



```
HttpServletRequest request , HttpServletResponse response, Student student){
    .");
}
```

为什么要添加这个参数呢，那是因为我们最终会通过form表单输入用户名和姓名进行一个模糊查询，所以要在controller层做一个数据绑定。

```
@RequestMapping("/getUsers")
@ResponseBody
public Map<String, Object> getUsers(HttpServletRequest request , HttpServletResponse response, Student student){
    System.out.println("开始搜索用户...");

    int page = Integer.parseInt(request.getParameter("page")); //第几页
    int rows = Integer.parseInt(request.getParameter("rows")); //每页多少行

    System.out.println(student);

    Map<String, Object> pageBean = userService.getUsers(student, page, rows);

    return pageBean;
}
```

再看这两个东西，因为datagrid已经设置了分页，所以加载数据的时候就会默认传过来这两个参数，它会自己告诉controller本次请求要查询的是第几页，每页多少条数据？

```
int page = Integer.parseInt(request.getParameter("page")); //第几页
int rows = Integer.parseInt(request.getParameter("rows")); //每页多少行

System.out.println(student);

Map<String, Object> pageBean = userService.getUsers(student, page, rows);

return pageBean;
```

接下来，我们调用service层的方法，因为是分页查询，所以我们不仅仅要传入需要查询的条件，还要传入分页参数。

最终，我们走到了dao层的方法：

```
@Override
public Map<String, Object> getUsers(Student student, int page, int rows) {
    String sql = "select * from t_student where 1=1 ";
    List args = new ArrayList();

    if(student.getId() != 0){
        sql += " and id = ?";
        args.add(student.getId());
    }

    if(student.getUsername() != null && !"".equals(student.getUsername())){
        sql += " and username = ?";
        args.add(student.getUsername());
    }

    if(student.getName() != null && !"".equals(student.getName())){
        sql += " and name = ?";
        args.add(student.getName());
    }

    SimpleDao dao = new SimpleDao();
    Map<String, Object> pageBean = dao.queryForPage(sql, page, rows, args.toArray());

    return pageBean;
}
```

相信只要你有一定的java基础和jdbc基础，就一定能看懂这个方法做了些什么？没错，我们挨个判断查询条件，如果有传过来，就拼接到sql语句中去。最后，用queryForPage方法去查询一个分页对象过来。


现在，我们来测试一下，刷新index.html。

新增用户

编辑用户

删除用户

密码重置

用户名 姓名  搜索

用户列表

	<input type="checkbox"/> 用户名	密码	姓名	性别
1	<input type="checkbox"/> zsf	123	张三丰	男
2	<input type="checkbox"/> gj	123456	郭靖	男
3	<input type="checkbox"/> aaa	123	啊啊啊	男
4	<input type="checkbox"/> pikaqiu	123	皮卡丘	男
5	<input type="checkbox"/> a	1	2	男

10 ▾

◀ ◻ ▶

第 1 共1页

▶ ▶ ◻

显示1到5, 共5记录

可以看到，数据成功的载入了我们的datagrid!

接下来，我们给搜索按钮添加点击事件，达到条件查询的目的：

```
//搜索用户
function searchUser(){
    var username = $('#username_search').val(); //用户名
    var name = $('#nickname_search').val(); //姓名

    $('#grid0').datagrid('load',{
        username : username,
        name : name
    });
}
```

效果（截图不清晰，请下载后观看）：

用户名 姓名  搜索

用户列表

	<input type="checkbox"/> 用户名	密码	姓名	性别
1	<input type="checkbox"/> zsf	123	张三丰	男
2	<input type="checkbox"/> gj	123456	郭靖	男
3	<input type="checkbox"/> aaa	123	啊啊啊	男
4	<input type="checkbox"/> pikaqiu	123	皮卡丘	男
5	<input type="checkbox"/> a	1	2	男

10 ▾

◀ ◻ ▶

第 1 共1页

▶ ▶ ◻

显示1到5, 共5记录

那如果是模糊查询的话，又该如何实现呢？这个就作为一个小小的作业留给大家，在下一讲我们来揭开答案！

[我要下载源码](#)

您的支持是我写作的最大动力：

打赏不分好坏，一毛也是真爱



支付宝



微信