MrmAccountReceivableComponent

August 18, 2026

MrmAccountReceivableComponent

Kind: Class

Source: `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-libary/widgets/mrm-account-receivable/mrm-account-receivable.component.ts` (line 74)

Part of: Pams

`MrmAccountReceivableComponent` is an Angular dashboard widget for presenting and maintaining MRM account-receivable dashboard data. It retrieves authorization and currency data, initializes or loads dashboard state, calculates totals by year and month, and saves dashboard updates.

Methods

Method	Signature	Returns
`getCurrencies`	`getCurrencies()`	`void`
`GetAuthorizationAccess`	`GetAuthorizationAccess(url: string)`	`void`
`ngOnInit`	`ngOnInit()`	`void`
`ngOnChanges`	`ngOnChanges()`	`void`
`ngOnDestroy`	`ngOnDestroy()`	`void`
`getDashboardData`	`getDashboardData()`	`void`
`initalizingData`	`initalizingData(value: BranchData)`	`void`
`saveDashboardData`	`saveDashboardData()`	`void`
`calculateSumByYear`	`calculateSumByYear()`	`void`
`calculateSumByMonth`	`calculateSumByMonth(monthIndex: number)`	`void`
`AddNewCurrency`	`AddNewCurrency(account: number, SalesCom: number)`	`void`

Method	Signature	Returns
`onCurrencyChanged`	`onCurrencyChanged(e: undefined, account: number, SalesCom: number, currencyIndex: number)`	`void`
`translateMonthName`	`translateMonthName(MonthId: number)`	`string`
`getpriceFormatted`	`getpriceFormatted(x: undefined)`	`string`
`onResize`	`onResize()`	`void`
`setChaching`	`setChaching()`	`void`
`getChaching`	`getChaching()`	`void`
`switchByGroupsHandler`	`switchByGroupsHandler()`	`void`
`switchBasedOnTendencyHandler`	`switchBasedOnTendencyHandler()`	`void`
`comparingWithPrevMonth`	`comparingWithPrevMonth(month: MonthData, accountIndex: number, salesComIndex: number, currencyIndex: number, monthIndex: number, propertyName: string)`	`string`
`comparingSummeryWithPrevMonth`	`comparingSummeryWithPrevMonth(month: MonthData, monthIndex: number, propertyName: string)`	`string`
`isCurrentMonth`	`isCurrentMonth(month: MonthData)`	`boolean`
`getpriceForecastAmount`	`getpriceForecastAmount(month: MonthData)`	`number`
`expectedCurrencyChange`	`expectedCurrencyChange(e: undefined)`	`void`
`selectText`	`selectText(e: undefined)`	`void`
`editSelectedMonthPrepare`	`editSelectedMonthPrepare(month: MonthData, supplierId: number)`	`void`
`mouseoverRowSpan`	`mouseoverRowSpan(e: undefined, bIndex: number)`	`void`
`mouseoutRowSpan`	`mouseoutRowSpan(e: undefined, bIndex: number)`	`void`
`openPopup`	`openPopup(n: string, t: string, i: string, btnTxt: string)`	`void`
`savePopup`	`savePopup()`	`void`
`ClosewithDonotSave`	`ClosewithDonotSave()`	`void`
`OnOpenedPopup`	`OnOpenedPopup()`	`void`

Method	Signature	Returns
`closePopup`	`closePopup()`	`void`

Properties

Property	Type
`sharedPopup`	`SharedPopupComponent`
`_widget`	`Widget`
`_widgetType`	`string`
`withoutOptionBar`	`boolean`
`editMode`	`boolean`
`bookingOrSales`	`number`
`byGroups`	`boolean`
`selectedTeam`	`number`
`forecastRuleId`	`number`
`data`	`BranchData`
`allowedtoEdit`	`boolean`
`allowedtoEnableEdting`	`boolean`
`widgetEditRequest`	`any`
`widgetEditEnd`	`any`
`forecastRules`	`any[]`
`currencies`	`CompanyCurrency[]`
`selectedYear`	`number`
`currentMonthIndex`	`number`
`numYearsDatasource`	`any[]`
`todayWidgetsEnum`	`any`
`cachingWidgetData`	`CachingWidgetData`
`originalData`	`BranchData`
`selectedData`	`BranchData`
`singleMonths`	`MonthData`

Property	Type
`monthsSummery`	`MonthsSum[]`
`nonGroupAccountReceivables`	`MonthData[]`
`branchCurrency`	`any`
`basedOnTendency`	`boolean`
`subscription`	`Subscription`
`SeriseRange`	`any[]`
`echartsIntance`	`any`
`totalTargets`	`number`
`yearBooking`	`number`
`totalForecasts`	`number`
`tempTotalTarget`	`TotalTargets`
`totalProfitAndLossAmount`	`number`
`totalOthers`	`number`
`totalTotalSGAndA`	`number`
`totalFORX`	`number`
`totalTotal`	`number`
`totalCumulative`	`number`
`totalPercentageUsedBudget`	`number`
`popup`	`any`
`searchText`	`string`

Where it refuses work

- `MrmAccountReceivableComponent` stops the work with an early return when `MonthId == 12`.
- `MrmAccountReceivableComponent` stops the work with an early return when `month[propertyName] < this.data.Products.Sales[currencyIndex].Details[monthIndex - 1][pro…`.
- `MrmAccountReceivableComponent` stops the work with an early return when `month[propertyName] < this.data.Products.Commission[currencyIndex].Details[monthIndex -`

`1…` .

- `MrmAccountReceivableComponent` stops the work with an early return when `month[propertyName] < this.data.Scrap.Sales[currencyIndex].Details[monthIndex - 1][proper…` .
- `MrmAccountReceivableComponent` stops the work with an early return when `month[propertyName] < this.data.Scrap.Commission[currencyIndex].Details[monthIndex - 1][p…` .
- `MrmAccountReceivableComponent` stops the work with an early return when `month[propertyName] < this.monthsSummery[monthIndex - 1][propertyName]` .

Diagram

```
graph LR
    Lifecycle[Angular lifecycle] --> Init[ngOnInit]
    Lifecycle --> Changes[ngOnChanges]
    Init --> Authorization[GetAuthorizationAccess]
    Init --> Currencies[getCurrencies]
    Authorization --> Data[getDashboardData]
    Currencies --> Data
    Data --> Initialize[initalizingData]
    Initialize --> YearTotals[calculateSumByYear]
    Initialize --> MonthTotals[calculateSumByMonth]
    YearTotals --> Save[saveDashboardData]
    MonthTotals --> Save
    Lifecycle --> Destroy[ngOnDestroy]
```

Usage

```
import { Component, ViewChild } from '@angular/core';
import { MrmAccountReceivableComponent } from './mrm-account-receivable.component';

@Component({
  selector: 'app-dashboard-page',
  template: `
    <!-- Render MrmAccountReceivableComponent in this dashboard view -->
  `
})
export class DashboardPageComponent {
  @ViewChild(MrmAccountReceivableComponent)
  accountReceivableWidget?: MrmAccountReceivableComponent;

  refreshAccountReceivableData(): void {
    this.accountReceivableWidget?.getDashboardData();
  }
```

```
  saveAccountReceivableData(): void {
    this.accountReceivableWidget?.saveDashboardData();
  }
}
```

AI Coding Instructions

- Let Angular manage `ngOnInit`, `ngOnChanges`, and `ngOnDestroy`; avoid invoking lifecycle methods directly from application code.
- Keep authorization and currency retrieval within the existing initialization flow before processing dashboard data.
- Update `calculateSumByYear` and `calculateSumByMonth` together when changing account-receivable aggregation rules.
- Route persisted dashboard changes through `saveDashboardData` rather than adding separate save paths.
- Preserve cleanup behavior in `ngOnDestroy` when adding subscriptions, timers, or other component resources.

Used by

3 references from 3 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (3)

- `TranslateHandler` — `Pams/Web/Pams.Web/Client/app/app.module.ts`:1
- `DashboardTabComponent` — `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-libary/dashboard-tab/dashboard-tab.component.ts`:1
- `ReportsTabComponent` — `Pams/Web/Pams.Web/Client/app/components/dashboard/reports-library/reports-tab/reports-tab.component.ts`:1