**缓存，整合spring+mybatis**

1. **缓存：**
2. **缓存：一级缓存（默认开启），二级缓存（手动设置）**
3. **一级缓存：一级缓存区域是根据SqlSession为单位划分的，同一个sqlSession下可以共享曾经查询过的数据。**
4. **二级缓存：不同sqlSession间可以共享各自曾经查询过的数据，区域是SessionFactory**
5. **如何开启二级缓存：**
6. **在全局配置文件的指定位置配置<setting>**

**<settings>**

**<setting name="cacheEnabled" value="true"/>**

**</settings>**

**properties,**

**settings,**

**typeAliases,**

**typeHandlers,**

**objectFactory,**

**objectWrapperFactory,**

**reflectorFactory,**

**plugins,**

**environments,**

**databaseIdProvider,**

**mappers".**

1. **在局部配置文件<mapper>标签中配置<cache>即可完成二级缓存的配置。**

**Ps:**

**Type属性可以更换成ehcache 完成二级缓存**

**更换为：<cache type="org.mybatis.caches.ehcache.EhcacheCache"></cache>**

**需要：**

**slf4j-api-1.7.5**

**slf4j-log4j12-1.7.5**

**ehcache-core-2.6.5.jar**

**mybatis-ehcache-1.0.2.jar**

**ehcache.xml**

**注意：**

1. **缓存：不同的方法带不同的缓存，就是说即便参数，返回值一致，只要不是不同的**

**方法，那么即便是内容指向同一个查询语句，也有着不同的缓存空间。如果insert,delete,update标签使用了flushCache=”false” 那么执行这些操作时的缓存不会清空。如果select标签使用了useCache=”false”,那么查询将不会从缓存中查找，会直接使用数据库查询。**

1. **一级缓存：如果SqlSession调用了close()方法，会释放掉一级缓存PerpetualCache对象，一级缓存将不可用，如果clearCache()，会清空PerpetualCache对象中的数据，但是该对象仍可使用。如果在insert,delete,update等操作后，一级缓存也会清空，不需要commit()，也就是说，执行这些操作但实际上并未写入到数据库中时，一级缓存就清空了。**
2. **二级缓存：**

**update，delete,insert ----- commit后清空缓存，openSession自动提交不会清掉缓存**

**select -------- close,commit都能提交缓存上去被使用**

1. **整合mybatis + spring**
2. **Jar:**

**aopalliance-1.0.0.jar**

**aspectjweaver-1.8.10.jar**

**c3p0-0.9.5.2.jar**

**commons-logging-1.2.jar**

**ehcache-core-2.6.5.jar**

**javax.annotation-api-1.3.2.jar**

**log4j-1.2.17.jar**

**log4j-api-2.0-rc1.jar**

**log4j-core-2.0-rc1.jar**

**mchange-commons-java-0.2.12.jar**

**mybatis-3.5.4.jar**

**mybatis-ehcache-1.0.2.jar**

**mybatis-spring-2.0.4.jar**

**mysql-connector-java-8.0.16.jar**

**slf4j-api-1.7.5.jar**

**slf4j-log4j12-1.7.5.jar**

**spring-aop-4.3.0.RELEASE.jar**

**spring-aspects-4.3.0.RELEASE.jar**

**spring-beans-4.3.0.RELEASE.jar**

**spring-context-4.3.0.RELEASE.jar**

**spring-core-4.3.0.RELEASE.jar**

**spring-expression-4.3.0.RELEASE.jar**

**spring-jdbc-4.3.0.RELEASE.jar**

**spring-tx-4.3.0.RELEASE.jar**

**spring-web-4.3.0.RELEASE.jar**

**spring-webmvc-3.2.0.RC2.jar**

1. **配置文件：**
2. **springConfig.xml**

**要点：**

* 1. **配置dataSource,需要：<context:property-placeholder location..>**
  2. **<bean class="com.mchange.v2.c3p0.ComboPooledDataSource" id="dataSource">以及内部的一系列<property>**
  3. **配置：**
     1. 使用SqlSessionFactoryBean,这个bean,OpenSession,selectOne-

1. <!--0. 使用这个bean,OpenSession,selectOne-->
2. <bean **class**="org.mybatis.spring.SqlSessionFactoryBean" id="factoryBean">
3. <property name="dataSource" ref="dataSource"></property>
4. <property name="configLocation" value="classpath:MyBatisMapConfig.xml"></property>
5. </bean>
   * 1. **Dao类** 继承SqlSessionDaoSupport，getBean这个dao
   1. <!-- 1. 继承SqlSessionDaoSupport, getBean这个dao-->
   2. <bean **class**="cn.cxd.impl.UserDaoImpl" id="dao">
   3. <property name="sqlSessionFactory" ref="factoryBean"></property>
   4. </bean>
      1. 由Spring产生MapperFactory，客户端getBean,openSession,getMapper
   5. <!-- 2. 由Spring产生MapperFactory,getBean,openSession,getMapper -->
   6. <bean **class**="org.mybatis.spring.mapper.MapperFactoryBean">
   7. <property name="sqlSessionFactory" ref="factoryBean"></property>
   8. <property name="mapperInterface" value="cn.cxd.mapper.UserMapper"></property>
   9. </bean>
      1. 由Spring产生MapperFactory..客户端getBean
   10. <!-- 3. 由Spring产生MapperFactory..getBean-->
   11. <bean id="userMapper" **class**="org.mybatis.spring.mapper.MapperFactoryBean">
   12. <property name="sqlSessionFactory" ref="factoryBean"></property>
   13. <property name="mapperInterface" value="cn.cxd.mapper.UserMapper"></property>
   14. </bean>
       1. 自动为包下配置mapperFactory,getBean
   15. <!--4. 自动为包下配置mapperFactory,getBean-->
   16. <bean **class**="org.mybatis.spring.mapper.MapperScannerConfigurer">
   17. <property name="basePackage" value="cn.cxd"></property>
   18. <property name="sqlSessionFactoryBeanName" value="factoryBean"></property>
   19. </bean>
6. **myBatisConfig.xml**

**要点：**

* + - 1. **不用配置<environment>，交给spring就行**

1. **userMapper.xml**

**要点：**

* + - 1. **没有新的要点，不再提及**

1. **基本思路**

**思路：spring做主导，配合myBatis**

**1. 由spring配置文件配置dataSource,mybatis 不用配置environment**

**2. Spirng配置文件也要配置SqlSessionFactoryBean，配上mybatis全局配置文件**

**3. mybatis的全局配置文件要配上局部配置文件**

**4. 其余就是getBean执行方法即可，namespace会将方法引到局部配置文件上。**