

WidgetBranchTendencyComponent

August 19, 2026

WidgetBranchTendencyComponent

Kind: Class

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

Part of: Pams

`WidgetBranchTendencyComponent` is an Angular dashboard widget responsible for loading and presenting branch tendency data in a chart. It initializes chart state, applies incoming values, refreshes dashboard data, and releases chart-related resources when the component is destroyed.

Implements: `OnInit`

Methods

Method	Signature	Returns
<code>getDashboardData</code>	<code>getDashboardData()</code>	<code>void</code>
<code>ngOnInit</code>	<code>ngOnInit()</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
<code>_widget</code>	<code>Widget</code>
<code>todayWidgetsEnum</code>	<code>any</code>
<code>Branches</code>	<code>BranchessTendency[]</code>

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

Diagram

```
graph LR
    Dashboard[Dashboard View] --> WidgetBranchTendancyComponent
    WidgetBranchTendancyComponent --> ngOnInit
    ngOnInit --> getDashboardData
    WidgetBranchTendancyComponent --> onChartInit
    getDashboardData --> setValue
    setValue --> Chart[Branch Tendency Chart]
    WidgetBranchTendancyComponent --> ngOnDestroy
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { WidgetBranchTendancyComponent } from './widget-branch-tendancy.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-widget-branch-tendancy></app-widget-branch-tendancy>

    <button type="button" (click)="refreshBranchTendency()">
      Refresh
    </button>
  `,
})
export class DashboardPageComponent {
  @ViewChild(WidgetBranchTendancyComponent)
  private branchTendencyWidget!: WidgetBranchTendancyComponent;

  refreshBranchTendency(): void {
```

```
this.branchTendencyWidget.getDashboardData();  
}  
}
```

AI Coding Instructions

- Keep initialization logic in `ngOnInit` and cleanup logic in `ngOnDestroy` .
- Call `onChartInit` only after the chart instance is available from the chart integration.
- Route refreshed dashboard responses through `setValue` so chart state is updated in one place.
- Avoid retaining chart instances, subscriptions, or timers after `ngOnDestroy` .
- Preserve the widget's role as a dashboard presentation component; fetch and map only the data needed for the branch tendency chart.

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