

Task 3c - SE Project 1

Team 8

2. Broken hierarchy observed in the code which creates unnecessary inheritance

LLM Generated Refactoring (ChatGPT o3 mini-high)

Summary of changes made during refactoring:

- Extracted `ResourceHelper` (a new helper class containing common authentication and permission methods).
- Removed `extends BaseResource` from all resource classes.
- Refactored method calls to use `ResourceHelper`.

Comparison to Manual Refactoring:

In comparison to the manual refactoring done, the method followed by OpenAI's ChatGPT follows a much more convoluted method that involves significantly more changes. Making changes directly to each `Resource` class runs the risk of affecting classes that may be dependent on their functionality and requires many more checks that could not be done with the context window of the LLM.

Manual refactoring differs from this approach mainly in the use of `BaseResource` as a superclass and adding a utility class `ResourceHelper`. This allowed the resource classes to remain unchanged while still benefiting from a cleaner, more modular design.

3. Broken Modularisation found in the following classes:

- `AtomArticleUrlGuesserStrategy.java`
- `AtomArticleCommentUrlGuesserStrategy.java`
- `AtomUrlGuesserStrategy.java`

LLM Generated Response (DeepSeek)

Summary of changes made during refactoring:

1. `BaseAtomUrlGuesserStrategy.java` :

- Introduced an abstract base class that encapsulates the common logic of iterating through `AtomLinks` and checking conditions defined by subclasses.
- This reduces code duplication and centralizes the iteration logic.

2. `AtomArticleCommentUrlGuesserStrategy.java` and `AtomArticleUrlGuesserStrategy.java` :

- These classes now extend the base class, defining their specific conditions using predicates.
- This eliminates duplicate loop structures and condition checks.

3. `AtomSiteUrlGuesserStrategy.java` and `AtomFeedUrlGuesserStrategy.java` :

- These new classes replace the original `AtomUrlGuesserStrategy` , each handling a single responsibility (guessing site URL and feed URL) by extending the base class.
- This adheres to the **Single Responsibility Principle** and improves modularity.

4. `RssReader.java` :

- Updated to use the new strategy classes, ensuring that the correct strategies are applied for site and feed URL guessing.
- This decouples `RssReader` from the specific strategy implementations, promoting flexibility.

By extracting common logic into a base class and utilizing predicates, the refactoring eliminates code duplication, improves maintainability, and enhances modularity. Each strategy now focuses on defining its specific conditions without worrying about the iteration logic, leading to cleaner and more scalable code.

Comparison to Manual Refactoring:

The approach by the LLM differs significantly from the manual refactoring done. It introduces changes to one existing class, splits the functionality of one existing class into two classes, and creates a new abstract class.

Manual refactoring combined the very similar functionality of the three similar classes into **one** class. This approach is simpler and achieves the same functionality.

4. Broken Modularisation is observed in the `StarredArticleImportedEvent` and `StarredReader` class.

LLM Generated Response (ChatGPT)

Summary of changes made during refactoring:

`StarredArticleImportedEvent.java`

- **Change:** Moved from `com.sismics.reader.core.dao.file.json` to `com.sismics.reader.core.event`.
- **Impact:** Separates domain events from file I/O logic without altering class functionality.

`StarredArticleImportedListener.java`

- **Change:** Also moved to the new `com.sismics.reader.core.event` package and updated documentation.
- **Impact:** Clearly isolates event-handling interfaces from JSON parsing, making them reusable across different modules.

`StarredReader.java`

Changes:

- **Updated Imports:** Now imports event classes from `com.sismics.reader.core.event`.
- **Helper Methods:** Added `createFeed(JsonNode origin)` and `createArticle(JsonNode itemNode)` to encapsulate JSON parsing logic.
- **Renamed Setter:** Changed from `setStarredArticleListener` to `setStarredArticleImportedListener` for clarity.

Impact:

- Simplifies the `read` method by isolating JSON extraction into dedicated methods.
- Decouples JSON parsing from event creation, improving modularity and maintainability.

Comparison to Manual Refactoring:

The response generated by the LLM differs from the approach taken during manual refactoring. The LLM opted to move event-related classes (`StarredArticleImportedEvent` and `StarredArticleImportedListener`) into a dedicated package.

The manual approach combined the functionality into **one class**, as `StarredArticleImportedEvent` and `StarredArticleImportedListener` were only used by `StarredReader`. This makes the code significantly **less complex and easier to understand**, while maintaining functionality.

A **pro** of the LLM-generated response is that it maintained **separation of concerns**, where one class only has **one responsibility**.