WidgetBacklogOverTheMonthsComponent

August 18, 2026

WidgetBacklogOverTheMonthsComponent

Kind: Class

Source: `Frontend/src/app/components/dashboard/dashboard-libary/widgets/widget-backlog-over-the-months/widget-backlog-over-the-months.component.ts` (line 44)

Part of: Frontend

`WidgetBacklogOverTheMonthsComponent` is an Angular dashboard widget that loads and prepares backlog data for display across months. It checks feature subscription access, retrieves dashboard and team lookup data, builds chart series ranges, and formats month labels and performance values.

Extends: `BaseWidgetComponent`

Implements: `OnInit`

Methods

Method	Signature	Returns
`checkFeatureSubscription`	`checkFeatureSubscription()`	`boolean`
`ngOnInit`	`ngOnInit()`	`void`
`ngDoCheck`	`ngDoCheck()`	`void`
`ngOnChanges`	`ngOnChanges()`	`void`
`getDashboardData`	`getDashboardData()`	`void`
`initalizingData`	`initalizingData(value: BranchData[])`	`void`
`getTeamLookup`	`getTeamLookup()`	`void`
`setSeriseRange`	`setSeriseRange()`	`any[]`
`calculatePerformancePer`	`calculatePerformancePer(month: BacklogData)`	`number`
`translateMonthName`	`translateMonthName(month: BacklogData)`	`string`

Method	Signature	Returns
`getpriceFormatted`	`getpriceFormatted(x: undefined)`	`string`
`onResize`	`onResize()`	`void`
`setChaching`	`setChaching()`	`void`
`getChaching`	`getChaching()`	`void`
`onValueChanged`	`onValueChanged(e: undefined)`	`void`
`onChartInit`	`onChartInit(e: any, index: number)`	`void`
`switchChartTypeHandler`	`switchChartTypeHandler()`	`void`
`switchBasedOnTendencyHandler`	`switchBasedOnTendencyHandler()`	`void`
`setValue`	`setValue(value: number)`	`void`
`setForecastValue`	`setForecastValue(value: number)`	`void`
`setMarginAvarage`	`setMarginAvarage(value: number)`	`void`
`setMarginAvarageColor`	`setMarginAvarageColor(value: number)`	`void`
`isForecast`	`isForecast(month: BacklogData)`	`boolean`
`isAchievedOrCurrent`	`isAchievedOrCurrent(month: BacklogData)`	`boolean`
`getpriceForecastAmount`	`getpriceForecastAmount(month: BacklogData)`	`number`
`expectedCurrencyChange`	`expectedCurrencyChange(e: undefined)`	`void`

Properties

Property	Type
`triangleNeedle`	`DxCircularGaugeComponent`
`originalWidget`	`Widget`
`_widget`	`Widget`
`_widgetType`	`string`
`withoutOptionBar`	`boolean`
`bookingOrSales`	`number`
`withNominal`	`boolean`
`selectedTeam`	`number`
`forecastRuleId`	`number`

Property	Type
`forecastRules`	`any[]`
`selectedYear`	`number`
`numYearsDatasource`	`any[]`
`todayWidgetsEnum`	`any`
`cachingWidgetData`	`CachingWidgetData`
`data`	`BranchData[]`
`originalData`	`BranchData[]`
`singlePrincipal`	`BacklogData`
`singleBacklogMonth`	`MonthData`
`singleBranch`	`BranchData`
`branchCurrency`	`any`
`basedOnTendency`	`boolean`
`SeriseRange`	`any[]`
`initOpts`	`any`
`headerData`	`any`
`echartsIntance`	`any`
`TargetBooking`	`string
`AchievedBooking`	`string
`teamsLookup`	`any[]`
`currenFeature`	`number`
`value`	`BranchData[]`

Where it refuses work

- `WidgetBacklogOverTheMonthsComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)`.
- `WidgetBacklogOverTheMonthsComponent` stops the work with an early return when `month.MonthId == 12`.
- `WidgetBacklogOverTheMonthsComponent` stops the work with an early return when `month.MonthId >= new Date().getMonth() + 1`.

- `WidgetBacklogOverTheMonthsComponent` stops the work with an early return when `month.MonthId <= new Date().getMonth() + 1`.

Diagram

```
graph LR
  A[Angular lifecycle] --> B[WidgetBacklogOverTheMonthsComponent]
  B --> C[checkFeatureSubscription]
  C --> D[getDashboardData]
  D --> E[initalizingData]
  E --> F[getTeamLookup]
  E --> G[setSeriseRange]
  G --> H[calculatePerformancePer]
  G --> I[translateMonthName]
  H --> J[Dashboard chart output]
  I --> J
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { WidgetBacklogOverTheMonthsComponent } from './widget-backlog-over-the-months.component

@Component({
  selector: 'app-dashboard-host',
  template: `
    <app-widget-backlog-over-the-months
      #backlogWidget>
    </app-widget-backlog-over-the-months>
  `
})
export class DashboardHostComponent {
  @ViewChild('backlogWidget')
  backlogWidget!: WidgetBacklogOverTheMonthsComponent;

  refreshBacklogData(): void {
    if (this.backlogWidget.checkFeatureSubscription()) {
      this.backlogWidget.getDashboardData();
    }
  }
}
```

AI Coding Instructions

- Keep data loading behind `checkFeatureSubscription()` so unavailable features do not request dashboard data.

- Preserve Angular lifecycle behavior in `ngOnInit`, `ngDoCheck`, and `ngOnChanges`; avoid moving lifecycle work into unrelated event handlers.
- Call `initalizingData()` after data retrieval when updating the widget data state; retain the existing method spelling for compatibility.
- Keep chart series construction in `setSeriseRange()` and use `translateMonthName()` for month labels rather than formatting month values in multiple locations.
- Treat `getTeamLookup()` as the integration point for team-related filtering or lookup data used by the dashboard widget.

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