

Java Web Uygulaması Hibernate ve Bonecp Kullanımı

Java'da Maven, Hibernate ve Bonecp kullanarak web uygulaması yapmak için bazı ayarların yapılması gereklidir. Bonecp, web uygulamaların vazgeçilmez olan jdbc veri havuzu modülü için yaygın olarak kullanılan açık kaynak ve en hızlı veri havuzu kütüphanesidir. Hibernate ise Java Frameworkleri arasında en yaygın kullanılan Object Relational Mapping(ORM) kütüphanesidir

WEB-INF/hibernate-cfg.xml Dosyası

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD//EN"
4     "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
5 <hibernate-configuration>
6     <session-factory>
7         <!--org.hibernate.dialect.DB2390Dialect is for DB2 Connection.
8         If you want to use mysql then write this: org.hibernate.dialect.MySQL5Dialect or org.hibernate.dialect.MySQLDialect -->
9         <property name="dialect">org.hibernate.dialect.DB2390Dialect</property>
10        <mapping class="edu.javaProgramming.jsfSpringHibernateBonecp.hibernateClasses.gender.Gender"/>
11    <!-- edu.javaProgramming.jsfSpringHibernateBonecp.hibernateClasses.gender.Gender is just a sample class.
12    This class is an Entity Class . This class is given at the bottom of this document.-->
13    </session-factory>
14 </hibernate-configuration>
```

WEB-INF/spring-servlet.xml Konfigurasyonu

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5
6     xmlns:tx="http://www.springframework.org/schema/tx"
7     xsi:schemaLocation="http://www.springframework.org/schema/beans
8     http://www.springframework.org/schema/beans/spring-beans.xsd
9     http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd
10    http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd">
11
12    <context:annotation-config/>
13    <context:component-scan base-package="edu"/> <!--edu is just a sample in our projects. -->
14
15    <!-- Hibernate SessionFactory -->
16    <bean id="sessionFactory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
17        <property name="hibernateProperties">
18            <props>
19                <prop key="hibernate.connection.provider_class">com.jolbox.bonecp.provider.BoneCPConnectionProvider
20            </prop>
21                <prop key="hibernate.connection.driver_class">com.ibm.db2.jcc.DB2Driver</prop>
22                <prop key="hibernate.connection.url">jdbc:db2://127.0.0.1:50000/BLOG</prop>
23                <prop key="hibernate.connection.username">root</prop>
24                <prop key="hibernate.connection.password">123456</prop>
25                <prop key="bonecp.idleMaxAge">240</prop>
26                <prop key="bonecp.idleConnectionTestPeriod">60</prop>
27                <prop key="bonecp.partitionCount">3</prop>
28                <prop key="bonecp.acquireIncrement">1</prop>
29                <prop key="bonecp.maxConnectionsPerPartition">10</prop>
30                <prop key="bonecp.minConnectionsPerPartition">2</prop>
31                <prop key="bonecp.statementsCacheSize">50</prop>
32                <prop key="bonecp.releaseHelperThreads">3</prop>
33                <prop key="show_sql">>true</prop>
34            </props>
35        </property>
36        <property name="configurationClass" value="org.hibernate.cfg.AnnotationConfiguration"/>
37        <property name="configLocation" value="/WEB-INF/hibernate.cfg.xml"/>
38
39    </bean>
40
41    <bean id="transactionManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">
42        <property name="sessionFactory" ref="sessionFactory"/>
43    </bean>
44
45    <!--View Scope provider. To use org.springframework.context.annotation.Scope annotation in a bean. Usage @Scope("view"
46    That's @Scope("view") equals to @ViewScoped in JSF. Note @ViewScoped annotation is defined in javax.faces.bean -->
47    <bean class="org.springframework.beans.factory.config.CustomScopeConfigurer">
48        <property name="scopes">
49            <map>
50                <entry key="view">
51                    <bean class="edu.javaProgramming.jsfSpringHibernateBonecp.configuration.ViewScope"/>
52                </entry>
53            </map>
54        </property>
55    </bean>
```

ViewScope Sınıfı

```
1 package edu.javaProgramming.jsfSpringHibernateBonecp.configuration;
2
3 import org.springframework.beans.factory.ObjectFactory;
4 import org.springframework.beans.factory.config.Scope;
5
6 import javax.faces.context.FacesContext;
7 import java.util.Map;
8
```

```

9  /**
10 * Implements the JSF View Scope for use by Spring. This class is registered as a Spring bean with the CustomScopeConfigurer
11 */
12 public class ViewScope implements Scope {
13
14     public Object get(String name, ObjectFactory<?> objectFactory) {
15         if (FacesContext.getCurrentInstance().getViewRoot() != null) {
16             Map<String, Object> viewMap = FacesContext.getCurrentInstance().getViewRoot().getViewMap();
17             if (viewMap.containsKey(name)) {
18                 return viewMap.get(name);
19             } else {
20                 Object object = objectFactory.getObject();
21                 viewMap.put(name, object);
22                 return object;
23             }
24         } else {
25             return null;
26         }
27     }
28
29     public Object remove(String name) {
30         if (FacesContext.getCurrentInstance().getViewRoot() != null) {
31             return FacesContext.getCurrentInstance().getViewRoot().getViewMap().remove(name);
32         } else {
33             return null;
34         }
35     }
36
37     public void registerDestructionCallback(String name, Runnable callback) {
38         // Do nothing
39     }
40
41     public Object resolveContextualObject(String key) {
42         return null;
43     }
44
45     public String getConversationId() {
46         return null;
47     }
48
49 }

```

Örnek Bir Entity Sınıfı

```

1  import javax.persistence.*;
2
3  @Entity //it must be javax.persistence.Entity
4  @Table(name = "GENDER", schema = "BLOG")
5  public class Gender {
6      private Integer id;
7      private String name;
8
9      public Gender(String name) {
10         this.name = name;
11     }
12
13     @Id
14     @GeneratedValue(strategy = GenerationType.AUTO)
15     public Integer getId() {
16         return id;
17     }
18
19     public void setId(Integer id) {
20         this.id = id;
21     }
22     @Column(name="AD")
23     public String getName() {
24         return name;
25     }
26
27     public void setName(String name) {
28         this.name = name;
29     }
30 }

```

Hibernate Maven Dependencies

```

1  <dependency>
2      <groupId>org.hibernate</groupId>
3      <artifactId>hibernate-annotations</artifactId>
4      <version>3.5.6-Final</version>
5  </dependency>
6  <dependency>
7      <groupId>javassist</groupId>
8      <artifactId>javassist</artifactId>
9      <version>3.12.1.GA</version>
10 </dependency>
11 <dependency>
12     <groupId>hibernate</groupId>
13     <artifactId>hibernate-tools</artifactId>
14     <version>3.2.3.GA</version>
15 </dependency>

```

Bonecp Maven Dependencies

```
1 <dependency>
2   <groupId>com.jolbox</groupId>
3   <artifactId>bonecp</artifactId>
4   <version>0.7.1.RELEASE</version>
5 </dependency>
6 <dependency>
7   <groupId>com.jolbox</groupId>
8   <artifactId>bonecp-provider</artifactId>
9   <version>0.7.1.RELEASE</version>
10 </dependency>
11 <dependency>
12   <groupId>com.jolbox</groupId>
13   <artifactId>bonecp-spring</artifactId>
14   <version>0.7.1.RELEASE</version>
15 </dependency>
```