

WidgetPrincipalsTendencyComponent

August 17, 2026

WidgetPrincipalsTendencyComponent

Kind: Class

Source: Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/widget-principals-tendency/widget-principals-tendency.component.ts (line 19)

Part of: [Pams](#)

WidgetPrincipalsTendencyComponent manages the principals tendency dashboard widget, including data retrieval, chart initialization, and value updates. It participates in the Angular component lifecycle by loading dashboard data on initialization and releasing resources during destruction.

Implements: OnInit

Methods

Method	Signature	Returns
getDashboardData	getDashboardData()	void
ngOnInit	ngOnInit()	void
onChartInit	onChartInit(e: any, index: number)	void
ngOnDestroy	ngOnDestroy()	void
setValue	setValue(value: number)	void

Properties

Property	Type
_widget	Widget
todayWidgetsEnum	any
Principals	PrincipalsTendency[]

Property	Type
subscription	Subscription
initOpts	any
series	any[]
headerData	any
echartsIntance	any
options	any
TargetBooking	`string
AchievedBooking	`string

Diagram

```
graph LR
    Dashboard[Dashboard View] --> Widget[WidgetPrincipalsTendancyComponent]
    Widget --> Init[ngOnInit]
    Init --> Load[getDashboardData]
    Load --> Values[setValue]
    Chart[Chart Integration] --> ChartInit[onChartInit]
    ChartInit --> Widget
    Widget --> Destroy[ngOnDestroy]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { WidgetPrincipalsTendancyComponent } from
  './components/dashboard/dashboard-library/widgets/widget-principals-tendancy/widget-principals

@Component({
  selector: 'app-dashboard-page',
  template: `
    <!-- Render WidgetPrincipalsTendancyComponent in this dashboard template -->
    <button type="button" (click)="refreshPrincipalsTendancy()">
      Refresh
    </button>
  `
})
export class DashboardPageComponent {
  @ViewChild(WidgetPrincipalsTendancyComponent)
  private principalsTendancyWidget?: WidgetPrincipalsTendancyComponent;

  refreshPrincipalsTendancy(): void {
```

```
this.principalsTendencyWidget?.getDashboardData();  
}  
}
```

AI Coding Instructions

- Let Angular invoke `ngOnInit` and `ngOnDestroy` ; do not call lifecycle methods directly from application code.
- Use `getDashboardData()` when the widget must refresh its dashboard content.
- Connect the chart integration to `onChartInit()` so chart-dependent state is available before values are applied.
- Keep value-mapping logic inside `setValue()` to avoid duplicating widget state updates in parent components.

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