



链滴

java 基础 |List 根据实体属性值搜索

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1575555938416>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



javaDEMO

本网站记录了最全的各种JavaDEMO,保证下载,复制就是可用的,包括基础的,集合的, spring的, Mybatis的等等各种,助力你从菜鸟到大牛,记得收藏哦~~

<https://www.javastudy.cloud>

List判断包含某个值

我们常常会用contains方法去判断一个list中是否包含某一串字符/数字,如下代码所示:

```
List<String> list = new ArrayList<>();  
list.add("java");  
list.add("study");  
list.add("cloud");  
boolean c = list.contains("java");  
System.out.println(c);
```

如果也不出意外的输出了true,如下图所示:

```
Debugger Console
/Library/Java/JavaVirtualMachines/jdk
Connected to the target VM, address:
true
Disconnected from the target VM, addr
Process finished with exit code 0
```

那如果List中不是简单的String/Integer等基本类型呢? 如下所示, 这时候要找list中是否包含 name="demo" 的实体,要怎么写?

```
public static void main(String[] args) {
    List<TestModel> list = new ArrayList<>();
    list.add(new TestModel("java","study"));
    list.add(new TestModel("demo","cloud"));
    list.add(new TestModel("demo","java技术大本营"));
    // 这时候要找list中是否包含 name="demo" 的实体,要怎么写?
    System.out.println();
}

static class TestModel {
    private String name;
    private String value;

    public TestModel(String name, String value) {
        this.name = name;
        this.value = value;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getValue() {
        return value;
    }

    public void setValue(String value) {
```

```
        this.value = value;
    }
}
```

List根据实体属性搜索

工具类如下所示,主要使用了泛型,反射

入参:

list,待查找的list

field: 要根据list中的哪个属性来查找,如上面的TestModel中的 name/value

value: 要查找的值为什么,如"demo", 查找list中name = demo的那个数据

clazz: 泛型类的描述信息,也就是TestModel

```
public static <T> List<T> containsByField(List<T> list ,String field,Object value,Class<T> clazz
{
    List<T> res = new ArrayList<>();

    try {
        // 通过反射获取对应的属性
        Field dbField = clazz.getDeclaredField(field);
        // 设置属性为可访问
        dbField.setAccessible(true);
        for (T t : list) {
            // 遍历list,获取到对应属性的值
            Object o = dbField.get(t);
            if(value.equals(o)){
                // 如值和预期想等的话,则加入到返回值中
                res.add(t);
            }
        }
    } catch (NoSuchFieldException e) {
        // 如果不存在这个字段,则直接返回空
        return res;
    } catch (IllegalAccessException e) {
        e.printStackTrace();
    }
    return res;
}
```

运行测试

```
public static void main(String[] args) {
    List<TestModel> list = new ArrayList<>();
    list.add(new TestModel("java", "study"));
    list.add(new TestModel("demo", "cloud"));
    list.add(new TestModel("demo", "java技术大本营"));
    List<TestModel> c = containsByField(list, "name", "demo", TestModel.class);
    System.out.println(c);
}
```

输出如下:

```
list.add(new TestModel( name: "java", value: "study"));
list.add(new TestModel( name: "demo", value: "cloud"));
list.add(new TestModel( name: "demo", value: "java技术大本营"));
List<TestModel> c = containsByField(list, field: "name", value: "demo");
System.out.println(c); c: "[com.platform.tools.jdk.ListUtils$TestModel@1c6b6478, com.platform.tools.jdk.ListUtils$TestModel@479]"
}
```

Variables

- args = {String[0]@473}
- list = {ArrayList@474} "[com.platform.tools.jdk.ListUtils\$TestModel@5577140b, com.platform.tools.jdk.ListUtils\$TestModel@1c6b6478, com.platform.tools.jdk.ListUtils\$TestModel@479]"
- c = {ArrayList@475} "[com.platform.tools.jdk.ListUtils\$TestModel@1c6b6478, com.platform.tools.jdk.ListUtils\$TestModel@479]"
- elementData = {Object[10]@478}
- Not showing null elements
- 0 = {ListUtils\$TestModel@479}
 - name = "demo"
 - value = "cloud"
- 1 = {ListUtils\$TestModel@480}
 - name = "demo"
 - value = "java技术大本营"
- size = 2
- modCount = 2

如上图所示,可以把两个实体给查找出来.

DEMO总评

这是一个小工具,今天工作中刚好用到,就顺手写了一个并整理了下,逻辑不麻烦,主要是反射的知识点,要好反射,实在是写工具类的一大利器! 加油!