# dubbo分享8--参数回调

 参数回调方式与调用本地callback或listener相同，只需要在Spring的配置文件中声明哪个参数是callback类型即可，Dubbo将基于长连接生成反向代理，这样就可以从服务器端调用客户端逻辑。

1、服务提供端及客户端共享代码

|  |  |
| --- | --- |
| 1  2  3  4  5 | package  com.alibaba.dubbo.demo;    public  interface  CallbackService {       void  addListener(String key, CallbackListener listener);  } |

|  |  |
| --- | --- |
| 1  2  3  4  5 | package  com.alibaba.dubbo.demo;    public  interface  CallbackListener {       void  changed(String msg);  } |

2、客户端代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | import  org.springframework.context.support.ClassPathXmlApplicationContext;    import  com.alibaba.dubbo.demo.CallbackListener;  import  com.alibaba.dubbo.demo.CallbackService;    public  class  Consumer {         /\*\*        \* @param args        \* @throws Exception        \*/       public  static  void  main(String[] args)  throws  Exception {           ClassPathXmlApplicationContext context =  new  ClassPathXmlApplicationContext(                   new  String[] {  "[classpath:consumer.xml](http://classpathconsumer.xml)"  });           context.start();             // 参数回调           CallbackService callbackService = (CallbackService) context.getBean( "callbackService" );             callbackService.addListener( "hanshubo" ,  new  CallbackListener() {               @Override               public  void  changed(String msg) {                   System.out.println( "callback1:"  + msg);               }           });             callbackService.addListener( "hanyiyi" ,  new  CallbackListener() {               @Override               public  void  changed(String msg) {                   System.out.println( "callback2:"  + msg);               }           });             while  ( true ) {               Thread.sleep( 1000 );           }       }  } |

 3、客户端配置文件

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | <?xml version= "1.0"  encoding= "UTF-8" ?>  <beans xmlns= "<http://www.springframework.org/schema/beans>"  [xmlns:xsi=](http://xmlnsxsi=) "<http://www.w3.org/2001/XMLSchema-instance>"  [xmlns:dubbo=](http://xmlnsdubbo=) "<http://code.alibabatech.com/schema/dubbo>"       xsi:schemaLocation="http: //[www.springframework.org/schema/beans](http://www.springframework.org/schema/beans)       http: //[www.springframework.org/schema/beans/spring-beans.xsd](http://www.springframework.org/schema/beans/spring-beans.xsd)       http: //[code.alibabatech.com/schema/dubbo](http://code.alibabatech.com/schema/dubbo)       http: //[code.alibabatech.com/schema/dubbo/dubbo.xsd](http://code.alibabatech.com/schema/dubbo/dubbo.xsd)">         <!-- 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 -->       <[dubbo:application](http://dubboapplication) name= "consumer-of-helloworld-app"   />         <!-- 使用zookeeper注册中心暴露发现服务地址 -->       <[dubbo:registry](http://dubboregistry) address= "<zookeeper://127.0.0.1:2181>"  />         <!-- 生成远程服务代理 -->       <[dubbo:reference](http://dubboreference) id= "demoService"  interface = "com.alibaba.dubbo.demo.DemoService"  retries= "2"  />         <!-- 生成远程服务代理 -->       <[dubbo:reference](http://dubboreference) id= "validationService"  interface = "com.alibaba.dubbo.demo.ValidationService"            retries= "2"  validation= "true"             />         <!-- 生成远程服务代理 -->       <[dubbo:reference](http://dubboreference) id= "demoService2"  interface = "com.alibaba.dubbo.demo.DemoService2"  async= "true"  />         <[dubbo:reference](http://dubboreference) id= "callbackService"  interface = "com.alibaba.dubbo.demo.CallbackService"  />    </beans> |

4、服务提供端代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | package  com.alibaba.dubbo.demo.provider;    import  java.text.SimpleDateFormat;  import  java.util.Date;  import  java.util.Map;  import  java.util.concurrent.ConcurrentHashMap;    import  com.alibaba.dubbo.demo.CallbackListener;  import  com.alibaba.dubbo.demo.CallbackService;    public  class  CallbackServiceImpl  implements  CallbackService {         private  final  Map<String, CallbackListener> listeners =  new  ConcurrentHashMap<String, CallbackListener>();         public  CallbackServiceImpl() {           Thread t =  new  Thread( new  Runnable() {               @Override               public  void  run() {                   while  ( true ) {                       try  {                           for  (Map.Entry<String, CallbackListener> entry : listeners.entrySet()) {                               try  {                                   entry.getValue().changed(getChanged(entry.getKey()));                               }  catch  (Throwable t) {                                   listeners.remove(entry.getKey());                               }                           }                           Thread.sleep( 5000 );  // 定时触发变更通知                       }  catch  (Throwable t) {  // 防御容错                           t.printStackTrace();                       }                   }               }           });           t.setDaemon( true );           t.start();       }         public  void  addListener(String key, CallbackListener listener) {           listeners.put(key, listener);           listener.changed(getChanged(key));  // 发送变更通知       }         private  String getChanged(String key) {           return  "Changed: "  +  new  SimpleDateFormat( "yyyy-MM-dd HH:mm:ss" ).format( new  Date());       }    } |

 5、服务提供端配置文件

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | <?xml version= "1.0"  encoding= "UTF-8" ?>  <beans xmlns= "<http://www.springframework.org/schema/beans>"  [xmlns:xsi=](http://xmlnsxsi=) "<http://www.w3.org/2001/XMLSchema-instance>"  [xmlns:dubbo=](http://xmlnsdubbo=) "<http://code.alibabatech.com/schema/dubbo>"       xsi:schemaLocation="http: //[www.springframework.org/schema/beans](http://www.springframework.org/schema/beans)       http: //[www.springframework.org/schema/beans/spring-beans.xsd](http://www.springframework.org/schema/beans/spring-beans.xsd)       http: //[code.alibabatech.com/schema/dubbo](http://code.alibabatech.com/schema/dubbo)       http: //[code.alibabatech.com/schema/dubbo/dubbo.xsd](http://code.alibabatech.com/schema/dubbo/dubbo.xsd)">         <!-- 提供方应用信息，用于计算依赖关系 -->       <[dubbo:application](http://dubboapplication) name= "hello-world"   />         <!-- 使用zookeeper注册中心暴露发现服务地址 -->       <[dubbo:registry](http://dubboregistry) address= "<zookeeper://127.0.0.1:2181>"  />         <!-- 用dubbo协议在 20880 端口暴露服务 -->       <[dubbo:protocol](http://dubboprotocol) name= "dubbo"  port= "20880"  />         <!-- 声明需要暴露的服务接口 -->       <[dubbo:service](http://dubboservice)  interface = "com.alibaba.dubbo.demo.DemoService"  ref= "demoService"  />         <!-- 和本地bean一样实现服务 -->       <bean id= "demoService"  class = "com.alibaba.dubbo.demo.provider.DemoServiceImpl"  />         <!-- 声明需要暴露的服务接口 -->       <[dubbo:service](http://dubboservice)  interface = "com.alibaba.dubbo.demo.ValidationService"  ref= "validationService"  />         <!-- 和本地bean一样实现服务 -->       <bean id= "validationService"  class = "com.alibaba.dubbo.demo.provider.ValidationServiceImpl"  />         <bean id= "cacheService"  class = "com.alibaba.dubbo.demo.provider.CacheServiceImpl"  />         <[dubbo:service](http://dubboservice)  interface = "com.alibaba.dubbo.demo.CacheService"  ref= "cacheService"  />         <!-- 声明需要暴露的服务接口 -->       <[dubbo:service](http://dubboservice)  interface = "com.alibaba.dubbo.demo.DemoService2"  ref= "demoService2"  />         <!-- 和本地bean一样实现服务 -->       <bean id= "demoService2"  class = "com.alibaba.dubbo.demo.provider.DemoService2Impl"  />         <bean id= "callbackService"  class = "com.alibaba.dubbo.demo.provider.CallbackServiceImpl"  />       <[dubbo:service](http://dubboservice)  interface = "com.alibaba.dubbo.demo.CallbackService"  ref= "callbackService"  connections= "1"  callbacks= "1000" >           <[dubbo:method](http://dubbomethod) name= "addListener" >               <[dubbo:argument](http://dubboargument) index= "1"  callback= "true"  />               <!--也可以通过指定类型的方式-->               <!--<[dubbo:argument](http://dubboargument) type= "com.alibaba.dubbo.demo.CallbackListener"  callback= "true"  />-->           </[dubbo:method](http://dubbomethod)>       </[dubbo:service](http://dubboservice)>    </beans> |

6、客户端控制台输出：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | callback1:Changed:  2015 - 02 - 13  13 : 58 : 20  callback2:Changed:  2015 - 02 - 13  13 : 58 : 20  callback2:Changed:  2015 - 02 - 13  13 : 58 : 20  callback1:Changed:  2015 - 02 - 13  13 : 58 : 20  callback2:Changed:  2015 - 02 - 13  13 : 58 : 25  callback1:Changed:  2015 - 02 - 13  13 : 58 : 25  callback2:Changed:  2015 - 02 - 13  13 : 58 : 30  callback1:Changed:  2015 - 02 - 13  13 : 58 : 30  callback2:Changed:  2015 - 02 - 13  13 : 58 : 35  callback1:Changed:  2015 - 02 - 13  13 : 58 : 35  callback2:Changed:  2015 - 02 - 13  13 : 58 : 40  callback1:Changed:  2015 - 02 - 13  13 : 58 : 40  callback2:Changed:  2015 - 02 - 13  13 : 58 : 45  callback1:Changed:  2015 - 02 - 13  13 : 58 : 45 |