

```
1 package net.wiredclub.translation;
2
3 import org.eclipse.jgit.api.Git;
4 import org.eclipse.jgit.api.errors.GitAPIException;
5 import org.eclipse.jgit.lib.ObjectId;
6 import org.eclipse.jgit.lib.ObjectLoader;
7 import org.eclipse.jgit.lib.Repository;
8 import org.eclipse.jgit.revwalk.RevCommit;
9 import org.eclipse.jgit.treewalk.TreeWalk;
10 import org.eclipse.jgit.treewalk.filter.PathFilter;
11
12 import java.io.ByteArrayOutputStream;
13 import java.io.File;
14 import java.io.IOException;
15 import java.nio.charset.StandardCharsets;
16 import java.nio.file.Files;
17 import java.nio.file.Path;
18 import java.nio.file.Paths;
19 import java.nio.file.StandardOpenOption;
20 import java.util.Set;
21 import java.util.stream.Collectors;
22 import java.util.stream.Stream;
23
24 /**
25  * A helper class for file operations, e.g.
26  * it reads a text file, writes back the output, or scans the language
27  * directory for existing languages.
28  * For a GIT repo, it traverses back one commit and returns the file
29  * in its previous version.
30  */
31 public class FileHelper {
32
33     // private static final Logger LOG = LoggerFactory.getLogger(FileHelper.class);
34
35     public String readFile(String fileName) throws TranslationFileNotFoundException {
36         try {
37             return Files.readString(Paths.get(fileName), StandardCharsets.UTF_8);
```

```
38     } catch (IOException e) {
39         throw new TranslationFileNotFoundException(fileName);
40     }
41 }
42
43 public void writeFile(String fileName, String content) throws IOException {
44     Files.writeString(Paths.get(new File(fileName).toURI()),
45         content,
46         StandardCharsets.UTF_8,
47         StandardOpenOption.CREATE,
48         StandardOpenOption.TRUNCATE_EXISTING);
49 }
50
51 public String readPreviousFileFromHistory(String repositoryPath, String fileName, int revRange)
52     throws IOException, TranslationFileNotFoundException {
53     try (Git git = Git.open(new File(repositoryPath))) {
54         Repository repository = git.getRepository();
55
56         Iterable<RevCommit> revCommits = git.log().setSkip(revRange).call();
57         for (RevCommit revCommit : revCommits) {
58             try (TreeWalk treeWalk = new TreeWalk(repository)) {
59                 treeWalk.addTree(revCommit.getTree());
60                 treeWalk.setRecursive(true);
61                 treeWalk.setFilter(PathFilter.create(fileName));
62
63                 if (!treeWalk.next()) {
64                     throw new TranslationFileNotFoundException(fileName);
65                 }
66                 ObjectId objectId = treeWalk.getObjectId(0);
67
68                 ObjectLoader loader = repository.open(objectId);
69                 ByteArrayOutputStream stream = new ByteArrayOutputStream();
70                 loader.copyTo(stream);
71
72                 return stream.toString(StandardCharsets.UTF_8);
73             }
74         }
75     }
```

```
75     } catch (GitAPIException e) {
76         throw new RuntimeException(e);
77     }
78
79     throw new TranslationFileNotFoundException(fileName);
80 }
81
82 public Set<String> discoverLanguageDirectories(String dir) throws IOException {
83     try (Stream<Path> stream = Files.list(Paths.get(dir))) {
84         return stream.filter(Files::isDirectory)
85             .map(Path::getFileName)
86             .map(Path::toString)
87             .collect(Collectors.toSet());
88     }
89 }
90 }
91
```