# dubbo分享9--事件通知

在调用之前，调用之后，出现异常时，会触发oninvoke, onreturn, onthrow三个事件，可以配置当事件发生时，通知哪个类的哪个方法。

1、服务提供端及客户端共享代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | package  com.alibaba.dubbo.demo;    public  interface  DemoService {       String sayHello(String name);         Person get( int  id);  } |

|  |  |
| --- | --- |
| 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 | package  com.alibaba.dubbo.demo;    import  java.io.Serializable;    public  class  Person  implements  Serializable {       private  static  final  long  serialVersionUID = -8994496944734041861L;       private  int  age;       private  String name;         public  int  getAge() {           return  age;       }         public  void  setAge( int  age) {           this .age = age;       }         public  String getName() {           return  name;       }         public  void  setName(String name) {           this .name = name;       }         @Override       public  String toString() {           return  "Person [age="  + age +  ", name="  + name +  "]" ;       }  } |

2、客户端代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | package  com.alibaba.dubbo.demo;    public  interface  Nofify {       public  void  onreturn(Person msg, Integer id);         public  void  onthrow(Throwable ex, Integer id);  } |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | import  java.util.HashMap;  import  java.util.Map;    import  com.alibaba.dubbo.demo.Nofify;  import  com.alibaba.dubbo.demo.Person;      public  class  NofifyImpl  implements  Nofify {       public  Map<Integer, Person> ret =  new  HashMap<Integer, Person>();       public  Map<Integer, Throwable> errors =  new  HashMap<Integer, Throwable>();         public  void  onreturn(Person msg, Integer id) {           System.out.println( "onreturn:"  + msg);           ret.put(id, msg);       }         public  void  onthrow(Throwable ex, Integer id) {           errors.put(id, ex);       }  } |

|  |  |
| --- | --- |
| 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 | import  java.util.Map;    import  org.springframework.context.support.ClassPathXmlApplicationContext;    import  com.alibaba.dubbo.demo.DemoService;  import  com.alibaba.dubbo.demo.Person;    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();             // 事件通知 示例           DemoService demoService = (DemoService) context.getBean( "demoService" );           NofifyImpl notify = (NofifyImpl) context.getBean( "demoCallback" );           int  requestId1 =  1 ;           Person ret1 = demoService.get(requestId1);           int  requestId2 =  2 ;           Person ret2 = demoService.get(requestId2);           int  requestId3 =  3 ;           Person ret3 = demoService.get(requestId3);             System.out.println( "查看返回结果：" );             for  (Map.Entry<Integer, Person> kv : notify.ret.entrySet()) {               System.out.println(kv.getValue());           }       }  } |

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  30  31  32  33 | <?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" >           <[dubbo:method](http://dubbomethod) name= "get"  onreturn =  "demoCallback.onreturn"  onthrow= "demoCallback.onthrow"  />       </[dubbo:reference](http://dubboreference)>         <!-- 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"  />         <bean id = "demoCallback"  class  =  "NofifyImpl"  />    </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 | package  com.alibaba.dubbo.demo.provider;    import  com.alibaba.dubbo.demo.DemoService;  import  com.alibaba.dubbo.demo.Person;    public  class  DemoServiceImpl  implements  DemoService {       @Override       public  String sayHello(String name) {           return  "Hello "  + name +  "==2" ;       }         @Override       public  Person get( int  id) {           Person persion =  new  Person();           persion.setAge(id);           persion.setName( "hanyiyi011" );           return  persion;       }  } |

 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 | onreturn:Person [age= 1 , name=hanyiyi011]  onreturn:Person [age= 2 , name=hanyiyi011]  onreturn:Person [age= 3 , name=hanyiyi011]  查看返回结果：  Person [age= 1 , name=hanyiyi011]  Person [age= 2 , name=hanyiyi011]  Person [age= 3 , name=hanyiyi011] |