[**Spring基于HTTP的远程方法调用**](https://www.oschina.net/code/snippet_1386372_48256)

|  |
| --- |
| Spring基于HTTP的远程方法调用  1. [代码]服务器spring applicationContext.xml  <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.springframework.org/schema/p"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.1.xsd">    <bean name="sayHelloService" **class**="springrmi.service.HelloServiceImpl"></bean>  <bean name="helloservice"  **class**="org.springframework.remoting.httpinvoker.HttpInvokerServiceExporter">  <property name="serviceInterface" value="springrmi.service.HelloService"></property>  <property name="service" ref="sayHelloService"></property>  </bean>    </beans>  2. [代码]服务器web.xml  <?xml version="1.0" encoding="UTF-8"?>  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  id="WebApp\_ID" version="3.0">  <display-name>springrmi</display-name>  <welcome-file-list>  <welcome-file>index.html</welcome-file>  <welcome-file>index.htm</welcome-file>  <welcome-file>index.jsp</welcome-file>  <welcome-file>**default**.html</welcome-file>  <welcome-file>**default**.htm</welcome-file>  <welcome-file>**default**.jsp</welcome-file>  </welcome-file-list>  <listener>  <listener-**class**>org.springframework.web.context.ContextLoaderListener</listener-**class**>  </listener>  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:applicationContext.xml</param-value>  </context-param>  <servlet>  <servlet-name>helloservice</servlet-name>  <servlet-**class**>org.springframework.web.context.support.HttpRequestHandlerServlet</servlet-**class**>  </servlet>  <servlet-mapping>  <servlet-name>helloservice</servlet-name>  <url-pattern>/helloservice</url-pattern>  </servlet-mapping>    </web-app>  3. [代码]服务器服务接口  **package** springrmi.service;    **public** **interface** HelloService {    String sayHello();  }  4. [代码]服务器服务实现类  **package** springrmi.service;    **public** **class** HelloServiceImpl **implements** HelloService {    @Override  **public** String sayHello() {    **return** "Hello";  }    }  5. [代码]客户端spring applicationContext.xml  <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.springframework.org/schema/p"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.1.xsd">    <bean name="helloservice"  **class**="org.springframework.remoting.httpinvoker.HttpInvokerProxyFactoryBean">  <property name="serviceUrl" value="http://localhost:8080/springrmi/helloservice"></property>  <property name="serviceInterface" value="springrmiclient.HelloService"></property>  </bean>    </beans>  6. [代码]客户端服务接口  **package** springrmiclient;    **public** **interface** HelloService {    String sayHello();  }  7. [代码]客户端依赖spring配置文件测试代码  **package** springrmiclient;    **import** org.springframework.context.support.ClassPathXmlApplicationContext;    **public** **class** ClientTest {    **public** **static** **void** main(String[] args) {    ClassPathXmlApplicationContext cac = **new** ClassPathXmlApplicationContext("classpath\*:/applicationContext.xml");  HelloService hello = (HelloService) cac.getBean("helloservice");  System.***out***.println(hello.sayHello());  }    }  8. [代码]客户端不依赖spring配置文件测试代码  **import** org.springframework.remoting.httpinvoker.HttpInvokerProxyFactoryBean;  **import** org.springframework.remoting.httpinvoker.HttpInvokerRequestExecutor;  **import** org.springframework.remoting.support.RemoteInvocation;  **import** org.springframework.remoting.support.RemoteInvocationResult;    **public** **class** SpringHttpClientTest {    **public** **static** **void** main(String[] args) {    HttpInvokerProxyFactoryBean proxyFactory = **new** HttpInvokerProxyFactoryBean();  proxyFactory.setServiceUrl("http://localhost:8080/springrmi/helloservice");  proxyFactory.setServiceInterface(HelloService.**class**);  proxyFactory.afterPropertiesSet();    // 1  HelloService helloservice = (HelloService) proxyFactory.getObject();  System.***out***.println(helloservice.sayHello());    // 2  RemoteInvocation remoteInvocation = **new** RemoteInvocation();  remoteInvocation.setArguments(**null**);  remoteInvocation.setMethodName("sayHello");  remoteInvocation.setParameterTypes(**null**);  HttpInvokerRequestExecutor httpExecutor = proxyFactory.getHttpInvokerRequestExecutor();  **try** {  RemoteInvocationResult result = httpExecutor.executeRequest(proxyFactory, remoteInvocation);  System.***out***.println(result.getValue());    } **catch** (Exception e) {    e.printStackTrace();  }  } |