

Custom Tip Tanımları



ColumnTransformer

- **@ColumnTransformer** anotasyonu ile Hibernate'in entity property'sini **okurken** ve **yazarken** custom bir **SQL ifadesi çalıştırması** sağlanabilir

```
@Entity
public class CreditCard {
    @Id
    private Long id;

    @ColumnTransformer(read="encrypt(card_number)",write="decrypt(?)")
    @Column(name="card_number")
    private String cardNumber;

    //...
}
```

ColumnTransformer

- @ColumnTransformer @Formula anotasyonuna alternatif olarak da kullanılabilir

```
@Entity
public class Item {
    @Id
    private Long id;

    @ColumnTransformer(read="PRICE + TAX_RATE * PRICE")
    @Column(insertable=false, updatable=false)
    private BigDecimal priceWithTax;

    //...
}
```

CompositeUserType

- **UserType**'in daha gelişmiş halidir, HQL sorgularında kompleks bir tipin property'lerine de refer etme imkanı sağlar

```
@org.hibernate.annotations.Type(type =
"com.example.MonetaryAmountUserType")
@org.hibernate.annotations.Columns(columns = {
    @Column(name="INITIAL_PRICE_AMOUNT"),
    @Column(name="INITIAL_PRICE_CURRENCY", length = 2)})
private MonetaryAmount initialPrice;
```

```
from Item i where i.initialPrice.amount > 100.0 and
i.initialPrice.currency = 'AUD'
```

Custom Tiplere Parametre Geçilmesi

- **ParameterizedType** arayüzünü implement eden custom tiplere eşleme sırasında parametre de geçmek mümkündür

```
@org.hibernate.annotations.TypeDefs({
    @org.hibernate.annotations.TypeDef(
        name="monetary_amount_usd",
        typeClass = MonetaryAmountParameterizedType.class,
        parameters = { @Parameter(name="convertTo", value="USD") }
    ),
    @org.hibernate.annotations.TypeDef(
        name="monetary_amount_eur",
        typeClass = MonetaryAmountParameterizedType.class,
        parameters = { @Parameter(name="convertTo", value="EUR") }
    })
public class Bid {

    @org.hibernate.annotations.Type(type = "monetary_amount_eur")
    @org.hibernate.annotations.Columns(columns={
        @Column(name = "BID_AMOUNT"),
        @Column(name = "BID_AMOUNT_CUR")})
    private MonetaryAmount bidAmount;
}
```

Custom Enumeration Handler Yazılması

- JPA ve Hibernate Annotasyon'ları ile çalışırken Enum'lar **built-in saklanabilir ve yüklenebilir**
- Enum Handler yazılması daha çok Enum tiplerin Hibernate **HBM XML dosyalarında** kullanım içindir
- **UserType** veya **ParameterizedType** kullanarak bir enum handler yazılabilir
- **org.hibernate.type.EnumType**, Hibernate ile hazır gelen böyle bir implemantasyondur

Custom Enumeration Handler Yazılması

```
<typedef name="phoneEnumType"  
class="org.hibernate.type.EnumType">  
  <param name="enumClass">  
    com.javaegitimleri.petclinic.model.PhoneType  
  </param>  
</typedef>
```

```
<property name="phoneType"  
  column="PHONE_TYPE"  
  type="phoneEnumType"  
  not-null="true"  
  update="false"  
  access="field" />
```

Custom Tipler ile Sorgulama

```
Query q =session.createQuery(  
    "from Person p where p.phoneType = :pt");  
  
Properties typeParams = new Properties();  
typeParams.put("enumClass",  
    "com.javaegitimleri.petclinic.model.PhoneType");  
  
q.setParameter("pt", PhoneType.WORK,  
    Hibernate.custom(PhoneEnumType.class, typeParams));
```


İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

