

CancelDetailBindingModel

August 17, 2026

CancelDetailBindingModel

Kind: Class

Source: Pams/Logic/Pams.Business/BindingModels/NewSales/CancelDetailBindingModel.cs (line 4)

Part of: [Pams](#)

`CancelDetailBindingModel` carries cancellation details for a new sale between the client and the PAMS business layer. It records the cancellation reason, date, comment, fees, archive permission, branch context, and audit-related user identifiers.

Diagram

```
graph LR
    Client[Client application] --> Model[CancelDetailBindingModel]
    Model --> Cancellation[Sale cancellation workflow]
    Model --> Archive[Archive permission check]
    Model --> Audit[Created and modified user tracking]

    Model --- Reason[CancelReason and CancelReasonName]
    Model --- Details[CancelDate, CancelComment, CancelFees]
    Model --- Context[CompanyBranchId]
```

Usage

```
const cancelDetail = {
  id: saleId,
  cancelReason: selectedReasonCode,
  isAllowedToArchive: allowArchive,
  cancelDate: new Date().toISOString(),
  cancelReasonName: selectedReasonName,
  lastModifiedById: currentUserId,
  companyBranchId: activeBranchId,
  cancelComment: cancellationComment,
  cancelFees: cancellationFee,
```

```
    createdById: currentUserId,  
};  
  
await salesApi.cancelSale(cancelDetail);
```

AI Coding Instructions

- Send `CancelReason` together with `CancelReasonName` when the client needs to display the selected reason after submission.
- Preserve nullable values for `LastModifiedById` , `CompanyBranchId` , and `CreatedById` when no related record is available.
- Send `CancelDate` in a format accepted by the API and keep timezone handling consistent across cancellation flows.
- Validate `CancelFees` and `CancelComment` according to business rules before passing the model to the cancellation workflow.
- Check `IsAllowedToArchive` before invoking any archive-related action after cancellation.

Relationships

- IMPORTS → `UserLogResultBindingModel`

Used by

24 references from 24 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (24)

- `MRQJobInquiryController` — `Pams/API/Pams.API/Controllers/PM/MRQJobInquiryController.cs` :12
- `MRQJobOfferController` — `Pams/API/Pams.API/Controllers/PM/MRQJobOfferController.cs` :9
- `MRQJobOrderController` — `Pams/API/Pams.API/Controllers/PM/MRQJobOrderController.cs` :12
- `PurchasingJobController` — `Pams/API/Pams.API/Controllers/Purchasing/PurchasingJobController.cs` :14
- `PurchasingOfferController` — `Pams/API/Pams.API/Controllers/Purchasing/PurchasingOfferController.cs` :14
- `PurchasingOrderController` — `Pams/API/Pams.API/Controllers/Purchasing/PurchasingOrderController.cs` :18

- `purchasingInquiryController` — `Pams/API/Pams.API/Controllers/Purchasing/purchasingInquiryController.cs :16`
- `SalesItemController` — `Pams/API/Pams.API/Controllers/Sales/SalesItemController.cs :14`

...and 16 more.