## 1、发布一个简单的终端

### 1、接口类HelloWS

|  |
| --- |
| @WebService  **public** **interface** HelloWS {  @WebMethod  **public** String sayHello(String name);  } |

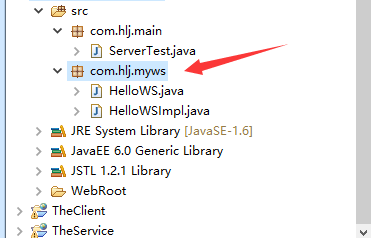
### 2、实现类

|  |
| --- |
| @WebService  **public** **class** HelloWSImpl **implements** HelloWS {  @Override  **public** String sayHello(String name) {  System.*out*.println("server sayHello()"+name);  **return** "Hello " +name;  }  } |

### 3、发布终端

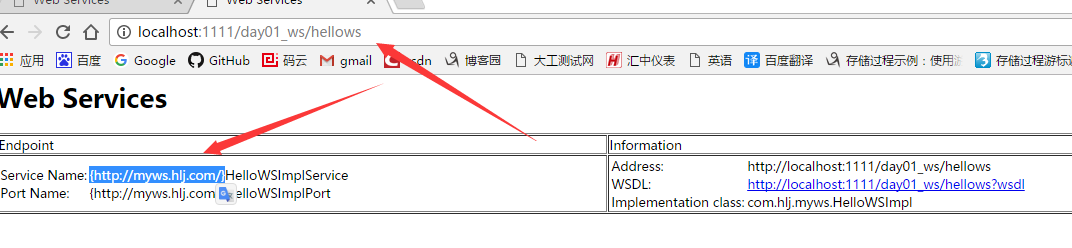
|  |
| --- |
| **public** **class** ServerTest {  **public** **static** **void** main(String[] args) {  String address = "http://localhost:1111/day01\_ws/hellows";    Endpoint.*publish*(address , **new** HelloWSImpl());  System.*out*.println("发布webservice成功!");  }  } |

### 4、项目结构如下



### 5、发布终端，会发现类似于targetNamespace的东西产生了，它其实就是我们的项目中接口所在包的名字

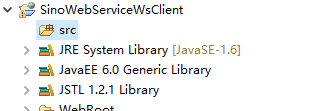
|  |
| --- |
| **public** **class** ServerTest {  **public** **static** **void** main(String[] args) {  String address = "http://localhost:1111/day01\_ws/hellows";    Endpoint.*publish*(address , **new** HelloWSImpl());  System.*out*.println("发布webservice成功!");  }  } |



|  |  |
| --- | --- |
| Service Name: | {http://myws.hlj.com/}HelloWSImplService |
| Port Name: | {http://myws.hlj.com/}HelloWSImplPort |

# 2、根据1中建立一个客户端web项目

## 1、项目建立好后

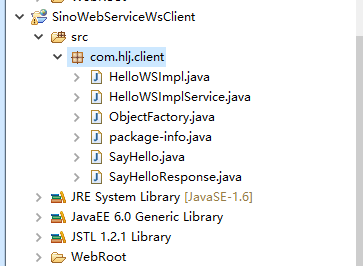


## 2、利用命令生成客户端代码，并建立包com.hlj.client

wsimport -s E:\\WorkSpace\\JspWorkSpace\\MyeclipseWebService\\SinoWebServiceWsClient\\src -p com.hlj.client -keep http://localhost:1111/day01\_ws/hellows?wsdl



## 3、刷新项目

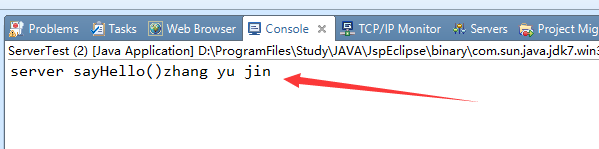


## 4、客户端建立测试类调用接口

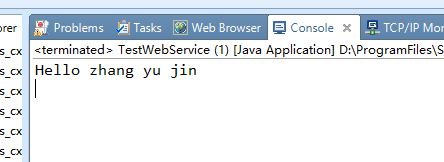
|  |
| --- |
| **public** **class** TestWebService {    **public** **static** **void** main(String[] args) {  HelloWSImpl hello = **new** HelloWSImplService().getHelloWSImplPort();    String name = hello.sayHello("zhang yu jin ");  System.*out*.println(name);  }  } |

## 5、控制台打印如下

### 服务的控制台

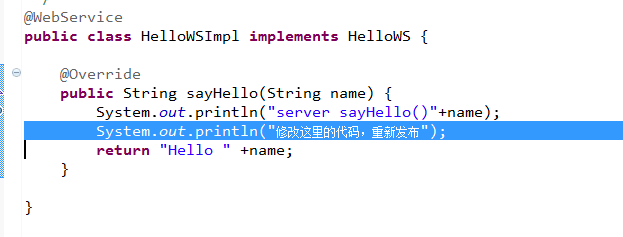


### 客户端控制台



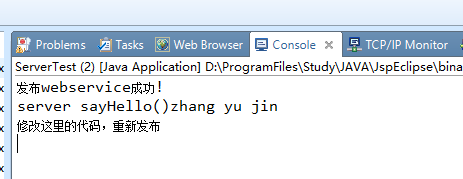
## 6、直接修改服务的接口中的实现代码重新发布，但是在客户端不重新获取wsdl文件，成功

### 1、服务中添加代码

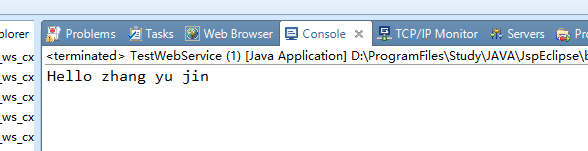


### 2、测试客户端重新调用，控制台如下

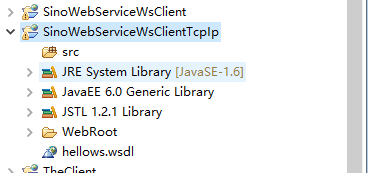
#### 1、服务端控制台



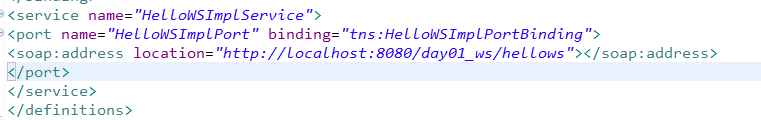
#### 2、客户端控制台



# 3、配置eclipse用来，监听请求，与代码无关，（重新建立一个项目）我这里的如下



## 1、复制wsdl文件，并修改改为8080（一共两处，修改最下面的）（这里的ip因为是本机测试所以才是localhost，实际项目中为其他的ip）

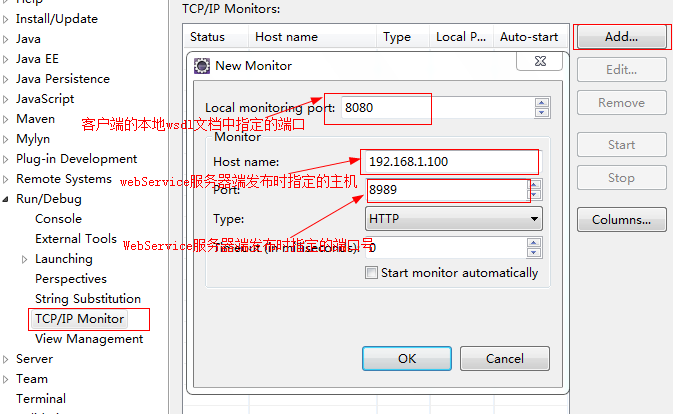


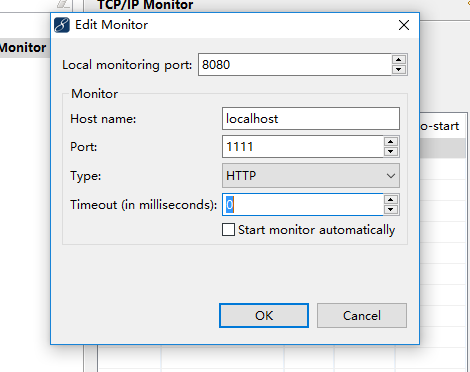
### <http://localhost:1111/day01_ws/hellows?wsdl>

## 2、cmd窗口中拖动这个wsdl文件创建



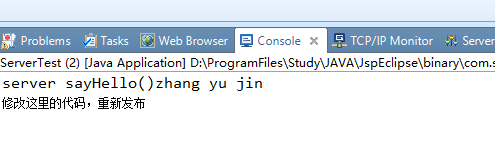
## 3、因为我们上面修改了端口，肯定是不行的，所以这里配置tcp/ip，端口转发器，用来监听这个端口，启动这个tcp/ip



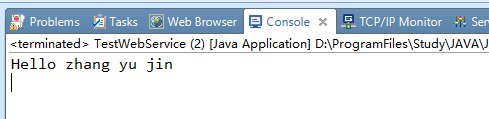


## 4、测试，客户端开始运行

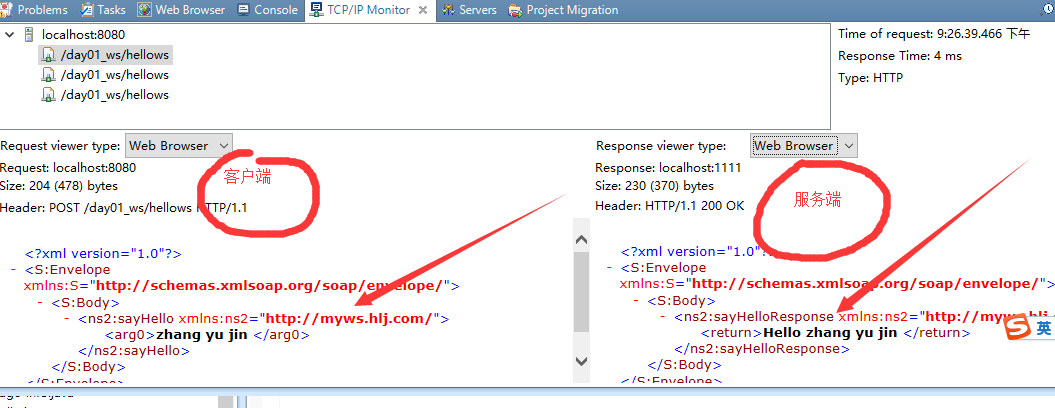
### 1、服务端控制台



### 2、客户端控制台



### 3、tcp/ip控制台，可以观察到客户端和服务端的东西



# 4、debug项目，只要服务端debug，客户端调用的时候就可以debug调试了。

