Hibernate Proxy Kabiliyeti

harezmi bilişim çözümleri

Hibernate ve Proxy Kabiliyeti

JAVA Eğitimleri

- Hibernate **LAZY M:1 ve 1:1 ilişkileri yönetmek için proxy** kabiliyetinden yararlanmaktadır
- Bunun için SessionFactory bootstrap sırasında **her bir entity sınıf için ondan türeyen bir de proxy sınıf** üretilir
- Hibernate bunun için **Javassit kütüphanesi**nden yararlanmaktadır
- Proxy sınıfların üretilebilmesi için entity sınıfların **final modifier ile tanımlanmaması** gerekir

Proxy Kabiliyetinin Devre Dışı Bırakılması

- Proxy üretimi entity düzeyinde **devre dışı** bırakılabilir

```
@Entity
@org.hibernate.annotations.Proxy(lazy = false)
public class Owner {

}
```

- Proxy devre dışı bırakıldığı vakit **load()** metodu proxy döndürmez, session.get() gibi davranır
- **M:1 ve 1:1** LAZY ilişkiler de EAGER gibi davranır

Proxy Sınıfın Oluşturulması

- Proxy sınıfın **extend edeceği** interface veya üst sınıf da explicit olarak belirtilebilir

```
@Entity
@org.hibernate.annotations.Proxy(
        proxyClass = OwnerInterface.class)
public class Owner implements OwnerInterface {

}
```

Proxy Nesneler ve Object Equality

- Proxy nesneler target nesne'nin **sınıfından türer**ler ve **target nesneyi wrap eder**ler
- Ancak **proxy nesnedeki attribute'lar NULL** veya sınıf tanımında atanmış **initial değer**leri içerirler
- Proxy nesne üzerinden **target nesnenin attribute değerlerine erişim** getter metotlar üzerinden olmalıdır
- Dolayısı ile **equals** ve **hashCode** metotlarında nesnelerin attribute'larına erişimde **hep getter metotlar üzerinden** gerçekleşmelidir

Proxy Nesneler ve Object Equality

JAVA Eğitimleri

```
@Entity
public class User {
    @Id
    @GeneratedValue
    private Long id;

    private String username;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }
    ...
}
```

Proxy Nesneler ve Object Equality

JAVA Eğitimleri

Proxy nesnelerin yer aldığı equality kontrollerinde veya Collection API işlemlerinde sorun olacaktır!

```
@Entity
public class User {
    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null) return false;
        if (getClass() != obj.getClass()) return false;
        User other = (User) obj;
        if (username == null) {
            if (other.username != null)
            return false;
        } else if (!username.equals(other.username)) return false;
        return true;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((username == null) ? 0 : username.hashCode());
        return result;
    }
}
```

Proxy Nesneler ve Object Equality


```
@Entity
public class User {
    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null) return false;
        if (!getClass().isAssignableFrom(obj.getClass())) return false;
        User other = (User) obj;
        if (getUsername() == null) {
            if (other.getUsername() != null)
            return false;
        } else if (!getUsername().equals(other.getUsername())) return false;
        return true;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((getUsername() == null) ? 0 :
                                                getUsername().hashCode());
        return result;
    }
}
```

Field/Property Düzeyinde Erişim ve Proxy Initialization

JAVA Eğitimleri

- **Field düzeyinde erişim** stratejisi kullanılırsa Proxy nesnenin **identifier metoduna erişim** (getId) dahi entity'nin DB'den **SELECT** edilmesini **tetikler**
- **Getter düzeyinde erişim**de ise identifier metoduna erişim **SELECT tetiklemez**, ancak diğer alanların metotlarına erişimde SELECT tetiklenir
- Bu durum hem Session.load() ile elde edilen hem de M:1 ve 1:1 lazy ilişkilerde dönülen proxy'lerde böyledir

Field/Property Düzeyinde Erişim ve Proxy Initialization

JAVA Eğitimleri

Field Level Access

```
@Entity
public class Foo {

    @Id
    @GeneratedValue
    private Long id;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }
}
```

Getter Level Access

```
@Entity
public class Foo {

    private Long id;

    @Id
    @GeneratedValue
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }
}
```

```
Foo foo = session.load(Foo.class, 1L);

long id = foo.getId();
```

Soldaki field level erişim tanımı söz konusu olduğu vakit getId() metoduna erişim anında bir SELECT çalıştırılır

Sağdaki getter level erişim tanımı söz konusu olduğu vakit ise getId() metoduna Erişim herhangi bir SELECT tetiklemeden id değeri dönülür

İletişim

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