Atitit.wrmi web rmi框架新特性

[1. V1d 新特性 1](#_Toc11375)

[1.1. 增加了精确参数 1](#_Toc17282)

[1.2. 增加了req参数，命名参数模式。。 1](#_Toc7382)

[1.3. 增加了globale 传递隐含参数req resp等 1](#_Toc5216)

[1.4. Cs bs两个版本的实现 1](#_Toc17825)

[2. V2 新特性 2](#_Toc17138)

[2.1. $req对象预定义参数的支持 2](#_Toc19890)

[2.2. 动态ioc IocXq214 2](#_Toc17292)

[2.3. end 4](#_Toc1310)

# V1d 新特性

## 增加了精确参数

## 增加了req参数，命名参数模式。。

增加了对req参数的处理

var mp=$("form").serialize();

mp=mp+"&$method=com.attilax.bet.BetService.bet&$callback=get\_posts\_callback&param1=$req";

## 增加了globale 传递隐含参数req resp等

## Cs bs两个版本的实现

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# V2 新特性

## $req对象预定义参数的支持

## 动态ioc IocXq214

**package** com.attilax.wrmi;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Map;

**import** javax.servlet.http.HttpServletRequest;

**import** org.apache.commons.lang3.reflect.ConstructorUtils;

**import** org.apache.commons.lang3.reflect.MethodUtils;

//import com.attilaax.encode.EncodeX; sould recomm q229 ati

**import** com.attilax.core;

**import** com.attilax.ioc.IocXq214;

**import** com.attilax.lang.Global;

**import** com.attilax.lang.ParamX;

**import** com.attilax.ref.refx;

**import** com.attilax.web.ReqX;

**import** com.google.common.collect.Maps;

/\*\*

\* $method=com.attilax.jsna.test.add

\* **@author** Administrator

\*

\*/

**public** **class** Wrmi {

/\*\*

\* def output raw type ..if list map object use json fmt

\*

\* **@param** req

\* **@return**

\*/

@SuppressWarnings("all")

**public** String exe(Map m) {

// Map m=Maps.newLinkedHashMap();

// if(req2 instanceof String) //cs envi

// m = new ParamX().urlParams2Map((String) req2);

//

Object[] params\_objArr = getParamsArr(m);

String meth = (String) m.get("$method");

String classname = refx.*getClassName*(meth);

String meth\_name = refx.*getMethodName*(meth);

Object o;

**try** {

// String apptype=System.getProperty("apptype");

o = IocXq214.*getBean*(classname);

Object invokeMethod\_ret = com.attilax.reflect.MethodUtils.*invokeMethod*(o, meth\_name,

params\_objArr);

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;

} **catch** (Throwable e) {

e.printStackTrace();

**return** core.*toJsonStrO88*(e);

}

}

**private** Object[] getParamsArr(Map m) {

List paras\_li = **new** ArrayList();

//m.get("param")

**if** (m.get("param") != **null**)

paras\_li.add(convertReqObjParams( m.get("param")));

**for** (**int** i = 0; i < 7; i++) {

String param\_index = "param" + String.*valueOf*(i);

**if** (m.get(param\_index) != **null**)

paras\_li.add(convertReqObjParams(m.get(param\_index)));

}

Object[] params\_objArr = paras\_li.toArray();

**return** params\_objArr;

}

**private** Object convertReqObjParams(Object object) {

**if**(object.toString().trim().equals("$req"))

**return** Global.*req*.get();

**return** object;

}

}

## end