java bean拷贝操作的一个工具类。

spring（org.springframework.beans.BeanUtils）

和apache commons-beanutils（org.apache.commons.beanutils.BeanUtils）中分别存在一个BeanUtils。

a，b为对象

BeanUtils.copyProperties(a,b);

1. spring下的BeanUtils是a拷贝到b
2. apache下的是b拷贝到a

spring中实现copyProperties的方法很简单,就是对两个对象中相同名字的属性进行简单get/set，仅检查属性的可访问性。

package org.springframework.beans;

private static void copyProperties(Object source, Object target, Class<?> editable, String... ignoreProperties)

throws BeansException {

Assert.notNull(source, "Source must not be null");

Assert.notNull(target, "Target must not be null");

Class<?> actualEditable = target.getClass();

if (editable != null) {

if (!editable.isInstance(target)) {

throw new IllegalArgumentException("Target class [" + target.getClass().getName() +

"] not assignable to Editable class [" + editable.getName() + "]");

}

actualEditable = editable;

}

PropertyDescriptor[] targetPds = getPropertyDescriptors(actualEditable);

List<String> ignoreList = (ignoreProperties != null ? Arrays.asList(ignoreProperties) : null);

for (PropertyDescriptor targetPd : targetPds) {

Method writeMethod = targetPd.getWriteMethod();

if (writeMethod != null && (ignoreList == null || !ignoreList.contains(targetPd.getName()))) {

PropertyDescriptor sourcePd = getPropertyDescriptor(source.getClass(), targetPd.getName());

if (sourcePd != null) {

Method readMethod = sourcePd.getReadMethod();

if (readMethod != null &&

ClassUtils.isAssignable(writeMethod.getParameterTypes()[0], readMethod.getReturnType())) {

try {

if (!Modifier.isPublic(readMethod.getDeclaringClass().getModifiers())) {

readMethod.setAccessible(true);

}

Object value = readMethod.invoke(source);

if (!Modifier.isPublic(writeMethod.getDeclaringClass().getModifiers())) {

writeMethod.setAccessible(true);

}

writeMethod.invoke(target, value);

}

catch (Throwable ex) {

throw new FatalBeanException(

"Could not copy property '" + targetPd.getName() + "' from source to target", ex);

}

}

}

}

}

}

而commons-beanutils则施加了很多的检验，包括类型的转换，甚至还会检验对象所属的类的可访问性。

package org.apache.commons.beanutils;

public void copyProperties(Object dest, Object orig)

throws IllegalAccessException, InvocationTargetException {

// Validate existence of the specified beans

if (dest == null) {

throw new IllegalArgumentException

("No destination bean specified");

}

if (orig == null) {

throw new IllegalArgumentException("No origin bean specified");

}

if (log.isDebugEnabled()) {

log.debug("BeanUtils.copyProperties(" + dest + ", " +

orig + ")");

}

// Copy the properties, converting as necessary

if (orig instanceof DynaBean) {

DynaProperty origDescriptors[] =

((DynaBean) orig).getDynaClass().getDynaProperties();

for (int i = 0; i < origDescriptors.length; i++) {

String name = origDescriptors[i].getName();

if (getPropertyUtils().isWriteable(dest, name)) {

Object value = ((DynaBean) orig).get(name);

copyProperty(dest, name, value);

}

}

} else if (orig instanceof Map) {

Iterator names = ((Map) orig).keySet().iterator();

while (names.hasNext()) {

String name = (String) names.next();

if (getPropertyUtils().isWriteable(dest, name)) {

Object value = ((Map) orig).get(name);

copyProperty(dest, name, value);

}

}

} else /\* if (orig is a standard JavaBean) \*/ {

PropertyDescriptor origDescriptors[] =

getPropertyUtils().getPropertyDescriptors(orig);

for (int i = 0; i < origDescriptors.length; i++) {

String name = origDescriptors[i].getName();

if ("class".equals(name)) {

continue; // No point in trying to set an object's class

}

if (getPropertyUtils().isReadable(orig, name) &&

getPropertyUtils().isWriteable(dest, name)) {

try {

Object value =

getPropertyUtils().getSimpleProperty(orig, name);

copyProperty(dest, name, value);

} catch (NoSuchMethodException e) {

; // Should not happen

}

}

}

}

}

commons-beanutils中的转换不支持java.util.Date的。除了支持基本类型以及基本类型的数组之外，还支持java.sql.Date，java.sql.TimeStamp，java.io.File，javaio.URL这些类的对象，其余一概不支持。不建议使用。