Atitit. Web RMI jna pinvoke 互操作中的自动类型转换原理

[1.1. 参数类型转换 1](#_Toc32624)

[1.2. 返回值类型转换 2](#_Toc13781)

## 参数类型转换

/wxb\_site/src/com/attilax/reflect/MethodUtils.java

**public** **static** Object invokeMethod(Object object, String methodName,

Object... args) **throws** NoSuchMethodException,

IllegalAccessException, InvocationTargetException {

**if** (args == **null**) {

args = ArrayUtils.***EMPTY\_OBJECT\_ARRAY***;

}

**int** arguments = args.length;

Class<?>[] parameterTypes = **new** Class[arguments];

**for** (**int** i = 0; i < arguments; i++) {

Object param = args[i];

**if** (param == **null**)

parameterTypes[i] = Object.**class**;

**else**

parameterTypes[i] = args[i].getClass();

}

**try** {

**return** *invokeMethod*(object, methodName, args, parameterTypes);

} **catch** (NoSuchMethodException e) {

Method m1 = refx.*getMeth*(object.getClass(), methodName, args);

Object[] args\_convertTypeed=*convertType*(args,m1);

Object ret = m1.invoke(object, args\_convertTypeed);

**return** ret;

}

}

**public** **static** Method getMeth(Class<? **extends** Object> class1, String meth, Object[] params) {

Class<?>[] paramTypes =refx. *getParamTypes*(params);

Method m=*getMethByParamTypes*(class1,meth, paramTypes);

**if**(m==**null**)

m=*getMethByParams*(class1,meth,params);

**return** m;

}

**private** **static** Object[] convertType(Object[] args, Method m1) {

Class[] paramTypes=m1.getParameterTypes();

List li=Lists.*newArrayList*();

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

Object arg\_converted=*convertType*(args[i], paramTypes[i]);

li.add(arg\_converted);

}

**return** li.toArray();

}

**private** **static** Object convertType(Object object, Class paramTypes) {

**if**(paramTypes==Integer.**class** ||paramTypes==**int**.**class** )

**return** Integer.*parseInt*(object.toString());

**return** object;

}

## 返回值类型转换

/wxb\_site/src/com/attilax/wrmi/Wrmi.java

String ret = "";

**if** (invokeMethod\_ret **instanceof** String)

ret = (String) invokeMethod\_ret;

**if** (invokeMethod\_ret **instanceof** Integer || invokeMethod\_ret **instanceof** Double)

ret = invokeMethod\_ret.toString();

**else**

ret = core.*toJsonStrO88*(invokeMethod\_ret);

**return** ret;