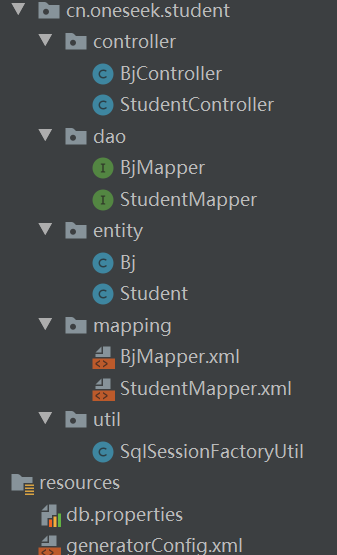
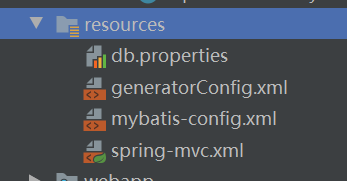
码云链接 <https://gitee.com/Chuangzw/JavaEE_Learn/tree/master/student-springmvc-maven>

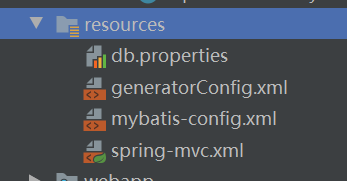
结构





# 改动代码

增加配置文件



generatorConfig.xml mybatis自动生成tool

mybatis-config.xml mybatis配置文件

spring-mvc.xml springmvc配置文件

自动生成mybatis插件的实现

Springmvc整合mybatis，需要引入mybatis包，在pom.xml中添加以下依赖包：

  <!--mybatis-->  
         <dependency>  
             <groupId>org.mybatis</groupId>  
             <artifactId>mybatis</artifactId>  
             <version>3.5.2</version>  
         </dependency>  
         <!--mybatis spring整合-->  
         <dependency>  
             <groupId>org.mybatis</groupId>  
             <artifactId>mybatis-spring</artifactId>  
             <version>2.0.3</version>  
         </dependency>

mybatis-generator插件

 <plugins>  
             <!--mybatis-generator插件-->  
             <plugin>  
                 <groupId>org.mybatis.generator</groupId>  
                 <artifactId>mybatis-generator-maven-plugin</artifactId>  
                 <version>1.3.2</version>  
                 <configuration>  
                     <verbose>true</verbose>  
                     <overwrite>true</overwrite>  
                 </configuration>  
             </plugin>  
         </plugins>

几个需要注意到的问题

mybatis-generator自动生成的代码 不能自动添加外键 所以要自己添加

Mapper.xml中的代码 自动生成的 参数都是bjid 需要改成 bj.ij

实现增删改查的两种方法

一是直接在Controller中 建立会话

 //   BjDao bjDao=new BjDao();  
 //   Reader reader;  
 //   {  
 //       try {  
 //           reader = Resources.getResourceAsReader("mybatis-config.xml");  
 //       } catch (IOException e) {  
 //           e.printStackTrace();  
 //       }  
 //   }  
 //  
 //   SqlSessionFactory sessionFactory=new SqlSessionFactoryBuilder().build(reader);  
 //   SqlSession session = sessionFactory.openSession();

二是建立SqlSessionFactoryUtil

 package cn.oneseek.student.util;  
 ​  
 import org.apache.ibatis.io.Resources;  
 import org.apache.ibatis.session.SqlSession;  
 import org.apache.ibatis.session.SqlSessionFactory;  
 import org.apache.ibatis.session.SqlSessionFactoryBuilder;  
 ​  
 import java.io.InputStream;  
 ​  
 public class SqlSessionFactoryUtil {  
 ​  
  private static SqlSessionFactory sqlSessionFactory;  
    
  private static SqlSessionFactory getSqlSessionFactory(){  
  if(sqlSessionFactory==null){  
  InputStream inputStream=null;  
  try{  
  inputStream= Resources.getResourceAsStream("mybatis-config.xml");  
  sqlSessionFactory=new SqlSessionFactoryBuilder().build(inputStream);  
  }catch(Exception e){  
  e.printStackTrace();  
  }  
  }  
  return sqlSessionFactory;  
  }  
    
  public static SqlSession openSession(){  
  return getSqlSessionFactory().openSession();  
  }  
 }

然后在Controller中

  private SqlSession session=SqlSessionFactoryUtil.openSession();  
  private BjMapper bjDao = session.getMapper(BjMapper.class);

其他包

引入datasource数据源的包，引入了2种方式，但本次项目中，我们使用的是alibaba的Druid DataSource。

 <!-- JDBC连接池 -->  
         <dependency>  
             <groupId>com.mchange</groupId>  
             <artifactId>c3p0</artifactId>  
             <version>0.9.5.2</version>  
         </dependency>  
         <!-- DruidDataSource，本工程的dataSource配置使用的Druid -->  
         <dependency>  
             <groupId>com.alibaba</groupId>  
             <artifactId>druid</artifactId>  
             <version>1.0.29</version>  
         </dependency>

pom.xml build添加

 <resources>  
             <!--表示把java目录下的有关xml文件,properties文件编译/打包的时候放在resource目录下-->  
             <resource>  
                 <directory>${basedir}/src/main/java</directory>  
                 <includes>  
                     <include>\*\*/\*.properties</include>  
                     <include>\*\*/\*.xml</include>  
                 </includes>  
             </resource>  
             <resource>  
                 <directory>${basedir}/src/main/resources</directory>  
             </resource>  
         </resources>

添加spring-jdbc

 <!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->  
 <dependency>  
     <groupId>org.springframework</groupId>  
     <artifactId>spring-jdbc</artifactId>  
     <version>5.2.0.RELEASE</version>  
 </dependency>  
 ​

问题

出现异常

  org.springframework.beans.factory.BeanDefinitionStoreException: Unexpected exception parsing XML document from class path resource [spring-mvc.xml]; nested exception is java.lang.NoSuchMethodError: org.springframework.beans.factory.xml.BeanDefinitionParserDelegate.getEnvironment()Lorg/springframework/core/env/Environment;  
 ​

原因：忘记在mybatis-config.xml中配置

 <typeAliases>  
         <typeAlias alias="Bj" type="cn.oneseek.student.entity.Bj"/>  
         <typeAlias alias="Student" type="cn.oneseek.student.entity.Student"/>  
     </typeAliases>  
 ​  
 <mappers>  
         <mapper resource="cn/oneseek/student/mapping/BjMapper.xml"/>  
         <mapper resource="cn/oneseek/student/mapping/StudentMapper.xml"/>  
     </mappers>

Controller 里数据操作的两种方法

一种是：

 session.selectList("cn.oneseek.student.dao.BjMapper.queryAll");

另一种

  private SqlSession session=SqlSessionFactoryUtil.openSession();  
  private BjMapper bjDao = session.getMapper(BjMapper.class);  
  List<Bj> bjList = bjDao.queryAll();

原生的mybatis-generator中没有一对一和一对多的关联关系的配置，所以反向生成代码的时候不会产生关联关系

手动添加一对多关系如下：

 package cn.oneseek.student.entity;  
 ​  
 import java.util.List;  
 ​  
 public class Bj {  
     private Integer id;  
 ​  
     private String bjname;  
       
     List<Student> students;  
 ​  
     public Integer getId() {  
         return id;  
    }  
 ​  
     public void setId(Integer id) {  
         this.id = id;  
    }  
 ​  
     public String getBjname() {  
         return bjname;  
    }  
 ​  
     public void setBjname(String bjname) {  
         this.bjname = bjname;  
    }  
     public List<Student> getStudents() {  
         return students;  
    }  
     public void setStudents(List<Student> students) {  
         this.students = students;  
    }  
     @Override  
     public String toString() {  
         return "Bj [id=" + id + ", bjname=" + bjname + "]";  
    }  
 ​  
 }

 package cn.oneseek.student.entity;  
 ​  
 public class Student {  
     private Integer id;  
 ​  
     private String sno;  
 ​  
     private String sname;  
 ​  
     private String sex;  
 ​  
 //   private Integer bjid;  
     private Bj bj;  
     public Integer getId() {  
         return id;  
    }  
 ​  
     public void setId(Integer id) {  
         this.id = id;  
    }  
 ​  
     public String getSno() {  
         return sno;  
    }  
 ​  
     public void setSno(String sno) {  
         this.sno = sno;  
    }  
 ​  
     public String getSname() {  
         return sname;  
    }  
 ​  
     public void setSname(String sname) {  
         this.sname = sname;  
    }  
 ​  
     public String getSex() {  
         return sex;  
    }  
 ​  
     public void setSex(String sex) {  
         this.sex = sex;  
    }  
 ​  
 //   public Integer getBjid() {  
 //       return bjid;  
 //   }  
 //  
 //   public void setBjid(Integer bjid) {  
 //       this.bjid = bjid;  
 //   }  
 ​  
     public Bj getBj() {  
         return bj;  
    }  
     public void setBj(Bj bj) {  
         this.bj = bj;  
    }  
     @Override  
     public String toString() {  
         return "Student [id=" + id + ", sno=" + sno + ", sname=" + sname + ", sex=" + sex + ", bj=" + bj + "]";  
    }  
 }

控制台乱码

VM options;

 -Dfile.encoding=UTF-8

问题一：学生列表不显示班级

出现原因：控制台打印发现bj=null

 [Student [id=1, sno=001, sname=����, sex=��, bj=null]]

未查询到班级

在Mapper里增加以下代码

 <resultMap type="cn.oneseek.student.entity.Student" id="StudentWithBj">  <association property="bj" column="bjid" select="cn.oneseek.student.dao.BjMapper.selectByPrimaryKey"/></resultMap>  
 <select id="queryAllWithBj"  resultMap="StudentWithBj">  
    select \* from student  
   </select>

问题二：增加学生时出错

原因:自动生成的entity没有bj.id属性

解决方法：

在StudentMapper.xml增加

 <insert id="add" parameterType="Student"  >  
    insert into student values(null,#{sno},#{sname},#{sex},#{bj.id})  
   </insert>

StudentMapping.java

     void add(Student student);

问题三：出现错误

 ### Cause: org.apache.ibatis.reflection.ReflectionException: Could not set property 'bjid' of 'class cn.oneseek.student.entity.Student' with value '1' Cause: org.apache.ibatis.reflection.ReflectionException: There is no setter for property named 'bjid' in 'class cn.oneseek.student.entity.Student'

原因：

Student实体没有bjid