**WSocket使用说明**

作者：旋律

版本：V1.2

Q Q ：535770212

Q 群：463603859

时间：2018-9-5

准备工作：

导入Jar包：WSocket.jar与json.jar.......等等包，如果与原有的jar包有冲突，那就不要导入冲突包就可以啦

服务器端：

服务类：WServerSocket

构造方法：

参数1：socket端口号（用于客户端连接）

参数2：最大连接数（用于控制客户端连接数）

参数3：扫描的注解类路径（用于客户端调用服务器端指定的注解类方法）

代码实例：

//初始化服务器端

WServerSocket wsocket = new WServerSocket(12306, 2,"com.test");

wsocket.start();

//是否开启WScoket控制台打印日志 true开启/false关闭

xl.sys.SysConfig.IsShowLogConsole = true;

//是否开启日志存储本地 true开启/false关闭

xl.sys.SysConfig.IsSaveLogToFile = false;

//日志储存路径：

Linux：/opt/

Windwos：C:\WSocket\Clent

//获得客户端长久连接(用于调用客户端指定注解类方法),

//需客户端连接入指定类型的连接才能获得数据

Map<String, Socket> map= wsocket.getClientSocket();

//客户端连接的key

String key = map.get("key");

//wsocket.method()为调用客户端方法,客户端的方法必须带有指定注解

//key用于调用key对应的客户端socket

//ClentTestClass为类注解上的值 如：@WSocket(value="ClentTestClass")

//sum为方法注解上的值 如：@WSocketMethod(value="sum")

//22和33为方法的两个参数,需与注解方法参数个数一致,类型均为String

//result为这个注解方法sum的返回值

String result= wsocket.method(key,"ClentTestClass","sum","22","33");

//结束指定的注解扫描客户端长连接

Wsocket.close(key);

客户端：

服务类：WClientSocket

构造方法：

/\*\*

\* @说明 ： 与服务器对接后仅用于服务器调用客户端注解类方法

\* @参数 ：@param ip 连接服务器socket地址

\* @参数 ：@param port 服务端口号

\* @参数 ：@param path 供给服务器调用的注解类包路径

\*\*/

public WClientSocket(String ip,int port,String path)

/\*\*

\* @说明 ： type=0 用于单次服务器请求 type=1 用于连续多次服务器请求

\* @参数 ：@param ip 连接服务器socket地址

\* @参数 ：@param port 服务端端口号

\* @参数 ：@param type 连接模式【0单次请求/1长连接】

\*\*/

public WClientSocket(String ip,int port,int type)

代码实例：

//单次请求服务器 参数三为0代表单次请求

WClientSocket socket = new WClientSocket("127.0.0.1", 12306,0);

//wsocket.method()为调用服务器端方法,服务器端的方法必须带有指定注解

//TestClass为类注解上的值 如：@WSocket(value="TestClass")

//sum为方法注解上的值 如：@WSocketMethod(value="sum")

//5和20为方法的两个参数,需与注解方法参数个数一致,类型均为String

//result为这个注解方法sum的返回值

String result = socket.method("TestClass", "sum", "5","20");

//多次请求服务器 参数三为1代表多次请求

WClientSocket socket = new WClientSocket("127.0.0.1", 12306,1);

//sum2方法无参数，所以后面不需要带有参数

String result1 = socket.method("TestClass", "sum2");

String result2 = socket.method("TestClass", "sum", "5","20");

//close()多次请求用完后需要关闭连接,单次无需手动关闭，会自动关闭

socket.close();

//建立服务器调用客户端方法连接(服务器端即可通过key调用指定的注解方法)

//参数三为：客户端扫描的注解类包路径（可供服务器端调用）

WClientSocket socket = new WClientSocket("127.0.0.1", 12306,"com.util");

//getKey()一般没用，只是用来看看，正常执行上方语句即可

System.out.println("key:"+socket.getKey());

注解类实例：

package com.test;

import xl.annotation.WSocket;

import xl.annotation.WSocketMethod;

//供调用的类

@WSocket(value="TestClass")

public class TestClass {

//供调用的sum方法

@WSocketMethod(value="sum")

public String sum(String str1,String str2){

int int1 =Integer.valueOf(str1);

int int2 =Integer.valueOf(str2);

return String.valueOf(int1+int2);

}

//供调用的sum2方法

@WSocketMethod(value="sum2")

public String sum2(){

int int1 =11;

int int2 =22;

return String.valueOf("11+22="+(int1+int2));

}

}

**实用技巧：**

1.灵活利用WUtil工具类可以实现调用的接口返回list或map集合

例如：

//方法

@WSocketMethod(value="test")

public String test(String str){

List result = new ArrayList();

result.add(str);

return WUtil.listToStr(result);

}

//调用

String result = socket.method("TestClass", "test", "testMethod");

List list = WUtil.strToList(result);

工具类：xl.util.WUtil

/\*\*

\* @说明 ： List集合转换成String (List可以存放实体对象，或其他类型)

\*\*/

public static String listToStr(List list)

/\*\*

\* @说明 ： String集合转换成List

\*\*/

public static List strToList(String str)

/\*\*

\* @说明 ： map转String (map可以存放实体对象，或其他类型)

\*\*/

public static String mapToStr(Map map)

/\*\*

\* @说明 ： String转map

\*\*/

public static Map strToMap(String str)