FxaAdditionsExclusionController.cs

This controller manages **additions and exclusions** on fixed assets. It provides endpoints to search, create, edit, and delete addition/exclusion records, and to populate dropdowns.

Dependencies:

- `IFxaServiceManager.FxaAdditionsExclusionService`
- `IAccServiceManager.AccFacilityService`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`
- `IApiDDLHelper`
- `IWebHostEnvironment`

□ Endpoints

Route	Method	Description
/Search	POST	Search additions/exclusions
/Add	POST	Create a new addition/exclusion
/DDLFxaAddExcType	GET	Load addition/exclusion type dropdown list
/GetByIdForEdit	GET	Retrieve a record for editing
/Edit	POST	Update an existing record
/Delete	DELETE	Delete a record

□ Search

Performs filtered search on non-deleted records belonging to the current facility.

- **Route:** `POST /Search`
- **Input DTO:** `FxaAdditionsExclusionFilterDto`
- **Key Logic:**
  - Permission check (`1986` , `Show`)
  - Default null numeric filters to `0`
  - Date range filter using `yyyy/MM/dd`
  - Map VW entity to filter DTO

```
[HttpPost("Search")]
public async Task<ActionResult> Search(FxaAdditionsExclusionFilterDto filter)
{
    // permission
    filter.Id      ??= 0;
    filter.FixedAssetNo ??= 0;
    var items = await fxaServiceManager.FxaAdditionsExclusionService
        .GetAllVW(a => !a.IsDeleted && a.FacilityId == session.FacilityId
                       && (filter.Id == 0 || a.Id == filter.Id)
                       && (filter.FixedAssetNo == 0 || a.FixedAssetNo == filter.FixedAssetNo)
                       && (string.IsNullOrEmpty(filter.FixedAssetName) || a.FixedAssetName == filter.FixedAssetName));

    // apply date filter, map to DTO, return
}
```

□ Add

Creates a new addition/exclusion record and generates a ZATCA QR code if successful.

- **Route:** `POST /Add`
- **Input DTO:** `FxaAdditionsExclusionDto`
- **Key Logic:**
  - Permission check (`1986` , `Add`)
  - Model-state validation
  - Call service `Add(obj)`
  - On success, fetch facility data and generate QR code

□ DDLFxaAddExcType

Loads a dropdown of addition/exclusion types by `TypeId` .

- **Route:** `GET /DDLFxaAddExcType?TypeId={TypeId}`
- **Returns:** `SelectList` of `FxaAdditionsExclusionTypeDto`

□ GetByIdForEdit

Fetches a record's details for editing, including related journal info.

- **Route:** `GET /GetByIdForEdit?id={id}`
- **Key Logic:**
  - Permission check (`1986` , `Edit`)
  - Validate `id` > 0
  - Fetch VW record
  - Map to `FxaAdditionsExclusionEditDto`
  - Fetch related `AccJournalMaster` and set `JId` , `JCode` , `PeriodId`

□ Edit

Updates an existing addition/exclusion and regenerates the ZATCA QR.

- **Route:** `POST /Edit`
- **Input DTO:** `FxaAdditionsExclusionEditDto`
- **Key Logic:**
  - Permission check (`1986` , `Edit`)
  - Model-state validation
  - Call service `Update(obj)`
  - On success, regenerate QR code

□ Delete

Soft-deletes a record by `id` .

- **Route:** `DELETE /Delete?id={id}`

- **Key Logic:**
  - Permission check (`1986` , `Delete`)
  - Validate `id` > 0
  - Call service `Remove(id)`

FxaFixedAssetTypeController.cs

Manages **fixed asset types**, enabling CRUD operations and hierarchical relationships.

Dependencies:

- `IFxaServiceManager.FxaFixedAssetTypeService`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`

□ Endpoints

Route	Method	Description
/GetAll	GET	Retrieve all asset types
/Search	POST	Filter asset types
/Add	POST	Create a new asset type
/GetByIdForEdit	GET	Fetch type for editing
/Edit	POST	Update an existing type
/Delete	DELETE	Remove a type

□ Class Overview

- **FxaFixedAssetTypeVm**: ViewModel
- **FxaFixedAssetTypeDto**: DTO for creation
- **FxaFixedAssetTypeEditDto**: DTO for editing

□ GetAll

Returns all non-deleted types for the current facility, populating `MainAssetName` for child types.

```
[HttpGet("GetAll")]
public async Task<ActionResult> GetAll()
{
    // permission(901, Show)
    var items = await fxaServiceManager.FxaFixedAssetTypeService.GetAllVW(
        t => !t.IsDeleted && t.FacilityId == session.FacilityId);

    // map to FxaFixedAssetTypeVm, fetch parent name if ParentId != 0
}
```

□ Search

Filters by code, name, parent, rates, age, and account codes.

- **Input DTO:** `FxaFixedAssetTypeFilterDto`
- **Logic:**
  - Default null fields to `0`
  - Build predicate for all filter fields
  - Map to `FxaFixedAssetTypeVm`

□ Add / □ Edit / □ Delete

Standard CRUD flows with permission checks (`Add=901` , `Edit=901` , `Delete=901`), model validation, and service calls.

FxaFixedAssetTransferController.cs

Handles **asset transfers** between branches, facilities, and employees.

Dependencies:

- `IFxaServiceManager.FxaFixedAssetTransferService`
- `IMainServiceManager.InvestEmployeeService`
- `IAccServiceManager`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`

□ Endpoints

Route	Method	Description
/Search	POST	Search transfer records
/Delete	DELETE	Remove a transfer
/GetAssetByNo	GET	Lookup asset data by number
/Add	POST	Create a new transfer
/GetByIdForEdit	GET	Fetch transfer details for editing
/Edit	POST	Update an existing transfer
/GetAssetsByEmpCode	GET	List assets owned by an employee

Route	Method	Description
/Add2	POST	Transfer multiple assets from one employee to another

□ Search & Delete

- **Permission:** `Show=799` , `Delete=799`
- **Filter:** Code, name, from/to locations, employees, date range
- **Mapping:** Map VW to `FxaFixedAssetTransferSearchVM`

□ GetAssetByNo

Lookup an active asset by its `No` and optionally override `facilityId` .

- **Returns:** `FixedAssetDataForTransferVm` with branch, facility, cost center, employee, location.

□ Add / □ Edit

- **DTOs:** `FxaFixedAssetTransferDto` , `FxaFixedAssetTransferEditDto`
- **Flow:** Permission → Validate → Service call

□□□ Add2

Special endpoint for transferring **multiple assets** between employees.

- **DTO:** `FxaFixedAssetTransferDto2`
- **Use case:** Bulk transfer in a single transaction

FxaDepreciationController.cs

Calculates and records **asset depreciation**.

Dependencies:

- `IFxaServiceManager.FxaTransactionService`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`
- `IMapper`

□ Endpoints

Route	Method	Description
/Search	POST	List depreciation transactions (TransTypeId = 5)
/GetLastDeprecDate	GET	Get date of last depreciation run

Route	Method	Description
/GetAssetForDeprec	POST	Compute depreciation for assets up to a given end date
/Add	POST	Record depreciation (transaction + asset entries)
/GetByIdForEdit	GET	Fetch depreciation record for editing
/Delete	DELETE	Remove a depreciation transaction

□ Search

- Filters by code and date range
- Maps to `FxaDepreciationFilterVM`

□ Asset Calculation

`GetAssetForDeprec` acts like a stored-procedure, gathering:

- **Balance** from credit/debit sums
- **LastDeprecDate** from previous records or asset start
- **InstallmentValue** by method
- **Month/day counts** to compute `DeprecValue` & `DeprecValueDialy`

□ Add

Delegates to `FxaTransactionService.Add_Depreciation(obj)`.

FxaPurchaseController.cs

Manages **asset purchase** records (TransTypeId = 2).

Dependencies:

- `IFxaServiceManager.FxaTransactionService`
- `IAccServiceManager`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`
- `ISysConfigurationHelper`

□ Endpoints

Route	Method	Description
/Search	POST	Filter purchase records

Route	Method	Description
/Delete	DELETE	Delete a purchase record

□ Search

- Filters by code range, payment terms, branch, supplier, reference
- Date range filter
- Optional filter by `FxCode` to show only records containing a specific asset

FxaFixedAssetController.cs

Core controller for **fixed asset** CRUD, search, and detailed edit view.

Dependencies:

- `IFxaServiceManager.FxaFixedAssetService`
- `IAccServiceManager`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`
- `ISysConfigurationHelper`

□ Endpoints

Route	Method	Description
/Search	POST	Search assets
/GetProperties	GET	Load asset-form properties (CC visibility)
/GetAssetTypeData	GET	Fetch type details for new asset entry
/Add	POST	Create a new asset
/GetByIdForEdit	GET	Fetch asset for view/edit
/FillGridViewsInEdit	GET	Populate grids: transactions, transfers, additions
/Edit	POST	Update asset
/Delete	DELETE	Remove an asset

□ Search

- Highly configurable filters: code, name, location, branch, status, classification, employee, addition type, date
- Hierarchical type filtering via `FxaFixedAssetTypeId_DF`

⚙ GetProperties

Fetches system config flags for cost-center fields visibility and titles.

□ FillGridViewsInEdit

Aggregates multiple data sources for the edit screen:

1. **Transactions** (`FxaTransactionsAssetVw`)
2. **Related fixed assets**
3. **Transfers**
4. **Additions/Exclusions**

FxaBarCodeController.cs

Generates **barcodes** and **QR codes** for fixed assets.

Dependencies:

- `IFxaServiceManager.FxaFixedAssetService`
- `IAccServiceManager`
- `IPermissionHelper`
- `ICurrentData`
- `IWebHostEnvironment`

□ Endpoints

Route	Method	Description
/Search	POST	Filter assets for barcode generation
/GetBarCodes	GET	Generate and return bar/QR codes URLs

□ Search

Filters assets similarly to the main asset search, returning minimal viewmodels.

□ GetBarCodes

- Accepts comma-separated `assetsIds`
- Fetches facility logo + generates facility QR
- Loops assets and generates EAN-style barcode URL

BaseFxaApiController.cs

Abstract base class applying the route prefix:

```
[Route("api/{ApiConfig.ApiVersion}/FXA/[controller]")]
[ApiController]
public abstract class BaseFxaApiController : ControllerBase { }
```

All FXA controllers inherit this for consistent routing and CORS.

FxaReportsController.cs

Provides **reporting** endpoints covering purchase, sale, depreciation, and category reports.

Dependencies:

- `IFxaServiceManager.FxaFixedAssetService`
- `IFxaServiceManager.FxaTransactionService`
- `IAccServiceManager`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`
- `IApiDDLHelper`

□ Endpoints

Route	Method	Description
/GetPurchaseAssetReport	POST	Report of purchased assets
/GetSaleAssetReport	POST	Report of sold assets
/GetDepreciationReport	POST	Asset depreciation report
/GetDepreciationReport3	POST	Depreciation by hierarchy level
/GetDepreciationReport2	POST	Alternate depreciation report
/GetDepreciationReportByCategory	POST	Depreciation aggregated by category
/DDLAssetType2	GET	Dropdown level-2 asset types
/DDLAssetType3	GET	Dropdown level-3 asset types (with descendants)

FxaAdditionsExclusionTypeController.cs

CRUD for **addition/exclusion types** (metadata).

Dependencies:

- `IFxaServiceManager.FxaAdditionsExclusionTypeService`
- `IPermissionHelper`
- `ILocalizationService`

□ Endpoints

Route	Method	Description
/Search	POST	List types
/Add	POST	Create a type
/GetByIdForEdit	GET	Fetch type for editing
/Edit	POST	Update a type
/Delete	DELETE	Remove a type

FxaRevaluationController.cs

Handles **asset revaluation** transactions (TransTypeId = 8).

Dependencies:

- `IFxaServiceManager.FxaTransactionService`
- `IFxaServiceManager.FxaTransactionsRevaluationService`
- `IAccServiceManager`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`

□ Endpoints

Route	Method	Description
/Search	POST	Filter revaluation records
/Add	POST	Record new revaluation (transaction & journal)

Route	Method	Description
/Delete	DELETE	Delete a revaluation transaction

FxaTransactionsPaymentController.cs

Manages **payments** against FXA transactions.

Dependencies:

- `IFxaServiceManager.FxaTransactionsPaymentService`
- `IAccServiceManager`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`
- `IApiDDLHelper`
- `IWebHostEnvironment`

□ Endpoints

Route	Method	Description
/Search	POST	List payment records
/Add	POST	Create a payment
/Update	POST	Modify an existing payment
/DDLFxaAddExcType	GET	Dropdown for addition/exclusion types
/GetByIdForEdit	GET	Fetch payment for editing
/Edit	POST	Update payment details
/Delete	DELETE	Remove a payment record

SharedController.cs

Provides **shared utilities** for FXA modules.

Dependencies:

- `IFxaServiceManager.FxaFixedAssetService`
- `ISysConfigurationHelper`

- `ICurrentData`

□ Endpoints

Route	Method	Description
/GetCostCenterProperties	GET	Load cost-center visibility & titles
/GetAssetDataByNo	GET	Lookup asset financial data by its number

FxaSaleController.cs

Manages **asset sale** lifecycle (TransTypeId = 4).

Dependencies:

- `IFxaServiceManager.FxaTransactionService`
- `IAccServiceManager`
- `IPermissionHelper`
- `ICurrentData`
- `ILocalizationService`

□ Endpoints

Route	Method	Description
/Search	POST	List sale transactions
/Add	POST	Record a new sale
/View	GET	View sale details (read-only)
/GetByIdForEdit	GET	Fetch sale for editing
/Edit	POST	Update sale record
/Delete	DELETE	Remove a sale transaction

Note: All controllers extend `BaseFxaApiController` and share the prefix `/api/{version}/FXA/[controller]`. They uniformly enforce permission checks, session-based facility restrictions, and return `Result<T>` wrappers.

Classes Summary

This table summarizes each controller class with its purpose and key aspects.

Class Name	Purpose	Key Methods	Core DTOs U
FxaAdditionsExclusionController	Manage additions and exclusions on assets	Search, Add, GetByIdForEdit, Edit, Delete, DDLFxaAddExcType	FxaAdditionsEx FxaAdditionsEx FxaAdditionsEx
FxaFixedAssetTypeController	CRUD for fixed asset types	GetAll, Search, Add, GetByIdForEdit, Edit, Delete	FxaFixedAsset FxaFixedAsset FxaFixedAsset
FxaFixedAssetTransferController	Transfer assets between locations and employees	Search, Add, GetByIdForEdit, Edit, Delete, GetAssetByNo, GetAssetsByEmpCode, Add2	FxaFixedAsset FxaFixedAsset FxaFixedAsset FxaFixedAsset
FxaDepreciationController	Compute and record depreciation	Search, GetLastDeprecDate, GetAssetForDeprec, Add, GetByIdForEdit, Delete	FxaDepreciatio AssetsForDepr AssetsDeprecA AssetsDeprecB
FxaPurchaseController	Search and delete purchase transactions	Search, Delete	FxaTransaction
FxaFixedAssetController	Asset CRUD, search, and edit views	Search, GetProperties, GetAssetTypeData, Add, GetByIdForEdit, FillGridViewsInEdit, Edit, Delete	FxaFixedAsset FxaFixedAsset FxaFixedAsset
FxaBarCodeController	Generate barcodes and facility QR codes	Search, GetBarCodes	FxaFixedAsset
BaseFxaApiController	Base API routing and attributes	N/A	N/A
FxaReportsController	Reporting for assets, sales, and depreciation	GetPurchaseAssetReport, GetSaleAssetReport, GetDepreciationReport, GetDepreciationReport2, GetDepreciationReport3, GetDepreciationReportByCategory, DDLAssetType2, DDLAssetType3	Multiple report

Class Name	Purpose	Key Methods	Core DTOs U
FxaAdditionsExclusionTypeController	CRUD for addition/exclusion types	Search, Add, GetByIdForEdit, Edit, Delete	FxaAdditionsEx FxaAdditionsEx
FxaRevaluationController	Manage revaluation transactions	Search, Add, Delete	FxaRevaluation FxaTransaction
FxaTransactionsPaymentController	Manage payments related to FXA transactions	Search, Add, Update, GetByIdForEdit, Edit, Delete, DDLFxaAddExcType	FxaTransaction FxaTransaction FxaTransaction
SharedController	Shared utilities for FXA modules	GetCostCenterProperties, GetAssetDataByNo	AddAssetPrope FxaFixedAsset
FxaSaleController	Manage asset sales	Search, Add, View, GetByIdForEdit, Edit, Delete	FxaSaleFilterV FxaTransaction

Detailed Class, Methods, and Properties Explanation

This section explains each class, its methods, and the relevant properties or fields.

FxaAdditionsExclusionController

Handles asset addition and exclusion lifecycle.

- Injected fields:
  - fxaServiceManager: Accesses FXA services.
  - accServiceManager: Accesses accounting services.
  - permission: Checks user permissions.
  - session: Provides current facility and branches.
  - localization: Resolves localized messages.
  - ddlHelper: Builds dropdown lists.
  - env: Exposes hosting environment for file paths.
- Methods:
  - Search(filter): Filters records by Id, asset number, name, and date range.
  - Add(obj): Adds a record and generates a ZATCA QR.
  - DDLFxaAddExcType(TypeId): Returns type dropdown for UI.
  - GetByIdForEdit(id): Returns edit data with journal references.
  - Edit(obj): Updates the record and regenerates the QR.
  - Delete(id): Soft deletes the record.
- Notable DTO properties:

  - FxaAdditionsExclusionFilterDto.Id: Filters by record Id.
  - FixedAssetNo, FixedAssetName: Filters by asset.
  - StartDate, EndDate: Limits records by date.
  - FxaAdditionsExclusionEditDto.Amount: Uses Credit or Debit based on type.
  - OperationType: Reflects addition or exclusion kind.

FxaFixedAssetTypeController

Handles hierarchical asset types.

- Injected fields:
  - fxaServiceManager, permission, session, localization.
- Methods:
  - GetAll(): Lists all types with parent names.
  - Search(filter): Filters by code, name, parent, rates, age, accounts.
  - Add(obj): Adds a type record.
  - GetByIdForEdit(id): Returns fields to edit.
  - Edit(obj): Updates the type record.
  - Delete(id): Removes the type.
- Notable DTO properties:
  - FxaFixedAssetTypeFilterDto.Code, TypeName: Text filters.
  - ParentId: Hierarchy filter.
  - DeprecYearlyRate, Age: Depreciation defaults.
  - AccountCode, AccountCode2, AccountCode3: Accounting mappings.

FxaFixedAssetTransferController

Transfers assets between employees, locations, and facilities.

- Injected fields:
  - fxaServiceManager, mainServiceManager, accServiceManager, permission, session, localization.
- Methods:
  - Search(filter): Finds transfers by route, employee, and date.
  - Delete(id): Removes a transfer.
  - GetAssetByNo(no, facilityId): Resolves current asset location and custodian.
  - Add(obj): Creates a transfer.
  - GetByIdForEdit(id): Returns transfer for edit.
  - Edit(obj): Updates transfer info.
  - GetAssetsByEmpCode(empCode, facilityId): Lists employee’s assets.
  - Add2(obj): Bulk transfer between employees.
- Notable DTO properties:
  - FxaFixedAssetTransferFilterDto.FromLocationId, ToLocationId: Movement filter.
  - FromEmpCode, ToEmpCode: Custodian change filter.
  - DateTransfer: Transfer date window.
  - FxaFixedAssetTransferEditDto.FromFacilityId, ToFacilityId: Facility movement.

FxaDepreciationController

Computes and records periodic depreciation.

- Injected fields:
  - fxaServiceManager, permission, session, localization, _mapper.
- Methods:
  - Search(filter): Lists depreciation transactions.
  - GetLastDeprecDate(): Finds last run end date.
  - GetAssetForDeprec(filter): Computes amounts per asset.
  - Add(obj): Saves depreciation transaction.
  - GetByIdForEdit(id): Loads transaction and lines.
  - Delete(id): Removes transaction and lines.
- Notable DTO properties:
  - FxaDepreciationFilterVM.StartDate, EndDate: Date window.
  - AssetsForDeprecFilter.TypeId, BranchId: Asset scope filters.
  - AssetsForDeprecVm.DeprecValue, DeprecValueDialy: Calculated outputs.

FxaPurchaseController

Searches and deletes purchase transactions.

- Injected fields:
  - fxaServiceManager, accServiceManager, permission, session, localization, configurationHelper.
- Methods:
  - Search(filter): Filters by code range, terms, branch, supplier.
  - Delete(id): Removes the purchase transaction.
- Notable DTO properties:
  - FxaTransactionFilterDto.Code, Code2: Range filtering.
  - PaymentTermsId, BranchId: Commercial filters.
  - StartDate, EndDate: Date constraints.
  - FxCode: Restricts to records with specific asset.

FxaFixedAssetController

Central controller for assets.

- Injected fields:
  - fxaServiceManager, accServiceManager, permission, session, localization, configurationHelper.
- Methods:
  - Search(filter): Complex filtering with hierarchy and dates.
  - GetProperties(): Returns cost center visibility and titles.
  - GetAssetTypeData(code): Returns defaults from type selection.
  - Add(obj): Creates a new asset.
  - GetByIdForEdit(id): Loads asset for edit or view.
  - FillGridViewsInEdit(id): Loads grids for transactions, transfers, children, additions.
  - Edit(obj): Updates an asset.
  - Delete(id): Removes an asset.
- Notable DTO properties:
  - FxaFixedAssetFilterDto.AdditionTypeFilter: Purchase vs addition flag.

  - BranchId, LocationId, StatusId, ClassificationId: Standard filters.
  - StartDate, EndDate: Purchase date range.
  - FxaFixedAssetEditDto fields: Includes accounting, CCs, depreciation, and status.

FxaBarCodeController

Generates barcodes and facility QR.

- Injected fields:
  - fxaServiceManager, accServiceManager, permission, session, env.
- Methods:
  - Search(filter): Finds assets to print codes for.
  - GetBarCodes(assetsIds): Returns URLs for barcodes and QR.
- Notable DTO properties:
  - FxaFixedAssetFilterDto.Code: Matches asset number exactly.
  - Name, Description, LocationId, BranchId: Standard filters.

BaseFxaApiController

Provides route prefix and base attributes.

- Properties and attributes:
  - Route: api/{ApiConfig.ApiVersion}/FXA/[controller].
  - ApiController: Enables automatic validation and binding.

FxaReportsController

Generates multiple asset-related reports.

- Injected fields:
  - fxaServiceManager, accServiceManager, permission, session, localization, ddlHelper.
- Methods:
  - GetPurchaseAssetReport(filter): Purchased assets.
  - GetSaleAssetReport(filter): Sold assets.
  - GetDepreciationReport(filter): Depreciation within period.
  - GetDepreciationReport3(filter): Depreciation by hierarchy level.
  - GetDepreciationReport2(filter): Alternate depreciation dataset.
  - GetDepreciationReportByCategory(filter): Aggregated by category.
  - DDLAssetType2(typeId), DDLAssetType3(typeId): Type dropdown helpers.
- Notable DTO properties:
  - Common filters: StartDate, EndDate, BranchId, LocationId, TypeId, StatusId.
  - ByCategory adds account codes grouping.

FxaAdditionsExclusionTypeController

Maintains metadata for operation types.

- Injected fields:
  - fxaServiceManager, permission, localization.

- Methods:
  - Search(filter): Filters by Id, TypeId, Name.
  - Add(obj): Creates a new type.
  - GetByIdForEdit(id): Loads type for editing.
  - Edit(obj): Updates type data.
  - Delete(id): Removes type.
- Notable DTO properties:
  - FxaAdditionsExclusionTypeFilterDto.TypeId: 1 addition, 2 exclusion.
  - Name: Display name of the operation subset.

FxaRevaluationController

Records revaluation changes and journals.

- Injected fields:
  - fxaServiceManager, accServiceManager, permission, session, localization.
- Methods:
  - Search(filter): Finds revaluation entries.
  - Add(obj): Creates revaluation transaction and journal.
  - Delete(id): Removes revaluation transaction.
- Notable DTO properties:
  - FxaRevaluationFilterVM: Code, FxNo, FxName, BranchId, date range.
  - FxaTransactionDto_Revaluation: TransTypeId set to 8.

FxaTransactionsPaymentController

Handles payments related to FXA transactions.

- Injected fields:
  - fxaServiceManager, accServiceManager, permission, session, localization, ddlHelper, env.
- Methods:
  - Search(filter): Filters by No, Code, BranchId, TransactionId, BankId, and date range.
  - Add(model): Adds payment after period check.
  - Update(model): Updates existing payment.
  - GetByIdForEdit(id): Loads payment record.
  - Edit(obj): Updates payment via service.
  - Delete(id): Removes payment.
  - DDLFxaAddExcType(TypeId): Helper dropdown endpoint.
- Notable DTO properties:
  - FxaTransactionsPaymentDto: AmountReceived, PaymentDate, BankId, PaymentMethodId.
  - ExchangeRate default is 1 if not provided.

SharedController

Exposes shared endpoints.

- Injected fields:

  - fxaServiceManager, session, configurationHelper.

- Methods:
  - GetCostCenterProperties(): Returns CC visibility and titles.
  - GetAssetDataByNo(no): Returns financial snapshot of asset.

- Notable DTO properties:
  - AddAssetProperties: CCnVisible flags and titles.
  - FxaFixedAssetVm2: Account codes, start/end dates, and computed totals.

FxaSaleController

Manages lifecycle of asset sales.

- Injected fields:
  - fxaServiceManager, accServiceManager, permission, session, localization.

- Methods:
  - Search(filter): Lists sale transactions by code and dates.
  - Add(obj): Creates sale transaction with TransTypeId 4.
  - View(id): Loads sale in read-only mode.
  - GetByIdForEdit(id): Loads sale for editing.
  - Edit(obj): Updates sale transaction.
  - Delete(id): Removes sale.

- Notable DTO properties:
  - FxaSaleFilterVM: Code and date range filters.
  - FxaTransactionDto_Sale: DebAccCode, ProfitLossAccCode, SaleAmount, CC codes, and journal fields.