CIDC之compass機制

以搜索新聞news為類：

1. 首先建立了News表的pojo類：
2. 建立Hibernate的映射到News的pojo類中。
3. 配置compass(有關資料顯示，compass配置，需要配置三個文件，.cfg.xml, .cmd.xml, .cpm.xml,在CIDC中，我只看到一個文件)

**.Cfg.xml(**compass-core-config**) 用來定義索引文件存放的路徑已經搜索所採用的算法**

 <?xml version="1.0" encoding="UTF-8"?>  
<compass-core-config  
 xmlns="<http://www.opensymphony.com/compass/schema/core-config>"  
 xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
 xsi:schemaLocation="<http://www.opensymphony.com/compass/schema/core-config>  
           <http://www.opensymphony.com/compass/schema/compass-core-config.xsd>">

 <compass name="default"> <!—这个名字随你取了，但它是必须的-->

  <connection>  
   <file path="target" /> <!—这里是索引文件的存放路径，我们设置的是当前工程的相对路径target-?  
  </connection>

  <searchEngine>  
    <!-- 因是使用自己的分词算法，所以这里的类型必须是CustomAnalyzer -->  
          <analyzer name="MMAnalyer" type="CustomAnalyzer" analyzerClass="jeasy.analysis.MMAnalyzer">  
              <stopWords>  
                  <stopWord value="test" />  
              </stopWords>  
          </analyzer>  
      </searchEngine>

 </compass>  
</compass-core-config>  
 在上面的配置中，我们使用的我们选用的一个中文分词算法库，你可以用compass自带的。具体compass提供了哪些分词算法，请查阅compass的手册。

**.cmd.xml(**compass-core-meta-data**)定義實體，在實體中可以定義所要搜索的所有實體以及要搜索的所有字段。**

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE compass-core-meta-data PUBLIC   
    "-//Compass/Compass Core Meta Data DTD 1.0//EN"  
    "<http://www.opensymphony.com/compass/dtd/compass-core-meta-data.dtd>">

<compass-core-meta-data>

<!-- 定义一个实体和字段组-->  
    <meta-data-group id="mycompass" displayName="My Compass">   
        <description>Mycompass Meta Data</description>         
        <uri>http://com/darkhe/sample/mycompass</uri>  
   <!-- 申明所有需要检索的实体-->          
        <alias id="Article" displayName="Article">  
            <description>Article alias</description>  
            <uri>http://com/darkhe/sample/mycompass/alias/Article</uri>  
            <name>Article</name>  
        </alias>  
          
 <!-- 申明所有需要检索的属性或者字段，而不管这些属性或者字段是哪个实体的 -->          
        <meta-data id="ArticleTitle" displayName="ArticleTitle">  
            <description>ArticleTitle</description>  
            <uri>http://com/darkhe/sample/mycompass/alias/ArticleTitle</uri>  
            <name>ArticleTitle</name>  
        </meta-data>  
          
        <meta-data id="PublishDate" displayName="PublishDate">  
            <description>PublishDate</description>  
            <uri>http://com/darkhe/sample/mycompass/alias/PublishDate</uri>  
            <name format="yyyy-MM-dd hh:mm:ss">date</name>  
        </meta-data>  
          
        <meta-data id="Summary" displayName="Summary">  
            <description>Summary</description>  
            <uri>http://com/darkhe/sample/mycompass/alias/Summary</uri>  
            <name>Summary</name>  
        </meta-data>  
          
        <meta-data id="Content" displayName="Content">  
            <description>Content</description>  
            <uri>http://com/darkhe/sample/mycompass/alias/Content</uri>  
            <name>Content</name>  
        </meta-data>

        <meta-data id="KeyList" displayName="KeyList">  
            <description>KeyList</description>  
            <uri>http://com/darkhe/sample/mycompass/alias/KeyList</uri>  
            <name>KeyList</name>  
        </meta-data>          
                          
    </meta-data-group>  
      
</compass-core-meta-data>

**.cpm.xml(compass-core-mapping)定義實體和字段的對應關系**<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE compass-core-mapping PUBLIC  
    "-//Compass/Compass Core Mapping DTD 1.0//EN"  
    "<http://www.opensymphony.com/compass/dtd/compass-core-mapping.dtd>">

<!-- 这里的包名必须和pojo的包名一致 -->  
<compass-core-mapping package="com.darkhe.sample.mycompass">

<!-- 定义实体及其字段的对应关系 -->

<!-- 注意实体及其字段的名称的大小写应当与pojo对象一致，而不是与数据库一致  
  关于pojo与数据库的对应表的一致性关系由hibernate的映谢文件定义，而不是这个文件  
  当前映射文件只定义compass与hibernate的关系 -->

 <class name="Article" alias="${mycompass.Article}">  
  <id name="ArticleId" />  
    
  <property name="ArticleTitle">  
   <meta-data>${mycompass.ArticleTitle}</meta-data>  
  </property>

  <property name="PublishDate">  
   <meta-data>${mycompass.PublishDate}</meta-data>  
  </property>

  <property name="Summary">  
   <meta-data>${mycompass.Summary}</meta-data>  
  </property>

  <property name="Content">  
   <meta-data>${mycompass.Content}</meta-data>  
  </property>

  <property name="KeyList">  
   <meta-data>${mycompass.KeyList}</meta-data>  
  </property>  
 </class>  
 </compass-core-mapping>

1. 配置compass與spring 和Hibernate的結合：

首先在applicationContext-resources.xml中配置了數據源

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

<property name="driverClassName" value="@DB-DRIVERNAME@"/>

<property name="url" value="@DB-URL@"/>

<property name="username" value="@DB-USERNAME@"/>

<property name="password" value="@DB-PASSWORD@"/>

<property name="maxActive" value="100"/>

<property name="maxIdle" value="30"/>

<property name="maxWait" value="1000"/>

<property name="defaultAutoCommit" value="true"/>

<property name="removeAbandoned" value="true"/>

<property name="removeAbandonedTimeout" value="60"/>

</bean>

然後在applicationContext-hibernate.xml中配置Hibernate的數據映射文件。

<bean id="sessionFactory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">

<property name="dataSource" ref="dataSource"/>

<property name="mappingResources">

<list>

<value>com/foxconn/cic/model/Role.hbm.xml</value>

<value>com/foxconn/cic/model/User.hbm.xml</value>

<value>com/foxconn/cic/model/Keyword.hbm.xml</value>

<value>com/foxconn/cic/model/Catalog.hbm.xml</value>

<value>com/foxconn/cic/model/CatalogType.hbm.xml</value>

<value>com/foxconn/cic/model/News.hbm.xml</value>

<value>com/foxconn/cic/model/NewsImage.hbm.xml</value>

<value>com/foxconn/cic/model/Website.hbm.xml</value>

<value>com/foxconn/cic/model/Quoter.hbm.xml</value>

<value>com/foxconn/cic/model/Material.hbm.xml</value>

<value>com/foxconn/cic/model/Price.hbm.xml</value>

<value>com/foxconn/cic/model/ExchangeRate.hbm.xml</value>

<value>com/foxconn/cic/model/InterestRate.hbm.xml</value>

<value>com/foxconn/cic/model/UserActivityLog.hbm.xml</value>

</list>

</property>

<property name="hibernateProperties">

<props>

<prop key="hibernate.show\_sql">false</prop>

<prop key="hibernate.dialect">@HIBERNATE-DIALECT@</prop>

<prop key="hibernate.query.substitutions">true 'Y', false 'N'</prop>

<!-- Create/update the database tables automatically when the JVM starts up

<prop key="hibernate.hbm2ddl.auto">update</prop> -->

<!-- Turn batching off for better error messages under PostgreSQL

<prop key="hibernate.jdbc.batch\_size">0</prop> -->

</props>

</property>

</bean>

然後在applicationContext-service.xml中開始配置compass

<bean id="compass" class="org.compass.spring.LocalCompassBean">

<property name="classMappings">

<list>

<value>com.foxconn.cic.model.InterestRate</value>

<value>com.foxconn.cic.model.NewsImage</value>

<value>com.foxconn.cic.model.Price</value>

<value>com.foxconn.cic.model.ExchangeRate</value>

</list>

</property>

<property name="resourceLocations">

<list>

<value>classpath:com/foxconn/cic/model/News.cpm.xml</value>

<value>classpath:com/foxconn/cic/model/Website.cpm.xml</value>

<value>classpath:com/foxconn/cic/model/Material.cpm.xml</value>

</list>

</property>

<property name="compassSettings">

<props>

<prop key="compass.engine.connection">

file://C:/FoxconnCIC/

</prop>

<prop key="compass.transaction.factory">

org.compass.spring.transaction.SpringSyncTransactionFactory

</prop>

<prop key="compass.engine.mergeFactor">

100

</prop>

<prop key="compass.engine.maxBufferedDocs">

100

</prop>

<!--

<prop key="compass.engine.store.wrapper.test.type">

org.compass.core.lucene.engine.store.wrapper.SyncMemoryMirrorDirectoryWrapperProvider

</prop>

-->

<prop key="compass.engine.analyzer.default.type">

org.apache.lucene.analysis.standard.StandardAnalyzer

</prop>

<prop key="compass.engine.analyzer.search.type">

org.apache.lucene.analysis.standard.StandardAnalyzer

</prop>

<prop key="compass.engine.analyzer.keyword.type">

org.apache.lucene.analysis.KeywordAnalyzer

</prop>

<prop key="compass.engine.highlighter.default.textTokenizer">

auto

</prop>

<prop key="compass.converter.html.type">

com.foxconn.cic.util.HTMLConverter

</prop>

<prop key="compass.engine.highlighter.default.formatter.simple.pre">

<![CDATA[<font color="red"><b>]]>

</prop>

<prop key="compass.engine.highlighter.default.formatter.simple.post">

<![CDATA[</b></font>]]>

</prop>

</props>

</property>

<property name="transactionManager" ref="transactionManager" />

</bean>