Subscription and Content Subsystem New

▼ public class FeedSubscription

1. Class Information
   - Class Name: FeedSubscription
   - Purpose:
     - Maintains user-feed relationships.
     - Supports categorization of feeds.
     - Tracks unread article counts.
     - Allows custom subscription titles.
2. Attributes and Methods
   - Attributes:

Attribute Name	Access Modifier	Type	Description
id	private	String	**Unique subscription ID** (Primary Key).
userId	private	String	**User who subscribed to the feed**.
feedId	private	String	**Feed being subscribed to**.
categoryId	private	String	**Category for this subscription**.
title	private	String	**Custom title for this subscription**.
order	private	Integer	**Display order in the category**.
unreadCount	private	Integer	**Number of unread articles**.
createDate	private	Date	**Date when the subscription was created**.
deleteDate	private	Date	**Date when the subscription was deleted**.

   - Methods:

Method Signature	Access Modifier	Description
String getId()	public	Returns the **subscription ID**.
void setId(String id)	public	Sets the **subscription ID**.
String getUserId()	public	Returns the **user ID**.
void setUserId(String userId)	public	Sets the **user ID**.
String getFeedId()	public	Returns the **feed ID**.
void setFeedId(String feedId)	public	Sets the **feed ID**.

String getCategoryId()	public	Returns the **category ID**.
void setCategoryId(String categoryId)	public	Sets the **category ID**.
String getTitle()	public	Returns the **custom title**.
void setTitle(String title)	public	Sets the **custom title**.
Integer getOrder()	public	Returns the **display order**.
void setOrder(Integer order)	public	Sets the **display order**.
Integer getUnreadCount()	public	Returns the **unread count**.
void setUnreadCount(Integer unreadCount)	public	Sets the **unread count**.
Date getCreateDate()	public	Returns the **creation date**.
void setCreateDate(Date createDate)	public	Sets the **creation 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 subscription.

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

▼ public class FeedSynchronization

1. Class Information
    - Class Name: FeedSynchronization
    - Purpose:
        - Logs synchronization attempts for feeds.
        - Provides debugging information when synchronization fails.
        - Stores performance metrics like synchronization duration.
2. Attributes and Methods
    - Attributes:

Attribute Name	**Access Modifier**	**Type**	**Description**
id	private	String	Unique identifier for feed synchronization.
feedId	private	String	Reference to the feed being synchronized.
success	private	boolean	True if synchronization succeeded, false otherwise.

message	private	String	Error message if synchronization failed (stored as LOB).
duration	private	Integer	Duration of synchronization in milliseconds.
createDate	private	Date	Timestamp of synchronization attempt.

- Methods:

Method Signature	Access Modifier	Description
String getId()	public	Returns the synchronization ID.
void setId(String id)	public	Sets the synchronization ID.
String getFeedId()	public	Returns the feed ID.
void setFeedId(String feedId)	public	Sets the feed ID.
boolean isSuccess()	public	Returns true if synchronization succeeded.
void setSuccess(boolean success)	public	Sets the success flag.
String getMessage()	public	Returns the error message.
void setMessage(String message)	public	Sets the error message.
Integer getDuration()	public	Returns the synchronization duration.
void setDuration(Integer duration)	public	Sets the synchronization duration.
Date getCreateDate()	public	Returns the creation timestamp.
void setCreateDate(Date createDate)	public	Sets the creation timestamp.
String toString()	public	Returns a string representation of the entity.

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

▼ public class FeedDao

1. Class Information
   - Class Name: FeedDao
   - Purpose: The `FeedDao` class manages data access operations for feed-related entities, including creation, retrieval, updates, and deletions of RSS feeds.
2. Attributes and Methods

- Attributes:

Attribute Name	Access Modifier	Type	Description
em	private	EntityManager	Manages persistence for feed entities

- Methods:

Method Signature	Access Modifier	Description
protected QueryParam getQueryParam(FeedCriteria, FilterCriteria)	protected	Builds the query for fetching feeds based on criteria
String create(Feed feed)	public	Creates a new feed and returns the newly generated feed ID
void delete(String id)	public	Deletes an existing feed by marking its delete date
Feed getByRssUrl(String rssUrl)	public	Retrieves a feed by its RSS URL
Feed update(Feed feed)	public	Updates the details of an existing feed

3. Relationships:
   - Inheritance:
     - Parent Class: BaseDao<FeedDto, FeedCriteria>
     - Child Classes: None
   - Associations:
     - Dependency: Feed, `EntityManager`, `ThreadLocalContext`, `FilterCriteria`, `QueryParam`
     - Aggregation: None
     - Composition: none

▼ public class FeedSubscriptionDao

1. Class Information
   - **Class Name:** FeedSubscriptionDao
   - **Purpose:** Manages data access operations for feed subscriptions, including CRUD operations, category-based retrieval, subscription reordering, and unread count updates.
2. Attributes and Methods

- **Attributes:**

Attribute Name	Access Modifier	Type	Description
em	private	EntityManager	Manages persistence for feed subscription entities

- **Methods:**

Method Signature	Access Modifier	Description

QueryParam getQueryParam(FeedSubscriptionCriteria, FilterCriteria)	protected	Builds query for fetching feed subscriptions based on criteria
String create(FeedSubscription)	public	Creates a new feed subscription and returns the generated ID
FeedSubscription update(FeedSubscription)	public	Updates the details of an existing feed subscription
void updateUnreadCount(String, Integer)	public	Updates the unread article count for a specific subscription
void reorder(FeedSubscription, int)	public	Moves and reorders subscriptions based on a display order
void delete(String)	public	Deletes a subscription by setting its deletion date
FeedSubscription getFeedSubscription(String, String)	public	Retrieves an active subscription by its ID and user ID
List<FeedSubscription> findByCategory(String)	public	Returns a list of active subscriptions in a category
int getCategoryCount(String, String)	public	Returns the count of subscriptions within a specific category

3. Relationships

- **Inheritance:**
  - Parent Class: `BaseDao<FeedSubscriptionDto, FeedSubscriptionCriteria>`
- **Associations:**
  - Dependency: Utilizes `EntityManager`, `FeedSubscriptionMapper`, `QueryParam` ,FeedSubscription, FeedSubscriptionDto, FeedSubscriptionCriteria
  - Composition: NA

▼ public class FeedSynchronizationDao

Class Documentation for `FeedSynchronizationDao`

1. Class Information

- **Class Name:** FeedSynchronizationDao
- **Purpose:** Manages data access operations for feed synchronization, including creation, deletion, and retrieval based on feed IDs.

2. Methods and Their Details

Method Signature	Access Modifier	Description
`String create(FeedSynchronization feedSynchronization)`	public	Creates a new feed synchronization record and returns its UUID.

`void deleteOldFeedSynchronization(String feedId, int minutes)`	public	Deletes old feed synchronization records older than a specified time period.
`List<FeedSynchronization> findByFeedId(String feedId)`	public	Retrieves all feed synchronization entries for a specific feed ID, ordered by creation date.

3. Relationships

- **Associations:**
  - Utilizes `EntityManager` for persistence operations.
  - Uses `DialectUtil` for database-specific date handling.
  - Works with the `FeedSynchronization` entity.

▼ public class FeedSubscriptionCriteria

---

1. Class Information

- **Class Name:** `FeedSubscriptionCriteria`
- **Purpose:**
  This class defines the criteria for filtering and managing feed subscriptions, enabling operations such as fetching specific subscriptions based on attributes like user, feed, category, and unread status.

---

2. Attributes and Methods

Attributes:

Attribute Name	Access Modifier	Type	Description
id	private	String	Unique identifier for the feed subscription.
userId	private	String	Identifier for the user owning the subscription.
feedId	private	String	Identifier for the subscribed feed.
categoryId	private	String	Identifier for the category under which the feed is organized.
feedUrl	private	String	URL of the subscribed feed.
unread	private	boolean	Indicates whether the subscription has unread articles.

Methods:

Method Signature	Access Modifier	Description
String getId()	public	Gets the subscription ID.
FeedSubscriptionCriteria setId(String id)	public	Sets the subscription ID.

String getUserId()	public	Gets the user ID.
FeedSubscriptionCriteria setUserId(String userId)	public	Sets the user ID.
String getFeedId()	public	Gets the feed ID.
FeedSubscriptionCriteria setFeedId(String feedId)	public	Sets the feed ID.
String getCategoryId()	public	Gets the category ID.
FeedSubscriptionCriteria setCategoryId(String categoryId)	public	Sets the category ID.
String getFeedUrl()	public	Gets the feed URL.
FeedSubscriptionCriteria setFeedUrl(String feedUrl)	public	Sets the feed URL.
boolean isUnread()	public	Checks if the subscription has unread articles.
FeedSubscriptionCriteria setUnread(boolean unread)	public	Sets the unread status for the subscription.

3. Relationships

- **Inheritance:**
  - **Parent Class:** None
  - **Child Classes:** None
- **Associations:**
  - **Dependency:** None explicitly defined but interacts with feed and user management components through attributes like `userId`, `feedId`, and `categoryId`.
  - **Aggregation:** None
  - **Composition:** None

▼ public class FeedSubscriptionMapper

1. Class Information

- **Class Name:** `FeedSubscriptionMapper`
- **Purpose:** Responsible for mapping raw database result sets to `FeedSubscriptionDto` objects, encapsulating detailed information about user feed subscriptions, such as unread article counts, feed metadata, and associated categories.

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 FeedSubscriptionDto map(Object[] o)`	`public`	Maps an array of raw database values to a `FeedSubscriptionDto` object.

3. Relationships

Inheritance:

- **Parent Class:** `ResultMapper<FeedSubscriptionDto>`
- **Child Classes:** None

Associations:

- **Dependency:** Depends on `FeedSubscriptionDto` for the target mapping type.

▼ public class FeedService

1. Class Information
   - Class Name: FeedService
   - Superclass: AbstractScheduledService
   - Purpose:
     - Manages scheduled updates of RSS feeds at a fixed interval.
     - Fetches new articles, removes obsolete content, and updates the feed metadata.
     - Ensures data integrity using transaction management via TransactionUtil.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
log	private stati	Logger	Logger for logging feed synchronization errors.

   - Methods:

Method Signature	Access Modifier	Description
void startUp()	protected	Called when the service starts (currently empty).
void shutDown()	protected	Called when the service stops (currently empty).
void runOneIteration()	protected	Runs one iteration of the feed synchronization process, handling exceptions explicitly.
Scheduler scheduler()	protected	Defines the update interval (every 10 minutes) for RSS synchronization.

void synchronizeAllFeeds()	public	Fetches all feeds with active subscribers, updates metadata, and stores synchronization status.
Feed synchronize(String url)	public	Fetches, parses, and stores a single RSS feed into the database.
void completeArticleList(List<Article> articleList)	private	Ensures each article has a valid publication date (prevents future-dated articles).
List<Article> getArticleToRemove(List<Article> articleList)	private	Identifies articles removed from the RSS feed (e.g., deleted posts).
boolean isFaviconUpdated(Feed feed)	private	Determines whether a favicon update is required (every 7 days).
RssReader parseFeedOrPage(String url, boolean parsePage)	private	Parses an RSS feed or webpage containing an RSS link, falling back if necessary.
void logParsingError(String url, Exception e)	private	Logs RSS parsing failures based on exception type.
void createInitialUserArticle(String userId, FeedSubscription feedSubscription)	public	Initializes unread articles for a new subscription, ensuring users have content to read.

3. Relationships:
    - Inheritance:
        - Parent Class: AbstractScheduledService
        - Child Classes:
    - Associations
        - Dependency:
            - TransactionUtil: Ensures safe transactions when updating feeds.
            - Logger : Used for logging errors but not owned.
            - AppContext : Used for event handling but not owned.
            - `ArticleCreatedAsyncEvent`, `ArticleDeletedAsyncEvent` - Used for event notifications but not owned.
        - Aggregation:
            - `FeedDao`, `FeedSynchronizationDao`, `FeedSubscriptionDao`, `ArticleDao`, `UserArticleDao` - Used for database operations but not owned permanently.
        - Composition:
            - FeedSynchronization - Created and managed entirely within `FeedService`.
            - ArticleSanitizer - Created and used exclusively within `synchronize()`

- FaviconUpdateRequestedEvent - Created within `FeedService` and does not exist independently.

▼ public class FeedChooserStrategy

1. Class Information
   - Class Name: FeedChooserStrategy
   - Purpose: This class implements a strategy to guess the correct RSS or Atom feed URL when multiple options are available. It selects the first feed URL from a given list.
2. Attributes and Methods
   - Attributes: NA
   - Methods:

Method Signature	Access Modifier	Description
String guess(List<String> feedList)	public	Selects and returns the most probable RSS/Atom feed URL from a list of provided URLs. The current implementation returns the first URL in the list.

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

▼ public class Feed

1. Class Information
   - Class Name: Feed
   - Purpose:
     - Stores RSS feed details.
     - Allows fetching and tracking of RSS articles.
     - Supports soft deletion through the deleteDate field.
2. Attributes and Methods
   - Attributes:

Attribute Name	**Access Modifier**	**Type**	**Description**
id	private	String	Unique feed identifier (Primary Key).
rssUrl	private	String	URL of the RSS feed.
url	private	String	Website URL of the feed.
baseUri	private	String	Relative base URI (Atom format).
title	private	String	Feed title.

language	private	String	Language of the feed (e.g., "en", "fr").
description	private	String	Description of the feed.
createDate	private	Date	Date when the feed was added.
lastFetchDate	private	Date	Date when the feed was last fetched.
deleteDate	private	Date	Date when the feed was marked for deletion.

- **Methods**:

Method Signature	Access Modifier	Description
public String getId()	public	Returns the feed ID.
public void setId(String id)	public	Sets the feed ID.
public String getRssUrl()	public	Returns the RSS feed URL.
public void setRssUrl(String rssUrl)	public	Sets the RSS feed URL.
public String getUrl()	public	Returns the website URL.
public void setUrl(String url)	public	Sets the website URL.
public String getBaseUri()	public	Returns the relative base URI.
public void setBaseUri(String baseUri)	public	Sets the relative base URI.
public String getTitle()	public	Returns the title of the feed.
public void setTitle(String title)	public	Sets the title of the feed.
public String getLanguage()	public	Returns the language of the feed.
public void setLanguage(String language)	public	Sets the language of the feed.
public String getDescription()	public	Returns the description of the feed.
public void setDescription(String description)	public	Sets the description of the feed.
public Date getCreateDate()	public	Returns the creation date.
public void setCreateDate(Date createDate)	public	Sets the creation date.
public Date getLastFetchDate()	public	Returns the last fetch date.
public void setLastFetchDate(Date lastFetchDate)	public	Sets the last fetch date.
public Date getDeleteDate()	public	Returns the deletion date.
public void setDeleteDate(Date deleteDate)	public	Sets the deletion date.
public String toString()	public	Returns a string representation of the feed

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

    - Child Classes: NA
- Associations
    - Dependency: NA
    - Aggregation:
        - Article: Feed Contains various article objects
    - Composition: NA

▼ public class FeedCriteria

1. Class Information

- **Class Name:**

  `FeedCriteria`
- **Purpose:**

  Encapsulates the parameters for filtering and querying feed data.

---

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`feedUrl`	private	`String`	URL of the feed.
`withUserSubscription`	private	`boolean`	Indicates whether to filter feeds with subscriptions.

Methods

Method Signature	Access Modifier	Description
`String getFeedUrl()`	public	Retrieves the feed URL.
`FeedCriteria setFeedUrl(String feedUrl)`	public	Sets the feed URL.
`boolean isWithUserSubscription()`	public	Checks if the filter for user subscriptions is enabled.
`FeedCriteria setWithUserSubscription(boolean val)`	public	Sets the filter to include feeds with user subscriptions.

---

3. Relationships

Inheritance

- **Parent Class:**

  None explicitly defined.
- **Child Classes:**

None.

Associations

- **Dependency:**
  - None specific beyond the use of primitive data types.

▼ public class FeedSubscriptionCriteria

1. Class Information

- **Class Name:**

  `FeedSubscriptionCriteria`

- **Purpose:**

  Defines filters and query parameters for retrieving feed subscription information, including criteria such as user, feed, category, and unread statuses.

---

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`id`	private	`String`	Unique identifier for the feed subscription.
`userId`	private	`String`	Identifier of the user owning the subscription.
`feedId`	private	`String`	Identifier of the feed being subscribed to.
`categoryId`	private	`String`	Category associated with the subscription.
`feedUrl`	private	`String`	URL of the feed.
`unread`	private	`boolean`	Filters subscriptions with unread articles if true.

Methods

Method Signature	Access Modifier	Description
`String getId()`	public	Retrieves the subscription ID.
`FeedSubscriptionCriteria setId(String id)`	public	Sets the subscription ID.
`String getUserId()`	public	Retrieves the user ID.
`FeedSubscriptionCriteria setUserId(String userId)`	public	Sets the user ID.
`String getFeedId()`	public	Retrieves the feed ID.
`FeedSubscriptionCriteria setFeedId(String feedId)`	public	Sets the feed ID.
`String getCategoryId()`	public	Retrieves the category ID.
`FeedSubscriptionCriteria setCategoryId(String id)`	public	Sets the category ID.
`String getFeedUrl()`	public	Retrieves the feed URL.

`FeedSubscriptionCriteria setFeedUrl(String feedUrl)`	public	Sets the feed URL.
`boolean isUnread()`	public	Checks if the criteria require unread subscriptions.
`FeedSubscriptionCriteria setUnread(boolean unread)`	public	Sets the unread filter criteria.

3. Relationships

Inheritance

- **Parent Class:**

  None explicitly defined.

- **Child Classes:**

  None.

Associations

- **Dependency: NA**

▼ public class FeedDto

1. Class Information

- **Class Name:** `FeedDto`
- **Purpose:** Represents feed-related data, such as the unique feed identifier and its corresponding RSS URL, enabling content retrieval and feed management.

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`id`	`private`	`String`	Unique identifier for the feed.
`rssUrl`	`private`	`String`	URL of the RSS feed.

Methods

Method Signature	Access Modifier	Description
`String getId()`	`public`	Gets the feed ID.
`void setId(String id)`	`public`	Sets the feed ID.
`String getRssUrl()`	`public`	Gets the feed RSS URL.
`void setRssUrl(String url)`	`public`	Sets the feed RSS URL.

3. Relationships

Inheritance

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

Associations

- **Dependency:** May depend on feed service classes for content updates
- **Aggregation:** None
- **Composition:** None

▼ public class FeedSubscriptionDto

---

1. Class Information

- **Class Name:** `FeedSubscriptionDto`
- **Purpose:** Represents a data model for feed subscriptions, including metadata about feeds, categories, and synchronization statuses for user subscriptions.

---

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`id`	`private`	`String`	Unique feed subscription ID.
`feedSubscriptionTitle`	`private`	`String`	Title of the feed subscription.
`feedTitle`	`private`	`String`	Title of the feed.
`userId`	`private`	`String`	Unique user ID associated with the subscription.
`feedId`	`private`	`String`	Unique ID of the feed.
`feedRssUrl`	`private`	`String`	RSS URL of the feed.
`feedUrl`	`private`	`String`	URL of the feed.
`feedDescription`	`private`	`String`	Description of the feed.
`unreadUserArticleCount`	`private`	`Integer`	Number of unread articles for the user.
`synchronizationFailCount`	`private`	`Integer`	Number of recent synchronization failures.
`createDate`	`private`	`Date`	Creation date of the subscription.
`categoryId`	`private`	`String`	ID of the associated category.
`categoryParentId`	`private`	`String`	Parent category ID.
`categoryName`	`private`	`String`	Name of the category.
`categoryFolded`	`private`	`boolean`	Indicates if the category is folded.

Methods

Method Signature	Access Modifier	Description

String getId()	public	Gets the subscription ID.
void setId(String id)	public	Sets the subscription ID.
String getFeedSubscriptionTitle()	public	Gets the feed subscription title.
void setFeedSubscriptionTitle(String title)	public	Sets the feed subscription title.
String getFeedTitle()	public	Gets the feed title.
void setFeedTitle(String feedTitle)	public	Sets the feed title.
String getUserId()	public	Gets the user ID.
void setUserId(String userId)	public	Sets the user ID.
String getFeedId()	public	Gets the feed ID.
void setFeedId(String feedId)	public	Sets the feed ID.
String getFeedRssUrl()	public	Gets the RSS URL.
void setFeedRssUrl(String feedRssUrl)	public	Sets the RSS URL.
String getFeedUrl()	public	Gets the feed URL.
void setFeedUrl(String feedUrl)	public	Sets the feed URL.
String getFeedDescription()	public	Gets the feed description.
void setFeedDescription(String feedDescription)	public	Sets the feed description.
Integer getUnreadUserArticleCount()	public	Gets the unread article count for the user.
void setUnreadUserArticleCount(Integer count)	public	Sets the unread article count for the user.
Integer getSynchronizationFailCount()	public	Gets the synchronization fail count.
void setSynchronizationFailCount(Integer count)	public	Sets the synchronization fail count.
Date getCreateDate()	public	Gets the subscription creation date.
void setCreateDate(Date createDate)	public	Sets the subscription creation date.
String getCategoryId()	public	Gets the category ID.
void setCategoryId(String categoryId)	public	Sets the category ID.
String getCategoryParentId()	public	Gets the parent category ID.
void setCategoryParentId(String parentId)	public	Sets the parent category ID.
String getCategoryName()	public	Gets the category name.
void setCategoryName(String categoryName)	public	Sets the category name.
boolean isCategoryFolded()	public	Checks if the category is folded.
void setCategoryFolded(boolean folded)	public	Sets the folded status of the category.

3. Relationships

Inheritance

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

Associations

- **Dependency:** None
- **Aggregation:** None

- **Composition:** None

▼ public class FeedMapper

1. Class Information

- **Class Name:** `FeedMapper`
- **Purpose:** Transforms raw database query results (`Object[]`) into structured `FeedDto` objects, mapping each value from the result set to corresponding attributes of the DTO.

---

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`dto`	private	`FeedDto`	Holds the mapped DTO instance
`i`	private	`int`	Iterates through input data indices

Methods

Method Signature	Access Modifier	Description
`FeedDto map(Object[] o)`	public	Maps database row to a `FeedDto` instance

---

3. Relationships

- **Inheritance:**
  - **Parent Class:** `ResultMapper<FeedDto>`
  - **Child Classes:** None
- **Associations:**
  - **Dependency:** `FeedDto` (as the output type)
  - **Aggregation:** None
  - **Composition:** None

▼ public class ArticleResource

1. Class Information

- Class Name: ArticleResource
- Purpose: Handles individual article operations and status management

2. Attributes and Methods

Methods

Method Signature	Access Modifier	Description
read(String id)	public	Marks single article as read

readMultiple(List<String> idList)	public	Marks multiple articles as read
unread(String id)	public	Marks single article as unread
unreadMultiple(List<String> idList)	public	Marks multiple articles as unread

3. Relationships

Inheritance

- Parent Class: BaseResource
- Child Classes: None

Associations

Dependencies:

- UserArticleDao
- ArticleDao
- FeedSubscriptionDao
- ArticleDto
- FeedSubscriptionDto
- ArticleCriteria
- FeedSubscriptionCriteria

▼ public class ArticleSanitizer

1. Class Information
   - Class Name: ArticleSanitizer
   - Purpose: The Article Sanitizer class ensures that article content retrieved from RSS feeds is **safe and properly formatted** before being displayed. It **sanitizes HTML content** by filtering allowed elements, transforming relative links into absolute URLs, and applying security policies to prevent malicious injections
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
log	private	Logger	Logger instance for logging errors and warnings.
INTEGER_POLICY	private	AttributePolicy	Attribute policy that ensures numeric values are valid integers.

   - Methods:

Method Signature	Access Modifier	Description

String sanitize(final String baseUri, String html)	public	Sanitizes the provided HTML content, ensuring safe display.

3. Relationships:
    - Inheritance:
        - Parent Class:
        - Child Classes:
    - Associations
        - Dependency:
            - UrlUtil: Used for converting relative urls to absolute urls
        - Aggregation:
        - Composition:

▼ public class ArticleMapper

1. Class Information

- **Class Name:** `ArticleMapper`
- **Purpose:** Transforms raw database query results (`Object[]`) into structured `ArticleDto` objects, mapping each value from the result set to corresponding attributes of the DTO.

---

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`dto`	private	`ArticleDto`	Holds the mapped DTO instance
`i`	private	`int`	Iterates through input data indices

Methods

Method Signature	Access Modifier	Description
`ArticleDto map(Object[] o)`	public	Maps database row to an `ArticleDto` instance

---

3. Relationships

- **Inheritance:**
    - **Parent Class:** `ResultMapper<ArticleDto>`
    - **Child Classes:** None
- **Associations:**
    - **Dependency:** `ArticleDto` (as the output type)

- **Aggregation:** None
- **Composition:** None

▼ public class Article

1. Class Information
   - Class Name: Article
   - Purpose:
     - Defines the structure for storing and retrieving articles from the database.
     - Supports CRUD operations via JPA repositories.
     - Provides metadata for RSS feed synchronization and user interactions.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
id	private	String	Unique identifier for the article (Primary Key).
feedId	private	String	ID of the RSS feed the article belongs to.
url	private	String	Direct URL to the article.
baseUri	private	String	Base URI for Atom feeds.
guid	private	String	Globally Unique Identifier (GUID) for the article.
title	private	String	Title of the article.
creator	private	String	Author or creator of the article.
description	private	String	HTML description or summary of the article.
commentUrl	private	String	URL for user comments on the article.
commentCount	private	String	Number of comments on the article.
enclosureUrl	private	String	URL of media attachments (images, videos, audio).
enclosureLength	private	integer	Size of enclosure (bytes).
enclosureType	private	String	MIME type of enclosure (e.g., image/png).
publicationDate	private	Date	Date the article was published
createDate	private	Date	Date the article was added to the system.
deleteDate	private	Date	Date the article was deleted (soft delete)

   - Methods:

Method Signature	**Access Modifier**	Description
String getId()	public	Returns the article ID.

void setId(String id)	public	Sets the article ID.
String getFeedId()	public	Returns the feed ID.
public void setFeedId(String feedId)	public	Sets the feed ID.
public String getUrl()	public	Returns the article URL.
public void setUrl(String url)	public	Sets the article URL.
public String getBaseUri()	public	Returns the base URI for Atom feeds.
public void setBaseUri(String baseUri)	public	Sets the base URI.
public String getGuid()	public	Returns the GUID of the article.
public void setGuid(String guid)	public	Sets the GUID.
public String getTitle()	public	Returns the article title.
public void setTitle(String title)	public	Sets the title.
public String getCreator()	public	Returns the author of the article.
public void setCreator(String creator)	public	Sets the creator.
public String getDescription()	public	Returns the article description.
public void setDescription(String description)	public	Sets the description.
public String getCommentUrl()	public	Returns the comment URL.
public void setCommentUrl(String commentUrl)	public	Sets the comment URL.
public Integer getCommentCount()	public	Returns the number of comments.
public void setCommentCount(Integer commentCount)	public	Sets the comment count.
public Date getPublicationDate()	public	Returns the publication date.
public void setPublicationDate(Date publicationDate)	public	Sets the publication date.
public Date getCreateDate()	public	Returns the creation date.
public void setCreateDate(Date createDate)	public	Sets the creation date.
public Date getDeleteDate()	public	Returns the deletion date.
public void setDeleteDate(Date deleteDate)	public	Sets the deletion date.
public String toString()	public	Returns a string representation of the article.

3. Relationships:
    - Inheritance:
        - Parent Class:
        - Child Classes:
    - Associations

- Dependency:
- Aggregation:
- Composition:

▼ public class ArticleDao

---

1. Class Information

- **Class Name:** `ArticleDao`
- **Purpose:**
  - Handles database operations for creating, reading, updating, and deleting article records.
  - Supports querying articles based on specific search criteria for efficient feed management.

---

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`em`	Private (through method access)	`EntityManager`	Handles interactions with the database using persistence context.
`criteriaList`	Local variable	`List<String>`	Holds conditions for dynamic SQL queries.
`parameterMap`	Local variable	`Map<String, Object>`	Stores key-value pairs for query parameters.

Methods

Method Signature	Access Modifier	Description
`protected QueryParam getQueryParam(ArticleCriteria criteria, FilterCriteria filterCriteria)`	Protected	Builds query parameters based on article criteria.
`public String create(Article article)`	Public	Inserts a new article into the database and returns the generated ID.
`public Article update(Article article)`	Public	Updates an existing article's attributes.
`public List<Article> findAll()`	Public	Fetches all non-deleted articles.
`public void delete(String id)`	Public	Marks an article and related user-article mappings as deleted.

---

3. Relationships

Inheritance

- **Parent Class:** `BaseDao<ArticleDto, ArticleCriteria>`
- **Child Classes:** None

Associations

- **Dependency:**
  - `QueryParam`, `SortCriteria`, `ArticleMapper`: Support query building and execution
  - ArticleCriteria
  - FilterCriteria
  - ArticleMapper
- **Aggregation:**
  - Article
- **Composition:**
  - EntityManager

▼ public class ArticleDto

---

1. Class Information

- **Class Name:** `ArticleDto`
- **Purpose:** Represents essential data about articles, including metadata such as title, creator, and publication date, enabling efficient data transfer between subsystems and the user interface.

---

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`id`	`private`	`String`	Unique identifier for the article.
`url`	`private`	`String`	The URL of the article.
`guid`	`private`	`String`	Globally unique identifier for the article.
`title`	`private`	`String`	Title of the article.
`creator`	`private`	`String`	Name of the person or entity that created the article.
`description`	`private`	`String`	Brief description of the article.
`commentUrl`	`private`	`String`	URL for the article's comments section.
`commentCount`	`private`	`Integer`	Number of comments on the article.
`enclosureUrl`	`private`	`String`	URL for associated media enclosure.
`enclosureCount`	`private`	`Integer`	Size of the media enclosure in bytes.

enclosureType	private	String	MIME type of the media enclosure.
publicationDate	private	Date	Date when the article was published.
createDate	private	Date	Date when the article was created in the system.
feedId	private	String	ID of the feed to which this article belongs.

Methods

Method Signature	Access Modifier	Description
String getId()	public	Gets the article ID.
void setId(String id)	public	Sets the article ID.
String getUrl()	public	Gets the article URL.
void setUrl(String url)	public	Sets the article URL.
String getGuid()	public	Gets the article GUID.
void setGuid(String guid)	public	Sets the article GUID.
String getTitle()	public	Gets the article title.
void setTitle(String title)	public	Sets the article title.
String getCreator()	public	Gets the article creator.
void setCreator(String creator)	public	Sets the article creator.
String getDescription()	public	Gets the article description.
void setDescription(String desc)	public	Sets the article description.
String getCommentUrl()	public	Gets the comment URL.
void setCommentUrl(String url)	public	Sets the comment URL.
Integer getCommentCount()	public	Gets the comment count.
void setCommentCount(Integer count)	public	Sets the comment count.
Date getPublicationDate()	public	Gets the publication date.
void setPublicationDate(Date date)	public	Sets the publication date.
Date getCreateDate()	public	Gets the creation date.
void setCreateDate(Date date)	public	Sets the creation date.
String getEnclosureUrl()	public	Gets the enclosure URL.
void setEnclosureUrl(String url)	public	Sets the enclosure URL.
Integer getEnclosureCount()	public	Gets the enclosure count.
void setEnclosureCount(Integer cnt)	public	Sets the enclosure count.
String getEnclosureType()	public	Gets the enclosure MIME type.
void setEnclosureType(String type)	public	Sets the enclosure MIME type.
String getFeedId()	public	Gets the feed ID.
void setFeedId(String feedId)	public	Sets the feed ID.

3. Relationships

Inheritance

- **Parent Class:** None

- **Child Classes:** None

Associations

- **Dependency:** None
- **Aggregation: None**
- **Composition:** None

▼ public class ArticleDeletedAsyncEvent

1. Class Information
   - Class Name: ArticleDeletedAsyncEvent
   - Purpose: This class represents an event that occurs when articles are deleted. It contains a list of deleted articles and provides getter and setter methods to access and modify the data.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
articleList	private	List	A list storing the articles that were deleted.

   - Methods:

Method Signature	Access Modifier	Description
List<Article> getArticleList()	public	Retrieves the list of deleted articles.
void setArticleList(List<Article> articleList)	public	Sets the list of deleted articles.
String toString()	public	Overrides toString to provide a string representation of the deleted articles count.

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

▼ public class ArticleUpdatedAsyncEvent

1. Class Information
   - Class Name: ArticleUpdatedAsyncEvent
   - Purpose: This class serves as an event model that carries a list of updated articles. It allows different components of the system to be notified of article updates asynchronously.

2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
articleList	private	List<Article>	Stores the list of updated articles.

    - Methods:

Method Signature	Access Modifier	Description
List<Article> getArticleList()	public	Retrieves the list of updated articles.
void setArticleList(List<Article> articleList)	public	Sets the list of updated articles.
String toString()	public	Returns a string representation of the object, including the number of updated articles

3. Relationships:
    - Inheritance: NA
    - Associations
        - Dependency: Article - The class holds a list of Article objects, representing updated articles.

▼ public class ArticleDeletedAsyncListener

1. Class Information
    - Class Name: ArticleDeletedAsyncListener
    - Purpose: This class listens for `ArticleDeletedAsyncEvent` events and triggers the deletion of corresponding indexed articles using the `ArticleDao` class. It logs the process to track deletion performance. It ensures that when an article is deleted, its index is also removed from the search database, thereby maintaining consistency in article management.
2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
log	private static final	Logger	Logger instance for logging information related to article deletion events.

    - Methods:

Method Signature	Access Modifier	Description

void onArticleDeleted(ArticleDeletedAsyncEvent articlesDeletedAsyncEvent) throws Exception	public	Listens for deleted article events, removes the article index, and logs the process.

3. Relationships:
    - Inheritance: NA
    - Associations
        - Dependency:
            1. `ArticleDeletedAsyncEvent` - The class listens to this event to trigger article deletion processes.
        - Aggregation :
            - ArticleDao

▼ public class ArticleUpdatedAsyncListener

1. Class Information
    - Class Name: ArticleUpdatedAsyncListener
    - Purpose: This class listens for asynchronous article update events and triggers the re-indexing process to maintain an up-to-date article index in the system. It plays a crucial role in maintaining an organized and searchable feed by processing updated articles asynchronously.
2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
log	private static final	Logger	Logger instance for logging messages related to article updates.

    - Methods:

Method Signature	Access Modifier	Description
void onArticleUpdated(ArticleUpdatedAsyncEvent articlesUpdatedAsyncEvent)	public	Handles article update events by updating the article index and logging the process duration.

3. Relationships:
    - Inheritance: NA
    - Associations
        - Dependency:
            1. `ArticleUpdatedAsyncEvent` - Listens for this event to detect updated articles.

2. `ArticleDao` - Calls `update()` method on `ArticleDao` to re-index the updated articles.
3. `Logger` - Uses `log` for debugging and logging event updates.

▼ public class ArticleCriteria

1. Class Information

- **Class Name:**
  `ArticleCriteria`
- **Role in Subsystem:**
  Part of the **Feed Organization Subsystem**, responsible for filtering and retrieving articles.
- **Purpose:**
  Encapsulates criteria for querying and filtering article data from the database.

---

2. Attributes and Methods

Attributes

Attribute Name	Access Modifier	Type	Description
`id`	private	`String`	Unique identifier of the article.
`guidIn`	private	`List<String>`	List of GUIDs to include in the search results.
`title`	private	`String`	Title of the article.
`url`	private	`String`	URL of the article.
`publicationDateMin`	private	`Date`	Minimum publication date for filtering articles.
`feedId`	private	`String`	Identifier of the feed associated with the article.

Methods

Method Signature	Access Modifier	Description
`String getId()`	public	Retrieves the article ID.
`ArticleCriteria setId(String id)`	public	Sets the article ID.
`List<String> getGuidIn()`	public	Retrieves the list of article GUIDs.
`ArticleCriteria setGuidIn(List guidIn)`	public	Sets the list of article GUIDs.
`String getTitle()`	public	Retrieves the title of the article.
`ArticleCriteria setTitle(String title)`	public	Sets the title of the article.
`String getUrl()`	public	Retrieves the URL of the article.
`ArticleCriteria setUrl(String url)`	public	Sets the URL of the article.
`Date getPublicationDateMin()`	public	Retrieves the minimum publication date.
`ArticleCriteria setPublicationDateMin(Date date)`	public	Sets the minimum publication date for filtering.

`String getFeedId()`	public	Retrieves the feed ID associated with the article.
`ArticleCriteria setFeedId(String feedId)`	public	Sets the feed ID for filtering articles.

3. Relationships

Inheritance

- **Parent Class:**

  None explicitly defined.

- **Child Classes:**

  None.

Associations

- **Dependency: NA**

▼ public class ArticleCreatedAsyncEvent

1. Class Information
   - Class Name: ArticleCreatedAsyncEvent
   - Purpose: This class represents an event triggered when new articles are created in the system. It encapsulates a list of newly created articles and provides methods to access and modify this list.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
articleList	Private	List	Stores a list of newly created articles

   - Methods:

Method Signature	Access Modifier	Description
List getArticleList()	public	Retrieves the list of newly created articles
void setArticleList(List articleList)	public	Sets the list of newly created articles.
String toString()	public	Returns a string representation of the event, including article count.

3. Relationships:
   - Inheritance : NA
   - Associations (Other components that listen for and process this event asynchronously.)

- Composition: Article (Composition: The event contains a list of Article objects.)

▼ public class ArticleCreatedAsyncListener

1. Class Information
   - Class Name: ArticleCreatedAsyncListener
   - Purpose: This class listens for events related to new article creation and processes them asynchronously by indexing them in the article database. It enhances the system's efficiency in handling new content.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
log	private static final	Logger	Logger instance to log processing details and errors.

   - Methods:

Method Signature	Access Modifier	Description
void onArticleCreated(ArticleCreatedAsyncEvent articlesCreatedAsyncEvent)	public	Handles the ArticleCreatedAsyncEvent, extracts the list of newly created articles, and indexes them using ArticleDao.

3. Relationships:
   - Inheritance: NA
   - Associations
     - Dependency:
       1. ArticleCreatedAsyncEvent- This class listens for `ArticleCreatedAsyncEvent` and extracts the list of new articles.
     - Aggregation :
       - ArticleDao

▼ public class SubscriptionImportedEvent

1. Class Information
   - Class Name: SubscriptionImportedEvent

- Purpose: This class encapsulates the details of a subscription import request, including the user making the request and the file to be imported. It is used to notify the system about an import operation and facilitate the handling of the subscription import process.

2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
user	private	User	Represents the user requesting the subscription import.
importFile	private	File	Represents the file that contains the subscription data to be imported.

   - Methods:

Method Signature	Access Modifier	Description
User getUser()	public	Returns the user who requested the subscription import.
void setUser(User user)	public	Sets the user who initiated the subscription import request.
File getImportFile()	public	Returns the file containing the subscriptions to be imported.
void setImportFile(File importFile)	public	Sets the file to be imported.
String toString()	public	Provides a string representation of the event, including user and import file details.

3. Relationships:
   - Inheritance: NA
   - Associations
     - Dependency:
       1. `User` - The class references a `User` object to identify who initiated the import request.
       2. `File` - The class holds a reference to a `File` object, which contains the subscriptions to be imported.

▼ public class SubscriptionImportAsyncListener

1. Class Information
   - Class Name: SubscriptionImportAsyncListener
   - Purpose: This class listens for subscription import events and processes OPML files or ZIP archives containing feed subscription data. It extracts feed information, starred articles, and updates the relevant user subscriptions and article data accordingly.

2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
log	private static final	Logger	Logger instance for logging events and errors.
FILE_STARRED_JSON	private static final	String	Constant for the starred articles file name in Google Takeout.
FILE_SUBSCRIPTIONS_XML	private static final	String	Constant for the subscriptions file name in Google Takeout.

    - Methods:

Method Signature	Access Modifier	Description
void onSubscriptionImport(SubscriptionImportedEvent subscriptionImportedEvent)	public	Handles the OPML import event, triggering the processing of the file.
Job createJob(User user, File importFile)	private	Reads the OPML file, determines the number of feeds/articles, and creates a job entry for tracking import progress.
long getFeedCount(List<Outline> outlineList)	private	Counts the number of feeds in an OPML tree structure.
void processImportFile(User user, File importFile, Job job)	private	Extracts and processes subscription data from the import file, handling OPML and ZIP file formats.
void importOutline(User user, List<Outline> outlineList, Job job)	private	Imports categories and feeds from an OPML file, creating necessary database records.
void importFeedFromStarred(User user, Feed feed, Article article)	private	Imports feeds and articles from starred article data in Google Takeout.

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

        - FeedSubscription
        - UserArticle
        - JobEvent
    - Aggregation :
        - User
        - `FeedSubscriptionDao`
        - `JobDao`,
        - `JobEventDao`
        - , `ArticleDao`,
        - `UserArticleDao`
        - FeedService
    - Dependency:
        1. SubscriptionImportedEvent - Listens to subscription import events and processes them.
        2. OpmlReader, OpmlFlattener - Parses and flattens OPML subscription files.
        3. FeedSubscriptionDao, CategoryDao - Manages user feed subscriptions and category assignments.
        4. EntityManagerUtil, TransactionUtil - Manages database transactions and entity persistence.
        5. MimeTypeUtil - Determines MIME type of files)
        6. StarredReader
        7. ArticleCreatedAsyncEvent

▼ public class SubscriptionParser

1. Class Information
    - Class Name: SubscriptionParser
    - Purpose: The class parses XML subscription data using the SAX parser approach. It processes elements related to subscriptions and extracts relevant attributes, storing them as `Subscription` objects in a list.
2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
`SUBSCRIPTION`	`private static final`	`String`	Constant representing the XML tag for a subscription entry.

`TITLE`	`private static final`	`String`	Constant representing the XML attribute for subscription title.
`SPECIALIZATION`	`private static final`	`String`	Constant representing the XML attribute for subscription specialization.
`URL`	`private static final`	`String`	Constant representing the XML attribute for subscription URL.
`HOMEPAGE`	`private static final`	`String`	Constant representing the XML attribute for subscription homepage.
`PREFIXES`	`private static final`	`String`	Constant representing the XML attribute for subscription prefixes.
`AUTHOR`	`private static final`	`String`	Constant representing the XML attribute for subscription author.
`subscriptions`	`private`	`List<Subscription>`	Stores the list of parsed subscriptions.
`currentSubscription`	`private`	`Subscription`	Temporary object used to store subscription details while parsing.

- Methods:

Method Signature	Access Modifier	Description
`SubscriptionParser(List<Subscription> subscriptions)`	`public`	Constructor initializing the parser with a list of subscriptions.
`void startElement(String uri, String localName, String qName, Attributes attributes)`	`public` (overridden)	Called at the start of an XML element, initializes a `Subscription` object when encountering a `<subscription>` tag and populates its attributes.
`void endElement(String uri, String localName, String qName)`	`public` (overridden)	Called at the end of an XML element, adds the parsed `Subscription` object to the list when encountering a `</subscription>` tag.

3. Relationships:
   - Inheritance:
     - Parent Class: `DefaultHandler` (from `org.xml.sax.helpers.DefaultHandler`)
   - Associations
     - Dependency: Subscription - The SubscriptionParser creates instances of Subscription to store parsed data.
     - Aggregation: List < Subscription > - A list is used to store multiple parsed subscription objects.

▼ public class Subscription

1. Class Information
   - Class Name: Subscription

- Purpose: The `Subscription` class stores metadata related to an Adblock Plus subscription, including its title, specialization, URL, homepage, prefixes, and author. It provides structured information about filter lists that can be used to block specific advertisements.

2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
`title`	`public`	`String`	The title or name of the subscription.
`specialization`	`public`	`String`	A brief description of the specialization of the subscription.
`url`	`public`	`String`	The URL from which the subscription can be retrieved.
`homepage`	`public`	`String`	The homepage of the subscription or its source.
`prefixes`	`public`	`String[]`	A list of URL prefixes that this subscription covers.
`author`	`public`	`String`	The author or maintainer of the subscription.

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

▼

▼ public class OpmlFlattener

1. Class Information
   - Class Name: OpmlFlattener
   - Purpose: This utility class flattens the hierarchical structure of an OPML outline tree into a map representation with a single category level, simplifying the structure for further processing within the RSS reader.
2. Attributes and Methods

- Attributes: NA
- Methods:

Method Signature	Access Modifier	Description
`public static Map<String, List<Outline>> flatten(List<Outline> outlineList)`	`public`	Flattens an OPML outline tree, returning a mapping of categories to their respective feed outlines.
`private static void flatten(List<Outline> outlineTree, Map<String, List<Outline>> outlineMap, String prefix)`	`private`	Recursively processes an outline tree, categorizing feeds under a single-level structure.
`private static String getPrefix(Outline outline, String prefix)`	`private`	Determines the category prefix for an outline entry based on its text or title.

3. Relationships:
   - Inheritance: NA
   - Associations
     - Dependency:
       1. Outline - `OpmlFlattener` depends on `Outline` to access information about OPML structures (such as feed URLs and category names).

▼ public class OpmlReader

1. Class Information
   - Class Name: OpmlReader
   - Purpose: The `OpmlReader` class is an OPML parser that reads an OPML file and converts it into a hierarchical tree structure of `Outline` objects, representing feed categories and subscriptions.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
`log`	private	Logger	Logger for debugging and error handling.
`content`	private	String	Stores the text content of the current XML element.
`rootOutline`	private	Outline	The root outline of the OPML tree.
`currentOutline`	private	Outline	Reference to the currently active `Outline` node during parsing.
`currentElement`	private	Element	Keeps track of the current XML element type.
`elementStack`	private	Stack<Element>	Stack used to track nested XML elements.

`outlineStack`	private	Stack<Outline>	Stack used to track nested `Outline` objects during parsing.

- Methods:

Method Signature	Access Modifier	Description
`OpmlReader()`	public	Constructor initializes the element and outline stacks.
`void read(InputStream is)`	public	Reads an OPML file from an input stream and parses it into an outline tree structure.
`void startElement(String uri, String localName, String qName, Attributes attributes)`	public	Called when the parser encounters the start of an XML element. It initializes `Outline` objects and sets their attributes.
`void endElement(String uri, String localName, String qName)`	public	Called when the parser encounters the end of an XML element. It updates the element stack and the outline hierarchy.
`void characters(char[] ch, int start, int length)`	public	Collects text content from XML elements.
`List<Outline> getOutlineList()`	public	Returns the list of outlines parsed from the OPML file.
`private void pushElement(Element newElement)`	private	Pushes a new XML element onto the element stack and updates `currentElement`.
`private void popElement()`	private	Pops the top element from the element stack and updates `currentElement`.

3. Relationships:
   - Inheritance:
     - Parent Class: `DefaultHandler` (from `org.xml.sax.helpers`)
   - Associations
     - Dependency:
       1. SAXParser,SAXParserFactory - `OpmlReader` uses `SAXParser` to process XML efficiently.
       2. Logger - Uses `Logger` for debugging and logging warnings/errors.
     - Aggregation:
       1. Outline - `OpmlReader` constructs a hierarchical tree of `Outline` objects while parsing OPML.

▼ public class RssReader

1. Class Information
   - Class Name: RssReader
   - Purpose:

The `RssReader` class reads and parses RSS and Atom feeds, extracting relevant metadata, articles, and content. It ensures compatibility with multiple feed formats, processes errors gracefully, and structures the parsed data for further use in the system.

2. Attributes and Methods

- Attributes:

Name	Access Modifier	Type	Description
`DF_RSS`	public static final	`DateTimeFormatter`	Common date formats for RSS feeds.
`DF_ATOM`	public static final	`DateTimeFormatter`	Common date formats for Atom feeds.
`DF_DC`	public static final	`DateTimeFormatter`	Common date formats for Dublin Core metadata.
`content`	private	`String`	Stores the contents of the current XML element being processed.
`feed`	private	`Feed`	Represents the parsed feed.
`article`	private	`Article`	Represents the currently parsed article.
`articleList`	private	`List<Article>`	Stores all parsed articles.
`atomLinkList`	private	`List<AtomLink>`	Stores Atom feed links.
`atomArticleLinkList`	private	`List<AtomLink>`	Stores Atom article links.
`fatalErrorCount`	private	`int`	Tracks the number of fatal parsing errors encountered.
`currentElement`	private	`Element`	Tracks the currently processed XML element.
`elementStack`	private	`Stack<Element>`	Maintains a stack of parsed XML elements.
`feedType`	private	`FeedType`	Identifies the type of the feed being processed (RSS, Atom, RDF).

- Methods:

Method Signature	Access Modifier	Description
`RssReader()`	public	Constructor that initializes the `elementStack`.
`void readRssFeed(InputStream is)`	public	Reads and parses an RSS/Atom feed from an input stream.
`void startElement(String uri, String localName, String qName, Attributes attributes)`	public	Handles XML opening tags and initializes feed/article objects.
`void endElement(String uri, String localName, String qName)`	public	Handles XML closing tags and assigns extracted data to objects.
`void characters(char[] ch, int start, int length)`	public	Reads character data inside XML elements.

`void fatalError(SAXParseException e)`	public	Handles and logs parsing errors.
`Feed getFeed()`	public	Returns the parsed feed object.
`List<Article> getArticleList()`	public	Returns the list of parsed articles.
`void validateFeed()`	private	Validates if the feed is properly structured.
`void fixGuid()`	private	Attempts to correct missing or invalid GUIDs in articles.
`void initFeed()`	private	Initializes a new feed and article list.
`void pushElement(Element newElement)`	private	Adds an element to the processing stack.
`void popElement()`	private	Removes the last processed element from the stack.
`String getContent()`	private	Retrieves and trims the current element's content.

3. Relationships:
   - Inheritance:
     - Parent Class: `DefaultHandler` (from `org.xml.sax.helpers.DefaultHandler`)
   - Associations
     - Dependency:
       1. StreamUtil - Utilized to detect and process compressed streams.
       2. DateUtil - Used for parsing and formatting dates in different feed formats.
       3. UrlUtil - Assists in handling URLs found in feeds.
     - Aggregation:
       1. AtomLink - Stores links related to Atom feeds.
       2. Feed
       3. Article

▼ public class RssExtractor

1. Class Information
   - Class Name: RssExtractor
   - Purpose: The class is designed to scan an HTML page and extract RSS or Atom feed URLs, enabling users to subscribe to feeds from websites that offer syndicated content.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description

log	private static final	Logger	Logger for error reporting and debugging.
url	private final	URL	The original page URL provided for extraction.
feedlist	private	List<String>	List of extracted RSS/Atom feed URLs.

- Methods:

Method Signature	Access Modifier	Description
RssExtractor(String url)	public	Constructor that initializes the extractor with a given webpage URL.
void readPage(InputStream is)	public	Reads and parses an HTML page to find RSS/Atom feed links.
void startElement(String uri, String localName, String qName, Attributes attributes)	public (overriden)	SAX parser method to detect <link> elements containing RSS/Atom feed URLs.
List<String> getFeedList()	public	Returns the list of extracted feed URLs.

3. Relationships:
   - Inheritance:
     - Parent Class: DefaultHandler (from org.xml.sax.helpers.DefaultHandler)
   - Associations
     - Dependency:
       1. `UrlUtil` (from `com.sismics.util`) - Used to resolve and complete extracted URLs.

▼ public class XmlReader

1. Class Information
   - Class Name: XmlReader
   - Purpose: The `XmlReader` class extends `Reader` and is designed to identify and use the correct character encoding for an XML input stream. It removes any Byte Order Mark (BOM) and, if necessary, determines the encoding from the XML declaration. This improves the handling of RSS feeds by ensuring proper character representation.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
`internalInputStreamReader`	private	`InputStreamReader`	A reader that reads characters from the input stream using the detected encoding.
`HEADER_SIZE`	private	`static final int`	Constant representing the number of bytes to read from

			the XML header to detect encoding.

- Methods:

Method Signature	Access Modifier	Description
`XmlReader(InputStream in, String defaultEnc)`	public	Constructor that reads an input stream, detects encoding via BOM or XML header, and initializes an `InputStreamReader` with the appropriate charset.
`void close()`	public	Closes the internal `InputStreamReader`, releasing resources.
`int read(char[] cbuf, int off, int len)`	public	Reads characters into an array, using the detected encoding.

3. Relationships:
   - Inheritance:
     - Parent Class: Reader
   - Associations
     - Dependency:
       1. PushbackInputStream - Used to read ahead in the stream and push back bytes when necessary.
       2. ByteStreams - Utility from Google's Guava library to read bytes efficiently.
       3. Pattern , Matcher - Used to extract encoding information from the XML declaration.
     - Aggregation:
     - Composition:
       1. InputStreamReader - `XmlReader` contains an instance of `InputStreamReader` to handle character conversion.

▼ public class AtomArticleCommentUrlGuesserStrategy

1. Class Information
   - Class Name: AtomArticleCommentUrlGuesserStrategy
   - Purpose: This class provides a strategy to determine the best URL for article comments from a list of Atom feed links. It prioritizes links with specific characteristics, such as "replies" with an "html" content type.
2. Attributes and Methods
   - Attributes: NA
   - Methods:

Method Signature	Access Modifier	Description

String guess(List<AtomLink> atomLinkList)	public	Iterates through a list of AtomLink objects to determine the best URL for article comments, prioritizing "replies" links with "text/html" content.

3. Relationships:
    - Inheritance: NA
    - Associations
        - Dependency:
            1. `AtomLink` – The `guess` method processes a list of `AtomLink` objects to extract a suitable comment URL.

▼

▼ public class AtomLink

1. Class Information
    - Class Name: AtomLink
    - Purpose: The `AtomLink` class encapsulates information about an Atom link, including its relationship type, content type, and reference URL. This class is used in parsing and handling RSS/Atom feed data within the RSS reader system.
2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
rel	private	String	Specifies the relationship type of the link (e.g., "self", "alternate").
type	private	String	Specifies the content type of the linked resource.
href	private	String	The reference URL of the linked resource.

    - Methods:

Method Signature	Access Modifier	Description
AtomLink(String rel, String type, String href)	public	Constructor to initialize the AtomLink object with rel, type, and href.
String getRel()	public	Returns the relationship type of the link.
void setRel(String rel)	public	Sets the relationship type of the link.
String getHref()	public	Returns the reference URL of the link.
void setHref(String href)	public	Sets the reference URL of the link.
String getType()	public	Returns the content type of the link.
void setType(String type)	public	Sets the content type of the link.

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

▼ public class AtomUrlGuesserStrategy

1. Class Information
    - Class Name: AtomUrlGuesserStrategy
    - Purpose: This class provides methods to extract and determine the appropriate site URL and feed URL from a list of `AtomLink` objects based on specific conditions.
2. Attributes and Methods
    - Attributes: NA
    - Methods:

Method Signature	Access Modifier	Description
String guessSiteUrl(List<AtomLink> atomLinkList)	public	Determines the correct site URL from a list of AtomLink objects. Prefers alternate links first and then returns the first valid link that is not self.
String guessFeedUrl(List<AtomLink> atomLinkList)	public	Identifies the correct feed URL by selecting the self link from the list of AtomLink objects.

3. Relationships:
    - Inheritance: NA
    - Associations
        - Dependency: AtomLink - The class depends on AtomLink objects, which provide the URL and rel attributes for determining the correct site and feed URLs.

▼ public class AtomArticleUrlGuesserStrategy

1. Class Information
    - Class Name: AtomArticleUrlGuesserStrategy
    - Purpose: The class provides a strategy for selecting the most relevant article URL from a list of Atom links associated with a feed entry. It follows a priority-based approach to determine the best URL.
2. Attributes and Methods
    - Attributes: NA
    - Methods:

Method Signature	Access Modifier	Description
String guess(List<AtomLink> atomLinkList)	public	Determines the most appropriate article URL from a list of Atom feed links using a priority-based approach.

3. Relationships:
    - Inheritance: NA
    - Associations
        - Dependency:
            1. AtomLink - The `guess` method relies on a list of `AtomLink` objects to determine the correct URL

▼

▼ public class FaviconUpdateRequestedAsyncListener

1. Class Information
    - Class Name: FaviconUpdateRequestedAsyncListener
    - Purpose: This class is an event listener responsible for handling favicon update requests for RSS feeds. It listens for `FaviconUpdateRequestedEvent` and attempts to download the corresponding favicon using `FaviconDownloader`.
2. Attributes and Methods
    - Attributes:

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

    - Methods:

Method Signature	Access Modifier	Description
void onFaviconUpdateRequested(FaviconUpdateRequestedEvent event)	public	Listens for a favicon update request and processes it asynchronously.
TransactionUtil.handle(Runnable task)	private static	Executes the favicon update logic within a transaction to ensure data integrity.

3. Relationships:
    - Inheritance: NA

- Associations
  - Dependency:
    1. FaviconUpdateRequestedEvent - The listener subscribes to this event and processes it when triggered.
    2. Feed - The feed object is extracted from the event to determine which favicon to update.
    3. DirectoryUtil - Retrieves the directory where favicons should be stored.
    4. TransactionUtil - Ensures the favicon update process occurs within a transaction.
  - Aggregation: FaviconDownloader - This class is responsible for downloading favicons.

▼ public class FaviconDownloader

1. Class Information
   - Class Name: FaviconDownloader
   - Purpose: The class provides methods to download favicons from a given webpage URL. It first tries to extract the favicon URL from the page metadata, then checks common locations like `/favicon.ico` if the favicon URL is not found. The downloaded favicon is saved locally for later use.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
log	private static final	Logger	Logger for logging messages and errors.
FAVICON_MIME_TYPE_MAP	final public	ImmutableMap<String, String>	Maps MIME types to file extensions for favicons.

   - Methods:

Method Signature	Access Modifier	Description
String downloadFaviconFromPage(String pageUrl, String directory, String fileName)	public	Downloads the favicon from a webpage by extracting the favicon URL or guessing its location.
String getFaviconUrl(String pageUrl, String fileName)	public	Constructs a possible favicon URL using the given webpage URL and common file names.
String downloadFavicon(String faviconUrl, String directory, String fileName)	public	Attempts to download a favicon from the provided URL and saves it locally.

3. Relationships:
    - Inheritance: NA
    - Associations
        - Dependency:
            1. ReaderHttpClient - Used to send HTTP requests and retrieve favicon files.
            2. FaviconExtractor - Used to parse HTML pages and extract favicon URLs from metadata.
            3. MimeTypeUtil - Determines the MIME type of downloaded files to ensure they are valid image files.
            4. LoggerFactory - Used for logging errors, warnings, and information messages.

▼ public class FaviconExtractor

1. Class Information
    - Class Name: FaviconExtractor
    - Purpose: The `FaviconExtractor` class parses an HTML page using SAX (Simple API for XML) and looks for the `<link>` tag that specifies a favicon. It extracts and constructs the favicon URL based on the provided page URL.
2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
log	private static final	Logger	Logger instance for logging errors and debugging information.
url	private final	URL	Stores the original page URL for reference.
favicon	private	String	Holds the extracted favicon URL if found.

    - Methods:

Method Signature	Access Modifier	Description
FaviconExtractor(String url)	public	Constructor that initializes the class with the provided page URL. It converts the given string into a URL object.
void readPage(InputStream is)	public	Parses the given HTML page input stream and looks for favicon links.
void startElement(String uri, String localName, String qName, Attributes attributes)	public (override)	SAX parser method that is triggered when a new HTML tag is encountered. It checks if the tag is a <link> element with a rel="icon" or rel="shortcut icon" attribute and extracts the favicon URL.
String getFavicon()	public	Returns the extracted favicon URL if found.

3. Relationships:
    - Inheritance:
        - Parent Class: DefaultHandler (from org.xml.sax.helpers.DefaultHandler)
        - Child Classes: NA
    - Associations
        - Dependency:
            1. StringUtils - Used to trim attribute values.
            2. Logger (SLF4J) - Used for logging errors and debugging.
        - Aggregation:
            1. `URL` – The `FaviconExtractor` holds a reference to a `URL` object representing the page URL.
        - Composition: The `favicon` URL is constructed dynamically using the provided page URL.

▼ public class FaviconUpdateRequestedEvent

1. Class Information
    - Class Name: FaviconUpdateRequestedEvent
    - Purpose: This class serves as an event object that encapsulates the details of a request to update a feed's favicon. It stores a reference to the Feed that requires an updated favicon. It plays a role in managing visual identifiers for feeds within the RSS reader application.
2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
feed	private	Feed	Stores the reference to the feed whose favicon needs updating.

    - Methods:

Method Signature	Access Modifier	Description
Feed getFeed()	public	Returns the Feed object associated with this event.
void setFeed(Feed feed)	public	Sets the Feed object associated with this event.
String toString()	public	Provides a string representation of the event, including the feedId.

3. Relationships:
    - Inheritance: NA

- Associations
  - Dependency: Feed - The `FaviconUpdateRequestedEvent` class has a direct dependency on the `Feed` class. This dependency exists because it holds a reference to a `Feed` object whose favicon update is requested.

▼ public class MimeTypeUtil

1. Class Information
   - Class Name: MimeTypeUtil
   - Purpose: The main purpose of this class is to guess the MIME type of a file using its header information (also known as a magic number). This is useful for ensuring proper file handling, security checks, and content validation.
2. Attributes and Methods
   - Attributes: NA
   - Methods:

Method Signature	Access Modifier	Description
`guessMimeType(File file) throws Exception`	`public static`	Attempts to determine the MIME type of a file based on its magic number (header bytes). Returns a MIME type constant from `MimeType` or `null` if the type cannot be determined.

3. Relationships:
   - Inheritance: NA
   - Associations
     - Dependency:
       - `MimeType`: Provides predefined MIME type constants.

▼ public class TextSanitizer

1. Class Information
   - Class Name: TextSanitizer
   - Purpose: Sanitizes text content by **removing HTML tags** using a predefined **HTML sanitization policy** and a **regular expression-based filter**.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
policy	private	PolicyFactory	Defines the HTML sanitization policy.

TAG_PATTERN	private	Pattern	Regular expression pattern for removing HTML tags.

- Methods:

Method Signature	Access Modifier	Description
String sanitize(String html)	public	Sanitizes the input string by applying an HTML policy and removing remaining tags.

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

▼ public class StreamUtil

1. Class Information
   - Class Name: StreamUtil
   - Purpose:
     - Detects whether an input stream is GZIP-compressed and returns an appropriate uncompressed stream.
     - Ensures that compressed RSS feeds or other GZIP-encoded content can be processed without errors.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description

   - Methods:

Method Signature	Access Modifier	Description
InputStream detectGzip(InputStream is) throws IOException	public static	Checks if an InputStream is GZIP-compressed and returns a decompressed stream if needed.

3. Relationships:

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

▼ public class Outline

1. Class Information
   - Class Name: Outline
   - Purpose: The `Outline` class is designed to model an OPML outline node, which may represent a feed or a folder of feeds. It includes attributes for text, title, type, XML and HTML URLs, and a list of child outlines to support hierarchical organization.
2. Attributes and Methods
   - Attributes:

Name	Access Modifier	Type	Description
text	private	String	The textual label of the outline node.
title	private	String	The title of the outline node.
type	private	String	The type of the outline (e.g., `rss`).
xmlUrl	private	String	The XML URL of the feed, if applicable.
htmlUrl	private	String	The HTML URL of the feed, if applicable.
outlineList	private	List	A list of child outlines (nested structure).

   - Methods:

Method Signature	Access Modifier	Description
`public Outline()`	public	Constructor initializing an empty `outlineList`.
`public String getText()`	public	Retrieves the text of the outline.
`public void setText(String text)`	public	Sets the text of the outline.
`public String getTitle()`	public	Retrieves the title of the outline.
`public void setTitle(String title)`	public	Sets the title of the outline.
`public String getType()`	public	Retrieves the type of the outline.
`public void setType(String type)`	public	Sets the type of the outline.
`public String getXmlUrl()`	public	Retrieves the XML URL of the outline.
`public void setXmlUrl(String xmlUrl)`	public	Sets the XML URL of the outline.
`public String getHtmlUrl()`	public	Retrieves the HTML URL of the outline.
`public void setHtmlUrl(String htmlUrl)`	public	Sets the HTML URL of the outline.

`public List<Outline> getOutlineList()`	public	Retrieves the list of child outlines.

3. Relationships:
    - Inheritance: NA
    - Associations
        - Composition: The `Outline` class contains a list of `Outline` objects (`outlineList`). This indicates a composition relationship where an `Outline` object consists of multiple child outlines, forming a tree-like structure.

▼ public class MimeType

1. Class Information
    - Class Name: MimeType
    - Purpose: The `MimeType` class defines a set of constant MIME type strings that are used in the application to identify and process different file types, particularly images and compressed files.
2. Attributes and Methods
    - Attributes:

Name	Access Modifier	Type	Description
`IMAGE_X_ICON`	public static final	String	Represents the MIME type for ICO (icon) images.
`IMAGE_PNG`	public static final	String	Represents the MIME type for PNG images.
`IMAGE_JPEG`	public static final	String	Represents the MIME type for JPEG images.
`IMAGE_GIF`	public static final	String	Represents the MIME type for GIF images.
`APPLICATION_ZIP`	public static final	String	Represents the MIME type for ZIP compressed files.

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

▼ public class GuidFixer

1. Class Information
   - Class Name: GuidFixer
   - Purpose: The class provides a utility function to generate a GUID for articles that lack one, ensuring each article has a unique identifier. It uses hashing techniques to create a unique string based on the article's content.
2. Attributes and Methods
   - Attributes: NA
   - Methods:

Method Signature	Access Modifier	Description
`fixGuid(Article article)`	`public static`	Ensures the article has a GUID by generating one if it is missing. Uses SHA-1 hashing based on the article's URL, title, or description.

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