

1, 分页查询要显示的内容

1, 第几页

2, 总页数

3, 每条页数

4, 总条数

5, 当页的数据 (页面上数据较多, 类型不统一, 所以要使用泛型, 并且把pojo中的PageBean类型页改为泛型)

2, 页面上要显示数据, 就要从数据库中获取, 数据库中获取需要一个载体来放置, 就产生了pojo下的PageBean对象

```
1 private int page;//第几页
2 private int totalPage;//总页数
3 private int count;//每页条数
4 private int totalCount;//总条数
5 private List<T> list;//当页的数据
6 //添加set和get方法
```

3, 在servlet自定义页数的每页的条数 (可以再页面显示用于测试), 以便通过这两个数据得到其他的三个数据

```
1 int page = 1;
2 if (request.getParameter("page") != null) {
3     //用户自定义
4     page = Integer.parseInt(request.getParameter("page"));
5 }
6 int count = 3;
7 if (request.getParameter("count") != null) {
8     count = Integer.parseInt(request.getParameter("count"));
9 }
```

4, 在servlet中调用创建service。在service中创建接口和实现类, 通过方法调用dao层, 得到数据库的数据的条数和每页的内容

```

1 //servlet中的内容
2 UserService userService = new UserServiceImpl();
3 PageBean<User> pageBean = userService.selectAll(page, count);
4
5 //service接口中的方法
6 public interface UserService {
7     PageBean<User> selectAll(int page, int count);
8 }
9 //方法的实现类
10 public class UserServiceImpl implements UserService {
11     private UserDao userDao = new UserDaoImpl();
12     @Override
13     public PageBean<User> selectAll(int page, int count) {
14         PageBean<User> pageBean = new PageBean<>();
15         //将页数存进pagebean
16         pageBean.setPage(page);
17
18         pageBean.setCount(count);
19
20         List<User> userList = userDao.selectAll((page - 1) * count, count);
21
22         pageBean.setList(userList);
23
24         int totalCount = userDao.selectCount();
25         pageBean.setTotalCount(totalCount);
26
27         int totalPage = totalCount % count == 0 ? totalCount / count : totalCount / count + 1;
28         /*if (totalCount % count == 0) {
29             totalPage = totalCount / count;
30         } else {
31             totalPage = totalCount / count + 1;
32         }*/
33         pageBean.setTotalPage(totalPage);
34         return pageBean;
35     }
36 }
37

```

5, 要想获取数据就要从数据库中获取内容, 就在dao层写接口和实现类 (impl), 接口中要写两个方法, 分页查询获取的内容和获取数据总条数 (为了显示总共的页数) 的方法

```
1 //接口中的方法
2 public interface UserDao {
3     //分页查询
4     List<User> selectAll(int m, int n);
5     //获取总条数
6     int selectCount();
7 }
8
9 //方法的实现类
10 public class UserDaoImpl implements UserDao {
11     private QueryRunner queryRunner = JDBCUtils.getQueryRunner();
12     @Override
13     public List<User> selectAll(int m, int n) {
14         List<User> userList = null;
15         try {
16             userList = queryRunner.query("select * from user limit ?,?",
17 ew BeanListHandler<User>(User.class), m, n);
18         } catch (SQLException e) {
19             e.printStackTrace();
20         }
21         return userList;
22     }
23     @Override
24     public int selectCount() {
25         Map<String, Object> map = null;
26         try {
27             map = queryRunner.query("select count(*) count from user", n
1 ew MapHandler());
28         } catch (SQLException e) {
29             e.printStackTrace();
30         }
31         return Integer.parseInt(map.get("count") + "");
32     }
33 }
```

6, 得到了五个数据, 将数据存进域对象中, 并跳转到显示页面

```
1 request.setAttribute("pageBean", pageBean);
2 request.getRequestDispatcher("/student.jsp").forward(request, response);
```

7, 在jsp页面中获取域对象中的值, 遍历之后显示在网页上

```
1 <table border="1px">
2     <tr>
3         <th>ID</th>
4         <th>用户名</th>
5         <th>密码</th>
6     </tr>
7     <c:forEach items="${requestScope.pageBean.list}" var="user">
8         <tr>
9             <th>${
10 user.id}</th>
11             <th>${user.username}</th>
12             <th>${user.password}</th>
13         </tr>
14     </c:forEach>
15 </table>
16
17 //判断是否为第一页, 如果是就让上一页链接失效
18 <c:if test="${requestScope.pageBean.page != 1}">
19     <a href="${pageContext.request.contextPath}/userList?page=${requestS
20 cope.pageBean.page - 1}">上一页</a>
21 </c:if>
22
23 //判断是否在当前页, 如果是当前页就让当前页的链接失效
24 <c:forEach begin="1" end="${requestScope.pageBean.totalPage}" var="i">
25     <c:if test="${i == requestScope.pageBean.page}">
26         ${i}
27     </c:if>
28     <c:if test="${i != requestScope.pageBean.page}">
29         <a href="${pageContext.request.contextPath}/userList?
30 page=${i}">${i}</a>
31     </c:if>
32 </c:forEach>
```

```
31
32 //判断是否为最后一页，如果是就让下一页链接失效
33 <c:if test="${requestScope.pageBean.page != requestScope.pageBean.totalPage}">
34     <a href="${pageContext.request.contextPath}/userList?page=${requestScope.pageBean.page + 1}">下一页</a>
35 </c:if>
```