**Spring + Hibernate**

**Проблемы с Lazy=true**

Свойство lazy=true – автоматическое запонение вложенных сущностей хибернейтом, если к ним обращаются в рамках одной сессии

Если печатать в консоль вложенные сущности из базы, то Lazy работает только внутри бина.

Если вызываем через getBean из ApplicationContext, то на этот момент сессия уже закрыта и поэтому не получается подтянуть вложенные сущности

**Удаление сирот и каскады**

Если установлено более лояльное свойство чем Cascade=all-delete-orphan, например Cascade=all или Cascade=save-update, то сирот придется удалять вручную - удаляя зависимые объекты.

**Транзакции + ошибки transaction-synchronized и org.hibernate.LazyInitializationException**

Метод помеченный @Transactional будет транзакцией. Если вызывать внутри него методы с такой же аннотацией, то внутренняя транзакция станет частью внешней (если свойство propagation не изменено).

Получение сессии методом getCurrentSession() необходимо осуществлять только в методе аннотированном @Transactional, иначе ошибка ... **transaction-synchronized**

Если внутри транзакции была получена сессия методом getCurrentSession(), то закроется она с завершением последнего метода с @Transactional. Таким образом, если в каком-то методе без @Transactional инициировать заполнение вложенных сущностей (только если lazy=true), то возникает **org.hibernate.LazyInitializationException,** т.к. сессия уже закрылась.

**Подключение к БД на виртуалке (с помощью туннеля ssh)**

1. Настраиваем сеть виртуалки: добавляем адаптер №2, прописываем ему ip
2. Этот же ip прописываем в файле настроек ssh-сервера в убунте, которая на виртуалке
3. В сети виртуалке(адаптер №1) делаем проброс порта 22 (или другого если изменили дефолтный)
4. Скачиваем putty, добавляем ip который указан в виртуалке и добавляем ssh-туннель для БД
5. Теперь можно подключаться к БД из кода, как будто она локальная

Примечание: можно и не создавать туннель для БД, тогда необходимо его создавать в коде или подключаться напрямую по TCP/IP.

Скриншоты процесса настроек в папке «настройка virtualbox как удаленного сервера»

**Подключение к БД на виртуалке (напрямую через tcp/ip)**

1. Настраиваем сеть виртуалки. Можно указать «сетевой мост» - тогда виртуалка будет доступна всем по локальной сети и всему инету. Либо можно указать «NAT» - тогда только на локальной машине (этот вариант требует каких-то дополнительных настроек, я не пробовал). В данном случае ip присваивается автоматически, посмотреть его можно на убунте командой ifconfig.

Есть еще вариант через host-only adapter (айпишник виртуалки будет доступен только на нашей локальной машине). Для этого в настройках виртуалки указываем ip 192.168.56.1, dhcp отключаем. В настройках убунту настраиваем айпишник 192.168.56.200 (или другой, но больше 192.168.56.1)

Примечание. В убунту названия сетевых портов (enp0s3, enp0s8 …) привязаны к адаптерам виртуалки, т.е. Адаптер1- enp0s3, Адаптер2- enp0s8 и т.д.

1. Настриваем БД так, чтобы вместо localhost можно было подключаться либо с определенного ip, либо с любого ip (для этого обычно указывается 0.0.0.0 или ‘\*’).
2. Теперь можно подключаться удаленно к БД так:

ip:port

где ip – это адрес виртуалки,

port – это порт БД.

Скриншоты процесса настроек в папке «настройка virtualbox как удаленного сервера»

**Настройка mysql на Ubuntu для подлкючения с удаленной машины (или с локальной к виртуалке)**

Изменяем адреса которые будет слушать mysql в файле в секции [mysqlid]:

*sudo nano /etc/mysql/my.cnf*

если там нет такой секции, то в файле:

/etc/mysql/mysql.conf.d/mysqld.cnf

**Вместо**:   
bind address=127.0.0.1  
**Вписываем**:   
bind address=0.0.0.0

И в файле /etc/mysql/mysql.conf.d/mysqld.cnf добавляем строку:

skip-grant-tables

Перезагружаем: *service mysql restart*

Чтобы подключиться из вне нам необходим ip виртаулки (можно посмотреть командой ifconfig), порт mysql (по умолчанию 3306) и созданные имя пользователя+пароль

**Настройка postgres на Ubuntu для подлкючения с удаленной машины (или с локальной к виртуалке)**

Раскоментить и заменить в файле (9.5 – это версия постгреса):

*sudo nano /etc/postgresql/9.5/main/postgresql.conf*

строку:

*listen\_addresses = '\*'*

Изменяем права для пользователей в файле:

*sudo nano /etc/postgresql/9.5/main/ pg\_hba.conf*

**Вместо**:   
# IPv4 local connections:  
host    all             all             127.0.0.1         md5  
**Вписываем**:   
# IPv4 local connections:  
host    all             all             0.0.0.0/0         md5

Создаем пользователя:

*su postgres*

*createuser -s -r -d -P admin*

*4 # это пароль*

*Exit*

Перезагружаем: *service postgresql restart*

Чтобы подключиться из вне нам необходим ip виртаулки (можно посмотреть командой ifconfig), порт постгреса (по умолчанию 5432) и созданные имя пользователя+пароль (в примере это – admin, 4)

**Проверка в какой кодировке таблицы в mysql**

SELECT

`tables`.`TABLE\_NAME`,

`collations`.`character\_set\_name`

FROM

`information\_schema`.`TABLES` AS `tables`,

`information\_schema`.`COLLATION\_CHARACTER\_SET\_APPLICABILITY` AS `collations`

WHERE

`tables`.`table\_schema` = DATABASE()

AND `collations`.`collation\_name` = `tables`.`table\_collation`;