УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Рефакторинг баз данных и приложений»

Лабораторная работа №2

Выполнили:

Бордун Анастасия

Иванов Евгений

P34111

Преподаватель

Логинов И. П.

Санкт-Петербург, 2023 г.

Задание:

Провести рефакторинг приложения(курсовая работа по ИСБД). Описать планируемые изменения, описать для чего они необходимы, провести их.

Выполнение:

В качестве изменений в системе, её рефакторинга было выбрано добавление в систему транзакций. В качестве менеджера транзакций выступает Bitronix, который несложно интегрируется с спрингом. Изменения:

1. Настройка менеджера транзакций (Bitronix). За транзакциями в системе следит отдельный компонент – менеджер транзакций, который надо реализовать(добавить).

Изменения в коде:

Добавление необходимых зависимостей в gradle.build

```
    implementation group: 'org.springframework.boot', name: 'spring-boot-starter-jta-bitronix', version: '2.3.8.RELEASE'
    implementation group: 'javax.transaction', name: 'javax.transaction-api', version: '1.3'
```

Настройка менеджера транзакций Bitronix

```
public class BitronixTransaction {
    @Bean
    public bitronix.tm.Configuration transactionManagerServices() {
        bitronix.tm.Configuration configuration =
                TransactionManagerServices.getConfiguration();
        configuration.setServerId("1");
        return configuration;
    }

    @Bean(name = "bitronixTransactionManager")
    public BitronixTransactionManager
                transactionManager(bitronix.tm.Configuration _c)
    {
        var transactionManager =
                TransactionManagerServices.getTransactionManager();
        try {
            transactionManager.setTransactionTimeout(60);
        } catch (SystemException e) {
            throw new RuntimeException(e);
        }
        return transactionManager;
    }
}
```

2. Добавление самих транзакций с помощью аннотаций ***@Transactional***. Для этого перед каждым методом, требующим, например, запись в БД, ставится аннотация ***@Transactional*** с указанием хар-тик транзакции при работе с данным методом. Транзакции необходимы в нашей системе, так как у нас есть множество методов, которые изменяют данные, и, если в момент выполнения некоторых из них произойдёт сбой в системе, то данные могут оказаться некорректными в базе. Для этого мы добавляем транзакции. Чтобы не указывать на все методы, которые используют только чтение данных(объектов) из базы, дополнительный параметр транзакций *readonly = true*, мы добавили аннотацию *@Transactional(readonly = true)* перед всем сервисом. В этом случае перед методами, которые изменяют данные в базе, ставится снова аннотация *@Transactional*, которая имеет значение по умолчанию для настройки *readonly* как *false*.

Добавление транзакций на весь сервис(и так для каждого сервиса)

```
import org.springframework.transaction.annotation.Transactional;

@Service
@Transactional(readOnly = true)
public class BookingService {
```

Добавление транзакций по умолчанию(с сброшенным флагом readonly) на все методы, которые “пишут” в базу

```
@Transactional
public Booking addBooking(BookingDTO bookingDTO)  {
```

И так для каждого метода.

Код изменений:

Всю проделанную работу(все изменения в коде) с кратким описанием изменений можно найти в PR на гитхабе проекта: https://github.com/3ilib0ba/ITMO-RDBaA/pull/2