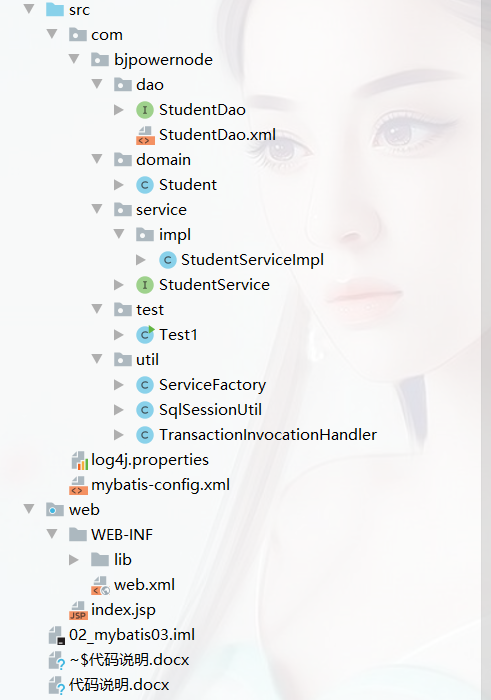
1、建项目

2、导入jar包

3、创建需要的包：更改了包名：com.cn.hm



4、在dao层删除impl，添加StudentMapper.xml，删除mapper包

（1）Studentdao接口中**方法的名字**要与Mapper映射文件中的**id的名字**必须相同；

（2）dao中有几个抽象方法，在mapper映射文件中就对应几套标签。一套标签对应一个方法。

（3）对于查询来讲，传递的参数类型和返回值类型必须跟接口中的参数类型和返回值类型保持一致。

5、业务层

（1）StudentServiceImpl

public class StudentServiceImpl implements StudentService {

*//StudentDao层的反射类型对象，语法约定*

private StudentDao studentDao = SqlSessionUtil.*getSession*().getMapper(StudentDao.class);

@Override

public Student getById(String id) {

Student student = studentDao.getById(id);

return student;

}

@Override

public void save(Student student) {

studentDao.save(student);

}

}

6、test测试

（1）测试根据id查询单条操作

public class Test1 {

public static void main(String[] args) {

*// StudentService studentService = new StudentServiceImpl();*

StudentService studentService = (StudentService) ServiceFactory.*getService*(new StudentServiceImpl());

*//测试添加操作*

*/\* Student student = new Student();*

*student.setId("A0006");*

*student.setName("cxk");*

*student.setAge(26);*

*studentService.save(student);\*/*

*//测试根据id查询单条操作*

Student student = studentService.getById("A0002");

System.*out*.println(student);

}

}

（2）查询所有记录

public class Test1 {

public static void main(String[] args) {

*// StudentService studentService = new StudentServiceImpl();*

StudentService studentService = (StudentService) ServiceFactory.*getService*(new StudentServiceImpl());

*//测试添加操作*

*/\* Student student = new Student();*

*student.setId("A0006");*

*student.setName("cxk");*

*student.setAge(26);*

*studentService.save(student);\*/*

*//测试根据id查询单条操作*

*/\*Student student = studentService.getById("A0002");*

*System.out.println(student);\*/*

*//**查询所有记录*

List<Student> studentList = studentService.getAll();*//控制层不干活，业务层拿数据*

for (Student student : studentList){

System.*out*.println(student);

}

}

}

7、在业务层创建getAll()

（1）逆向生成——在StudentService接口中生成getAll();（双击StudentService，连敲击两下shift，找到实现类）

public interface StudentService {

public Student getById(String id);

public void save(Student student);

List<Student> getAll();

}

（2）在StudentServiceImpl实现类重写方法getAll()

public class StudentServiceImpl implements StudentService {

*//StudentDao层的反射类型对象，语法约定*

private StudentDao studentDao = SqlSessionUtil.*getSession*().getMapper(StudentDao.class);

@Override

public Student getById(String id) {

Student student = studentDao.getById(id);

return student;

}

@Override

public void save(Student student) {

studentDao.save(student);

}

@Override

public List<Student> getAll() {

List<Student> studentList = studentDao.getAll();

return studentList;

}

}

8、开发dao层

（1）在StudentDao接口中生成getAll();（双击StudentDao，连敲击两下shift，找到xml文件）

public interface StudentDao {

public Student getById(String id);

public void save(Student student);

List<Student> getAll();

}

（2）在StudentMapper.xml文件中写sql语句

<mapper namespace="com.cn.hm.dao.StudentDao">

<select id="getAll" resultType="com.cn.hm.domain.Student">

select *\** from tbl\_student

</select>

<select id="getById" parameterType="java.lang.String" resultType="com.cn.hm.domain.Student">

select *\** from tbl\_student where id = #{id}

</select>

<insert id="save">

insert into tbl\_student(id,name,age) values(#{id},#{name},#{age})

</insert>

</mapper>