```java
package net.wiredclub.translation;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.core.util.DefaultIndenter;
import com.fasterxml.jackson.core.util.DefaultPrettyPrinter;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.SerializationFeature;
import com.fasterxml.jackson.databind.node.ArrayNode;
import com.fasterxml.jackson.databind.node.ObjectNode;

import java.util.HashSet;
import java.util.Set;

/**
 * A helper class that provides a few utility methods, e.g.
 * it serializes and deserializes strings to json and back,
 * extracts responses from DeepL and returns the desired return type,
 * or creates patch objects.
 */
public class JsonHelper {

    // private static final Logger LOG = LoggerFactory.getLogger(JsonHelper.class);

    private final ObjectMapper objectMapper;
    private final DefaultPrettyPrinter printer;

    public JsonHelper() {
        this.objectMapper = new ObjectMapper().enable(SerializationFeature.INDENT_OUTPUT);

        printer = new DefaultPrettyPrinter();
        DefaultPrettyPrinter.Indenter indenter = new DefaultIndenter("\t", DefaultIndenter.SYS_LF);
        printer.indentObjectsWith(indenter);
        printer.indentArraysWith(indenter);
    }

    public JsonNode convertStringToJson(String json) throws TranslationJsonProcessingException {
```

```
38          try {
39              return objectMapper.readTree(json);
40          } catch (JsonProcessingException e) {
41              throw new TranslationJsonProcessingException(e.getMessage());
42          }
43      }
44
45      public String convertJsonToString(JsonNode jsonNode) throws TranslationJsonProcessingException {
46          try {
47              String content = objectMapper.writer(printer).writeValueAsString(jsonNode);
48              // remove spaces between '"' and ':' or '{', jacksons output is '"key" : "value"'.
49              // we want it '"key": "value"' and '"key": {'. This is a simple approach but could be also dangerous.
50              return content.replace("\" : \"", "\": \"").replace("\" : {", "\": {");
51          } catch (JsonProcessingException e) {
52              throw new TranslationJsonProcessingException(e.getMessage());
53          }
54      }
55
56      public ArrayNode createNewTranslationPatch() {
57          return objectMapper.createArrayNode();
58      }
59
60      public ObjectNode createPatchOperationAdd(String path, JsonNode value) {
61          ObjectNode objectNode = objectMapper.createObjectNode();
62
63          objectNode.put("op", "add");
64          objectNode.put("path", path);
65          objectNode.set("value", value);
66
67          return objectNode;
68      }
69
70      public ObjectNode createPatchOperationRemove(String path) {
71          ObjectNode objectNode = objectMapper.createObjectNode();
72
73          objectNode.put("op", "remove");
74          objectNode.put("path", path);
```

```java

        return objectNode;
    }

    public ObjectNode createPatchOperationReplace(String path, String value) {
        ObjectNode objectNode = objectMapper.createObjectNode();

        objectNode.put("op", "replace");
        objectNode.put("path", path);
        objectNode.put("value", value);

        return objectNode;
    }

    public Set<String> extractLanguages(JsonNode json) {
        Set<String> languages = new HashSet<>();
        if (json.isArray()) {
            ArrayNode languagesArray = (ArrayNode) json;
            for (int i = 0; i < languagesArray.size(); i++) {
                languages.add(languagesArray.get(i).get("language").asText());
            }
        }
        return languages;
    }

    public String extractTranslation(JsonNode json, String defaultText) {
        JsonNode translations = json.get("translations");
        String translation = defaultText;
        if (translations.isArray() && translations.size() == 1) {
            translation = translations.get(0).get("text").asText();
        }
        return translation;
    }
}

```