**[阿里巴巴dubbo处理文件上传下载](http://dreamoftch.iteye.com/blog/1987922)**

**博客分类：**

* [dubbo](http://dreamoftch.iteye.com/category/294527)
* [hessian](http://dreamoftch.iteye.com/category/263236)

[dubbo](http://www.iteye.com/blogs/tag/dubbo)[hessian](http://www.iteye.com/blogs/tag/hessian)[inputstream](http://www.iteye.com/blogs/tag/inputstream)[upload](http://www.iteye.com/blogs/tag/upload)[download](http://www.iteye.com/blogs/tag/download)

**注：使用InputStream 作为方法参数的时候，需要将该参数作为方法的最后一个参数，否则会有问题**

dubbo和hessian的maven依赖：

**Java代码  [收藏代码](javascript:void())**

1. <dependency>
2. <groupId>com.alibaba</groupId>
3. <artifactId>dubbo</artifactId>
4. <version>2.5.3</version>
5. </dependency>
6. <dependency>
7. <groupId>com.caucho</groupId>
8. <artifactId>hessian</artifactId>
9. <version>4.0.7</version>
10. </dependency>

服务提供者（项目名称：provider）

首先是web.xml配置（使用spring）：

**Xml代码  [收藏代码](javascript:void())**

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<web-app** version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5. http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"**>**
6. **<context-param>**
7. **<param-name>**contextConfigLocation**</param-name>**
8. **<param-value>**classpath\*:applicationContext.xml**</param-value>**
9. **</context-param>**
10. **<listener>**
11. **<listener-class>**org.springframework.web.context.ContextLoaderListener**</listener-class>**
12. **</listener>**
13. **<servlet>**
14. **<servlet-name>**dubbo**</servlet-name>**
15. **<servlet-class>**com.alibaba.dubbo.remoting.http.servlet.DispatcherServlet**</servlet-class>**
16. **<load-on-startup>**1**</load-on-startup>**
17. **</servlet>**
18. **<servlet-mapping>**
19. **<servlet-name>**dubbo**</servlet-name>**
20. **<url-pattern>**/\***</url-pattern>**
21. **</servlet-mapping>**
23. **</web-app>**

然后是服务接口：

**Java代码  [收藏代码](javascript:void())**

1. **package** com.tch.test.dubbo.service;

4. **public** **interface** DemoService {
6. String sayHello(String name);
7. /\*\*
8. \*  Created on:             2013-12-11
9. \* <p>Discription:            下载</p>
10. \* @return              InputStream
11. \*/
12. **public** InputStream download(String path);
13. /\*\*
14. \* 处理上传
15. \* @param in
16. \* @return
17. \*/
18. String upload(String destPath,InputStream in);
19. }

服务实现类：

**Java代码  [收藏代码](javascript:void())**

1. **package** com.tch.test.dubbo.service;

4. **import** java.io.FileInputStream;
5. **import** java.io.InputStream;
7. **public** **class** DemoServiceImpl **implements** DemoService {
8. @Override
9. **public** InputStream download(String path) {
10. System.out.println(path+"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");
11. **try** {
12. InputStream inputStream = **new** FileInputStream(path);
13. **return** inputStream;
14. } **catch** (Exception e) {
15. e.printStackTrace();
16. }
17. **return** **null**;
18. }
19. @Override
20. **public** String sayHello(String name) {
21. **return** "Hello " + name;
22. }
23. @Override
24. **public** String upload(String destPath,InputStream in) {
25. **try** {
26. //FileOutputStream out = new FileOutputStream("E:\\temp\\a.js");
27. **int** n = -1;
28. **byte**[] b = **new** **byte**[10240];
29. **while**((n=in.read(b)) != -1){
30. System.out.println(**new** String(b,0,n,"utf-8"));
31. //out.write(b, 0, n);
32. }
33. //out.close();
34. //out.flush();
35. in.close();
36. } **catch** (Exception e) {
37. e.printStackTrace();
38. }
39. **return** "upload complete !";
40. }
42. }

最重要的applicationContext.xml ：

参考：[dubbo hessian协议](http://code.alibabatech.com/wiki/display/dubbo/Hessian+Protocol)

**Xml代码  [收藏代码](javascript:void())**

1. **<beans** xmlns="http://www.springframework.org/schema/beans"
2. xmlns:dubbo="http://code.alibabatech.com/schema/dubbo" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3. xmlns:context="http://www.springframework.org/schema/context" xmlns:tx="http://www.springframework.org/schema/tx"
4. xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
5. http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-2.5.xsd
6. http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd
7. http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd"**>**
8. <!-- 提供方应用信息，用于计算依赖关系 -->
9. **<dubbo:application** name="hello-world-app"  **/>**
10. <!-- 使用multicast广播注册中心暴露服务地址 -->
11. **<dubbo:registry** address="multicast://224.5.6.7:1234" **/>**
12. <!-- 用hessian协议在8080(<span style="font-family: Arial, Helvetica, FreeSans, sans-serif; font-size: 13px; line-height: 17.328125px; white-space: normal; background-color: #ffffff;">servlet容器的端口</span>)端口暴露服务，contextpath是项目名称，server这里是使用的web.xml里面配置的servlet -->
13. **<dubbo:protocol** name="hessian" port="8080" contextpath="dubbo" server="servlet"**/>**
14. <!-- 声明需要暴露的服务接口 -->
15. **<dubbo:service** interface="com.tch.test.dubbo.service.DemoService" ref="demoService" **/>**
16. <!-- 和本地bean一样实现服务 -->
17. **<bean** id="demoService" class="com.tch.test.dubbo.service.DemoServiceImpl" **/>**
19. **</beans>**

启动项目之后，就同时启动了服务。。。

下面是 consumer ：

首先是服务接口：

**Java代码  [收藏代码](javascript:void())**

1. **package** com.tch.test.dubbo.service;

4. **public** **interface** DemoService {
6. String sayHello(String name);
7. /\*\*
8. \*  Created on:             2013-12-11
9. \* <p>Discription:            下载</p>
10. \* @return              InputStream
11. \*/
12. **public** InputStream download(String path);
13. /\*\*
14. \* 处理上传
15. \* @param in
16. \* @return
17. \*/
18. String upload(String destPath,InputStream in);
19. }

然后是applicationContext.xml ：

**Java代码  [收藏代码](javascript:void())**

1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
4. xsi:schemaLocation="http://www.springframework.org/schema/beans
5. http://www.springframework.org/schema/beans/spring-beans.xsd
6. http://code.alibabatech.com/schema/dubbo
7. http://code.alibabatech.com/schema/dubbo/dubbo.xsd
8. ">
9. <!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->
10. <dubbo:application name="consumer-of-helloworld-app" />
12. <!-- 使用multicast广播注册中心暴露发现服务地址 -->
13. <dubbo:registry address="multicast://224.5.6.7:1234" />
15. <!-- 生成远程服务代理，可以和本地bean一样使用demoService -->
16. <dubbo:reference id="demoService"
17. **interface**="com.tch.test.dubbo.service.DemoService" />
19. </beans>

好了，开始调用provider提供的服务：

**Java代码  [收藏代码](javascript:void())**

1. **package** com.alibaba.dubbo;
3. **import** java.io.FileInputStream;
4. **import** java.io.FileNotFoundException;
5. **import** java.io.InputStream;
7. **import** org.springframework.context.support.ClassPathXmlApplicationContext;
9. **import** com.alibaba.dubbo.demo.DemoService;
10. **public** **class** Consumer {
11. **public** **static** **void** main(String[] args) **throws** Exception {
12. ClassPathXmlApplicationContext context = **new** ClassPathXmlApplicationContext(**new** String[] {"applicationContext.xml"});
13. context.start();
14. DemoService demoService = (DemoService)context.getBean("demoService"); // 获取远程服务代理
15. sayHello(demoService);  //sayHello
16. upload(demoService);    //upload
17. download(demoService); //download
18. context.close();
19. }
20. //调用sayHello
21. **public** **static** **void** sayHello(DemoService demoService){
22. String result = demoService.sayHello(" dubbo "); // 执行远程方法
23. System.out.println(result); // 显示调用结果
24. }
25. //调用upload
26. **public** **static** **void** upload(DemoService demoService) **throws** FileNotFoundException{
27. String result = demoService.upload("E:\\temp\\a.js",**new** FileInputStream("D:\\Program files\\apache-maven-3.1.0\\repository\\com\\caucho\\hessian\\4.0.7\\hessian-4.0.7.pom")); // 执行远程方法
28. System.out.println(result); // 显示调用结果
29. }
31. //调用upload
32. **public** **static** **void** download(DemoService demoService) **throws** Exception{
33. String srcPath = "D:\\Program files\\apache-maven-3.1.0\\repository\\com\\caucho\\hessian\\4.0.7\\hessian-4.0.7.pom";
34. InputStream inputStream = demoService.download(srcPath); // 执行远程方法
35. **byte** b[] = **new** **byte**[1024];
36. **int** n;
37. **try** {
38. **while** ((n = inputStream.read(b)) != -1) {
39. System.out.print(**new** String(b,0,n,"utf-8"));
40. }
41. } **catch** (Exception e) {
42. e.printStackTrace();
43. }**finally**{
44. inputStream.close();
45. }
46. }
47. }

 如果调用到download方法的时候，出现**java.io.IOException: stream is closed**

 则此时应该是jar包的问题，貌似这是hessian的一个bug ： [关于该bug](http://bugs.caucho.com/view.php?id=3655)

解决方法下载hessian.jar的源码包，修改 com.caucho.hessian.client.HessianProxy ：

在   Object value = in.readReply(method.getReturnType()); 这一句话之后加上下面代码：

**Java代码  [收藏代码](javascript:void())**

1. **if** (value **instanceof** InputStream) {
2. value = **new** ResultInputStream(conn, is, in, (InputStream) value);
3. is = **null**;
4. conn = **null**;
5. }

然后重新打包，使用这个新的jar包就没问题了。。。

然后，就可以看到sayHello、下载、上传成功的结果了。。。

 最后提供一个我自己修改过之后的hessian.jar

* [hessian-4.0.7.jar](http://dl.iteye.com/topics/download/4ac85b78-4d11-3703-a80e-ab3bb7d8658b) (398.4 KB)
* 下载次数: 163