

WidgetTeamsTendencyComponent

August 18, 2026

WidgetTeamsTendencyComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/widget-teams-tendency/widget-teams-tendency.component.ts (line 21)

Part of: Pams

WidgetTeamsTendencyComponent manages dashboard data and chart behavior for the teams tendency widget. It handles data loading, cached state, value changes, chart initialization, resize events, lifecycle cleanup, and chart type switching.

Implements: OnInit , DoCheck

Methods

Method	Signature	Returns
getDashboardData	getDashboardData()	void
onResize	onResize()	void
setChaching	setChaching()	void
getChaching	getChaching()	void
onValueChanged	onValueChanged(e: undefined)	void
ngOnInit	ngOnInit()	void
ngDoCheck	ngDoCheck()	void
onChartInit	onChartInit(e: any, index: number)	void
ngOnDestroy	ngOnDestroy()	void
switchChartTypeHandler	switchChartTypeHandler()	void
switchBasedOnTendencyHandler	switchBasedOnTendencyHandler()	void

Method	Signature	Returns
setValue	setValue(value: number)	void

Properties

Property	Type
triangleNeedle	DxCircularGaugeComponent
originalWidget	Widget
_widget	Widget
numYears	number
numYearsDatasource	any[]
todayWidgetsEnum	any
cachingWidgetData	CachingWidgetData
teams	TeamsTendancy[]
withNominal	boolean
basedOnTendency	boolean
subscription	Subscription
initOpts	any
series	any[]
headerData	any
echartsIntance	any
options	any
TargetBooking	`string
AchievedBooking	`string

Where it refuses work

- WidgetTeamsTendancyComponent stops the work with an early return when isNumber(value) .

Diagram

```
graph LR
  Dashboard[Dashboard] --> WidgetTeamsTendancyComponent
  WidgetTeamsTendancyComponent --> Data[getDashboardData]
  WidgetTeamsTendancyComponent --> Cache[setChaching / getChaching]
  WidgetTeamsTendancyComponent --> Chart[onChartInit]
  WidgetTeamsTendancyComponent --> Resize[onResize]
  WidgetTeamsTendancyComponent --> ChartType[switchChartTypeHandler]
  WidgetTeamsTendancyComponent --> Cleanup[ngOnDestroy]
```

Usage

```
import { AfterViewInit, Component, ViewChild } from '@angular/core';
import { WidgetTeamsTendancyComponent } from './widget-teams-tendancy.component';

@Component({
  selector: 'app-dashboard-host',
  template: `
    <app-widget-teams-tendancy #teamsWidget></app-widget-teams-tendancy>
    <button type="button" (click)="refreshTeamsWidget()">Refresh</button>
  `,
})
export class DashboardHostComponent implements AfterViewInit {
  @ViewChild('teamsWidget')
  teamsWidget?: WidgetTeamsTendancyComponent;

  ngAfterViewInit(): void {
    this.teamsWidget?.getDashboardData();
  }

  refreshTeamsWidget(): void {
    this.teamsWidget?.getDashboardData();
  }
}
```

AI Coding Instructions

- Keep dashboard data retrieval inside `getDashboardData()` so refresh behavior remains in one place.
- Use `setChaching()` and `getChaching()` together when preserving widget state between dashboard updates.
- Do not call Angular lifecycle methods such as `ngOnInit()`, `ngDoCheck()`, or `ngOnDestroy()` from application code.
- Route chart setup through `onChartInit()` and chart mode changes through `switchChartTypeHandler()`.

- Keep resize-related chart updates in `onResize()` to avoid duplicating chart layout logic.

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`