwitch[ChartSwitch[switchChartTypeHandler]]
```

```
Widget --> Formatting[getpriceFormatted]
Widget --> Values[setValue]
```

Usage

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { WidgetForecastVsTargetThisYearComponent } from './widget-forecast-vs-target-this-year.

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

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

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

  it('loads and updates forecast data', () => {
    component.ngOnInit();

    if (component.checkFeatureSubscription()) {
      component.getDashboardData();
      component.switchIsForecastHandler();
      component.switchChartTypeHandler();

      const formattedPrice = component.getpriceFormatted();
      const forecastVsTarget = component.calculatForecastVsTarget();

      expect(formattedPrice).toBeDefined();
      expect(forecastVsTarget).toBeDefined();
    }
  });
});
```

AI Coding Instructions

- Keep subscription checks in place before requesting or displaying widget data.
- Call `initalizingData()` after dashboard data is available so chart values reflect the current response.
- Update chart state through `switchIsForecastHandler()`, `switchChartTypeHandler()`, and `setValue()` rather than changing chart-related fields directly.

- Preserve the existing `getpriceFormatted()` and `calculatForecastVsTarget()` behavior when changing displayed values.

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`