

WidgetTopOffersByMarketSegmentsLastNyearsComponent

August 17, 2026

WidgetTopOffersByMarketSegmentsLastNyearsComponent

Kind: Class**Source:** `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/widget-top-offers-by-market-segments-last-nyears/widget-top-offers-by-market-segments-last-nyears.component.ts` (line 22)**Part of:** [Pams](#)

`WidgetTopOffersByMarketSegmentsLastNyearsComponent` is an Angular dashboard widget that loads and displays top offers grouped by market segment across a selected year range. It reacts to lifecycle and value changes, initializes chart behavior, handles chart clicks, assigns year colors, and stores or retrieves cached widget state.

Implements: `OnInit`

Methods

Method	Signature	Returns
<code>getDashboardData</code>	<code>getDashboardData()</code>	<code>void</code>
<code>afterResponse</code>	<code>afterResponse(Data: any[])</code>	<code>void</code>
<code>ngOnChanges</code>	<code>ngOnChanges()</code>	<code>void</code>
<code>ngOnInit</code>	<code>ngOnInit()</code>	<code>void</code>
<code>setYearColor</code>	<code>setYearColor(year: number)</code>	<code>string</code>
<code>onChartInit</code>	<code>onChartInit(e: any)</code>	<code>void</code>
<code>onPlannedVisitsChartClick</code>	<code>onPlannedVisitsChartClick(e: undefined)</code>	<code>void</code>
<code>setChaching</code>	<code>setChaching()</code>	<code>void</code>
<code>getChaching</code>	<code>getChaching()</code>	<code>void</code>
<code>onValueChanged</code>	<code>onValueChanged(e: undefined)</code>	<code>void</code>
<code>switchChartTypeHandler</code>	<code>switchChartTypeHandler()</code>	<code>void</code>
<code>switchChartRefernceCommissionHandler</code>	<code>switchChartRefernceCommissionHandler()</code>	<code>void</code>
<code>toggoleOnlyCurrentOffers</code>	<code>toggoleOnlyCurrentOffers()</code>	<code>void</code>
<code>toggoleBudgetOffers</code>	<code>toggoleBudgetOffers()</code>	<code>void</code>
<code>toggoleFirmOffers</code>	<code>toggoleFirmOffers()</code>	<code>void</code>
<code>switchFirmOffers</code>	<code>switchFirmOffers()</code>	<code>void</code>
<code>switchBudgetOffers</code>	<code>switchBudgetOffers()</code>	<code>void</code>

Method	Signature	Returns
nextPage	nextPage()	void
prevPage	prevPage()	void
ngOnDestroy	ngOnDestroy()	void

Properties

Property	Type
_widget	Widget
meAsVendor	boolean
isRefernceCommission	boolean
withoutOptionBar	boolean
onlyCurrentOffers	boolean
numYears	number
_FirmOffers	boolean
_BudgetOffers	boolean
numYearsDatasource	any[]
todayWidgetsEnum	any
subscription	Subscription
branchName	string
maxValueForYaxis	number
initOpts	any
series	any[]
headerData	any[]
echartsIntance	any
cachingWidgetData	CachingWidgetData
originalDatasource	WidgetTopOffers[]
datasource	WidgetTopOffers[]
originalHeader	any[]
options	any
skip	number

Diagram

```
graph LR
  Inputs --> ngOnChanges
  ngOnInit --> getChaching
  getChaching --> getDashboardData
```

```

ngOnChanges --> getDashboardData
onValueChanged --> setChaching
onValueChanged --> getDashboardData
getDashboardData --> afterResponse
afterResponse --> Chart
onChartInit --> Chart
setYearColor --> Chart
Chart --> onPlannedVisitsChartClick

```

Usage

```

import { AfterViewInit, Component, ViewChild } from '@angular/core';
import { WidgetTopOffersByMarketSegmentsLastNyearsComponent } from './widget-top-offers-by-market-segments-last-nyears';

@Component({
  selector: 'app-dashboard-host',
  template: `
    <app-widget-top-offers-by-market-segments-last-nyears>
    </app-widget-top-offers-by-market-segments-last-nyears>
  `,
})
export class DashboardHostComponent implements AfterViewInit {
  @ViewChild(WidgetTopOffersByMarketSegmentsLastNyearsComponent)
  offersWidget!: WidgetTopOffersByMarketSegmentsLastNyearsComponent;

  ngAfterViewInit(): void {
    this.offersWidget.getChaching();
    this.offersWidget.getDashboardData();
  }

  refreshOffers(): void {
    this.offersWidget.onValueChanged();
  }
}

```

AI Coding Instructions

- Keep data loading in `getDashboardData()` and apply response-dependent state updates in `afterResponse()` .
- Call `setChaching()` when widget selections change so cached state matches the current chart state.
- Use `getChaching()` during initialization before requesting dashboard data when cached selections are supported.
- Keep chart setup in `onChartInit()` and route chart interaction handling through `onPlannedVisitsChartClick()` .
- Use `setYearColor()` for chart series colors rather than assigning year colors in multiple locations.

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `TranslateHandler` — `Pams/Web/Pams.Web/Client/app/app.module.ts :1`

- `WidgetTopOffersByVariantsComponent` — `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-library/widgets/widget-top-offers-by-variants/widget-top-offers-by-variants.component.ts :1`