Feed Organization Subsystem New

▼ public class UserArticleDao

1. Class Information

- **Class Name:** `UserArticleDao`
- **Parent Class:** `BaseDao<UserArticleDto, UserArticleCriteria>`
- **Purpose:**
  - The class provides methods to interact with the database for operations related to user-article relationships, including retrieving user-specific articles, updating read and starred statuses, and managing subscriptions.

Attributes： This class does not define instance variables but relies on inherited properties from `BaseDao` and database interactions.

2. Methods and Their Details

Method Signature	Access Modifier	Description
`protected QueryParam getQueryParam(UserArticleCriteria criteria, FilterCriteria filterCriteria)`	Protected	Constructs a dynamic SQL query based on user article criteria for filtering, sorting, and retrieving user-specific articles.
`public String create(UserArticle userArticle)`	Public	Generates a new UUID and inserts a new `UserArticle` record into the database. Returns the newly created article's ID.
`public UserArticle update(UserArticle userArticle)`	Public	Updates an existing `UserArticle` entity, modifying its read or starred status.
`public void markAsRead(UserArticleCriteria criteria)`	Public	Marks all articles in a specified category as read based on filtering criteria.
`public void delete(String id)`	Public	Deletes a user-article subscription by setting its `deleteDate` timestamp.
`public UserArticle getUserArticle(String id, String userId)`	Public	Retrieves a `UserArticle` entity based on the article ID and user ID. Returns `null` if not found.

3. Relationships

Inheritance:

- **Parent Class:** `BaseDao<UserArticleDto, UserArticleCriteria>`
- **Child Classes:** None
- **Association:**
  - Dependencies :

- UserArticle , UserArticleCriteria , and UserArticleDto,UserArticleMapper
- QueryParam

▼ public class UserArticleDto

1. Class Information

- **Class Name:** UserArticleDto
- **Purpose:** Represents detailed information about an article read or starred by a user, along with metadata related to the article and its associated feed.

2. Attributes and Methods

Attributes

Attribute Name	Type	Description
id	String	User article unique identifier.
readTimestamp	Long	Timestamp when the user read the article.
starTimestamp	Long	Timestamp when the user starred the article.
feedTitle	String	Title of the feed.
feedSubscriptionId	String	Subscription identifier for the feed.
feedSubscriptionTitle	String	Title of the feed subscription.
articleId	String	Unique article identifier.
articleUrl	String	URL of the article.
articleGuid	String	Globally unique identifier for the article.
articleTitle	String	Title of the article.
articleCreator	String	Creator of the article.
articleDescription	String	Description of the article.
articleCommentUrl	String	URL for article comments.
articleCommentCount	Integer	Number of comments on the article.
articleEnclosureUrl	String	URL of the article enclosure.
articleEnclosureLength	Integer	Length of the enclosure in bytes.
articleEnclosureType	String	MIME type of the enclosure.
articlePublicationTimestamp	Long	Timestamp of the article's publication.

Methods:

Method Signature	Access Modifier	Description
String getId()	public	Returns the user-article ID.
void setId(String id)	public	Sets the user-article ID.
Long getReadTimestamp()	public	Returns the read timestamp.
void setReadTimestamp(Long readTimestamp)	public	Sets the read timestamp.
Long getStarTimestamp()	public	Returns the star timestamp.

`void setStarTimestamp(Long starTimestamp)`	public	Sets the star timestamp.
`String getFeedTitle()`	public	Returns the feed title.
`void setFeedTitle(String feedTitle)`	public	Sets the feed title.
`String getFeedSubscriptionId()`	public	Returns the feed subscription ID.
`void setFeedSubscriptionId(String feedSubscriptionId)`	public	Sets the feed subscription ID.
`String getFeedSubscriptionTitle()`	public	Returns the feed subscription title.
`void setFeedSubscriptionTitle(String feedSubscriptionTitle)`	public	Sets the feed subscription title.
`String getArticleId()`	public	Returns the article ID.
`void setArticleId(String articleId)`	public	Sets the article ID.
`String getArticleUrl()`	public	Returns the article URL.
`void setArticleUrl(String articleUrl)`	public	Sets the article URL.
`String getArticleGuid()`	public	Returns the article GUID.
`void setArticleGuid(String articleGuid)`	public	Sets the article GUID.
`String getArticleTitle()`	public	Returns the article title.
`void setArticleTitle(String articleTitle)`	public	Sets the article title.
`String getArticleCreator()`	public	Returns the article creator.
`void setArticleCreator(String articleCreator)`	public	Sets the article creator.
`String getArticleDescription()`	public	Returns the article description.
`void setArticleDescription(String articleDescription)`	public	Sets the article description.
`String getArticleCommentUrl()`	public	Returns the comment URL.
`void setArticleCommentUrl(String articleCommentUrl)`	public	Sets the comment URL.
`Integer getArticleCommentCount()`	public	Returns the article comment count.
`void setArticleCommentCount(Integer articleCommentCount)`	public	Sets the article comment count.
`String getArticleEnclosureUrl()`	public	Returns the enclosure URL.
`void setArticleEnclosureUrl(String articleEnclosureUrl)`	public	Sets the enclosure URL.
`Integer getArticleEnclosureLength()`	public	Returns the enclosure length.
`void setArticleEnclosureLength(Integer articleEnclosureLength)`	public	Sets the enclosure length.
`String getArticleEnclosureType()`	public	Returns the enclosure type.
`void setArticleEnclosureType(String articleEnclosureType)`	public	Sets the enclosure type.
`Long getArticlePublicationTimestamp()`	public	Returns the publication timestamp.
`void setArticlePublicationTimestamp(Long articlePublicationTimestamp)`	public	Sets the publication timestamp.

3. Relationships

Inheritence : NA

Associations : NA

▼ public class UserArticleMapper

1. Class Information

- **Class Name:** UserArticleMapper
- **Purpose:** Maps raw database result sets to UserArticleDto objects, encapsulating information about user interactions with articles, including read and starred timestamps, article metadata, and subscription details.

2. Attributes and Methods

Attributes:

This class does not have instance attributes besides those inherited from its superclass ResultMapper .

Methods:

Method Signature	Access Modifier	Description
public UserArticleDto map(Object[] o)	public	Maps an array of raw database values to a UserArticleDto object.

3. Relationships

Inheritance:

- **Parent Class:** ResultMapper<UserArticleDto>
- **Child Classes:** None

Associations:

- **Dependency:** Depends on UserArticleDto for the target mapping type.
- **Aggregation:** Not applicable
- **Composition:** Not applicable

▼ public class UserArticleCriteria

Class Documentation for UserArticleCriteria

1. Class Information

- **Class Name:** UserArticleCriteria
- **Purpose:** To define filtering and sorting criteria for fetching user-specific articles, enabling features like filtering unread, starred, and categorized articles.

---

2. Attributes and Methods

- **Attributes:**

Attribute Name	Access Modifier	Type	Description

userId	private	String	The ID of the user requesting the articles.
feedId	private	String	The ID of the feed associated with the articles.
visible	private	boolean	Indicates whether the article is visible to the user.
subscribed	private	boolean	Indicates if the user is subscribed to the feed.
feedSubscriptionId	private	String	The ID of the feed subscription.
categoryId	private	String	The ID of the category for organizing articles.
unread	private	boolean	Filters unread articles if set to true.
starred	private	boolean	Filters starred articles if set to true.
articleId	private	String	The ID of a specific article.
articleIdIn	private	List<String>	A list of specific article IDs to filter.
userArticleId	private	String	The ID of the user-specific article.
articlePublicationDateMax	private	Date	Filters articles published before a specific date.
userArticleStarredDateMax	private	Date	Filters articles starred before a specific date.
articleIdMax	private	String	Filters articles before a specific article ID.
userArticleIdMax	private	String	Filters user articles before a specific user article ID.
fetchAllFeedSubscription	private	boolean	Fetches all subscriptions if true.

- **Methods:**

Method Signature	Access Modifier	Description
String getUserId()	public	Gets the user ID.
UserArticleCriteria setUserId(String userId)	public	Sets the user ID.
String getFeedId()	public	Gets the feed ID.
UserArticleCriteria setFeedId(String feedId)	public	Sets the feed ID.
boolean isVisible()	public	Checks if the article is visible.
UserArticleCriteria setVisible(boolean v)	public	Sets the visibility of an article.
boolean isSubscribed()	public	Checks if the user is subscribed.
UserArticleCriteria setSubscribed(boolean s)	public	Sets the subscription status.
String getFeedSubscriptionId()	public	Gets the feed subscription ID.
UserArticleCriteria setFeedSubscriptionId()	public	Sets the feed subscription ID.
boolean isUnread()	public	Checks if the article is unread.
UserArticleCriteria setUnread(boolean u)	public	Sets the unread status of the article.
String getCategoryId()	public	Gets the category ID.
UserArticleCriteria setCategoryId(String c)	public	Sets the category ID.
boolean isStarred()	public	Checks if the article is starred.
UserArticleCriteria setStarred(boolean s)	public	Sets the starred status of an article.
String getArticleId()	public	Gets the article ID.

`UserArticleCriteria setArticleId(String a)`	public	Sets the article ID.
`List<String> getArticleIdIn()`	public	Gets a list of article IDs for filtering.
`UserArticleCriteria setArticleIdIn(List a)`	public	Sets the list of article IDs for filtering.
`String getUserArticleId()`	public	Gets the user article ID.
`UserArticleCriteria setUserArticleId(String)`	public	Sets the user article ID.
`Date getArticlePublicationDateMax()`	public	Gets the max publication date for filtering.
`UserArticleCriteria setArticlePublicationDateMax(Date d)`	public	Sets the max publication date for filtering.
`Date getUserArticleStarredDateMax()`	public	Gets the max starred date for filtering user articles.
`UserArticleCriteria setUserArticleStarredDateMax(Date d)`	public	Sets the max starred date for filtering user articles.
`String getArticleIdMax()`	public	Gets the max article ID for filtering.
`UserArticleCriteria setArticleIdMax(String)`	public	Sets the max article ID for filtering.
`String getUserArticleIdMax()`	public	Gets the max user article ID for filtering.
`UserArticleCriteria setUserArticleIdMax()`	public	Sets the max user article ID for filtering.
`boolean isFetchAllFeedSubscription()`	public	Checks whether all subscriptions should be fetched.
`UserArticleCriteria setFetchAllFeedSubscription(boolean)`	public	Sets whether all subscriptions should be fetched.

3. Relationships

- **Inheritance:**
  - Parent Class: None
  - Child Classes: None
- **Associations:**
  - NA

▼ public class UserArticle

1. Class Information
   - Class Name: UserArticle
   - Purpose:
     - Stores data related to a user's interaction with articles.
     - Tracks whether an article has been read or starred by a specific user.
2. Attributes and Methods
   - Attributes:

Attribute Name	Access Modifier	Type	Description
id	private	String	Unique identifier for the user-article relationship.
userId	private	String	The ID of the user interacting with the article.

Attribute Name	Access Modifier	Type	Description
articleId	private	String	The ID of the article being interacted with.
createDate	private	Date	Timestamp when the relationship was created.
readDate	private	Date	Timestamp when the user read the article.
starredDate	private	Date	Timestamp when the user starred the article.
deleteDate	private	Date	Timestamp when the relationship was marked as deleted.

- Methods:

Method Signature	Access Modifier	Description
String getId()	public	Returns the ID of the user-article relationship.
void setId(String id)	public	Sets the ID of the user-article relationship.
String getUserId()	public	Returns the user ID.
void setUserId(String userId)	public	Sets the user ID.
String getArticleId()	public	Returns the article ID.
void setArticleId(String articleId)	public	Sets the article ID.
Date getCreateDate()	public	Returns the creation date.
void setCreateDate(Date createDate)	public	Sets the creation date.
Date getReadDate()	public	Returns the read date.
void setReadDate(Date readDate)	public	Sets the read date.
Date getStarredDate()	public	Returns the starred date.
void setStarredDate(Date starredDate)	public	Sets the starred date.
Date getDeleteDate()	public	Returns the deletion date.
void setDeleteDate(Date deleteDate)	public	Sets the deletion date.
String toString()	public	Returns a string representation of the entity.

3. Relationships:
   - Inheritance: NA
     - Parent Class:
     - Child Classes:
   - Associations : NA
     - Dependency:
     - Aggregation:
     - Composition:

▼

▼ public class IndexingService

1. Class Information
   - Class Name: IndexingService
   - Purpose:
     - Maintains a searchable Lucene index of RSS articles.
     - Supports scheduled updates and manual index rebuilding.
     - Ensures efficient search queries for users based on article content.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
log	private static	Logger	Logger for indexing errors.
directory	private	Directory	Represents Lucene's index storage.
directoryReader	private	DirectoryReader	Allows reading the Lucene index.
luceneStorageConfig	private	String	Determines whether RAM or File storage is used.

   - Methods:

Method Signature	Access Modifier	Description
void startUp()	protected	Initializes Lucene storage, setting it to RAM or File storage.
void shutDown()	protected	Closes the Lucene index and reader on shutdown.
void runOneIteration()	protected	Runs one scheduled update (currently does nothing).
Scheduler scheduler()	protected	Schedules indexing every hour.
PaginatedList<UserArticleDto> searchArticles(String userId, String searchQuery, Integer offset, Integer limit)	public	Executes a search query on articles, returning paginated results.
void rebuildIndex()	public	Destroys and rebuilds the Lucene index asynchronously.
Directory getDirectory()	public	Returns the Lucene directory instance.
DirectoryReader getDirectoryReader()	public	Returns a valid Lucene reader, reopening it if needed.

3. Relationships:

- Inheritance:
  - Parent Class: AbstractScheduledService
  - Child Classes:
- Associations
  - Dependency:
    - TransactionUtil: Ensures safe indexing transactions.
    - ArticleDao: Searches articles using Lucene queries.
    - UserArticleDao: Retrieves user-specific article metadata.
    - PaginatedLists: Handles paginated search results.
    - RebuildIndexAsyncEvent: Triggers index rebuilding when needed.
    - UserArticleCriteria
    - LoggerFactory
    - PaginatedList
  - Aggregation:
    - Directory: Manages Lucene's index storage.
    - DirectoryReader: Reads existing indexes.
  - Composition:

▼ public class RebuildIndexAsyncEvent

1. Class Information
   - Class Name: RebuildIndexAsyncEvent
   - Purpose: This class serves as an event object indicating that the system should initiate an asynchronous rebuild of the index of stored RSS feed articles. It is useful for ensuring that the indexing system is kept up-to-date, allowing for fast retrieval and searching of articles.
2. Attributes and Methods
   - Attributes: NA
   - Methods:

Method Signature	Access Modifier	Description
String toString()	public	Overrides the default toString() method to provide a string representation of the object using Objects.toStringHelper() from Guava.

3. Relationships:
   - Inheritance:
     - Parent Class:
   - Associations
     - Dependency: NA

▼ public class RebuildIndexAsyncListener

1. Class Information
   - Class Name: RebuildIndexAsyncListener
   - Purpose: This class listens for `RebuildIndexAsyncEvent` and triggers the process of rebuilding the article index. It retrieves all articles from the database and re-indexes them to improve search efficiency.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
log	private static final	Logger	Logger instance for logging rebuild index events.

   - Methods:

Method Signature	Access Modifier	Description
void onArticleCreated(RebuildIndexAsyncEvent rebuildIndexAsyncEvent)	public	Listens for RebuildIndexAsyncEvent, fetches all articles, and triggers re-indexing.

3. Relationships:
   - Inheritance: NA
   - Associations
     - Aggregation :
       - ArticleDao
     - Dependency:
       1. RebuildIndexAsyncEvent - Listens for the event and processes it when triggered.
       2. TransactionUtil - Manages transactions while fetching articles and rebuilding indexes.

▼ public class SortCriteria

1. Class Information
   - Class Name: SortCriteria
   - Purpose: Defines sorting criteria for database queries
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
column	private	int	Index of the column to be sorted

asc	private	boolean	Defines whether sorting is ascending (true) or descending (false).
sortQuery	private	String	Optional custom query string for sorting.

- Methods:

Method Signature	Access Modifier	Description
SortCriteria(String sortQuery)	public	Constructor that initializes sorting using custom query string
SortCriteria(Integer column, Boolean asc)	public	Constructor that initializes sorting based on column index and order.
String getSortQuery()	public	Returns the custom sorting query string.
int getColumn()	public	Returns the column index to sort by.
boolean isAsc()	public	Returns true if sorting is ascending, otherwise false

3. Relationships:
   - Inheritance:
     - Parent Class:
     - Child Classes:
   - Associations : NA
     - Dependency:
     - Aggregation:
     - Composition:

▼ public class QueryParam

1. Class Information
   - Class Name: QueryParam
   - Purpose:
     - Encapsulates parameters required for executing database queries.
     - Supports sorting, filtering, and grouping criteria for constructing complex queries.
     - Allows mapping query results using a ResultMapper.
2. Attributes and Methods
   - Attributes:

Attribute Name	Access Modifier	Type	Description
queryString	private	String	SQL or JPQL query string.
criteriaList	private	List<String>	Stores additional **query conditions**.
parameterMap	private	Map<String, Object>	Maps **query parameters** to their values.
sortCriteria	private	SortCriteria	Defines sorting rules for the query.
filterCriteria	private	FilterCriteria	Defines filtering rules for the query.

groupByList	private	List<String>	Stores **grouping criteria** for the query.
resultMapper	private	ResultMapper	Maps **query results** for native queries.

- Methods:

Method Signature	Access Modifier	Description
`QueryParam(String queryString, List<String> criteriaList, Map<String, Object> parameterMap, SortCriteria sortCriteria, FilterCriteria filterCriteria, List<String> groupByList, ResultMapper resultMapper)`	public	Constructor to initialize a query with full parameter options.
`QueryParam(String queryString, List<String> criteriaList, Map<String, Object> parameterMap, SortCriteria sortCriteria, FilterCriteria filterCriteria, ResultMapper resultMapper)`	public	Constructor for queries without groupBy options.
`String getQueryString()`	public	Returns the query string.
`void setQueryString(String queryString)`	public	Sets the query string.
`SortCriteria getSortCriteria()`	public	Retrieves sorting criteria.
`void setSortCriteria(SortCriteria sortCriteria)`	public	Sets sorting criteria.
`List<String> getCriteriaList()`	public	Retrieves filtering criteria.
`Map<String, Object> getParameterMap()`	public	Retrieves query parameters.
`FilterCriteria getFilterCriteria()`	public	Retrieves filter criteria.
`List<String> getGroupByList()`	public	Retrieves group-by columns.
`ResultMapper getResultMapper()`	public	Retrieves the result mapper for native queries.

3. Relationships:
   - Inheritance: NA
     - Parent Class:
     - Child Classes:
   - Associations
     - Dependency:
       - SortCriteria
       - FilterCriteria
       - Mapper
     - Aggregation:
     - Composition:

▼ public class FilterCriteria

1. Class Information

- Class Name: FilterCriteria
- Purpose:
    - Encapsulates a list of FilterColumn instances, representing filtering conditions for a database query.
    - Provides a structured approach to apply multiple filtering conditions dynamically.

2. Attributes and Methods
    - Attributes:

Attribute Name	Access Modifier	Type	Description
filterColumnList	private	List<FilterColumn>	A list of filtering conditions to be applied to a query.

- Methods:

Method Signature	Access Modifier	Description
FilterCriteria(List<FilterColumn> filterColumnList)	public	Initializes the filtering criteria with a given list of filters.
List<FilterColumn> getFilterColumnList()	public	Returns the list of filter conditions.

3. Relationships:
    - Inheritance:
        - Parent Class:
        - Child Classes:
    - Associations
        - Dependency:
        - Aggregation:
            - FilterColumn: Contains FilterColumn objects representing filtering conditions
        - Composition:

▼ public class FilterColumn

1. Class Information
    - Class Name: FilterColumn
    - Role in Subsystem:
        - Used forr filtering database query results within the JPA layer.
        - It is used to dynamically generate filter conditions for queries.
    - Purpose:
        - Represents a filter condition applied to a specific database column.
        - Supports dynamic query filtering by generating predicates and parameters.

- Designed as an **abstract class**, requiring subclasses to define how filtering is applied.

2. Attributes and Methods

- Attributes:

Attribute Name	Access Modifier	Type	Description
column	protected	String	The name of the column to be filtered.
filter	protected	String	The filter expression or value to apply.

- Methods:

Method Signature	Access Modifier	Description
FilterColumn(String column, String filter)	public	Initializes the column name and filter expression.
String getColumn()	public	Returns the column being filtered.
String getFilter()	public	Returns the filter expression.
abstract String getPredicate()	public abstract	Subclasses must implement this to generate the SQL predicate for the filter.
abstract Object getParamValue()	public abstract	Subclasses must implement this to return the value associated with the filter parameter.
String getParamName()	public	Generates a unique parameter name based on the column name.
boolean hasParam()	public	Returns true if the column name is not null, indicating a valid filter.

3. Relationships:

- Inheritance:
  - Parent Class:
  - Child Classes:
- Associations
  - Dependency:
  - Aggregation:
  - Composition:

▼ public class SearchResource

1. Class Information

- **Class Name**: SearchResource
- **Role in Subsystem**: REST API endpoint handler for search functionality
- **Purpose**: Handles search requests for articles in the reader application

2. Attributes and Methods

Attributes

None explicitly defined

Methods

Method	Access Modifier	Description
get(String query, Integer limit, Integer offset)	Public	- Handles GET requests to search articles - Takes query parameters for pagination and search terms - Returns JSON response with search results

3. Relationships

Inheritance

- **Parent Class**: BaseResource
- **Child Classes**: None

Associations

Dependencies

- IndexingService
- ArticleAssembler
- ValidationUtil

Aggregation :

IndexingService

PaginatedList<UserArticleDto>

▼ public class PaginatedLists

1. Class Information
   - Class Name: PaginatedLists
   - Purpose: A utility class for handling paginated database queries using JPA. It constructs paginated lists, executes queries efficiently, and manages sorting, filtering, and counting of results.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
DEFAULT_PAGE_SIZE	private	int	Default size of a page
MAX_PAGE_SIZE	private	int	Maximum size of a page

   - Methods:

Method Signature	Access Modifier	Description
public static <E> PaginatedList<E> create(Integer pageSize, Integer offset)	public	Creates a paginated list with the specified page

		size and offset.
public static <E> PaginatedList<E> create()	public	Creates a paginated list with default page size and offset.
public static <E> List<E> executeQuery(QueryParam queryParam)	public	Executes a non-paginated query and returns the result list.
public static <E> void executePaginatedQuery(PaginatedList<E> paginatedList, QueryParam queryParam, SortCriteria sortCriteria)	public	Executes a paginated query by counting and retrieving results.
private static <E> void executeCountQuery(PaginatedList<E> paginatedList, QueryParam queryParam)	private	Counts the number of records matching the query
private static <E> void executeResultQuery(PaginatedList<E> paginatedList, QueryParam queryParam)	private	Executes the paginated query and retrieves results
private static String getQueryString(QueryParam queryParam)	private	Constructs the SQL query string with filtering(WHERE) and grouping(GROUP BY)
private static String getNativeCountQuery(QueryParam queryParam)	private	Creates a native SQL query to count the number of records.
private static String getOrderByClause(SortCriteria sortCriteria)	private	Generates an ORDER BY clause for sorting results based on SortCriteria
private static void mapQueryParam(Query query, QueryParam queryParam)	private	Maps query parameters for filtering
private static void mapFilterColumn(Query query, QueryParam queryParam)	private	Maps filter criteria to query parameters.

3. Relationships:
    - Inheritance:
        - Parent Class:
        - Child Classes:
    - Associations
        - Dependency:
            - PaginatedList<E>: Uses it to store paginated results
            - QueryParam: Holds query details such as filtering, sorting, and parameters.
            - EntityManager: Executes database queries.
            - SortCriteria: Defines sorting order of query results.
            - ThreadLocalContext: Provides a thread-local EntityManager instance.
            - FilterColumn: Handles query filtering conditions.
        - Aggregation:

    - Composition:

- ▼ public class PaginatedList<T>

  1. Class Information

     - Class Name: PaginatedList
     - Purpose: The class provides a structured way to handle pagination for lists of feed articles or other records within the system.

  2. Attributes and Methods

     - Attributes:

Name	Access Modifier	Type	Description
limit	private	int	Maximum number of records per page.
offset	private	int	Starting point of the current page in the dataset.
resultCount	private	int	Total number of records available.
resultList	private	List<T>	The list of records currently being displayed.

     - Methods:

Method Signature	Access Modifier	Description
PaginatedList(int pageSize, int offset)	public	Constructor that initializes the pagination settings.
int getResultCount()	public	Returns the total number of records.
void setResultCount(int resultCount)	public	Sets the total number of records.
List<T> getResultList()	public	Returns the list of records on the current page.
void setResultList(List<T> resultList)	public	Sets the list of records for the current page.
int getLimit()	public	Returns the maximum number of records per page.
int getOffset()	public	Returns the starting offset of the current page.

  3. Relationships:

     - Inheritance: NA
       - Parent Class:
       - Child Classes:
     - Associations : NA
       - Dependency:
       - Aggregation:
       - Composition:

▼ public class LuceneUtil

1. Class Information
   - Class Name: LuceneUtil
   - Purpose:
     - Provides a utility method for safely executing Lucene indexing operations.
     - Ensures that index writers are properly created, used, and closed, reducing the risk of locked indexes or incomplete transactions.
     - Supports custom indexing logic via the LuceneRunnable interface.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
log	private static	Logger	Logger for logging errors related to Lucene indexing.

   - Methods:

Method Signature	Access Modifier	Description
void handle(LuceneRunnable runnable)	public static	Encapsulates a Lucene indexing operation, ensuring proper error handling and transaction rollback.

   - Nested Interface:

Interface	Description
LuceneRunnable	Defines a functional interface for executing custom indexing logic within the Lucene context.

   - Interface Method:

Method Signature	Access Modifier	Description
void run(IndexWriter indexWriter)	public abstract	Defines custom indexing logic that runs within a Lucene transaction.

3. Relationships:
   - Inheritance:
     - Parent Class:
     - Child Classes:
   - Associations
     - Dependency:
       - LuceneRunnable: Allows passing custom indexing logic to the handle method.
       - ReaderStandardAnalyzer

        - IndexWriterConfig
        - SerialMergeScheduler
        - Directory
        - IndexWriter
    - Aggregation:
    - Composition:
        - Logger

▼

▼ public class StarredArticleImportedEvent

1. Class Information
    - Class Name: StarredArticleImportedEvent
    - Purpose: This class acts as an event model for starred articles that are imported into the system. It holds references to both the article and the feed it belongs to, facilitating tracking and management of user-starred content.
2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
feed	private	Feed	The feed to which the starred article belongs.
article	private	Article	The article that has been starred and imported.

    - Methods:

Method Signature	Access Modifier	Description
Feed getFeed()	public	Returns the Feed object associated with this event.
void setFeed()	public	Sets the Feed object associated with this event.
Article getArticle()	public	Returns the Article object associated with this event.
void setArticle(Article article)	public	Sets the Article object associated with this event.
String toString()	public	Generates a string representation of the event using Objects.toStringHelper(), displaying feed and article details.

3. Relationships:
    - Inheritance: NA
    - Associations
        - Dependency:

1. Feed - The event contains a reference to a Feed, indicating that it is associated with but not dependent on a specific feed.
2. Article - The event references an Article, representing the starred article being imported.

▼ public class StarredReader

1. Class Information
   - Class Name: StarredReader
   - Purpose: The `StarredReader` class processes JSON input from a starred article export, extracts feed and article details, and notifies listeners of imported starred articles.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
log	private static final	Logger	Logger instance for logging information and errors.
starredArticleImportedListener	private	StarredArticleImportedListener	Listener that handles the import event for starred articles.

   - Methods:

Method Signature	Access Modifier	Description
void setStarredArticleListener(StarredArticleImportedListener starredArticleListener)	public	Reads a starred JSON file, validates and extracts article details, and triggers the import event.
void setStarredArticleListener(StarredArticleImportedListener starredArticleListener)	public	Sets the listener for starred article import events.

3. Relationships:
   - Inheritance: NA
   - Associations
     - Dependency:
       1. JsonValidationUtil - Used for validating JSON data fields.
       2. ObjectMapper - Reads and processes JSON content.
       3. Logger

4. MessageFormat

- Aggregation :
  1. StarredArticleImportedListener- Used to notify when a starred article is successfully imported.
- Composition:
  1. Feed - A `Feed` object is created for each starred article's source.
  2. Article - An `Article` object is created for each starred entry.

▼ public class StarredResource

1. Class Information

- **Class Name**: StarredResource
- **Purpose**: Manages user's starred articles including retrieval, starring, and unstarring operations

2. Attributes and Methods

Attributes

None explicitly defined

Methods

Method	Access Modifier	Description
`get(Integer limit, String afterArticle)`	Public	Retrieves starred articles with pagination
`star(String id)`	Public	Marks an article as starred
`unstar(String id)`	Public	Removes star marking from an article
`starMultiple(List<String> idList)`	Public	Stars multiple articles at once
`unstarMultiple(List<String> idList)`	Public	Unstars multiple articles at once

3. Relationships

Inheritance

- **Parent Class**: BaseResource
- **Child Classes**: None

Associations

Dependencies

- Context
- JsonHttpResponseHandler
- RequestParams

▼

▼