JPA и Hibernate

№ урока: 4 **Курс:** JDBC & Hibernate

Средства обучения: JDK, Intelij Idea

Обзор, цель и назначение урока

Рассмотрение разницы между «Чистым» JPA и Hibernate. Выполнение CRUD на JPA. Добавление и вставка с использованием Hibernate.

Изучив материал данного занятия, учащийся сможет:

- Поймет разницу, где лучше использовать JPA, а где Hibernate.
- Научиться подключать напрямую библиотеки JPA и Hibernate.
- Научиться выполнять CRUD на JPA.
- Разберется, как происходит выборка данных в JPA.
- Выполнять вставку и выборку используя Hibernate.

Содержание урока

1. Рассмотрение JPA и его настроек.
2. Рассмотрение основных объектов для работы с JPA и Hibernate.
3. Рассмотрение понятия Mapping.
4. Подключение jar-библиотек в обычный java-проект
5. Выполнение CRUD-операций используя JPA
6. Рассмотрение способов выборки в JPA.
7. Выполнение выборки и вставки используя Hibernate

Резюме

- **JPA (Java Persistance API) –** спецификация, документ, в котором описаны правила и API для реализации принципов ORM для Java (аннотации, настройки, подход).
- **Hibernate** – фреймворк между БД и приложением, в котором не нужно создавать прямые SQL запросы через JDBC.
  Может применятся в любом типе приложения (web/desktop/Spring и т.д.).
  Основная цель – работа с таблицами БД как с объектами.
  Легче создавать правильный абстрактный уровень для приложения (применять принципы ООП).
  На низком уровне Hibernate выполняет запросы через JDBC.
- **persistence.xml** – для любой реализации JPA (в том числе Hibernate) – ограничен спецификацией.
- **hibernate.cfg.xml** – для Hibernate –может содержать дополнительные настройки, которые будут работать только в Hibernate реализации.
- Каждая СУБД имеет свой dialect-дополнения и отличия для SQL запросов (на основе ANSI SQL). Hibernate умеет определять автоматически нужный диалект.
- **SessionFactory** – создается только 1 раз при запуске приложения и это настройка и работа с сессиями. При создании SessionFactory – считываются настройки hibernate.cfg.xml
- **Session** — Часто используется термин «persistence context». Для манипуляции с персистентными объектами. Является расширенным интерфейсом для EntityManager (EntityManager + свое API) (аналогично для SessionFactory). Если в примерах документации или статьях встречаете переменную entityManager, вместо нее можно подставлять объект Session (только для Hibernate реализации JPA).
- **Mapping** – процесс описания связей между POJO и таблицей БД для создания persistent object («persist» -сохранять). **Persistent Object (персистентный объект)** – **POJO**, который представляет таблицу из БД. Для маппинга можно использовать 2 способа:
  - С помощью XML файла (Java отдельно –mapping конфиг отдельно).

CyberBionic systematics

CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str.,5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

    - С помощью аннотаций в POJO (все вместе в одном классе).
    - У каждого способа есть свои плюсы и минусы
- **@Entity** – главная аннотация, которая делает Entity из обычного POJO класса
- **@Table** – не обязательная аннотация, если только не нужно уточнить параметры mapping-таблицы.
- **@Id** – помечает первичный ключ.
- **@GeneratedValue** – генерация ID при добавлении новой записи
- **@Column** – не обязательная аннотация, если только не нужно уточнить параметры столбца
- **@Transient** – если нужно исключить поле.
- **Criteria API** – Использование объектов вместо написания SQL(ООП ориентированный вариант).Более динамичный способ создания запросов (условия, параметры и пр.).Многие ошибки предотвращаются на этапе компиляции (type safe).Не совсем интуитивный API. Из-за сложного API –возможна громоздкость кода.
- **HQL (Hibernate Query Language)** - работает с персистентными объектами, а не таблицами БД. Легче для чтения. В JPA используется JPQL -Java Persistence Query Language (HQL – более расширенный вариант JPQL). В конечном итоге преобразуется в нужный SQL формат, согласно диалекту СУБД и похож на «обычный» SQL
- **Native SQL** – «Обычные» SQL запросы, как при JDBC. Самостоятельно нужно соблюдать диалект СУБД. Не универсальный, как HQL. Работает с таблицами БД, а не персистентными объектами
- **Работа с данными** – создать нужный запрос любым способом (Criteria, HQL, Native SQL). Если запрос типа select -преобразовать к нужному типу:
    - Коллекция объектов.
    - Уникальный объект.
    - Примитивное значение.
- **Коллекции** – самый частый вариант типа данных при запросах в БД. Тип коллекции при объявлении должен быть типом интерфейса (List, Set…). Обязательно используем типизированные коллекции (List<Author>)
- **Одиночный объект** - получение по ID (готовые методы в объекте Session). Получение по любым полям (создавать SQL запросы вручную).

Закрепление материала

- Что такое JPA?
- Какую API (JPA и Hibernate) где лучше использовать?
- Какие основные объекты для работы с JPA и Hibernate?
- Что такое Mapping?
- Какие основные способы выборки в JPA?

Дополнительное задание

Задание
Создать новую базу данных с помощью MySQLWorkbench. Создать обычный java-проект и подключить к нему библиотеки JPA и Hibernate. Создать файл с настройками persistence.xml в папке META-INF. И создать класс Animal(int age, String name, boolean tail) с методами get и set, как сущность к нашей таблице. И с помощью jpa сделать crud в классе AnimalHelper.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные понятия, рассмотренные на уроке.

Задание 2
Изучить раздел 23. Configurations с возможными настройками Hibernate
http://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html#configurations

CyberBionic systematics
CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str.,5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Задание 3
Пройти раздел 3. Bootstrap из документации Hibernate. Прочитать https://vaughnvernon.co/?page_id=31
Что такое Service (из другой документации –«Hibernate Integrations Guide») http://docs.jboss.org/hibernate/orm/5.1/integrationsGuide/html_single//

* Поначалу можно не изучать досконально все классы для инициализации сессии

Задание 4
Пройти Chapter 2 Entities из спецификации (разделы 2.1 и 2.2)
Прочитать раздел 2.5. Entity types, главы с 2.5.1 по 2.5.

Задание 5
Начать изучать разделы из документации:
15 HQL и JPQL
16 Criteria
17 Native SQL Queries

*Не нужно сразу пытаться за один раз пройти все разделы (многое выветрится) –лучше применять по необходимости на практике, когда есть конкретная задача.

Задание 6
Создать Gradle-проект и настроить его под Hibernate. Взять пример ex_003_hibernate_get_and_insert. Получить объект Book и коллекцию объектов. Получить конкретный Book по id. Добавить Новый Book. И это все реализовать в классе BookHelper.

Рекомендуемые ресурсы

Поддержка диалектов для различных СУБД:
http://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html#database-dialect

23. Configurations с возможными настройками Hibernate
http://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html#configurations

Диалект:
http://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html#database-dialect

Основы UML
https://vaughnvernon.co/?page_id=31

Что такое Service
http://docs.jboss.org/hibernate/orm/5.1/integrationsGuide/html_single/

CyberBionic systematics
CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str.,5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com