

今日内容

1. 综合练习

1. 简单功能

1. 列表查询
2. 登录
3. 添加
4. 删除
5. 修改

2. 复杂功能

1. 删除选中
2. 分页查询
 - * 好处:
 1. 减轻服务器内存的开销
 2. 提升用户体验
3. 复杂条件查询

2. 登录

1. 调整页面，加入验证码功能
2. 代码实现

姓名
李
籍贯
北京
邮箱

<input type="checkbox"/>	编号	姓名	性别	年龄	籍贯	QQ	邮箱
<input type="checkbox"/>	1	李四	女	15	北京	88888	ls@itcast.cn
<input type="checkbox"/>	2	张三	男	13	陕西	12345	zhangsan@itcast.cn
<input type="checkbox"/>	3	李四	女	15	北京	88888	ls@itcast.cn

«
1
2
3
4
5
6
7
8
»

共38条记录，共8页

map

name	李
address	北京
email	"/null

输出

PageBean

输入

表单参数

根据map中的value值动态生成sql

```
int totalCount;
```

```
List list;
```

PageBean

```
select count(*) from user where name like ? and address like ?
```

动态sql

```
select * from user where name like ? and address like ? limit ?, ?
```

1. 定义初始化sql select count(*) from user where 1 = 1

2. 遍历map, 判断其value 是否有值
sb.append("and key like ?")

add.jsp

添加联系人页面

姓名：

请输入姓名

性别：☒男 ☐女

年龄：

请输入年龄

籍贯：

陕西

QQ：

请输入QQ号码

Email：

请输入邮箱地址

提交

重置

返回

AddUserServlet

1. 设置编码
2. 获取所有数据
3. 封装对象
4. 调用service完成保存
5. 跳转到userListServlet, 再次查询

UserService

```
public void add(User user) {  
    dao.add(user)  
}
```

UserDao

```
public void add(  
    User user) {  
    //操作数  
    数据库保存  
}
```

添加联系人



DelUserServlet

1. 获取参数, id
2. 调用service删除
3. 跳转查询所有Servlet

UserService

```
public void deleteUser(String id) {  
    dao.delete(Integer.parseInt(id));  
}
```

UserDao

```
public void delete(int id) {  
    操作数据库删除  
}
```

删除

绑定参数: 这条记录的id

public void deleteUser(String id) {

dao.delete(Integer.parseInt(id));

}

public void delete(int id) {

操作数据库删除

}

update.jsp

UpdateUserServlet

UserService

findUserServlet

1. 获取用户id
2. 根据id查询用户信息 User
3. 将user对象存到request
4. 转发到update.jsp页面

修改联系人 回显信息

姓名：

请输入姓名

性别：☐男 ☐女

年龄：

请输入年龄

籍贯：

QQ：

请输入QQ号码

Email：

请输入邮箱地址

1. 设置编码
2. 获取表单数据 map
3. 封装User对象
4. 调用Service完成修改
5. 跳转到查询所有Servlet

void update(User user)

UserDao

```
void update(User user) {  
  
    update user set name=  
    ?, age = ?...  
    where id = ?  
}
```

修改

绑定用户id

DelSelectedServlet

1. 获取id数组
`getParameterValues()`
2. 调用service删除
3. 跳转到查询所有的Servlet

UserService

```
void deleteUsers(String[] ids) {  
    1. 遍历id  
    2. 调用dao删除方法  
}
```

UserDao

```
void delete(int id);
```

删除选中

1. 获取选中条目的id
2. 将这些id提交到
DelSelectedServlet

?id=1&id=2

<input type="checkbox"/>	编号	姓名	性别	年龄	籍贯	QQ	邮箱	操作
<input type="checkbox"/>	1	张三	男	13	陕西	12345	zhangsan@itcast.cn	<input type="button" value="修改"/> <input type="button" value="删除"/>
<input type="checkbox"/>	2	李四	女	15	北京	88888	ls@itcast.cn	<input type="button" value="修改"/> <input type="button" value="删除"/>
totalCount = select count(*) from user; totalPage = 提供每页显示条数给服务器 : rows list = select * from user limit ?, ? * 第一个?:开始查询的索引客户端 * 第二个?: rows 每页显示的条数								

↑ 输出 PageBean

↓ 输入

currentPage = 提供当前页码给服务器

开始查询的索引 = (currentPage - 1) * rows
服务器

```
int totalCount; // 总记录数  
int totalPage; // 总页码 = 总记录数 % 每页显示条数 == 0 ? 总记录数 / 每页显示条数 : 总记录数 / 每页显示条数 + 1  
List list; // 每页的数据 list集合  
int currentPage; // 当前页码  
int rows; // 每页显示的条数
```

PageBean 分页对象

1 : 0 , 5
2 : 5 , 5
3 : 10 , 5

FindUserByPageServlet

1. 接受请求参数 currentPage , rows
2. 调用Service查询 PageBean
3. 将PageBean存入request
4. 转发list.jsp展示

1

currentPage=1&rows=5

UserService

```
PageBean<User> findUserByPage(int currentPage ,  
int rows) {
```

```
    //1. 创建空的PageBean对象  
    //2. 设置当前页面属性和rows属性  
    //3. 调用dao查询totalCount,总记录数  
    dao.findTotalCount();  
    //4. start = (currentPage - 1)* rows  
    //5. 调用dao查询list集合  
    dao.findByPage(int start,int rows)  
    //6. 计算总页码
```

```
    //7. 返回PageBean对象
```

```
}
```

UserDao

```
//查询总记录数  
int findTotalCount()
```

```
//分页查询List  
List findByPage(int start,int rows)
```