Elasticsearch Project

Documentation

Erasmus Students

09/01/2015

Zawartość

1. Introduction 3
2. Task assignment 4
3. Application architecture 4
   3.1. UML diagram 4
   3.2. Class description 5
   3.3. Tests 6
4. Conclusion and work evaluation 7

1. Introduction

Our project was about NoSQL-Databases - elasticsearch and openEHR. We must created application, which is able to store all openEHR entry types (observation, action, instruction, evaluation). For this goal we used JAVA API and Maven.

The main goals to do in our project:

- Create application in Java using JAVA API for elasticsearch.
- Create client application, which is able to connect with elasticsearch server/cluster.
- openEHR's types (archetypes) must be load from hard disc, from a specified location on disc (directory).
- Create mapping for each loaded archetype (according to particular archetype's definition).
- Prepare JSON documents with the structure of some selected archetype and store it.
- Perform test of our application.
- Create documentation for our project.

2. Task assignment

"Elasticsearch project" was created by two Erasmus students – Karol Kalaga and Mateusz Łysień. The project required us good distribution of tasks, because project was quite complicated. The task assignment was follows:

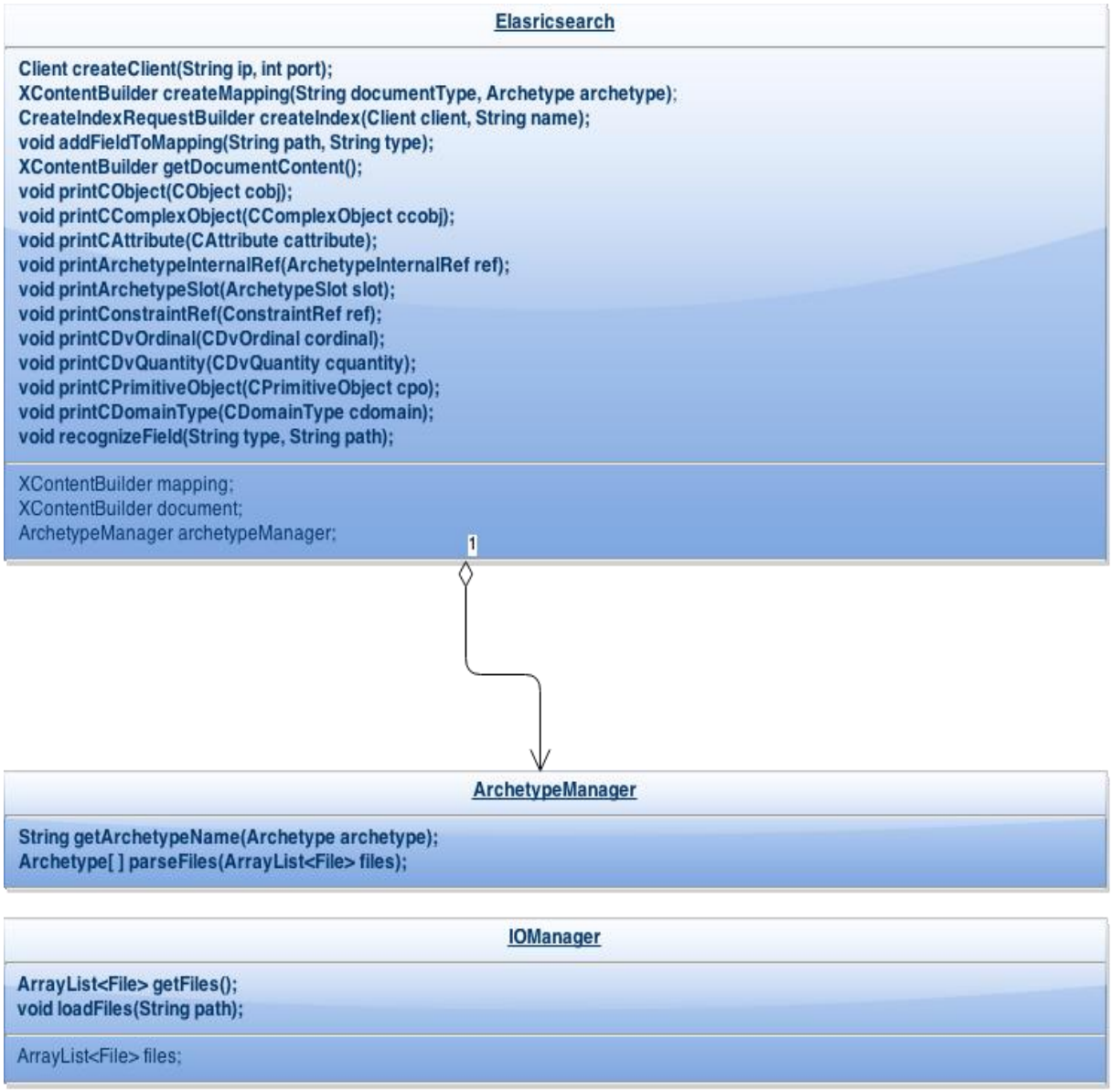
- **Karol Kalaga:**
    - Study Elasticsearch and OpenEHR documentation,
    - Create project documentation,
    - Help with designing application algorithms,
    - Create tests for application.
- **Mateusz Łysień:**
    - Study Elasticsearch and OpenEHR documentation,
    - Create project documentation,
    - Design application structure,
    - Design application algorithms.

3. Application architecture

3.1. UML diagram

Our project include 3 basic classes, of course we also using another classes, but these classes are from standard Java libs, we also use classes included in openEHR and elasticsearch libs:

- Elasticsearch
- ArchetypeManager
- IOManager



Screen 1 UML class diagram

3.2. Class description

Elasticsearch class:

This class include methods to connect with elasticsearch server/cluster. “Elasticsearch” class have also methods to create new index on the server, if current index is already exist, is deleting and creating new index. The main goal of this class is creating mapping and creating documents. All of this things are generating dynamic from archetype’s definition.

ArchetypeManager:

This class is responsible for getting every important things form archetypes, its names, numbers of fields, values of fields, etc..

IOManager:

Class responsible for loading archetypes from hard disc – from specific directory.

3.3. Tests

Our project has few test. We test some methods from class included in project. We tested method to creating client, we prepared some tests cases, with correct and incorrect data in this part of testing we also test time of creation client. We also tested class responsible for loading archetypes form hard disc, we prepared some tests cases, like incorrect directory. All of tests were successful.

```
/**
 * Test of LoadFiles method, of class IOManager.
 */
@Test
public void testLoadFiles() {
    IOManager instance = new IOManager();


    instance.loadFiles(fileName);
    ArrayList<File> result = instance.getFiles();
    assertNotNull(result.size());
}
```

```
/**
 * Test of LoadFiles method, of class IOManager.
 */
@Test
public void testLoadFiles() {
    IOManager instance = new IOManager();


    instance.loadFiles(fileName);
    ArrayList<File> result = instance.getFiles();
    assertNotNull(result.size());
}
```

4. Conclusion and work evaluation

Working with this project was opportunity to gain new experience. We learned how to work and how to using NoSQL-databases thanks Elasticsearch. Thanks this project we get new skills in designing application, like projecting algorithms, projecting tests and UML diagrams, also we created project documentation, which was good opportunity to learn how to created document, which will use by other people. The project was quite difficult for us, because we never before worked with Elasticsearch, during the entire project we must learning new things, and from time to time we must changing all code in our project. The very important new experience was worked with openEHR. We must studying code from “Java-libs” (because on website wasn’t any documentation) and trying a lot of methods in our code, thanks that we learned how to work with code, which come from new libraries. The project was new experience for us, we are thinking the project was done quite good, we are feeling quite satisfaction for this project.