# 删除操作：

## 2、创建持久层

**public** **interface** BookDataBaseDao {

**public** List<Book> queryBookListByPage(Page page);

**public** **int** deleteBook(Integer id);

}

## 3、编写Mapper

<mapper namespace=*"com.hxzy.dao.BookDataBaseDao"*>

<!-- 对象参数传递到了语句中，id、username 和 password 属性将会被查找，然后将它们的值传入预处理语句的参数中 -->

<select id=*"queryBookListByPage"* resultType=*"com.hxzy.entity.Book"*

parameterType=*"com.hxzy.entity.Page"*>

select `id`,`name`,`author`,`publish`,left(publish\_date,10) as publishDate,`page`,`price`,`content`

from `book`

<if test=*"offset>=0 and limit!=0"*>

limit #{offset},#{limit}

</if>

</select>

<!-- 删除一本书 -->

<delete id=*"deleteBook"* parameterType=*"int"*>

DELETE FROM `book` WHERE id = #{id}

</delete>

</mapper>

## 4、编写service层接口

**public** **interface** BookDataBaseService {

**public** List<Book> queryBookListByPage(Page page);

**public** **int** deleteBook(Integer id);

}

## 5、实现service层接口

/\*\*

\* 删除指定ID的book

\*/

@Override

**public** **int** deleteBook(Integer id) {

**return** dao.deleteBook(id);

}

## 6、编写controller层

/\*\*

\* 删除数据

\* **@param** session

\* **@param** book

\* **@param** map

\* **@return**

\*/

@RequestMapping(value = "/bookDelete.do")

**public** String delete(HttpSession session, HttpServletRequest resq, ModelMap map) {

Object obj = session.getAttribute("loginUser");

**if** (obj == **null**) {

map.put("message", "登入已过期，请重新登入！");

//return "book\_db/index";

}

String[] ids = resq.getParameterValues("id");

**for** (String id : ids) {

**if** (service.deleteBook(**new** Integer(id)) == 1) {

System.***out***.println("数据删除成功：ID="+id);

**return** "{\"status\":\"success\",\"data\":\"数据删除成功\"}";

}

}

**return** "{\"status\":\"\",\"data\":\"\"}";

}

# 插入操作：

## 2、在BookDataBaseDao.java文件中增加方法申明：

**public** **int** insertBook(Book book);

## 3、在BookDataBaseMapper.xml文件中增加插入的SQL语句

<!-- 插入一本书 -->

<insert id=*"insertBook"* parameterType=*"com.hxzy.entity.Book"*>

insert into `book` ( `name`,`author`,`publish`,`publish\_date`,`page`,`price`,`content`

)

values

( #{name},#{author},#{publish},#{publishDate},#{page},#{price},#{content}

)

</insert>

## 4、在BookDataBaseService.java文件中增加方法申明

**public** **int** insertBook(Book book);

## 5、在BookDataBaseServiceImpl.java文件中增加实现方法

/\*\*

\* 插入一条数据

\*/

@Override

**public** **int** insertBook(Book book) {

**return** dao.insertBook(book);

}

## 6、在BookDataBaseController.java文件中增加插入方法

/\*\*

\* 插入一条数据

\* **@param** session

\* **@param** book

\* **@param** map

\* **@return**

\*/

@RequestMapping(value = "/bookInsert.do",produces="text/html;charset=UTF-8")

**public** String bookUpdate(HttpSession session, Book book, ModelMap map) {

Object obj = session.getAttribute("loginUser");

**if** (obj == **null**) {

map.put("message", "登入已过期，请重新登入！");

//return "book\_db/index";

}

**if**(service.insertBook(book) == 1){

System.***out***.println("数据增加成功："+book.toString());

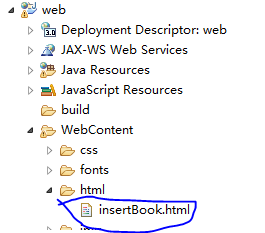
**return** "<h2>数据增加成功</h2>";

}

**return** "{\"status\":\"success\",\"data\":\"\"}";

}

## 修改前端访问地址：



如上图文件，打开



把action="/m/bookUpdate.do"改成：action="/web/bookInsert.do"