

JDBC & Hibernate

JPA и Hibernate

JDBC & Hibernate

Автор курса



Бондаренко Андрей

JDBC & Hibernate

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

JDBC & Hibernate

Тема

JPA и Hibernate

What is JPA

Что такое JPA?

JPA (Java Persistence API) – спецификация, документ, в котором описаны правила и API для реализации принципов ORM для Java (аннотации, настройки, подход).

JPA имеет множество реализаций:

- OpenJPA (Apache)
- Hibernate
- TopLink (Oracle)
- EclipseLink

Часто используется запись JPA API.

Configurations

XML настройки

- При запуске Hibernate считывает настройки из специальных файлов конфигурации XML (либо из Java конфигурации)
- Можно использовать любой из 2-х файлов:
- **hibernate.cfg.xml** –для Hibernate –может содержать дополнительные настройки, которые будут работать только в Hibernate реализации.
- **persistence.xml** –для любой реализации JPA (в том числе Hibernate) –ограничен спецификацией

Minimum Configurations

Минимальные настройки

- Для соединения с БД – десктоп проекты:
- `<property name="connection.url">`
- `<property name="connection.username">`
- `<property name="connection.password">`
- Либо, для веб приложений:
- `<property name="hibernate.connection.datasource">`


SQL-Dialect

Диалект

- Каждая СУБД имеет свой dialect-дополнения и отличия для SQL запросов (на основе ANSI SQL)
- Hibernate умеет определять автоматически нужный диалект.

API

Какое API использовать?



«Чистый»
JPA API

Если нужна
универсальность

Hibernate
API

Если не планируете
переходить на другую
JPA реализацию

Session Factory

SessionFactory (в JPA – EntityManagerFactory)

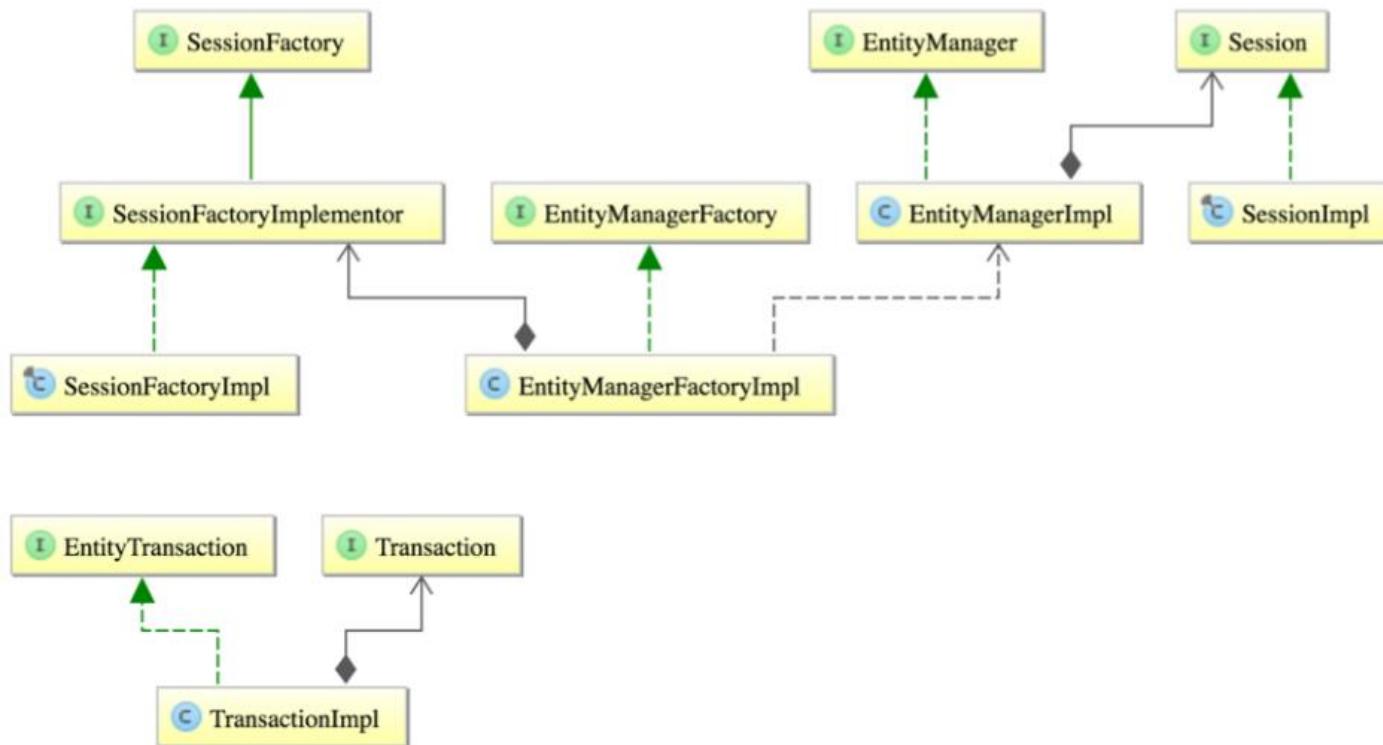
- Создается только 1 раз при запуске приложения
- Настройка и работа с сессиями
- При создании SessionFactory – считываются настройки hibernate.cfg.xml

Session (в JPA – EntityManager)

- 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).

API

Связь между API



- Hibernate создал «надстройку» над реализацией JPA и добавил свои классы и интерфейсы.
- Как указано в документации Hibernate – желательно уметь «читать» диаграммы зависимостей, чтобы понимать структуру классов.

Mapping

Mapping

- 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 конфиготдельно)
- С помощью аннотаций в POJO (все вместе в одном классе).
- У каждого способа есть свои плюсы и минусы

JPA Annotations

Аннотации JPA

- @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

Notations

Заметки

- JPA API -пакеты `javax.persistence.*`
- Hibernate API -пакеты `org.hibernate.*`
- Entity класс должен быть не `final`, иметь конструктор без параметров (подробнее в разделе «2.1 The Entity Class»)

API

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)

- Если вам удобней работать с SQL запросами, чем с объектами
- 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 ☐ 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

Какой способ использовать?

- Все способы обладают практически одинаковыми возможностями
- Разница – в синтаксисе
- В последних версиях Hibernate разницы в производительности минимальны

SELECTION

Работа с данными

1. Создать нужный запрос любым способом (Criteria, HQL, Native SQL)
2. Если запрос типа select -преобразовать к нужному типу:
 - Коллекция объектов.
 - Уникальный объект.
 - Примитивное значение.

Collections

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>)

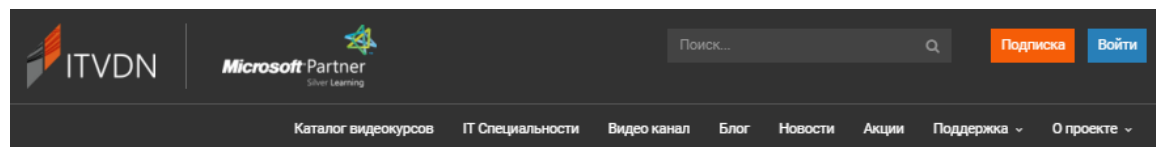
Single Object

Одиночный объект

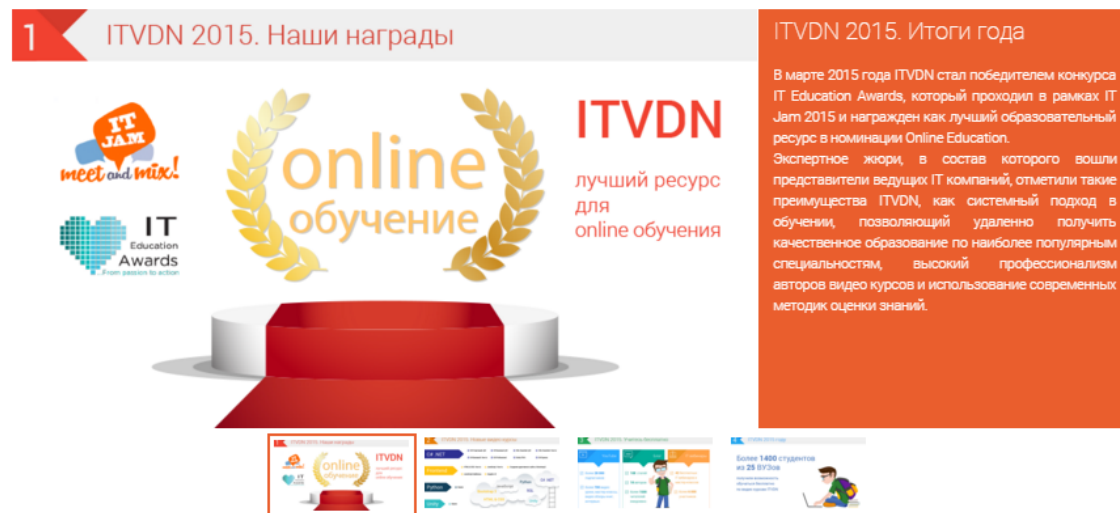
- Получение по ID (готовые методы в объекте Session).
- Получение по любым полям (создавать SQL запросы вручную).

Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.



Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics

Новые видео

Исключения	0
Итераторы и генераторы	0

Популярные видео курсы

Видео курс С# Стартовый (для начинающих)	9 уроков (16 ч. 3 мин.)
Видео курс по шаблонам проектирования	29 уроков (16 ч. 7 мин.)

Теги

.NET Developer
Frontend Developer



Проверка знаний

TestProvider.com



The screenshot shows the TestProvider.com website. At the top, there is a navigation bar with the TestProvider logo, a tagline "Мы помогаем людям оценить себя", and links for "Регистрация" and "Войти". Below this is a secondary navigation bar with links: "Главная", "Каталог", "Сертификация Microsoft", "Поддержка", and "О нас". The main content area features a large blue banner with the text "Тестирование" and "Языки программирования и информационные технологии". Below this is the Microsoft logo and a list of technologies: C#, ASP.NET, MVC, JavaScript, Patterns, Of, Design, SQL, Architecture, Guide, WCF, HTML&CSS, XML, SEO, WPF, HTML5&CSS3, JQuery, XNA, SharePoint, GUI for Android, Windows, Azure, Platform, Microsoft Patterns&Practices, TFS, SCRUM, ReSharper, TDD, WWF, LINQ, Entity Framework, Windows Forms, Refactoring, Microsoft Expression Blend 4, Windows Phone 8, Windows 8, AppStore, Visual Studio, Tips&Tricks, MSF, MEF, SilverLight, AJAX, MEF, Service Oriented Architecture. A "Пройти тест" button is at the bottom of the banner. To the right of the banner is a photo of a woman smiling while using a laptop. Below the banner is a section titled "Наши партнеры" with logos for Microsoft Partner, CyberBionic, ITVDN, PROMETRIC TEST CENTER, PEARSON VUE Authorized Test Center, Windows Azure Cloud Partner, and EBA. At the bottom, there is a blue bar with the text "Дополнительные ресурсы:" and links for "Очное обучение", "On-line обучение", and "Видео обучение".

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A

Информационный видеосервис для разработчиков программного обеспечения

