

ExportToExcelService

August 17, 2026

ExportToExcelService

Kind: Class

Source: Frontend/src/app/services/common/export-to-excel.service.ts (line 13)

Part of: Frontend

ExportToExcelService handles Excel export actions for dashboard widgets. It exposes separate methods for standard widget exports and advanced widget exports, allowing dashboard components to trigger the appropriate export flow.

Methods

Method	Signature	Returns
exportingDashboardWidgetToExcel	exportingDashboardWidgetToExcel(header: any[], data: any[], widgetName: string, isArrayIncludingTotal: boolean)	void
exportingDashboardWidgetToExcelAdvanced	exportingDashboardWidgetToExcelAdvanced(header: any[], headerProperty: string, data: any[], dataProperties: string[], widgetName: string)	void

Properties

Property	Type
xlxsdata	any[]
ExlRow	ExcelRow[]
header	any
value	any
subscription	Subscription

Diagram

```
graph LR
  DashboardWidget --> ExportToExcelService
  ExportToExcelService --> exportingDashboardWidgetToExcel
  ExportToExcelService --> exportingDashboardWidgetToExcelAdvanced
  exportingDashboardWidgetToExcel --> ExcelFile
  exportingDashboardWidgetToExcelAdvanced --> ExcelFile
```

Usage

```
import { ExportToExcelService } from '@app/services/common/export-to-excel.service';

export class DashboardComponent {
  constructor(private readonly exportToExcelService: ExportToExcelService) {}

  exportWidget(): void {
    this.exportToExcelService.exportingDashboardWidgetToExcel();
  }

  exportAdvancedWidget(): void {
    this.exportToExcelService.exportingDashboardWidgetToExcelAdvanced();
  }
}
```

AI Coding Instructions

- Inject `ExportToExcelService` through the Angular constructor rather than creating it directly.
- Call `exportingDashboardWidgetToExcel()` for the standard dashboard widget export path.
- Call `exportingDashboardWidgetToExcelAdvanced()` only for widgets that require the advanced export path.
- Keep widget-specific data preparation outside the service unless the export logic requires shared handling.

Used by

63 references from 63 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (63)

- `WidgetAccountsRunningOffersComponent` — `Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-accounts-running-offers/widget-accounts-running-offers.component.ts :1`
- `DataResponse` — `Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-aging-receivables/widget-aging-receivables.component.ts :1`

- `WidgetBacklogByAccountManagersComponent` — `Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-backlog-by-account-managers/widget-backlog-by-account-managers.component.ts :1`
- `WidgetBacklogByClientsComponent` — `Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-backlog-by-clients/widget-backlog-by-clients.component.ts :1`
- `WidgetBacklogByMarketSegmentsComponent` — `Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-backlog-by-market-segments/widget-backlog-by-market-segments.component.ts :1`
- `WidgetBacklogByMonthsComponent` — `Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-backlog-by-months/widget-backlog-by-months.component.ts :1`
- `WidgetBacklogByPrincipalsComponent` — `Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-backlog-by-principals/widget-backlog-by-principals.component.ts :1`
- `WidgetBacklogByProductTypesComponent` — `Frontend/src/app/components/dashboard/dashboard-library/widgets/widget-backlog-by-product-types/widget-backlog-by-product-types.component.ts :1`

...and 55 more.