

NewJobService

August 17, 2026

NewJobService

Kind: Class**Source:** Frontend/src/app/services/sales/new-job/new-job.service.ts (line 17)**Part of:** Frontend

`NewJobService` manages data access and actions for sales job and inquiry workflows. It retrieves branch, responsible, product type, time frame, and status data sources, then saves, updates, or deletes job-related records.

Methods

Method	Signature	Returns
<code>getCurrentBranch</code>	<code>getCurrentBranch()</code>	<code>void</code>
<code>getSalesResponsiblesDataSource</code>	<code>getSalesResponsiblesDataSource(companyBranchForData: number)</code>	<code>void</code>
<code>getSalesResponsiblesDataSourceWithEnum</code>	<code>getSalesResponsiblesDataSourceWithEnum(departmentIdEnum: number, companyBranchForData: number)</code>	<code>void</code>
<code>getProductTypesDataSource</code>	<code>getProductTypesDataSource()</code>	<code>void</code>
<code>getPurchasingProductTypesDataSource</code>	<code>getPurchasingProductTypesDataSource()</code>	<code>void</code>
<code>getTimeFramePeriodsDataSource</code>	<code>getTimeFramePeriodsDataSource()</code>	<code>void</code>
<code>getStatusDataSource</code>	<code>getStatusDataSource(stage: undefined)</code>	<code>void</code>
<code>saveJob</code>	<code>saveJob(job: undefined)</code>	<code>void</code>
<code>updateJob</code>	<code>updateJob(job: undefined)</code>	<code>void</code>
<code>deleteInquiry</code>	<code>deleteInquiry(id: undefined)</code>	<code>void</code>
<code>getsalesjobbyid</code>	<code>getsalesjobbyid(id: undefined)</code>	<code>void</code>
<code>getDeliveriesForInstandSalesJob</code>	<code>getDeliveriesForInstandSalesJob(Jobid: undefined)</code>	<code>void</code>
<code>createInquiry</code>	<code>createInquiry(inquiry: undefined)</code>	<code>void</code>
<code>getInquiry</code>	<code>getInquiry(inquiryId: number)</code>	<code>void</code>
<code>updateInquiry</code>	<code>updateInquiry(inquiry: undefined)</code>	<code>void</code>
<code>deleteJob</code>	<code>deleteJob(id: undefined)</code>	<code>void</code>

Method	Signature	Returns
changeJobStatus	changeJobStatus(job: undefined)	void
regretJob	regretJob(id: undefined, regretDetails: undefined)	void
unRegretPotentialJob	unRegretPotentialJob(id: undefined)	void
cancelJob	cancelJob(id: undefined, cancelDetails: undefined)	void
unCancelPotentialJob	unCancelPotentialJob(id: undefined)	void
convertJobToOffer	convertJobToOffer(id: undefined)	void
regretOfferedJob	regretOfferedJob(id: undefined, regretDetails: undefined)	void
unRegretOfferedJob	unRegretOfferedJob(id: undefined)	void
cancelOfferedJob	cancelOfferedJob(id: undefined, cancelDetails: undefined)	void
unCancelOfferedJob	unCancelOfferedJob(id: undefined)	void
convertJobToPotential	convertJobToPotential(id: undefined)	void
LostOfferedJob	LostOfferedJob(id: number, LossDetails: undefined)	void
unLossOfferedJob	unLossOfferedJob(id: number)	void
unRejectOrderedJob	unRejectOrderedJob(id: number)	void
getSingleOffer	getSingleOffer(offerId: number)	void
createSingleOffer	createSingleOffer(offer: SalesOffers)	void
approveOfferArchiveRequest	approveOfferArchiveRequest(offerId: number)	void
disapproveOfferArchiveRequest	disapproveOfferArchiveRequest(offerId: number)	void
changeStageSingleOffer	changeStageSingleOffer(offer: SalesOffers)	void
requestSalesOfferPrices	requestSalesOfferPrices(offer: SalesOffers)	void
updateSingleOffer	updateSingleOffer(offer: SalesOffers)	void
regretSingleOffer	regretSingleOffer(id: number, regretDetails: undefined)	void
getregrettedOfferReason	getregrettedOfferReason(archivedId: number)	void
updateRegrettedOfferReason	updateRegrettedOfferReason(Id: number, regretDetails: RegretReasonModel)	void
unRegretSingleOffer	unRegretSingleOffer(id: number)	void
cancelSingleOffer	cancelSingleOffer(id: number, cancelDetails: undefined)	void
getCancelledOfferReason	getCancelledOfferReason(archivedId: number)	void
updateCancelledOfferReason	updateCancelledOfferReason(Id: number, cancelDetails: CancelReasonModel)	void
unCancelSingleOffer	unCancelSingleOffer(id: number)	void

Method	Signature	Returns
lostSingleOffer	lostSingleOffer(id: number, lostDetails: undefined)	void
getLostOfferReason	getLostOfferReason(archivedId: number)	void
updateLostOfferReason	updateLostOfferReason(Id: number, lostDetails: LostReasonModel)	void
unLostSingleOffer	unLostSingleOffer(id: number)	void
deleteSingleOffer	deleteSingleOffer(id: number)	void

Properties

Property	Type
datasourceBySearchEnum	any

Diagram

```
graph LR
  Component[Sales Job Component] --> Service[NewJobService]

  Service --> Branch[getCurrentBranch]
  Service --> SalesResponsibles[getSalesResponsiblesDataSource]
  Service --> SalesResponsiblesEnum[getSalesResponsiblesDataSourceWithEnum]
  Service --> ProductTypes[getProductTypesDataSource]
  Service --> PurchasingProductTypes[getPurchasingProductTypesDataSource]
  Service --> TimeFrames[getTimeFramePeriodsDataSource]
  Service --> Statuses[getStatusDataSource]

  Service --> Save[saveJob]
  Service --> Update[updateJob]
  Service --> Delete[deleteInquiry]
```

Usage

```
import { Component, inject } from '@angular/core';
import { NewJobService } from '@app/services/sales/new-job/new-job.service';

@Component({
  selector: 'app-new-job-form',
  template: `<!-- form template -->`,
})
export class NewJobFormComponent {
  private readonly newJobService = inject(NewJobService);

  readonly salesResponsibles =
    this.newJobService.getSalesResponsiblesDataSource();

  readonly productTypes =
    this.newJobService.getProductTypesDataSource();
```

```

readonly statuses =
  this.newJobService.getStatusDataSource();

save(job: unknown): void {
  this.newJobService.saveJob(job);
}

update(job: unknown): void {
  this.newJobService.updateJob(job);
}

deleteInquiry(inquiry: unknown): void {
  this.newJobService.deleteInquiry(inquiry);
}
}

```

AI Coding Instructions

- Use the existing data-source methods when binding branch, responsible, product type, time frame, and status controls.
- Keep job payload shapes aligned with the contracts expected by `saveJob()` and `updateJob()`.
- Call `getSalesResponsiblesDataSourceWithEnum()` only where the consuming control requires enum-backed responsible values.
- Route inquiry deletion through `deleteInquiry()` rather than duplicating deletion logic in components.

Used by

52 references from 52 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (52)

- `MrqDetailsComponent` — `Frontend/src/app/components/MRQ/mrq-details/mrq-details.component.ts :1`
- `BillOfMaterialsDetailsComponent` — `Frontend/src/app/components/bill-of-materials/bill-of-materials-details/bill-of-materials-details.component.ts :1`
- `BlogPopupComponent` — `Frontend/src/app/components/blogs/blog-popup/blog-popup.component.ts :1`
- `BlogsComponent` — `Frontend/src/app/components/blogs/blogs.component.ts :1`
- `WarehouseConfigDetailsComponent` — `Frontend/src/app/components/configuration/company/warehouse/warehouse-config-details/warehouse-config-details.component.ts :1`
- `DashboardSettingsComponent` — `Frontend/src/app/components/configuration/dashboard-settings/dashboard-settings/dashboard-settings.component.ts :1`
- `UsersLayoutsListComponent` — `Frontend/src/app/components/configuration/default-layouts/users-layouts-list/users-layouts-list.component.ts :1`
- `ProductsDetailsComponent` — `Frontend/src/app/components/configuration/products/products-details/products-details.component.ts :1`

...and 44 more.