BeanUtils组件

由于java对javaBean的操作很频繁，所以使用BeanUtils组件进行javaBean操作

从apache下载BeanUtils组件。然后项目中引入jar文件

使用BeadUtils组件

1. 引入commons-beanutils-1.8.3.jar核心包
2. 引入日志支持包commons-loggings-1.1.3.jar

**如果缺少日志jar文件，报错：**

java.lang.NoClassDefFoundError: org/apache/commons/logging/LogFactory

at org.apache.commons.beanutils.ConvertUtilsBean.<init>(ConvertUtilsBean.java:157)

at org.apache.commons.beanutils.BeanUtilsBean.<init>(BeanUtilsBean.java:117)

at org.apache.commons.beanutils.BeanUtilsBean$1.initialValue(BeanUtilsBean.java:68)

at

基本用法

1. 对象属性的拷贝

//这里基本等同于给对象赋值

BeanUtils.copyProperty(admin , "userName" , "jack");

BeanUtils.setProperty(admin , "age" , 18);

1. 对象的拷贝

注意，这里的newAdmin也是admin类的实例化对象，这里的拷贝是指把admin对象中的属性拷贝到newAdmin中

BeadUtils.copyProperties(newAdmin , admin);

1. Map数据拷贝到javaBean中 【注意，map中的key要和javabean的属性名称一致】

注意，这里的newAdmin也是admin类的实例化对象，这里的拷贝是指把Map对象中的属性拷贝到newAdmin中，Map对象中的键和值对应admin类的属性和值类型

BeanUtils.populate(adminMap , map);

1. 自定义类型转换器

在使用属性赋值的时候，经常涉及到各种类型转换，除了常规类型转换之外，要转特殊的或者自定义的数据类型要使用赋值转换器

下面以日期类型举例

1. 首先注册转换器类型

BeanUtils.register(

new Converter(){

//重写转换方法

public Object convert(Class type , Object value){

if(type == null){

return null ;

}

if( value == null || "".equals(value.toString().trim())){

return null;

}

//字符串转为日期

try{

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");

return sdf.Parse(value.toString());

}catch(ParseException e){

throw new RunTimeException(e);

}

}

} ， Date.class

);

2,调用转换方法

BeanUtils.copyProperty(admin , "birty" , birty); //这里进行属性赋值时会自动调用上面的转换方法