Task 2 - Design Smells, Code Metrics

Design Smells

Note: Design Smells were taken from a mixture of Sonarqube, Designite and Human Observation

Insufficient Modularization

- `com.sismics.reader.core.service.FeedService` - The `synchronizeAllFeeds` method combines database operations, HTTP requests, and event notifications instead of separating concerns into smaller modules

  Metrics:

  - CBO - 12
  - RFC - 20
  - WMC - 13
  - LOC -268
  - LCAM - 0.417

- `com.sismics.reader.core.dao.file.rss.RssReader` - The class combines XML parsing, date format handling, feed validation, and article creation, making it overly complex.

  Metrics

  - CBO - 5
  - RFC - 13
  - SRFC - 2
  - WMC - 16
  - LOC - 362
  - LCAM - 0.656

- `com.sismics.reader.resource.SubscriptionResource` - The class acts as both an API client and a controller, lacking separation between request construction and response handling.

  - CBO - 14
  - RFC - 17
  - WMC - 23
  - LOC - 365
  - LCAM - 0.444

Multifaceted Abstraction

- `com.sismics.reader.core.service.FeedService` - The class has multiple responsibilities: managing feed synchronization, updating articles, handling user subscriptions, and interacting with event-driven mechanisms.
- `com.sismics.reader.core.dao.file.rss.RssReader` - The class is responsible for both XML parsing and feed/article creation, violating the Single Responsibility Principle
- `com.sismics.reader.resource.SubscriptionResource` - The class handles multiple responsibilities, including managing subscriptions, fetching feeds, marking articles as read, and handling request cancellation.

Unexploited Encapsulation

- `com.sismics.reader.core.dao.file.rss.RssReader` - The parsing logic for different feed formats (RSS, Atom, RDF) is handled within the same class instead of being encapsulated into separate format-specific parsers.

Missing Abstraction

- `com.sismics.reader.core.dao.file.rss.RssReader` - The `Element` enum attempts to represent RSS, RDF and Atom feeds, instead of separating them into classes.

Deficient Encapsulation

- `com.sismics.util.adblock.Subscription` - The class has all its attributes public

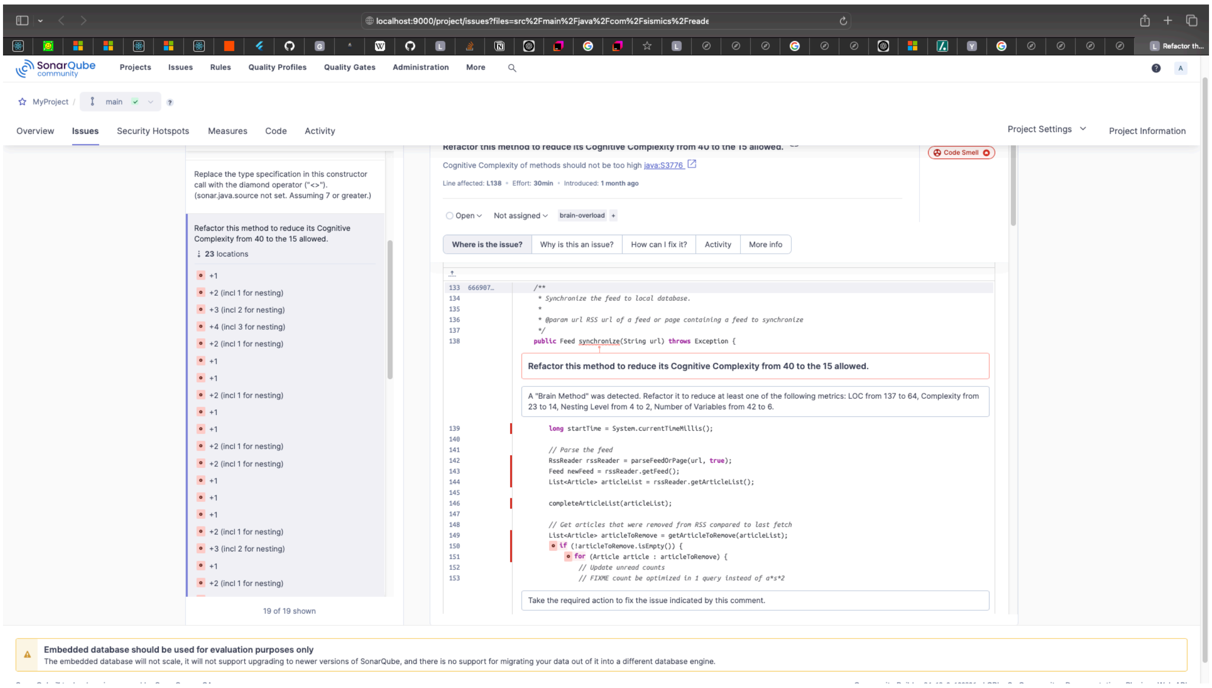
Package metrics

- `com.sismics.reader.core.service`
  - WMC -33

    - LOC - 360
- `com.sismics.reader.core.dao.file.rss`
    - WMC - 23
    - LOC - 479
- `com.sismics.reader.resource`
    - WMC - 73
    - LOC - 1333

Sonarqube References

1. Insufficient Modularisation in `com.sismics.reader.core.service.FeedService`



Replace the type specification in this constructor call with the diamond operator ("<>"). (sonar.java.source not set. Assuming 7 or greater.)

Refactor this method to reduce its Cognitive Complexity from 40 to the 15 allowed.

23 locations

- +1
- +2 (incl 1 for nesting)
- +3 (incl 2 for nesting)
- +4 (incl 3 for nesting)
- +2 (incl 1 for nesting)
- +1
- +1
- +2 (incl 1 for nesting)
- +1
- +1
- +2 (incl 1 for nesting)
- +2 (incl 1 for nesting)
- +1
- +1
- +1
- +2 (incl 1 for nesting)
- +3 (incl 2 for nesting)
- +1
- +2 (incl 1 for nesting)

19 of 19 shown

Refactor this method to reduce its Cognitive Complexity from 40 to the 15 allowed.

Cognitive Complexity of methods should not be too high java:S3776

Line affected: L138 • Effort: 30min • Introduced: 1 month ago

```
133 666907…    /**
134             * Synchronize the feed to local database.
135             *
136             * @param url RSS url of a feed or page containing a feed to synchronize
137             */
138            public Feed synchronize(String url) throws Exception {
```

Refactor this method to reduce its Cognitive Complexity from 40 to the 15 allowed.

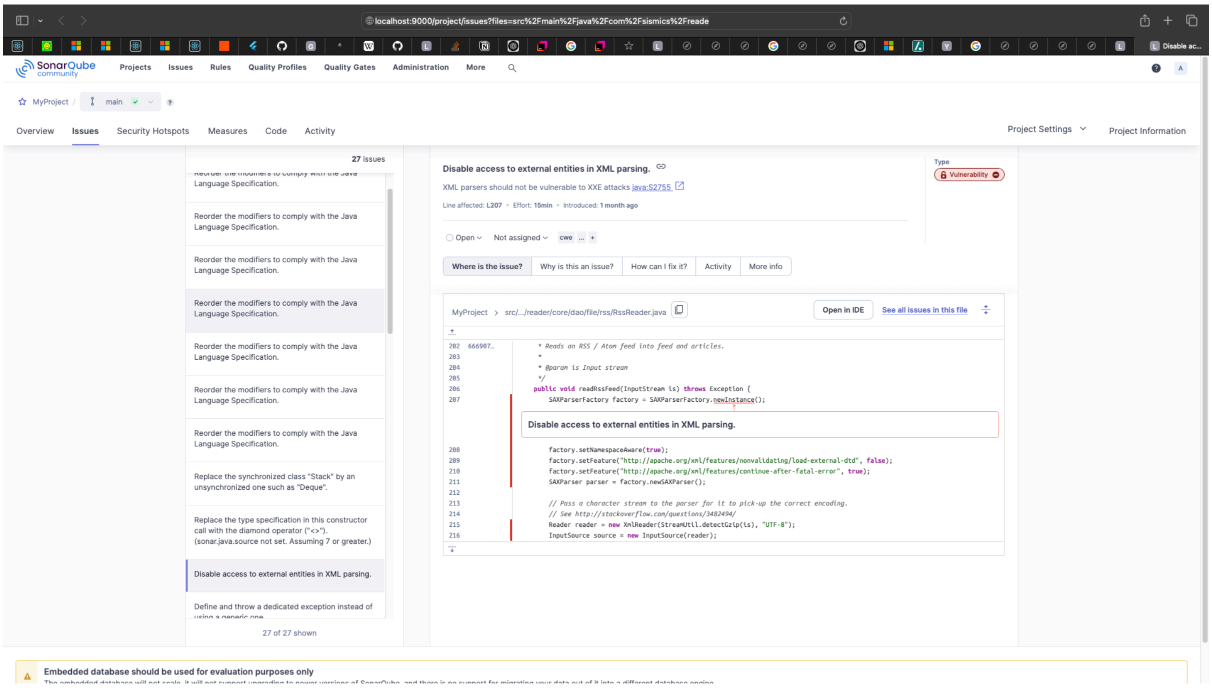
A "Brain Method" was detected. Refactor it to reduce at least one of the following metrics: LOC from 137 to 64, Complexity from 23 to 14, Nesting Level from 4 to 2, Number of Variables from 42 to 6.

```
139                long startTime = System.currentTimeMillis();
140
141                // Parse the feed
142                RssReader rssReader = parseFeedOrPage(url, true);
143                Feed newFeed = rssReader.getFeed();
144                List<Article> articleList = rssReader.getArticleList();
145
146                completeArticleList(articleList);
147
148                // Get articles that were removed from RSS compared to last fetch
149                List<Article> articleToRemove = getArticleToRemove(articleList);
150                if (!articleToRemove.isEmpty()) {
151                    for (Article article : articleToRemove) {
152                        // Update unread counts
153                        // FIXME count be optimized in 1 query instead of a*s*2
```

Take the required action to fix the issue indicated by this comment.

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

2. Multifaceted Abstraction in RssReader



27 issues

- Reorder the modifiers to comply with the Java Language Specification.
- Reorder the modifiers to comply with the Java Language Specification.
- Reorder the modifiers to comply with the Java Language Specification.
- Reorder the modifiers to comply with the Java Language Specification.
- Reorder the modifiers to comply with the Java Language Specification.
- Reorder the modifiers to comply with the Java Language Specification.
- Reorder the modifiers to comply with the Java Language Specification.
- Replace the synchronized class "Stack" by an unsynchronized one such as "Deque".
- Replace the type specification in this constructor call with the diamond operator ("<>"). (sonar.java.source not set. Assuming 7 or greater.)
- Disable access to external entities in XML parsing.
- Define and throw a dedicated exception instead of using a generic one.

27 of 27 shown

Disable access to external entities in XML parsing.

XML parsers should not be vulnerable to XXE attacks java:S2755

Line affected: L207 • Effort: 15min • Introduced: 1 month ago

MyProject > src/.../reader/core/dao/file/rss/RssReader.java

```
     * Reads an RSS / Atom feed into feed and articles.
     *
     * @param is Input stream
     */
    public void readRssFeed(InputStream is) throws Exception {
        SAXParserFactory factory = SAXParserFactory.newInstance();
```

Disable access to external entities in XML parsing.

```
        factory.setNamespaceAware(true);
        factory.setFeature("http://apache.org/xml/features/nonvalidating/load-external-dtd", false);
        factory.setFeature("http://apache.org/xml/features/continue-after-fatal-error", true);
        SAXParser parser = factory.newSAXParser();

        // Pass a character stream to the parser for it to pick-up the correct encoding.
        // See http://stackoverflow.com/questions/3482494/
        Reader reader = new XmlReader(StreamUtil.detectGzip(is), "UTF-8");
        InputSource source = new InputSource(reader);
```

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.