

WidgetPrincipalsTendencyComponent

August 18, 2026

WidgetPrincipalsTendencyComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-principals-tendency/widget-principals-tendency.component.ts (line 17)

Part of: Frontend

WidgetPrincipalsTendencyComponent manages dashboard data and chart presentation for principal tendency information. It checks feature subscription access, initializes the chart, and derives values, colors, and chart height for rendering.

Extends: BaseWidgetComponent

Implements: OnInit

Methods

Method	Signature	Returns
getDashboardData	getDashboardData()	void
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
onChartInit	onChartInit(e: any, index: number)	void
setValue	setValue(value: number)	void
setColor	setColor(value: number)	string
chartHeight	chartHeight()	string

Properties

Property	Type
<code>_widget</code>	<code>Widget</code>
<code>todayWidgetsEnum</code>	<code>any</code>
<code>Principals</code>	<code>PrincipalsTendency[]</code>
<code>initOpts</code>	<code>any</code>
<code>series</code>	<code>any[]</code>
<code>headerData</code>	<code>any</code>
<code>echartsIntance</code>	<code>any</code>
<code>options</code>	<code>any</code>
<code>TargetBooking</code>	<code>`string</code>
<code>AchievedBooking</code>	<code>`string</code>
<code>currenFeature</code>	<code>number</code>

Where it refuses work

- `WidgetPrincipalsTendencyComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `WidgetPrincipalsTendencyComponent` stops the work with an early return when `value > 150` .

Diagram

```
graph LR
  Init[ngOnInit] --> Subscription[checkFeatureSubscription]
  Subscription -->|Allowed| Data[getDashboardData]
  Data --> ChartInit[onChartInit]
  ChartInit --> Value[setValue]
  ChartInit --> Color[setColor]
  ChartInit --> Height[chartHeight]
  Subscription -->|Not allowed| Restricted[Render restricted state]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { WidgetPrincipalsTendencyComponent } from '../widget-principals-tendency.component';
```

```

@Component({
  selector: 'app-dashboard-host',
  template: `
    <app-widget-principals-tendancy
      #principalsWidget>
    </app-widget-principals-tendancy>
  `,
})
export class DashboardHostComponent {
  @ViewChild('principalsWidget')
  principalsWidget!: WidgetPrincipalsTendancyComponent;

  refreshWidget(): void {
    if (this.principalsWidget.checkFeatureSubscription()) {
      this.principalsWidget.getDashboardData();
      this.principalsWidget.onChartInit();
    }
  }
}

```

AI Coding Instructions

- Keep feature-access checks through `checkFeatureSubscription()` before requesting or rendering subscription-dependent dashboard data.
- Call `getDashboardData()` as part of component initialization or refresh flows; avoid duplicating its data-loading logic in parent components.
- Keep chart setup in `onChartInit()` and use `setValue()`, `setColor()`, and `chartHeight()` for chart-specific display state.
- Preserve Angular lifecycle behavior by keeping initialization work in `ngOnInit()` rather than requiring callers to construct the component directly.

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`