

Hibernate Spesifik Veri Erişim Yöntemleri

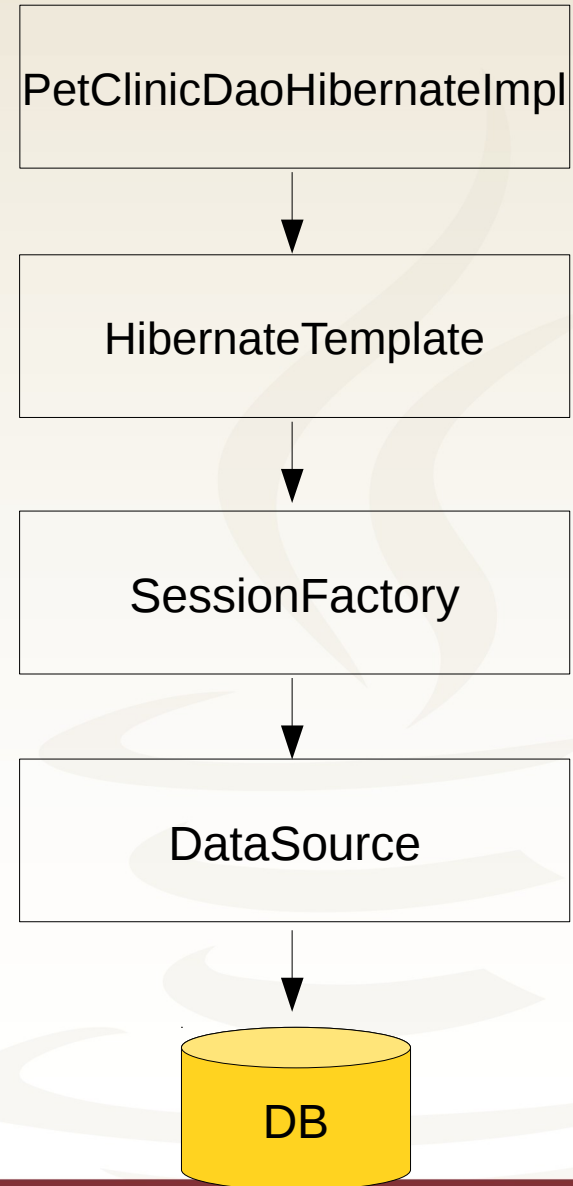


HibernateTemplate ve HibernateDaoSupport Sınıfları

- Spring **Hibernate 3 öncesi dönemde** Hibernate ile ORM işlemlerini kolaylaştırmak için bu sınıfları sunmuştur
- **HibernateTemplate**, template method yaklaşımına sahip bir **utility sınıf**tır
- **HibernateDaoSupport** ise DAO bean'leri için **base sınıf** olarak tasarlanmıştır
- Hibernate 3 ile birlikte **bu sınıflara ihtiyaç kalmamıştır**

HibernateTemplate Konfigürasyonu

- HibernateTemplate, **Session** arayüzünün sahip olduğu metotları sunmaktadır
- Buna ilaveten pek çok **helper metoda** da sahiptir
- Session'a **HibernateCallback** ile doğrudan da erişmek mümkündür



HibernateTemplate Kullanımı

```
public class ProductDaoImpl implements ProductDao {  
  
    private HibernateTemplate hibernateTemplate;  
  
    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {  
        this.hibernateTemplate = hibernateTemplate;  
    }  
  
    public Collection loadProductsByCategory(String category) throws DataAccessException {  
        return hibernateTemplate.find("from Product p where p.category=?", category);  
    }  
}
```

find
findByCriteria
findByNameQuery
findByNameQueryAndNamedParam
...

get
load
save
saveOrUpdate
merge
delete
...

HibernateTemplate ve HibernateCallback

```
public class ProductDaoImpl implements ProductDao {  
  
    private HibernateTemplate hibernateTemplate;  
  
    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {  
        this.hibernateTemplate = hibernateTemplate;  
    }  
  
    public Collection loadProductsByCategory(final String category) throws  
    DataAccessException {  
  
        return hibernateTemplate.execute(new HibernateCallback() {  
            public Object doInHibernate(Session session) {  
                Criteria criteria = session.createCriteria(Product.class);  
                criteria.add(Restrictions.eq("category", category));  
                criteria.setMaxResults(6);  
                return criteria.list();  
            }  
        });  
    }  
}
```

HibernateTemplate'in sunduğu yardımcı metotlar yetersiz kaldığı vakit Her zaman bir HibernateCallback nesnesi üzerinden Session'a erişip İstenen her türlü persistence işlemi gerçekleştirilebilir

HibernateDaoSupport Kullanımı

- Uygulamanın DAO sınıfları Spring'in base sınıfı olan **HibernateDaoSupport**'u extend eder
- **HibernateCallback** kullanımına gerek duymaksızın transactional current **Session**'a erişim imkanı sunar
- Data access exception'larını otomatik olarak dönüştürür

HibernateDaoSupport Kullanımı

```
public class HibernateProductDao extends HibernateDaoSupport implements
ProductDao {

    public Collection loadProductsByCategory(String category) throws DataAccessException {
        Session session = getSession(false);

        Query query = session.createQuery("from Product p where p.category=?");
        query.setString(0, category);
        List result = query.list();
        return result;
    }
}
```

allowCreate:false değeri transactional Session nesnesinin Kullanılmasını sağlar. Session kapatma işi ile DAO içerisinde uğraşmaya gerek yoktur.

İletişim

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

