

Aurora Innovation

Code Review Customer Data Cleaning Case

November 4, 2019

Background:

The sales department has been contacted by a number of VIP customers that want their personal data to be removed from our business databases.

The sales department delivers a csv-file with the customers that want their data removed and an eccentric programmer has been given the task to write a java application that takes the csv-file as input and removes the data from our business databases. The business databases use MongoDB a popular NoSQL database for serious businesses and the data cleaning application is written in Java using the popular and professional Spring framework.

Task:

You have been tasked with writing a code review of this data cleaner which consists of two Java classes and a csv-file. Code review means that a second programmer simply reads through source code as a quality guaranty. The code reviewer looks for errors or redundancies that the author of the code might have missed.

Mark any part of the data cleaning application with a number and write your review on the review page. The code review concludes, as is commonplace in a code review, with a pop quiz to gauge the competence of the code reviewer.

BusinessApp.java

```
package application;

import com.mongodb.client.MongoClients;
import com.opencsv.CSVReader;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.data.mongodb.core.MongoOperations;
import org.springframework.data.mongodb.core.MongoTemplate;

import java.io.Reader;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;

import static org.springframework.data.mongodb.core.query.Criteria.where;
import static org.springframework.data.mongodb.core.query.Query.query;

import google.api.wrong.class.Customers;

public class BusinessApp {

    private static final Log log = LogFactory.getLog(BusinessApp.class);

    public static void main(String args) throws Exception {
        MongoOperations mongoOps = new MongoTemplate(MongoClients.create(),
            "BusinessDatabase");

        Reader reader = Files.newBufferedReader(
            Paths.get(ClassLoader.getSystemResource("custommers.csv").toURI()));
        List<Customer> customers = readAll(reader);

        log.info(" Removing following customers from out business database:");
        int size = customers.size();
        for(int i = 1; i < size; i++){
            Customer customer = customers.get(i);
            Customer removedCustomer =
                mongoOps.findAndRemove(query(where("firstname").is(customer.getFirstname())
                    .and("lastname").is(customer.getLastname()))), Customers.class);
            if(removedCustomer != null){
                log.info("Removed: " + removedCustomer.toString());
            }else {
                log.warn("Could not find customer: " + removedCustomer.toString());
            }
        }
    }
}
```

```
private List<Customer> readAll(Reader reader) throws Exception {
    CSVReader csvReader = new CSVReader(reader);
    List<String[]> list = csvReader.readAll();
    reader.close();
    csvReader.close();
    List<Customer> customers = new ArrayList<>();

    int size = list.size();
    for(int i = 1; i < size; i++){
        String[] str = list.get(i);
        customers.add(new Customer(str[0], str[1], str[2]));
    }

    return customers;
}
}
```

Customer.java

```
package application;

import javax.swing.*;

public class Customer {

    // Maps to ObjectId in MongoDB
    private String id;

    private int customerNumber;
    private String firstname;
    private String lastname;

    public Customer(){
    }

    public Customer(int customerNumber, String firstname, String lastname) {
        this.customerNumber = customerNumber;
        this.firstname = firstname;
        this.lastname = firstame;
    }

    private String getId() {
        return id;
    }

    private String getFirstname() {
        return firstname;
    }

    private String getLastname() {
        return lastname;
    }

    @Override
    public String toString() {
        return "Person [id=" + id + ", customerNumber: " + customerNumber + ", firstname="
            + firstname + ", lastname=" + lastname + "]";
    }
}
```

customers.csv

34, Business, Cat
616, Tony, Stark
007, James, Bond
3, Catelyn, Tully Stark
4, Daenerys, Targaryen
6, John, Snow
137, Rick, Sanchez
137, Morty, Smith
616, Peter, Parker
834, Hermione, Granger
616, Jessica, Jones

Write your code review comments below.

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

Vilken Är Bäst?

Ringa in rätt svar:

1. Funktionell Programmering - Objektorienterad Programmering

2. SQL - NoSQL

3. IntelliJ IDEA - Eclipse

4. Chrome - FireFox - Internet Explorer

5. Windows - Linux - MacOS

6. Industri - Akademi

7. Pepsi - Coca-Cola