Bytecode Enhancement Kabiliyeti

harezmi bilişim çözümleri

ByteCode Enhancement Nedir?

- Entity sınıfların **bytecode içeriğinin dinamik olarak değiştirilmesi** işlemidir
- Bu sayede entity sınıfların kaynak kodunda yer almayan **davranışlar bytecode içerisine dinamik olarak dahil** edilebilir
- **Derleme aşamasında veya çalışma zamanında** sınıflar JVM tarafından yüklenirken gerçekleştirilebilir

Hibernate 5 ve ByteCode Enhancement

JAVA Eğitimleri

- Hibernate 5, bytecode enhancement yöntemi ile üç farklı işlemi daha verimli biçimde gerçekleştirebilmektedir
  - **Lazy Initialization**: Entity attribute'larının lazy biçimde yüklenmesi mümkün olabilmektedir
  - **Dirty Tracking**: Persistence context içerisinde state'i değişmiş entity'leri tespit etmek için kullanılabilmektedir
  - **Association Management**: Çift yönlü ilişkilerin bir tarafında yapılan ekleme/çıkarmanın diğer tarafa otomatik yansıtılmasını sağlamaktadır

ByteCode Enhancement ve Lazy Attribute Initialization

JAVA Eğitimleri

- Entity içerisindeki attribute'ların teker teker veya grup şeklinde lazy biçimde yüklenebilmesini sağlar

```
@Entity
public class Image {

    @Column
    @Basic(fetch=FetchType.LAZY)
    private String description;

    @Column
    @Basic(fetch=FetchType.LAZY)
    @LazyGroup("lobs")
    private Blob content;
    ...
}
```

description ve content attribute'ları entity yüklenirken değil, ilk erişildikleri vakit DB'den yükleneceklerdir.

description attribute'unun yüklenmesi content alanının yüklenmesini tetiklemeyecektir.

Her ikisi ayrı lazy gruplarda yer aldıklarından ayrı ayrı erişildiklerinde yükleneceklerdir

ByteCode Enhancement ve Dirty Tracking

JAVA Eğitimleri

- Hibernate default olarak DB'den yüklenen entity'lerin state'lerinin bir kopyasını (**snapshot**) Session'da tutar
- Flush aşamasında da entity'nin **current state'i ile snapshot state'i karşılaştırarak** state değişikliklerini tespit eder
- **Date tipli property**'lerin state değişikliğini tespit edebilmek için de en iyi yöntem budur
- Çünkü bu tür property'lerin state'i **entity dışında doğrudan** değiştirilebilmektedir

ByteCode Enhancement ve Dirty Tracking

JAVA Eğitimleri

- Ancak attribute sayısının çok olduğu, Session'a çok fazla sayıda entity yüklendiği durumlarda bu yöntem **performans açısından dezavantaj** yaratmaktadır
- Bytecode enhancement sayesinde **state değişiklikleri entitynin kendi içinde** takip edilir
- Flush aşamasında da Hibernate **sadece entity'ye değişip değişmediğini sorarak** dirty kontrolü yapabilir
- Snapshot yine vardır, ancak dirty kontrolünde kullanılmaz

ByteCode Enhancement ve Assoc Mgmt

JAVA Eğitimleri

- Owner – Pet arasında çift yönlü 1:M bir ilişki olsun
- Uygulama tarafında düzgün çalışabilmek için bu **ilişki iki entity üzerinde de yönetilmeli**dir

```
Owner owner = session.get(Owner.class, 1L);
Pet pet = session.get(Pet.class, 101L);

owner.getPets().add(pet);
pet.setOwner(owner);
```

- Bytecode enhancement sayesinde ilişkinin bir tarafında yapılan değişiklik **otomatik diğer tarafa** da yansıtılır

ByteCode Enhancement Nasıl Gerçekleştirilir?

- **Runtime** ve **build-time** şeklinde iki farklı biçimde gerçekleştirilebilir
- **Runtime sadece JPA managed ortamda (JavaEE)** gerçekleşebilir
- Ayrıca persistent unit içerisinde **hibernate.ejb.use_class_enhancer=true** property'si tanımlı olmalıdır
- Diğer bir kısıt ise **sadece anotasyonlarla işaretlenmiş entity'ler**de devreye girmesidir

ByteCode Enhancement Nasıl Gerçekleştirilir?

- Build-time için ise Hibernate **gradle** ve **maven plugin'leri** sunmaktadır

```
<plugin>
    <groupId>org.hibernate.orm.tooling</groupId>
    <artifactId>hibernate-enhance-maven-plugin</artifactId>
    <version>${hibernate.version}</version>
    <executions>
        <execution>
            <configuration>
                <failOnError>true</failOnError>
                <enableLazyInitialization>true</enableLazyInitialization>
                <enableDirtyTracking>true</enableDirtyTracking>
                <enableAssociationManagement>true</enableAssociationManagement>
            </configuration>
            <goals>
                <goal>enhance</goal>
            </goals>
        </execution>
    </executions>
</plugin>
```

ByteCode Enhancement ve Proxy

JAVA Eğitimleri

- Build-time bytecode enhancement gerçekleştirildiği takdirde Session.load(), EntityManager.getReference() metotlarından **proxy dönülmesi devre dışı** kalmaktadır
- Bu metotlar Session.get() veya EntityManager.find() gibi davranacaklardır

İletişim

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