

WidgetBookingSummaryTodayComponent

August 18, 2026

WidgetBookingSummaryTodayComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-booking-summary-today/widget-booking-summary-today.component.ts (line 36)

Part of: Frontend

WidgetBookingSummaryTodayComponent is an Angular dashboard widget that loads and displays booking summary data for the current view. It checks feature subscription access, synchronizes branch-specific views, formats pricing values, and manages chart initialization and planned-visit interactions.

Extends: BaseWidgetComponent

Implements: OnInit

Methods

Method	Signature	Returns
checkFeatureSubscription	checkFeatureSubscription()	boolean
ngOnInit	ngOnInit()	void
getDashboardData	getDashboardData()	void
displayDashboardData	displayDashboardData()	void
syncWithSeparateBranchesViews	syncWithSeparateBranchesViews()	void
getpriceFormatted	getpriceFormatted(x: undefined)	string
onChartInit	onChartInit(e: any)	void
onPlannedVisitsChartClick	onPlannedVisitsChartClick(e: undefined)	void
setValue	setValue(value: number)	void

Properties

Property	Type
<code>_widget</code>	<code>Widget</code>
<code>todayWidgetsEnum</code>	<code>any</code>
<code>allData</code>	<code>MonthData[]</code>
<code>data</code>	<code>Data[]</code>
<code>tempData</code>	<code>Data</code>
<code>separateBranchesViews</code>	<code>boolean</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>number</code>
<code>AchievedBooking</code>	<code>number</code>
<code>ForecastBooking</code>	<code>number</code>
<code>BlindForecastBooking</code>	<code>number</code>
<code>TargetBookingFormatted</code>	<code>any</code>
<code>AchievedBookingFormatted</code>	<code>any</code>
<code>ForecastBookingFormatted</code>	<code>any</code>
<code>BlindForecastFormatted</code>	<code>any</code>
<code>branchTendency</code>	<code>number</code>
<code>branchCurrency</code>	<code>any</code>
<code>currenFeature</code>	<code>number</code>

Where it refuses work

- `WidgetBookingSummaryTodayComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .

Diagram

```
graph LR
  Init[ngOnInit] --> Subscription[checkFeatureSubscription]
  Subscription --> Data[getDashboardData]
  Data --> Display[displayDashboardData]
  Display --> Branches[syncWithSeparateBranchesViews]
  Display --> Price[getpriceFormatted]
  Display --> Chart[onChartInit]
  Chart --> Click[onPlannedVisitsChartClick]
  Click --> Value[setValue]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { WidgetBookingSummaryTodayComponent } from '../widget-booking-summary-today.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-widget-booking-summary-today
      #bookingSummary>
    </app-widget-booking-summary-today>

    <button type="button" (click)="refreshBookingSummary()">
      Refresh
    </button>
  `,
})
export class DashboardPageComponent {
  @ViewChild('bookingSummary')
  bookingSummary!: WidgetBookingSummaryTodayComponent;

  refreshBookingSummary(): void {
    if (this.bookingSummary.checkFeatureSubscription()) {
      this.bookingSummary.getDashboardData();
    }
  }

  get formattedPrice(): string {
    return this.bookingSummary.getpriceFormatted();
  }
}
```

AI Coding Instructions

- Check `checkFeatureSubscription()` before requesting or displaying booking summary data.
- Keep data retrieval in `getDashboardData()` and presentation updates in `displayDashboardData()` .
- Call `syncWithSeparateBranchesViews()` when branch selection or branch-scoped dashboard data changes.
- Initialize chart behavior through `onChartInit()` and route planned-visit selections through `onPlannedVisitsChartClick()` .
- Keep price display formatting inside `getpriceFormatted()` rather than formatting values in templates.

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`