

# Hibernate Konfigürasyonu



# Hibernate Konfigürasyonu

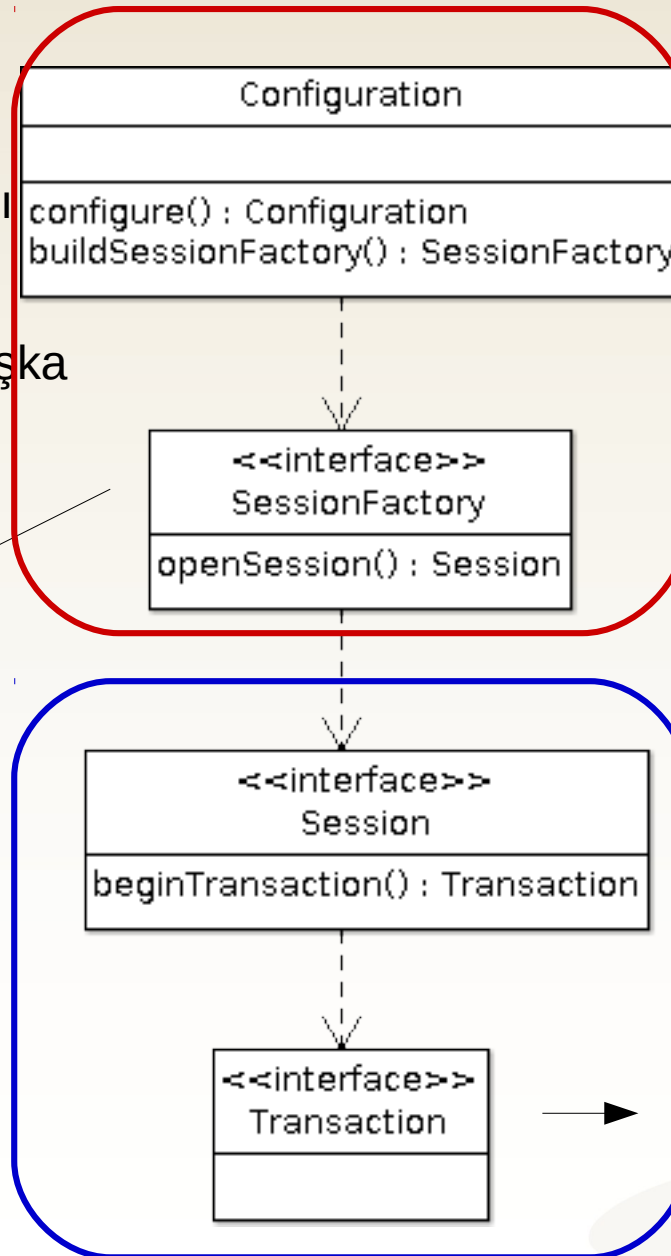
- Hibernate ile çalışmaya başlayabilmek için öncelikle **Configuration** nesnesi ile konfigürasyon bilgileri yüklenmelidir
- Ardından uygulama düzeyinde bir **SessionFactory** nesnesine ihtiyaç vardır
- Artık çalışma zamanında bu konfigürasyon üzerinden elde edilen **Session** nesneleri üzerinden de persistence işlemleri gerçekleştirilir
- Persistence işlemleri gerçekleştirirken de **Transaction**'a ihtiyaç duyulur

# Hibernate Konfigürasyonu

new Configuration()  
root classpath'de  
**hibernate.properties** dosyasını  
arar

Opsiyoneldir, varsa yüklenir, başka  
bir yerden yüklenemez

SessionFactory nesnesi  
uygulama genelinde bir defa  
oluşturulur



Configure() metodu  
root classpath'de  
**hibernate.cfg.xml** dosyasını arar  
bulamaz ise hata verir

Başka bir lokasyondan da  
yüklenebilir

Persistence işlemleri için  
her ihtiyaç duyulduğunda  
Session nesneleri yaratılır

Persistence işlemleri mutlaka  
bir transaction içerisinde  
gerçekleştirilmelidir

# Hibernate Konfigürasyonu

- **SessionFactory** nesnesi uygulama genelinde sadece bir kere oluşturulur
- SessionFactory **thread-safe**'tir
- SessionFactory nesnesi **JNDI veya JMX** ile merkezi bir yerden yönetilebilir
- Bu opsiyonlar **full Java EE ortamında** mevcuttur
- Spring uygulamalarında SessionFactory ApplicationContext'de **singleton bir bean** olarak tanımlanır
- Bütün **Session** nesneleri aynı **SessionFactory**'den elde edilir

# Hibernate'in Başlatılması ve SessionFactory Build İşlemi

```
Configuration cfg = new Configuration();  
  
cfg.configure();  
  
SessionFactory sessionFactory = cfg.buildSessionFactory();
```

Hibernate SessionFactory oluşturma işlemi uygulama genelinde tek sefer yapılacak bir işlemdir

Bu kısım uygulama içerisinde statik initialization yapan bir yerde bir defalığına yapılabilir. Daha sonra SessionFactory nesnesi uygulama genelinde erişilir kılınmalıdır

# hibernate.properties

```
hibernate.connection.driver_class=org.h2.Driver
hibernate.connection.url=jdbc:h2:tcp://localhost/~ /test
hibernate.connection.username=sa
hibernate.connection.password=
```

# hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.hbm2ddl.auto">create</property>
    <property name="hibernate.show_sql">true</property>
    <property name="hibernate.format_sql">true</property>
    <property name="hibernate.use_sql_comments">true</property>

    <mapping class="com.javaegitimleri.petclinic.model.Owner"/>
    <mapping class="com.javaegitimleri.petclinic.model.Pet"/>

  </session-factory>
</hibernate-configuration>
```

hibernate.properties'de tanımlanan bütün property'ler burada da tanımlanabilir

Buradaki tanımlar hibernate.properties'deki tanımları ezer

hibernate.cfg.xml'de ayrıca metadata lokasyonları ve security, cache, event listener gibi diğer konfigürasyon bilgileri de tanımlanır

# Hibernate ve Veritabanı Bağlantıları

- Hibernate konfigürasyonu veritabanı bağlantılarını kimin yöneteceğine göre iki farklı biçimde yapılabilir
  - Veritabanı bağlantılarını **Hibernate yönetebilir**
    - Çok tercih edilmez
    - Hibernate property tanımlarında driver\_class, url, username, password tanımlanarak yapılır
  - Ya da veritabanı bağlantıları **harici bir DataSource** nesnesi üzerinden elde edilebilir
    - En çok tercih edilen yöntemdir
    - Spring gibi frameworklerle birlikte çalışırken de bu yöntem kullanılır



# Hibernate Konfigürasyonu ve Managed DataSource

- Hibernate **harici bir JDBC DataSource** nesnesi ile ayarlanarak kullanılır
- Kurumsal uygulamalarda **yaygın kullanım biçimi** böyledir
- DataSource nesnesi genellikle uygulama sunucusunun **JNDI repository**'sinde tanımlanır
- Hibernate DataSource'a **JNDI lookup** ile erişir
- **Spring uygulamalarında** ise DataSource nesnesi Spring Container içerisinde Hibernate SessionFactory konfigürasyonuna enjekte edilir

# Hibernate Konfigürasyonu ve Managed DataSource

```
<property name="hibernate.connection.datasource">  
    java:/comp/env/jdbc/DataSource  
</property>  
<property name="hibernate.jndi.url">  
    jnp://localhost:1099  
</property>  
<property name="hibernate.jndi.class">  
    org.jnp.interfaces.NamingContextFactory  
</property>  
<property name="hibernate.dialect">  
    org.hibernate.dialect.H2Dialect  
</property>
```



Eğer harici datasource nesnesi kullanılıyor ise Hibernate **DIALECT** bilgisine ihtiyaç duyar. Connection'ın hibernate tarafından oluşturulduğu durumda ise JDBC Driver vasıtası ile DIALECT otomatik olarak tespit edilebildiği için bu property'ye gerek yoktur

# Hibernate Logları ve SQL İfadeleri

- SQL ifadelerinin CONSOLE'a loglanması **hibernate property tanımları** ile de yönetilebilir
  - `hibernate.show_sql=true`
  - `hibernate.format_sql=true`
  - `hibernate.use_sql_comments=true`

# Hibernate Logları ve SQL İfadeleri

- Hibernate log mesajları, SQL ifadeleri vs **log4j** ile de takip edilebilir
- **log4j.xml** içerisinde **TRACE** seviyesinde bir **org.hibernate.type** logger tanımı ile SQL parametrelerini loglamak da mümkündür

```
<logger name="org.hibernate">  
  <level value="DEBUG" />  
</logger>
```

```
<logger name="org.hibernate.type">  
  <level value="TRACE" />  
</logger>
```

# Otomatik Şema Yönetimi

- Uygulamanın başlatılması sırasında **hibernate.hbm2ddl\_auto** property değerine göre DB şeması ORM metadatalara göre otomatik olarak yaratılabilir veya güncellenebilir
- hbm2ddl\_auto property'sinin alabileceği değerler
  - none
  - create
  - create-drop
  - update
  - validate

# Şema Yönetimi ve Örnek Data Import İşlemi

- Hibernate root classpath'de yer alan **import.sql** isimli bir dosyadaki SQL ifadelerini **şema export işlemi sonrası** çalıştırabilir
- Bunun için **hibernate.hbm2ddl\_auto** değeri **“create”** veya **“create-drop”** olmalıdır

# İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

