



# Introduction to Hibernate



Presented by Hosein Zare  
Twitter: @zare88r






- در نرم افزار دو دنیای متفاوت وجود دارد:
  - یکی در دنیای شیء گرایی که هیچ چیز به جز اشیا در آن وجود ندارد
  - و دیگری دنیای پایگاه های داده ی رابطه ای که در آن دیتا حکمفرماست
- ذخیره سازی اشیاء یکی از مهمترین مباحث برنامه های نوشته شده با جاوا میباشد.



# ORM

- از طرفی در پایگاه داده باید دیتا ها را در قالب ستون و سطر همراه با روابط خاص خود ذخیره کرد
- تبدیل اشیاء در جاوا به داده ها در دیتابیس همیشه با چالش هایی روبرو بوده است و این امر را به وظیفه ای پیچیده مبدل ساخته است.
- به این فرآیند Object-Relational Mapping، میگویند



# چگونه میتوانیم یک شیء را در دیتابیس ذخیره کنیم؟

- دنیای اشیا با دنیای داده های رابطه ای تفاوت هایی دارد:

1. **ارث بری** یکی از اصول اولیه ی برنامه نویسی شی گراست ولی این مفهوم در پایگاه داده معنایی ندارد

2. **در رابطه ی** بین اشیا ارتباط هایی چون یک به چند ، یک به یک و چند به چند میبینیم ، در پایگاه داده تمامی این رابطه ها پشتیبانی نمیشود

3. اشیا قابلیت های **یکتایی و تساوی** را در خود به طور ضمنی دارند در حالی که رکوردهای اطلاعاتی در پایگاه داده توسط مقادیر ستونهایش متمایز میشود



- جاوا ابزاری جهت ارتباط با پایگاه داده به نام JDBC دارد.
- این API برای پروژه های کوچک مناسبتر میباشد چرا که در مقیاس بالتر که دامنه ی مدل برنامه بزرگ میباشد پیچیدگی زیادی اضافه میکند
- Hibernate چارچوبی است که به شما قابلیت ذخیره سازی اشیا را می دهد
- به هر حال Hibernate برای اینکه نحوه ی تعامل با پایگاه داده را بداند نیاز به یک سری تنظیمات و نگاشت ها دارد



مراحل راه اندازی Hibernate

- 1) پیکربندی ارتباط با پایگاه داده
- 2) ساخت تعاریف نگاشت
- 3) ذخیره سازی آبجکت ها

# پیکربندی ارتباط با پایگاه داده

```
<hibernate-configuration>
  <session-factory>
    <property name="connection.url">
      jdbc:mysql://localhost:3307/JH
    </property>
    <property name="connection.driver_class">
      com.mysql.jdbc.Driver
    </property>
    <property name="connection.username">
      mkonda
    </property>
    <property name="connection.password">
      password
    </property>
    <property name="dialect">
      org.hibernate.dialect.MySQL5Dialect
    </property>
    <mapping resource="Movie.hbm.xml" />
  </session-factory>
</hibernate-configuration>
```

اطلاعات پیکربندی (معمولاً) در یک  
فایلی به نام hibernate.cft.xml  
نگهداری میشود همچنین میتوان از یم  
فایل hibernate.properties هم  
استفاده کرد



# ساخت تعاریف نگاشت

```
@Table(name = "available_languages")
@Entity
public class AvailableLanguages {
    @Id
    private String language;
    @Column(name = "default_flag")
    private Character defaultFlag;
    @Column(name = "created_by")
    private String createdBy;
    @Column(name = "last_updated_by")
    private String lastUpdatedBy;
    @Column(name = "creation_date")
    private Date creationDate;
    @Column(name = "last_update_date")
    private Date lastUpdateDate;
    @Column(name = "object_version_id")
    private Long objectVersionId;
    //getter and setters ...
}
```





# اولین کد

```
private static final SessionFactory ourSessionFactory;

static {
    try {
        ourSessionFactory = new Configuration().
            configure("hibernate.cfg.xml").
            addAnnotatedClass(AvailableLanguages.class).
            buildSessionFactory();
    } catch (Exception ex) {
        throw new ExceptionInInitializerError(ex);
    }
}

public static void main(final String[] args) {
    try (Session session = ourSessionFactory.openSession()) {
        session.createQuery("from AvailableLanguages").list();
    }
}
```



# ذخیره سازی اشیا

```
public class AvailableLanguagesDAO {  
  
    public void persistAvLang(AvailableLanguages availableLanguages) {  
        try (Session session = ORMConfiguration.openSession()) {  
            session.beginTransaction();  
            session.save(availableLanguages);  
            session.getTransaction().commit();  
        }  
    }  
}
```



# واکشی یک شیء توسط id

```
public AvailableLanguages findAvLangById(String id) {  
    try (Session session = ORMConfiguration.openSession()) {  
        AvailableLanguages avLang = session.load(AvailableLanguages.class, id);  
        return avLang;  
    }  
}
```



# ها Query

```
public List<AvailableLanguages> findAllAvLang() {  
    try (Session session = ORMConfiguration.openSession()) {  
        List<AvailableLanguages> avLangList  
            = session.createQuery("from AvailableLanguages").list();  
        return avLangList;  
    }  
}
```



# تنظیمات Hibernate در پروژه

تنظیم کردن Hibernate ساده است کافی است وابستگی آن را در Maven تعریف کنید به این صورت

```
<dependency>  
  <groupId>org.hibernate</groupId>  
  <artifactId>hibernate-core</artifactId>  
  <version>5.2.12.Final</version>  
</dependency>  
<dependency>  
  <groupId>com.oracle</groupId>  
  <artifactId>ojdbc6</artifactId>  
  <version>11.2.0.2.0</version>  
</dependency>
```



# تنظیمات Hibernate در پروژه

Hibernate قابلیت‌هایی دارد که می‌تواند جداول شما را از روی نگاشت شما طراحی کند کافی است در فایل پیکربندی اضافه کنید

```
<property name="hbm2ddl.auto">create</property>  
<property name="hbm2ddl.auto">create-drop</property>  
<property name="hbm2ddl.auto">update</property>
```