

WidgetForecastBacklogPaymentComponent

August 17, 2026

WidgetForecastBacklogPaymentComponent

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-forecast-backlog-payment/widget-forecast-backlog-payment.component.ts (line 26)

Part of: Frontend

WidgetForecastBacklogPaymentComponent manages dashboard data for the forecast backlog payment widget. It checks feature subscription access, loads dashboard or My Disk data, prepares chart state, and handles chart click events.

Extends: BaseWidgetComponent

Implements: OnInit

Methods

Method	Signature	Returns
viewData	viewData()	void
getDashboardData	getDashboardData()	void
getMyDiskData	getMyDiskData()	void
checkFeatureSubscription	checkFeatureSubscription()	boolean
afterResponse	afterResponse(Data: WidgetBacklogInvoices[])	void
borderColor	borderColor(stage: string)	string
ngOnInit	ngOnInit()	void
onChartInit	onChartInit(e: any)	void
onSalesPipelineChartClick	onSalesPipelineChartClick(e: undefined)	void

Method	Signature	Returns
onSalesPipelineChartInMyDiskTabClick	onSalesPipelineChartInMyDiskTabClick(e: undefined)	void
switchCommissionInvoicesOff	switchCommissionInvoicesOff()	void
switchSalesInvoicesOff	switchSalesInvoicesOff()	void

Properties

Property	Type
_widget	Widget
_myDiskFilterObj	MyDiskFilterObj
ChartInMyDiskTabClick	any
todayWidgetsEnum	any
maxValueForYaxis	number
initOpts	any
series	any[]
header	any[]
data	any[]
datasource	WidgetBacklogInvoices[]
originalDatasource	WidgetBacklogInvoices[]
echartsIntance	any
CommissionInvoices	boolean
SalesInvoices	boolean
options	any
currenFeature	number
seriesData	any
headerData	any[]
showDataOnChart	boolean

Where it refuses work

- `WidgetForecastBacklogPaymentComponent` stops the work with an early return when `this.dashboardLibraryService.checkFeatureSubscription(this.widget.chartEnumId)` .
- `WidgetForecastBacklogPaymentComponent` stops the work with an early return when `stage == "Overdue"` .

Diagram

```
graph LR
  Init[ngOnInit] --> Subscription[checkFeatureSubscription]
  Subscription -->|Allowed| View[viewData]
  View --> Dashboard[getDashboardData]
  View --> MyDisk[getMyDiskData]
  Dashboard --> Response[afterResponse]
  MyDisk --> Response
  Response --> Chart[onChartInit]
  Chart --> Color[borderColor]
  Chart --> PipelineClick[onSalesPipelineChartClick]
  Chart --> MyDiskClick[onSalesPipelineChartInMyDiskTabClick]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { WidgetForecastBacklogPaymentComponent } from '../widget-forecast-backlog-payment.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <app-widget-forecast-backlog-payment
      #forecastBacklogPayment
    ></app-widget-forecast-backlog-payment>

    <button (click)="refreshWidget()">Refresh</button>
  `,
})
export class DashboardPageComponent {
  @ViewChild('forecastBacklogPayment')
  forecastBacklogPayment?: WidgetForecastBacklogPaymentComponent;

  refreshWidget(): void {
    if (this.forecastBacklogPayment?.checkFeatureSubscription()) {
      this.forecastBacklogPayment.viewData();
    }
  }
}
```

AI Coding Instructions

- Check `checkFeatureSubscription()` before requesting or displaying widget data.
- Use `viewData()` as the entry point for refreshing the widget instead of calling data-loading methods from unrelated components.
- Keep response-dependent chart setup in `afterResponse()` and `onChartInit()`.
- Route chart interactions through `onSalesPipelineChartClick()` or `onSalesPipelineChartInMyDiskTabClick()` based on the active data source.
- Keep chart border styling centralized in `borderColor()`.

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`