



链滴

Lists.transform 的使用和采坑

作者: [jackssybin](#)

原文链接: <https://ld246.com/article/1564569411804>

来源网站: [链滴](#)

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



Lists.transform的使用

Lists.transform: 能够轻松的从一种类型的list转换为另一种类型的list。

```
Map<String,String> map = Maps.newHashMap();
map.put("a","testa");
map.put("b","test2");
map.put("c","test3");
Map<String,String> map2 = Maps.newHashMap();
map2.put("a","test3");
map2.put("b","testb");
map2.put("d","testc");
List<Map<String,String>> list2=Lists.newArrayList();
list2.add(map);
list2.add(map2);
List<String> list3=Lists.transform(list2,s->s.get("a"));
list3.forEach(s -> System.out.println(s));
```

结果如下

```
testa
testa2
```

还有采坑的，得需要注意下

```
List<User> userdbs = Lists.newArrayList(new User("zhangsan", "20"),
new User("lisi", "24"), new User("wangwu", "30"));

List<UserVo> userVos = Lists.transform(userdbs,xx ->{
return new UserVo(xx.getName(),xx.getId());
```

```
});  
List<UserVo> userVosNew =userdbs.stream().map(xx->{  
    return new UserVo(xx.getName(),xx.getId());  
}).collect(Collectors.toList());  
  
userdbs.forEach(xx->xx.setName(xx.getName()+"_test"));  
userVos.forEach(xx -> System.out.println(xx.getName()));  
System.out.println("====change lambda 方式");  
userVosNew.forEach(xx -> System.out.println(xx.getName()));
```

结果

```
zhangsan_test  
lisi_test  
wangwu_test  
====change lambda 方式  
zhangsan  
lisi  
wangwu
```

引用实体如下，未添加set,get方法

```
class UserVo{  
    private String id;  
    private String name;  
}  
class User{  
    private String id;  
    private String name;  
}
```

**对原数据集users的修改会直接影响到Lists.transform方法返回的结果userVos，
解决方案还是用 lambda 把方便快捷**

参考<https://blog.csdn.net/mnmlist/article/details/53870520>