Atitit mybatis spring整合。读取spring、yml、文件的mysql url

步骤，读取yml，文件，使用ongl定位到url pwd usr

读取mybatis模板配置，，替换其中的mysql url等参数。。

注意xml转义符号

使用SqlSessionFactoryBuilder().build(is2); 加在加载

*package* org.chwin.firefighting.apiserver.data;  
  
*import* java.io.ByteArrayInputStream;  
*import* java.io.InputStream;  
*import* java.io.InputStreamReader;  
*import* java.util.List;  
*import* java.util.Map;  
  
*import* com.google.common.base.Charsets;  
*import* com.google.common.io.CharStreams;  
*import* groovy.xml.XmlUtil;  
*import* ognl.Ognl;  
*import* ognl.OgnlException;  
*import* org.apache.ibatis.session.SqlSession;  
*import* org.apache.ibatis.session.SqlSessionFactory;  
*import* org.apache.ibatis.session.SqlSessionFactoryBuilder;  
  
*import* com.alibaba.fastjson.JSON;  
  
@SuppressWarnings("rawtypes")  
*public class* mybatisdemo {  
 *public static void* main(String[] args) *throws* Exception {  
  
 String resource = "/mybatis.xml";  
 *// ����mybatis �������ļ�����Ҳ���ع�����ӳ���ļ���  
 //ClassLoader classLoader = .getClassLoader();* InputStream is = mybatisdemo.*class*.getResourceAsStream(resource);  
 String mybatisCfg\_result = CharStreams.*toString*(*new* InputStreamReader(is, Charsets.UTF\_8));  
  
 org.yaml.snakeyaml.Yaml yaml = *new* org.yaml.snakeyaml.Yaml();  
 Object mObject=yaml.load(mybatisdemo.*class*.getResourceAsStream("/application-test.yml"));  
 Object expression = Ognl.*parseExpression*("spring.datasource.url");  
 Object url = Ognl.*getValue*(expression, mObject);  
 Object usr = Ognl.*getValue*(Ognl.*parseExpression*("spring.datasource.username"), mObject);  
 Object pwd = Ognl.*getValue*(Ognl.*parseExpression*("spring.datasource.password"), mObject);  
*if*(pwd==*null*)pwd="";  
 url= XmlUtil.*escapeXml*(url.toString());  
  
 mybatisCfg\_result=mybatisCfg\_result.replaceAll("\\$\\{mysql.url}",url.toString());  
 mybatisCfg\_result=mybatisCfg\_result.replaceAll("\\$\\{mysql.username}",usr.toString());  
  
 mybatisCfg\_result=mybatisCfg\_result.replaceAll("\\$\\{mysql.password}",pwd.toString());  
  
  
  
 System.out.println(mybatisCfg\_result);  
 InputStream is2=*new* ByteArrayInputStream(mybatisCfg\_result.getBytes());  
 *// ����sqlSession �Ĺ���* SqlSessionFactory sqlSessionFactory = *new* SqlSessionFactoryBuilder().build(is2);  
  
 SqlSession session = sqlSessionFactory.openSession(*true*);  
 *// api ��Ϊ[ openSession(boolean autoCommit) ]���ò���ֵ���������Ƹ� sqlSession �Ƿ��Զ��ύ��true��ʾ�Զ��ύ��false��ʾ���Զ��ύ[���޲εķ�������һ�£������Զ��ύ]* MybatisMapperCls mapper = session.getMapper(MybatisMapperCls.*class*);  
  
 *// List li =mapper.queryall();* List<Map> li = mapper.query("select \* from tab1");  
 System.out.println(JSON.*toJSONString*(li, *true*));  
 session.close();  
 *// = session.selectList(arg0);* }