Operations

August 17, 2026

Operations

Kind: Class

Source: `Pams/Web/Pams.Web/Client/app/components/dashboard/dashboard-operations.ts` (line 15)

Part of: Pams

`Operations` contains dashboard data helpers for converting enums, grouping and sorting records, preparing grouped data, and calculating derived values. It also includes display-oriented helpers for remaining-day values, sparkline data, and random colors.

Methods

Method	Signature	Returns
`EnumToArray`	`EnumToArray(Enum: any)`	`EnumItem[]`
`GroupBy_v1`	`GroupBy_v1(GroupFields: dash.GroupOperation[], DataSource: any[], removeNull: boolean)`	`any[]`
`GroupBy_Single`	`GroupBy_Single(GroupQuery: dash.Query, DataSource: any[], removeNull: boolean)`	`any[]`
`ConstructGroups_WithFields`	`ConstructGroups_WithFields(GroupFields: dash.GroupOperation[], DataSource: any[], Qtype: dash.QueryTypeEnum, removeNull: boolean, sortGroups: undefined)`	`PreparedDataGroups`
`SortGroups`	`SortGroups(Groups: any[], type: dash.SortingType)`	`void`
`SortGroupedObjectKeys`	`SortGroupedObjectKeys(data: any[], seperated: any[])`	`void`

Method	Signature	Returns
`getRemainingDays`	`getRemainingDays(CompareDate: any, OrderActivationDate: any, DaystoDelivery: number)`	`number`
`SparkLine`	`SparkLine(Field: dash.MeasureOperation, Agru: dash.GroupOperation, data: any[])`	`void`
`getRandomColor`	`getRandomColor()`	`void`
`CalculateExpression`	`CalculateExpression(exp: dash.CalculatedField, data: any[])`	`void`
`PrepareGroups`	`PrepareGroups(Queries: dash.Query[], Data: any[], removeNull: boolean)`	`any[]`
`PrepareSerise`	`PrepareSerise(Queries: dash.Query[], Data: any[], removeNull: boolean)`	`any[]`
`PrepareGroups_withFields`	`PrepareGroups_withFields(Queries: dash.Query[], Data: any[], removeNull: boolean, sortGroups: boolean)`	`PreparedDataGroups`
`PrepareSerise_withFields`	`PrepareSerise_withFields(Queries: dash.Query[], Data: any[], removeNull: boolean, sortGroups: boolean)`	`PreparedDataGroups`
`BuildGrid`	`BuildGrid(Quries: dash.Query[], Datasource: any[])`	`any[]`
`BuildPieChart`	`BuildPieChart(Quries: dash.Query[], Datasource: any[])`	`any[]`
`BuildPieChart_chartjs`	`BuildPieChart_chartjs(Quries: dash.Query[], DatasourceOrg: any[])`	`widgetData`
`BuildWidget`	`BuildWidget(widget: DashboardWidget)`	`void`
`BuildGauge`	`BuildGauge(Queries: dash.Query[], dataSource: any[])`	`void`
`reSortLabels`	`reSortLabels(widget: DashboardWidget)`	`void`
`construct_layer`	`construct_layer(data: any[], agrument: any)`	`any[]`
`buildActiveTotalChart`	`buildActiveTotalChart(Quries: dash.Query[], Datasource: any[])`	`any`
`buildFlatCharts`	`buildFlatCharts(Quries: dash.Query[], Datasource: any[])`	`any`

Method	Signature	Returns
`handleSerise`	`handleSerise(ser: dash.Query[], data: any[], argu: any)`	`any`
`cloneDeepNotIndexed`	`cloneDeepNotIndexed(arr: undefined)`	`void`
`ConcatDistinctroups`	`ConcatDistinctroups(T: DimensionField[], S: DimensionField[])`	`DimensionField[]`
`buildDigitChart`	`buildDigitChart(Quries: dash.Query[], DatasourceOrg: any[])`	`number`
`buildFlatCharts_chartjs`	`buildFlatCharts_chartjs(Quries: dash.Query[], DatasourceOrg: any[])`	`widgetData`
`buildExpBarChart`	`buildExpBarChart(Quries: dash.Query[], DatasourceOrg: any[])`	`widgetData`
`calculatExp`	`calculatExp(groups: dash.Query[], datasets: any[])`	`any`
`constructorExpLabels`	`constructorExpLabels(type: dash.DateGroupEnum)`	`void`
`arrangColors`	`arrangColors(dataset: any[])`	`void`
`ApplyFilters`	`ApplyFilters(widget: DashboardWidget, Updating: boolean)`	`void`
`formatDate`	`formatDate(date: any, Type: dash.DateGroupEnum, dashes: boolean)`	`void`
`FilterByString`	`FilterByString(FilterString: string, data: any[])`	`any[]`
`FilterAndGroup`	`FilterAndGroup(FilterString: string, GroupOptions: DashboardDataFields[], Data: any[], FilterFirst: undefined)`	`void`
`FilterLists`	`FilterLists(FilterList: any[], Datasource: any[])`	`any[]`
`GroupBy`	`GroupBy(fields: DashboardDataFields[], Data: any[])`	`any[]`

Properties

Property	Type
`MasterFilter`	`any[]`

Property	Type
`MasterFilterListIDs`	`any[]`

Where it refuses work

- `Operations` stops the work with an early return when `!groupField || !actual || !target`.
- `Operations` stops the work with an early return when `!groups || !groups.length`.
- `Operations` stops the work with an early return when `!actual`.
- `Operations` stops the work with an early return when `!labels`.
- `Operations` stops the work with an early return when `type == dash.DateGroupEnum.Month`.
- `Operations` stops the work with an early return when `FilterString.length > 3`.

Diagram

```
graph LR
  Operations --> EnumToArray
  Operations --> GroupBy_v1
  Operations --> GroupBy_Single
  Operations --> ConstructGroups_WithFields
  Operations --> SortGroups
  Operations --> SortGroupedObjectKeys
  Operations --> CalculateExpression
  Operations --> getRemainingDays
  Operations --> SparkLine
  Operations --> getRandomColor

  GroupBy_v1 --> ConstructGroups_WithFields
  GroupBy_Single --> ConstructGroups_WithFields
  ConstructGroups_WithFields --> SortGroups
  SortGroups --> SortGroupedObjectKeys
```

Usage

```
import { Operations } from "./components/dashboard/dashboard-operations";

const operations = new Operations();

const enumItems = operations.EnumToArray();
const groupedRecords = operations.GroupBy_v1();
const preparedGroups = operations.ConstructGroups_WithFields();
```

```
operations.SortGroups();
operations.SortGroupedObjectKeys();

const remainingDays = operations.getRemainingDays();
const sparkLineData = operations.SparkLine();
const color = operations.getRandomColor();
const calculatedValue = operations.CalculateExpression();

console.log({
  enumItems,
  groupedRecords,
  preparedGroups,
  remainingDays,
  sparkLineData,
  color,
  calculatedValue,
});
```

AI Coding Instructions

- Keep grouping logic inside `Operations` so dashboard components consume prepared group data rather than duplicate grouping behavior.
- Call sorting methods after building grouped data when the dashboard depends on ordered group output.
- Preserve the return shapes of `EnumToArray`, grouping methods, and `ConstructGroups_WithFields` when changing dashboard data handling.
- Treat `SparkLine`, `getRandomColor`, and `CalculateExpression` as display and derived-value helpers; keep their output compatible with current dashboard bindings.

Relationships

- IMPORTS → `DashboardDataFields`
- IMPORTS → `OperationTypeEnum`
- IMPORTS → `DateGroupEnum`
- IMPORTS → `FilterOptions`
- IMPORTS → `DataTypeEnum`
- IMPORTS → `Measure`
- IMPORTS → `data`
- IMPORTS → `EnumItem`
- IMPORTS → `DimensionField`

- IMPORTS → `PreparedDataGroups`
- IMPORTS → `DashboardWidgetTypeEnum`
- IMPORTS → `DateGroupEnum`
- IMPORTS → `DashboardWidget`
- IMPORTS → `widgetData`
- IMPORTS → `AppSettings`

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)

- `DashboardWidgetOperationsService` — `Pams/Web/Pams.Web/Client/app/services/dashboard/dashboard-widget-operations.service.ts`:1