

WidgetForecastVsTargetThisYearComponent

August 18, 2026

WidgetForecastVsTargetThisYearComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/widget-forecast-vs-target-this-year/widget-forecast-vs-target-this-year.component.ts (line 31)

Part of: [Pams](#)

`WidgetForecastVsTargetThisYearComponent` manages the dashboard widget that compares this year's forecast with its target. It loads dashboard data, prepares display values, calculates the forecast-versus-target result, and responds to forecast and chart-type switches.

Implements: `OnInit`

Methods

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

Properties

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

Diagram

```
graph LR
    Dashboard[Dashboard data source] --> Load[Load[getDashboardData]]
    Load --> Init[Init[initializingData]]
    Init --> Calculate[Calculate[calculatForecastVsTarget]]
    Init --> Format[Format[getpriceFormatted]]
    ToggleForecast[ToggleForecast[switchIsForecastHandler]] --> Calculate
    ToggleChart[ToggleChart[switchChartTypeHandler]] --> Chart[Chart instance]
    ChartInit[ChartInit[onChartInit]] --> Chart
    SetValue[SetValue[setValue]] --> Chart
    Destroy[Destroy[ngOnDestroy]] --> Chart
```

Usage

```

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

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('updates the forecast view', () => {
  component.ngOnInit();
  component.switchIsForecastHandler();
  component.switchChartTypeHandler();

  const formattedPrice = component.getPriceFormatted();
  const comparison = component.calculatForecastVsTarget();

  expect(formattedPrice).toBeDefined();
  expect(comparison).toBeDefined();
});

```

AI Coding Instructions

- Call `getDashboardData()` and `initalizingData()` through the component lifecycle flow rather than duplicating dashboard-data preparation.
- Keep forecast comparison logic in `calculatForecastVsTarget()` so displayed values and chart values remain aligned.
- Update chart state through `onChartInit()` and `setValue()` after the chart instance is available.
- Preserve cleanup behavior in `ngOnDestroy()` when adding subscriptions, timers, or chart resources.
- Keep the existing `initalizingData` and `calculatForecastVsTarget` method names when calling them, including their current spelling.

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