

Федеральное государственное автономное образовательное
учреждение высшего образования Национальный
исследовательский университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3

Информационные системы

Вариант: 367039

Студент: Абдуллин Игорь Эдуардович

Номер группы: Р3308

Преподаватель: Николаев Владимир Вячеславович

Санкт-Петербург, 2024

Текст задания

Доработать ИС из ЛР2 следующим образом:

- Реализовать сохранение загруженных на сервер файлов, используемых для импорта данных, в файловом хранилище MinIO (можно взять любое другое S3-совместимое хранилище). Поднять и настроить MinIO требуется самостоятельно. Загруженные файлы должны быть доступны для скачивания из таблицы с логом импорта.
- Сохранение загруженных файлов в файловом хранилище должно быть реализовано транзакционно по отношению к операциям, реализующим непосредственную вставку объектов в БД при импорте.
- Для реализации распределенной транзакции из пункта 2 разрешается использовать любые инструменты. Рекомендуются решать задачу при помощи собственной реализации двух фазного коммита.
- Необходимо на защите быть готовым продемонстрировать корректность реализованной распределенной транзакции в следующих условиях:
 - отказ файлового хранилища (БД продолжает работать)
 - отказ БД (файловое хранилище продолжает работать)
 - ошибка в бизнес-логике сервера (работают и БД, и файловое хранилище, однако в коде сервера вылетает RuntimeException между запросами в разные источники данных)
- Необходимо на защите быть готовым продемонстрировать корректность работы распределенной транзакции в условиях параллельных запросов от нескольких пользователей (реализованный в ЛР 2 сценарий для Apache JMeter, тестирующий функцию импорта, должен продолжать корректно обрабатывать).

Содержание отчёта:

1. Текст задания.
2. UML-диаграммы классов и пакетов разработанного приложения.
3. Исходный код системы или ссылка на репозиторий с исходным кодом.
4. Выводы по работе.

Ссылка на PR с изменениями: [merge-request](#)

Ссылка на отчеты прошлых работ: [Лаб-1](#), [Лаб-2](#)

Сценарии:

- 1) Отказ файлового хранилища (БД продолжает работать) - [Логи](#)

```
(0 rows)

is=# select * from movie;
is=# select * from movie;
is=# select * from import;
  id | creation_date | created_by | is_success | count | url
-----+-----+-----+-----+-----+-----
(0 rows)

is=# select * from import;
  id |      creation_date      | created_by | is_success | count | url
-----+-----+-----+-----+-----+-----
 108 | 2024-12-24 12:22:10.757183 |          51 | f          |      0 |
(1 row)

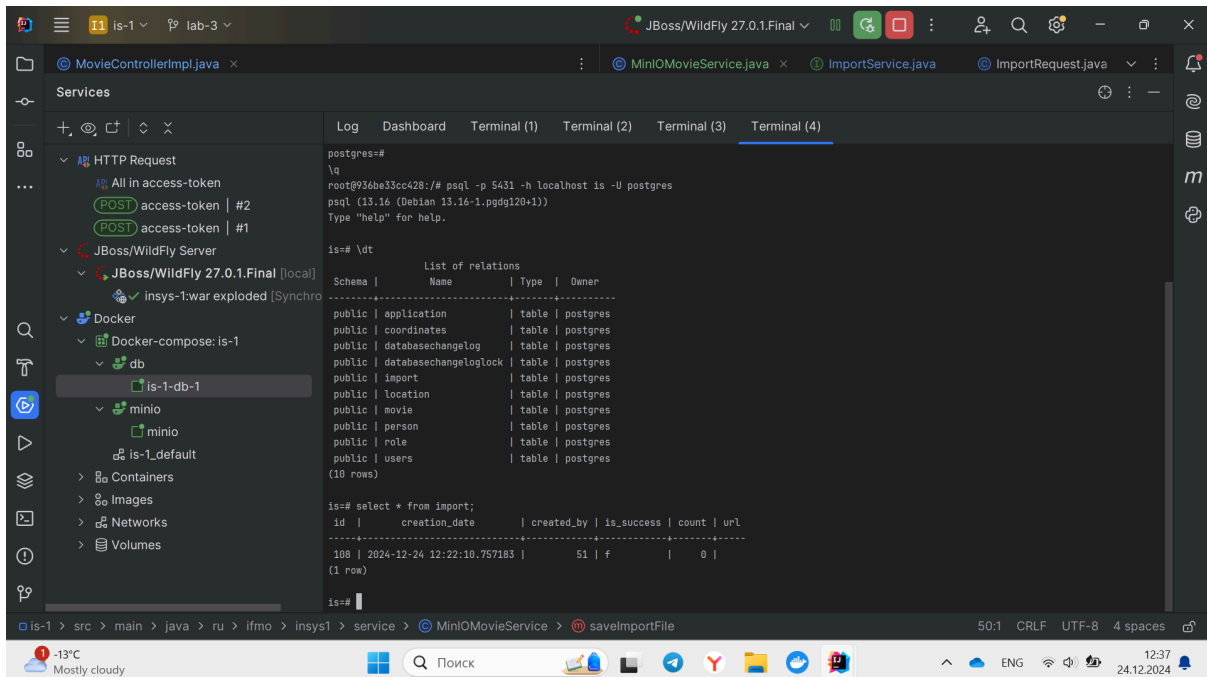
is=# select * from import;
  id |      creation_date      | created_by | is_success | count | url
-----+-----+-----+-----+-----+-----
 108 | 2024-12-24 12:22:10.757183 |          51 | f          |      0 |
(1 row)

is=# select * from movie;
is=# select * from movie;
is=# \x
Expanded display is on.
is=# select * from movie;
(0 rows)

is=#
```

Результат: Запись в бд о неуспешном импорте, файл в S3 не сохранен

2) Отказ БД (файловое хранилище продолжает работать) - [Логи](#)



The screenshot shows an IDE with a terminal window open. The terminal displays the following commands and output:

```
postgres=# \q
root@936be33cc428:/# psql -p 5431 -h localhost is -U postgres
psql (13.16 (Debian 13.16-1.pgdg120+1))
Type "help" for help.

is=# \dt
          List of relations
Schema | Name          | Type | Owner
-----|-----|-----|-----
public | application   | table | postgres
public | coordinates   | table | postgres
public | databasechangelog | table | postgres
public | databasechangelock | table | postgres
public | import         | table | postgres
public | location      | table | postgres
public | movie         | table | postgres
public | person        | table | postgres
public | role          | table | postgres
public | users         | table | postgres
(10 rows)

is=# select * from import;
 id | creation_date | created_by | is_success | count | url
-----|-----|-----|-----|-----|-----
108 | 2024-12-24 12:22:10.757183 | 51 | f | 0 |
(1 row)
```

Результат: в таблице import нет записи, файл в S3 не сохранен (запись в таблице из прошлого пункта)

3) Ошибка в бизнес-логике сервера (работают и БД, и файловое хранилище, однако в коде сервера вылетает RuntimeException между запросами в разные источники данных) - [Логи](#)

```
@Override
@Transactional
public Response upload(FileUploadForm form) {
    Pair<String, String> fileUrl = null;
    try {
        List<UploadMovie> parsedMovies =
movieRequestFileReader.read(form);
        fileUrl = minIOMovieService.saveImportFile(form);
        doThrow();
        Pair<List<Long>, Long> pair =
movieService.uploadAll(parsedMovies, fileUrl);
        log.info("Successfully uploaded movies");
        return Response.status(Response.Status.CREATED)
            .entity(pair.getLeft())
            .build();
    } catch (RuntimeException e) {
        log.error("Error while uploading movies, save failed
import", e);
        if (fileUrl != null) {
```

```

        minIOMovieService.deleteImportFile(fileUrl.getLeft());
    }
    importService.persistInNewTransaction(new
ImportRequest(false, 0, null));
    throw e;
}
}

void doThrow() {
    throw new RuntimeException();
}

```

The screenshot shows an IDE with several open files: `MovieControllerImpl.java`, `MovieServiceImpl.java`, and `MiniOMovieService.java`. The `Services` sidebar on the left lists various services including HTTP Request, JBoss/WildFly Server, Docker, and minio. The terminal window at the bottom shows a series of SQL queries and their results. The first query is a `psql` command to connect to a database. The subsequent queries are `select` statements that retrieve data from the `import` and `movie` tables. The results show a single row of data for the `import` table and a single row for the `movie` table.

```

root@936a33cc428:/# psql -p 5431 -h localhost -U postgres
psql (13.16 (Debian 13.16-1.pgdg120+1))
Type "help" for help.

is=# select * from import;
 id | creation_date | created_by | is_success | count | url
-----+-----+-----+-----+-----+-----
(0 rows)

is=# select * from movie;
is=# select * from import;
 id | creation_date | created_by | is_success | count | url
-----+-----+-----+-----+-----+-----
111 | 2024-12-24 14:28:42.581392 | 51 | f | 0 |
(1 row)

is=#

```

The screenshot shows a web browser with the MinIO Console open at `localhost:9001`. The console displays the `Object Browser` for the `movies-import-logs` bucket. The bucket was created on `Tue, Dec 24 2024 09:33:33 (GMT+3)` and is `PUBLIC`. The console shows a list of objects, including `movies-import-logs / movies158a5ef1-0b0c-4d46-8b23-c15fa298ed1a.json`. The console also includes a search bar, a `Create new path` button, and a `Upload` button. The sidebar on the left shows the `Object Store` navigation menu with options like `User`, `Object Browser`, `Access Keys`, `Documentation`, `Administrator`, `Buckets`, `Policies`, `Identity`, and `Monitoring`.

Результат: сохранена неуспешная запись о импорте, не сохранен файл в S3

Выводы по работе

1. Реализация транзакционного сохранения

В ходе работы было успешно реализовано транзакционное сохранение данных, включающее синхронную вставку записей в базу данных и загрузку файлов в файловое хранилище MinIO.

2. Демонстрация отказоустойчивости

Были протестированы различные сценарии сбоев, включая отказ БД, отказ MinIO, а также возникновение исключений в бизнес-логике. Во всех случаях система корректно откатывала частично выполненные операции, что подтверждает соблюдение принципов ACID.

3. Поддержка параллельных запросов

Разработанная система успешно обрабатывала конкурентные запросы от нескольких пользователей благодаря использованию JPA для управления транзакциями и механизмов синхронизации при записи файлов в MinIO. Тестирование с помощью Apache JMeter подтвердило отсутствие нарушений целостности данных и корректную обработку всех запросов.

4. Интеграция MinIO

Поднятое и настроенное файловое хранилище MinIO доказало свою эффективность в качестве S3-совместимого решения. Использование MinIO API упростило работу с объектами, включая загрузку, генерацию временных URL для скачивания и удаление файлов.

5. Опыт и навыки

Выполнение работы позволило углубить знания в области транзакционного менеджмента, работы с S3-совместимыми хранилищами, а также разработки RESTful приложений на основе JAX-RS. Этот опыт также подчеркнул важность тестирования системы в условиях реального использования, включая тестирование на отказоустойчивость и конкурентные нагрузки.