DashboardLibraryService

August 18, 2026

DashboardLibraryService

Kind: Class

Source: `Frontend/src/app/services/dashboard/dashboard-library.service.ts` (line 228)

Part of: Frontend

`DashboardLibraryService` manages dashboard widgets, tab mappings, role-based dashboard data, and saved dashboard changes. It connects dashboard UI state with widget data, subscription checks, role endpoints, and dashboard persistence methods.

Methods

Method	Signature	Returns
`getAllWidgets`	`getAllWidgets()`	`Widget[]`
`mappingDashboard`	`mappingDashboard(OldDashboardTabs: OldDashboardTabs[], disabling: boolean, isReport: boolean)`	`DashboardTabs[]`
`unmappingDashboard`	`unmappingDashboard(newModelDashboard: DashboardTabs[])`	`OldDashboardTabs[]`
`checkFeatureSubscription`	`checkFeatureSubscription(chartEnumId: undefined)`	`boolean`
`getWidgetData`	`getWidgetData(widgetEnum: number, useNominal: boolean, useLocalVendor: boolean, yearId: number, yearEnum: number, useTendency: boolean, skip: number, take: number, teamId: number, widgetTargetId: number, forecastRuleId: number, OnlyActiveOffers: boolean, useForecast: boolean, getRateByVal: boolean)`	`void`
`getRolesDashboard`	`getRolesDashboard()`	`void`
`getRoleDashboardById`	`getRoleDashboardById(id: number)`	`void`
`savelatestmodifications`	`savelatestmodifications(logHistory: DashboardLog)`	`void`
`saveRoleDashboard`	`saveRoleDashboard(roleDashboard: RoleDashboard)`	`void`
`saveDashboardTabs`	`saveDashboardTabs(tabs: DashboardTabs[])`	`void`
`getDashboardTabs`	`getDashboardTabs()`	`void`
`changeActiveDashboard`	`changeActiveDashboard(roleId: number)`	`void`
`toggoleSeparateBranchesViews`	`toggoleSeparateBranchesViews(separateBranchesViews: boolean)`	`void`

Method	Signature	Returns
`onReportsParamsChanged`	`onReportsParamsChanged(reportsParamsChanged: ReportParams)`	`void`
`getRolesReports`	`getRolesReports()`	`void`
`getRoleReportsById`	`getRoleReportsById(id: number)`	`void`
`getReportsTabs`	`getReportsTabs()`	`void`
`saveRoleReports`	`saveRoleReports(roleDashboard: RoleDashboard)`	`void`
`savelatestmodificationsForReport`	`savelatestmodificationsForReport(logHistory: DashboardLog)`	`void`
`changeActiveReport`	`changeActiveReport(roleId: number)`	`void`
`checkForcountOfEditablePrevMonthsRole`	`checkForcountOfEditablePrevMonthsRole(yearId: number, monthId: number, countOfPrevMonths: number)`	`boolean`
`getBookingByProductType`	`getBookingByProductType(useNominal: boolean, yearId: number, periodId: number)`	`void`
`getBookingByTeam`	`getBookingByTeam(useNominal: boolean, yearId: number, periodId: number)`	`void`
`getBookingBySupplier`	`getBookingBySupplier(yearId: number, periodId: number)`	`void`
`getMRMBookingByNonSalesTeam`	`getMRMBookingByNonSalesTeam(yearId: number, periodId: number, useNominal: boolean)`	`void`
`saveMRMBookingByNonSalesTeam`	`saveMRMBookingByNonSalesTeam(monthData: any[])`	`void`
`getMRMBacklogByProductType`	`getMRMBacklogByProductType(vendorTypeId: number, yearId: number, periodId: number)`	`void`
`getMRMForLGs`	`getMRMForLGs(yearId: number, periodId: number)`	`void`
`getMRMWrehouse`	`getMRMWrehouse(yearId: number, periodId: number)`	`void`
`saveMRMWrehouse`	`saveMRMWrehouse(monthData: any[])`	`void`
`getMRMScrapStock`	`getMRMScrapStock(yearId: number, periodId: number)`	`void`
`saveMRMScrapStock`	`saveMRMScrapStock(monthData: any[])`	`void`
`getMRMTotalSales`	`getMRMTotalSales(calculateBy: number, yearId: number, periodId: number)`	`void`
`saveMRMTotalSales`	`saveMRMTotalSales(monthData: any[])`	`void`
`getMRMCOGS`	`getMRMCOGS(yearId: number, periodId: number)`	`void`
`saveMRMCOGS`	`saveMRMCOGS(monthData: any[])`	`void`
`getMRMSGAndA`	`getMRMSGAndA(yearId: number, periodId: number)`	`void`
`saveMRMSGAndA`	`saveMRMSGAndA(monthData: any[])`	`void`
`getMRMDepreciation`	`getMRMDepreciation(yearId: number, periodId: number)`	`void`
`saveMRMDepreciation`	`saveMRMDepreciation(monthData: any[])`	`void`

Method	Signature	Returns
`getMRMProfitAndLoss`	`getMRMProfitAndLoss(yearId: number, periodId: number)`	`void`
`saveMRMProfitAndLoss`	`saveMRMProfitAndLoss(monthData: any[])`	`void`
`getMRMCurrenciesRate`	`getMRMCurrenciesRate(yearId: number, periodId: number)`	`void`
`saveMRMCurrenciesRate`	`saveMRMCurrenciesRate(monthData: any[])`	`void`
`getMRMPMBacklog`	`getMRMPMBacklog(yearId: number, periodId: number)`	`void`
`getMRMPMShipment`	`getMRMPMShipment(yearId: number, periodId: number)`	`void`
`getMRMCashPosition`	`getMRMCashPosition(yearId: number, periodId: number)`	`void`
`saveMRMCashPosition`	`saveMRMCashPosition(monthData: any[])`	`void`
`getMRMAccountReceivable`	`getMRMAccountReceivable(yearId: number, periodId: number)`	`void`
`saveMRMAccountReceivable`	`saveMRMAccountReceivable(monthData: any[])`	`void`

Properties

Property	Type
`yearDropdown`	`any[]`
`yearSinceJanDropdown`	`any[]`
`AccountTypes`	`any[]`
`noResize`	`boolean`
`noMove`	`boolean`
`forecastRules`	`any[]`
`_separateBranchesViews`	`boolean`
`separateBranchesViews`	`any`
`_reportsParamsChanged`	`ReportParams`
`reportsParamsChanged`	`any`
`allWidgets`	`Widget[]`
`MRMReportTabs`	`DashboardTabs[]`
`MRMWidgets`	`Widget[]`

Where it refuses work

- `DashboardLibraryService` stops the work with an early return when `AppSettings.ifSubscriptionInclude(ItemEnum)`.

Diagram

```
graph LR
  UI[Dashboard UI] --> Service[DashboardLibraryService]
  Service --> Widgets[getAllWidgets]
  Service --> Mapping[mappingDashboard]
  Service --> Unmapping[unmappingDashboard]
  Service --> Subscription[checkFeatureSubscription]
  Service --> Roles[getRolesDashboard]
  Service --> RoleById[getRoleDashboardById]
  Service --> Data[getWidgetData]
  Service --> Save[savelatestmodifications]
  Service --> SaveRole[saveRoleDashboard]
  Service --> SaveTabs[saveDashboardTabs]
```

Usage

```
import { Component, OnInit } from '@angular/core';
import { DashboardLibraryService } from
  'src/app/services/dashboard/dashboard-library.service';

@Component({
  selector: 'app-dashboard',
  template: ''
})
export class DashboardComponent implements OnInit {
  widgets = [];
  dashboardTabs = [];

  constructor(
    private readonly dashboardLibraryService: DashboardLibraryService
  ) {}

  ngOnInit(): void {
    if (this.dashboardLibraryService.checkFeatureSubscription()) {
      this.widgets = this.dashboardLibraryService.getAllWidgets();
      this.dashboardTabs = this.dashboardLibraryService.mappingDashboard();
      this.dashboardLibraryService.getRolesDashboard();
    }
  }

  saveChanges(): void {
    this.dashboardLibraryService.savelatestmodifications();
    this.dashboardLibraryService.saveDashboardTabs();
  }
}
```

AI Coding Instructions

- Call `checkFeatureSubscription()` before exposing dashboard features that depend on subscription state.
- Use `mappingDashboard()` for current dashboard tab data and `unmappingDashboard()` when working with legacy tab structures.
- Keep widget selection aligned with the `Widget[]` returned by `getAllWidgets()`.
- Use the save methods after dashboard, tab, or role changes so UI changes are persisted.

- Keep role dashboard calls grouped with dashboard access logic when using `getRolesDashboard()` or `getRoleDashboardById()`.

Used by

116 references from 116 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (116)

- `DashboardSettingsComponent` — `Frontend/src/app/components/configuration/dashboard-settings/dashboard-settings/dashboard-settings.component.ts`:1
- `DashboardSettingsDetailsComponent` — `Frontend/src/app/components/configuration/dashboard-settings/dashboard-settings-details/dashboard-settings-details.component.ts`:1
- `ReportsSettingsComponent` — `Frontend/src/app/components/configuration/reports-settings/reports-settings/reports-settings.component.ts`:1
- `ReportsSettingsDetailsComponent` — `Frontend/src/app/components/configuration/reports-settings/reports-settings-details/reports-settings-details.component.ts`:1
- `DashboardLibaryComponent` — `Frontend/src/app/components/dashboard/dashboard-libary/dashboard-libary.component.ts`:1
- `DashboardTabComponent` — `Frontend/src/app/components/dashboard/dashboard-libary/dashboard-tab/dashboard-tab.component.ts`:1
- `BranchData` — `Frontend/src/app/components/dashboard/dashboard-libary/widgets/mrm-account-payable/mrm-account-payable.component.ts`:1
- `BranchData` — `Frontend/src/app/components/dashboard/dashboard-libary/widgets/mrm-account-receivable/mrm-account-receivable.component.ts`:1

…and 108 more.