

Operations

August 17, 2026

Operations

Kind: Class

Source: Frontend/src/app/components/dashboard/dashboard-operations.ts (line 15)

Part of: Frontend

`Operations` contains dashboard data helpers for converting enums, grouping and sorting records, preparing field-based groups, and calculating display values. It also includes helpers for remaining-day values, sparkline data, random colors, and expression calculation.

Methods

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

Method	Signature	Returns
	DaystoDelivery: 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[]
PrepareSerie	PrepareSerie(Queries: dash.Query[], Data: any[], removeNull: boolean)	any[]
PrepareGroups_withFields	PrepareGroups_withFields(Queries: dash.Query[], Data: any[], removeNull: boolean, sortGroups: boolean)	PreparedDataGroups
PrepareSerie_withFields	PrepareSerie_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
handleSerie	handleSerie(ser: dash.Query[], data: any[], argu: any)	any

Method	Signature	Returns
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[]
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 --> getRemainingDays
    Operations --> SparkLine
    Operations --> getRandomColor
    Operations --> CalculateExpression

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

Usage

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

const operations = new Operations();

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

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

const remainingDays = operations.getRemainingDays();
```

```
const sparkline = operations.SparkLine();
const color = operations.getRandomColor();
const calculatedValue = operations.CalculateExpression();

console.log({
  enumItems,
  groupedData,
  preparedGroups,
  remainingDays,
  sparkline,
  color,
  calculatedValue,
});
```

AI Coding Instructions

- Keep grouping changes aligned with `GroupBy_v1` , `GroupBy_Single` , and `ConstructGroups_WithFields` so prepared dashboard data keeps the expected shape.
- Apply sorting through `SortGroups` and `SortGroupedObjectKeys` after changing group construction logic.
- Treat `EnumToArray` output as `EnumItem[]` when binding enum values to dashboard controls.
- Keep display helpers such as `SparkLine` , `getRandomColor` , `getRemainingDays` , and `CalculateExpression` separate from grouping logic.

Relationships

- 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` — `Frontend/src/app/services/dashboard/dashboard-widget-operations.service.ts :1`