☆雪无痕☆

Dubbo与Zookeeper、SpringMVC整合和使用

作为dubbo框架初学者,能让框架跑起来非常不容易,非常感谢网上诸多大神提供的文章,本人参考文章地址是: https://my.oschina.net/xshuai/blog/891281

不过别人的记录终究不适合自己,所以还是按照自己的风格简单记录下学习dubbo整合的步骤。

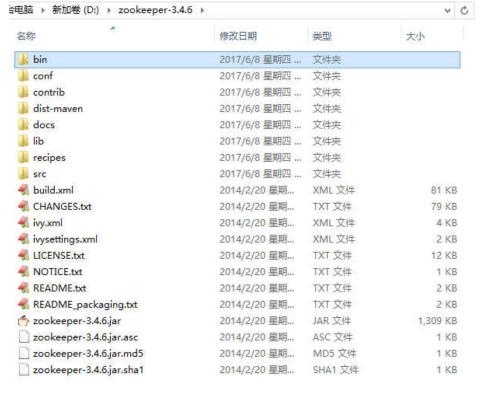
windows环境介绍:

myeclipse 10

jdk1.6

tomcat 6.0.35

- 一、安装Zookeeper
 - 1.通过链接下载对应的包 http://www.apache.org/dist/zookeeper/
 - 2.Zookeeper下载后解压即可,见下图



3.进入到conf里面,会看到zoo_sample.cfg文件。将zoo_sample.cfg改成bak文件,并复制一个修改为zoo.cfg,修改相关配置内容, 注意修改的日志文件夹需要自己手动创建

```
---+---1----2---+---3----+---4----+---5----+----6----+---7----+
 1 # The number of milliseconds of each tick
 2 tickTime=2000
 3 # The number of ticks that the initial
 4 # synchronization phase can take
 5 initLimit=10
 6 # The number of ticks that can pass between
   # sending a request and getting an acknowledgement
 8 syncLimit=5
 9 # the directory where the snapshot is stored.
10 # do not use /tmp for storage, /tmp here is just
11 # example sakes.
12 dataDir=d:\\zookeeper\\tmp
   dataLogDir=d:\\zookeeper\\logs
   # the port at which the clients will connect
15 clientPort=2181
16 # the maximum number of client connections.
   # increase this if you need to handle more clients
17
18 #maxClientCnxns=60
19
20 # Be sure to read the maintenance section of the
21
   # administrator guide before turning on autopurge.
22
23
   # http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc maintenance
24
25 # The number of snapshots to retain in dataDir
26 #autopurge.snapRetainCount=3
   # Purge task interval in hours
28 # Set to "0" to disable auto purge feature
29 #autopurge.purgeInterval=1
```

4.进入D:\zookeeper-3.4.6\bin,双击zkServer.cmd,见到如下界面,就表示zookeeper已启动成功

<			18年7	'月		>
日	_	=	Ξ	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22 29	23 30	24 31	25 1	26 2	27 3	28 4
导情首新联订管 统一随文评引公 昵园粉关的客页随系阅理计 笔章论用告告:::::::::::::::::::::::::::::::::::	7 0 0 0 1年1· 2					
+加关 搜索	<i>i</i> ±			大找看 谷歌搜		
常用链 我的链 我的评 我的参 最新评 我的标	笔论与论签					
我的标 dubbo (1) redis3 zooke zooke zooke zooke	文装 部 eper(eper z eper z	署(1) 1) zkclier 原生 ap	nt api(pi 调用		 隆合入i	刀级
随笔分 dubbo java(i redis(zooke)(1) L) 1)	4)				
随笔 2018 ² 2017 ² 2017 ² 2017 ²	∓1月(∓12月	(1) (3)				

阅读排行榜
1. Dubbo与Zookeeper、SpringMVC整合和
使用(34568)
2. zookeeper zkClient api 使用(422)
3. zookeeper环境搭建及使用(232)

- 4. zookeeper集群搭建(120)
- 5. redis安装(linux)(49)

2017年6月 (1)

推荐排行榜

1. Dubbo与Zookeeper、SpringMVC整合和

使用(3)

```
C:4.
                    C:\Windows\system32\cmd.exe
2017-06-08 16:58:29,228 [myid:] - INFO | [main:Environment@100] - Server environm ^
ent:java.io.tmpdir=C:\Users\zhaoj\AppData\Local\Temp\
2017-06-08 16:58:29,229 [myid:] — INFO [main:Environment@100] — Server environ
ent:java.compiler=<NA>
ent:os.name=Windows 8.1
ent:os.arch=amd64
2017-06-08 16:58:29,233 [myid:l - INFO [main:Environment@100] - Server environm
ent:os.version=6.3
2017-06-08 16:58:29,233 [myid:] - INFO [main:Environment@100] - Server environm
ent:user.name=zhaoj
ent:user.home=C:∖Users\zhaoj
2017-06-08 16:58:29,234 [myid:] - INFO
                              [main:Environment@100] - Server environm
ent:user.dir=D:\zookeeper-3.4.6\bin
2017-06-08 16:58:29,242 [myid:] - INFO
                              [main:ZooKeeperServer@755] - tickTime se
t to 2000
imeout set to −1
2017-06-08 16:58:29,252        [myid:] - INFO        [main:ZooKeeperServer@773] - maxSessionT
imeout set to −1
2017-06-08 16:58:29,520 [myid:] - INFO [main:NIOServerCnxnFactory094] - binding
to port 0.0.0.0/0.0.0.0:2181
```

二、安装dubbo

- 1.本人使用的是dubbo-admin-2.5.3.war, 下载地址: http://pan.baidu.com/s/1eSnuqEQ
- 2. 拷贝一个新的tomcat , 并将tomcat/webapps里面的ROOT文件夹删掉
- 3.将dubbo-admin-2.5.3.war重命名为ROOT.war , 并拷贝到tomcat/webapps目录下
- 4.启动tomcat

```
INFO context.WebxComponentsContext - Bean '(inner bean)#11' is not eligible for getting processed by all BeanPostProcessc INFO context.WebxComponentsContext - Bean 'com.alibaba.dubbo.config.RegistryConfig' is not eligible for getting processcd INFO context.WebxComponentsContext - Bean 'com.alibaba.dubbo.config.RegistryConfig' is not eligible for getting processor INFO context.WebxComponentsContext - Bean 'registryService' is not eligible for getting processed by all BeanPostProcessor INFO context.WebxComponentsContext - Bean '(inner bean)#11' is not eligible for getting processed by all BeanPostProcessor INFO context.InheritableListableBeanFactory - Pre-instantiating singletons in com.alibaba.citrus.springext.support.com.org. INFO context.InheritableListableBeanFactory - Pre-instantiating singletons in com.alibaba.citrus.springext.support.com.org. INFO velocity.VelocityEngine - SpringResourceLoaderAdapter : initialization starting.
INFO velocity.VelocityEngine - SpringResourceLoaderAdapter : set path '/templates/common/'
INFO velocity.VelocityEngine - SpringResourceLoaderAdapter : initialization complete.
INFO rule.ExtensionMappingRule - Initialized extension.input:ExtensionMappingRule with cache disabled
INFO rule.ExtensionMappingRule - Initialized extension.output:ExtensionMappingRule with cache disabled
INFO rule.DirectModuleMappingRule - Initialized action:DirectModuleMappingRule with cache disabled
INFO rule.FallbackModuleMappingRule - Initialized screen.rotemplate:DirectModuleMappingRule with cache enabled
INFO rule.FallbackTemplateMappingRule - Initialized control.template:DirectModuleMappingRule with cache disabled
INFO rule.DirectModuleMappingRule - Initialized control.template:DirectModuleMappingRule with cache disabled
INFO rule.DirectTemplateMappingRule - Initialized control.template:DirectModuleMappingRule with cache disabled
INFO rule.DirectTemplateMappingRule - Initialized control.template:DirectModuleMappingRule with cache disabled
INFO rule.DirectTemplateMappingRule - Initialized control.templa
```

出现上述问题表示你没有启动zookeeper

5.dubbo发布成功后,输入http://localhost:8889/dubbo-admin-2.5.3/,此时出现登录页面,输入root/root进行登录,登录成功后如图



统计: 服务数:0 应用数:0 提供者数:0 消费者数:0

三、创建服务提供服务(provider)

项目结构如下图

```
provider
  ▲ 进 src

    ■ com.provider

     DemoService.java
      DemoServiceImpl.java
      ▶ ☑ ProviderTestAction.java
     applicationContext-servlet.xml
     applicationContext.xml
     dubbo-provider.xml
  ▶ Mark JRE System Library [jdk1.6.0_43]
  Web App Libraries
   Diagram Commons-io-1.4.jar - D:\myeclipse1(

    javassist-3.7.ga.jar - D:\myeclipse10\

   Dog4j-1.2.16.jar - D:\myeclipse10\pro
   netty-3.2.5.Final.jar - D:\myeclipse10'
   org.springframework.beans-3.1.4.RE
   Dog.springframework.core-3.1.4.REL
   Digital org.springframework.expression-3.1
   org.springframework.web-3.1.4.RELI
   org.springframework.web.portlet-3.
   org.springframework.web.servlet-3.*

▶ Slf4j-api-1.6.6.jar - D:\myeclipse10\p

   Diagram vookeeper-3.4.6.jar - D:\myeclipse1(

■ WEB-INF

     D 🗁 lib
       x web.xml
     j index.jsp
```

相应的类中的代码如下:

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
   xmlns="http://java.sun.com/xml/ns/javaee"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
   http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
 <display-name></display-name>
 <listener-class>
       org.springframework.web.context.ContextLoaderListener
   </listener-class>
       <servlet-name>provider</servlet-name>
       <init-param>
           <param-name>contextConfigLocation</param-name>
              classpath:applicationContext.xml,classpath:applicationContext-servlet.xml
           </param-value>
       </init-param>
       <load-on-startup>1</load-on-startup>
   </servlet>
   <servlet-mapping>
       <servlet-name>provider</servlet-name>
       <url-pattern>*.do</url-pattern>
   </servlet-mapping>
   <!-- 字符过滤器 -->
   <filter>
       <filter-name>Set Character Encoding</filter-name>
       <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
       <init-param>
           <param-name>encoding</param-name>
           <param-value>UTF-8</param-value>
       </init-param>
   </filter>
   <filter-mapping>
       <filter-name>Set Character Encoding</filter-name>
       <url-pattern>/*</url-pattern>
   </filter-mapping>
 <welcome-file-list>
   <welcome-file>index.jsp</welcome-file>
 </welcome-file-list>
</web-app>
```

DemoService.java

```
package com.provider;

public interface DemoService {
    String sayHello(String name);
}
```

${\tt DemoServiceImpl.java}$



```
package com.provider;
import org.springframework.stereotype.Service;

@Service(value="demoService")
public class DemoServiceImpl implements DemoService{

   public String sayHello(String name) {
      return "Hello Dubbo, Hello " + name;
   }
}
```

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:jdbc="http://www.springframework.org/schema/jdbc"
    \verb|xmlns:context="http://www.springframework.org/schema/context"|\\
    xsi:schemaLocation="
      http://www.springframework.org/schema/context <a href="http://www.springframework.org/schema/context/spring-">http://www.springframework.org/schema/context/spring-</a>
context-3.1.xsd
      http://www.springframework.org/schema/beans <a href="http://www.springframework.org/schema/beans/spring-beans-">http://www.springframework.org/schema/beans/spring-beans-</a>
3.1.xsd
      http://www.springframework.org/schema/tx
      http://www.springframework.org/schema/tx/spring-tx-3.1.xsd
      http://www.springframework.org/schema/aop
      http://www.springframework.org/schema/aop/spring-aop-3.1.xsd"
    <context:component-scan base-package="com.**"></context:component-scan>
    <!-- 引入服务提供者配置文件 -->
    <import resource="dubbo-provider.xml" />
</beans>
```

dubbo-provider.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
   \verb|xsi:schemaLocation="http://www.springframework.org/schema/beans| \\
   http://www.springframework.org/schema/beans/spring-beans.xsd
   http://code.alibabatech.com/schema/dubbo
   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 name="hello-world-provider" />
   <!-- 使用multicast广播注册中心暴露服务地址 -->
  <!-- <dubbo:registry address="multicast://224.5.6.7:1234" /> -->
    <!-- 使用zookeeper注册中心暴露服务地址 -->
   <dubbo:registry address="zookeeper://127.0.0.1:2181" />
   <!-- 用dubbo协议在20880端口暴露服务 -->
   <dubbo:protocol name="dubbo" port="20880" />
   <!-- 声明需要暴露的服务接口 -->
   <dubbo:service interface="com.provider.DemoService" ref="demoService" />
   <!-- 和本地bean—样实现服务 -->
   <bean id="demoService" class="com.provider.DemoServiceImpl" />
</heans>
```

application Context-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:aop="http://www.springframework.org/schema/aop"
   \verb|xmlns:tx="http://www.springframework.org/schema/tx"|
   xmlns:jdbc="http://www.springframework.org/schema/idbc"
   xmlns:mvc="http://www.springframework.org/schema/mvc"
   xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="
     http://www.springframework.org/schema/context <a href="http://www.springframework.org/schema/context/spring-">http://www.springframework.org/schema/context/spring-</a>
     http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-
     http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.1.xsd
     http://www.springframework.org/schema/mvc_http://www.springframework.org/schema/mvc/spring-mvc-3.1.xsd
   default-autowire="byName">
   <!-- 默认的注解映射的支持 -->
   <mvc:annotation-driven>
       <mvc:message-converters register-defaults="true">
            <bean class ="org.springframework.http.converter.StringHttpMessageConverter">
               cproperty name ="supportedMediaTypes">
                        <value>text/plain;charset=UTF-8</value>
               </property>
```

```
</bean>
       </mvc:message-converters>
   </mvc:annotation-driven>
   <!-- 视图解释类 -->
   <bean id="viewResolver"</pre>
       class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        cproperty name="prefix" value="/WEB-INF/jsp/" />
       property name="suffix" value=".jsp" />
       property name="viewClass"
           value="org.springframework.web.servlet.view.JstlView" />
   </bean>
   <!-- 加载静态资源 -->
   <mvc:resources mapping="/css/**" location="/css/" />
   <mvc:resources mapping="/js/**" location="/js/" />
   <mvc:resources mapping="/images/**" location="/images/" />
</beans>
```

四、创建调用服务(customer)

customer服务架构和provider—致,拷贝一个即可

相应的代码如下:

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
\verb|\color| \verb| smlns="http://www.springframework.org/schema/beans"|
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:aop="http://www.springframework.org/schema/aop"
   xmlns:tx="http://www.springframework.org/schema/tx"
   xmlns:jdbc="http://www.springframework.org/schema/jdbc"
   xmlns:context="http://www.springframework.org/schema/context"
   xsi:schemaLocation="
     http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-
     http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-
3.1.xsd
     http://www.springframework.org/schema/tx
     http://www.springframework.org/schema/tx/spring-tx-3.1.xsd
     http://www.springframework.org/schema/aop
     http://www.springframework.org/schema/aop/spring-aop-3.1.xsd"
   <context:component-scan base-package="com.**"></context:component-scan>
   <!-- 引入消费者配置文件 -->
   <import resource="dubbo-consumer.xml" />
</beans>
```

dubbo-consumer.xml

```
4
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
   \verb|xsi:schemaLocation="http://www.springframework.org/schema/beans| \\
   http://www.springframework.org/schema/beans/spring-beans.xsd
   \verb|http://code.alibabatech.com/schema/dubbo|
   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 name="hello-world-customer"/>
   <!-- 使用multicast广播注册中心暴露发现服务地址 -->
   <!-- <dubbo:registry address="multicast://224.5.6.7:1234" /> -->
   <dubbo:registry address="zookeeper://127.0.0.1:2181" check="false"/>
   <!-- 生成远程服务代理,可以和本地bean一样使用demoService -->
   <dubbo:reference id="demoService" interface="com.provider.DemoService" check="false"/>
</beans>
```

applicationContext-servlet.xml 和provider中的代码一致,就不贴出来了,本项目中其实这个文件可以不用

CustomerAction.java测试类

```
package com.customer;
import javax.annotation.Resource;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;
import com.provider.DemoService;
@Controller
@RequestMapping(value="/customerTest")
public class CustomerAction {
   @Resource(name="demoService")
   private DemoService demoService;
   @RequestMapping(value="/test.do")
   public ModelAndView test(HttpServletRequest request, HttpServletResponse response) {
       System.out.println("成功");
       String result = demoService.sayHello("world");
       System.out.println(result);
       return null;
```

五、共享服务

注意:需要将服务提供的接口打成jar包,放入customer中

六、测试

步骤:1、先启动zookeeper服务

- 2、启动dubbo服务
- 3、启动provider服务
- 4、启动customer服务

正常会出现如下界面

错误总结:

若按照上述测试步骤分别在不同的tomcat中启动服务,应该一切正常

若2、3、4这三步放在同一个tomcat中启动,会出现如下错误。

```
tom.cat7Server[Remote Java Application] D\Program Files\Java\jdk1.6.0_43\bin\javaw.exe (2017-6-9 下午1:29:11)
Iog4j:WARN No appenders could be found for logger (org.springframework.web.context.support.StandardServletEnvironment).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
2017-6-9 13:29:14 org.apache.catalina.core.ApplicationContext log
信息: Initializing Spring FrameworkServlet 'customer'
2017-6-9 13:29:17 org.apache.catalina.core.ApplicationContext log
 严重: StandardWrapper.Throwable
<u>org.springframework.beans.factory.BeanCreationException</u>: Error creating bean with name 'customerAction': Injection of resource d at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(<a href="CommonAnnotationBe">CommonAnnotationBe</a>
                                at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(\underline{AbstractAutowireCapableBean} at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(\underline{AbstractAutowireCapableBean})
                                 at\ org. spring framework. beans. factory. support. Abstract Autowire Capable Bean Factory. create Bean (\underline{Abstract Autowire Capable Bean Factory). create Bean (\underline{Abstract Autowire Capable Bean Factory). }
                                at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(<u>AbstractBeanFactory.java:294</u>) at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(<u>DefaultSingletonBeanRegistry.java</u>
                                  at\ org. spring framework. beans. factory. support. Abstract Bean Factory. do Get Bean (\underline{Abstract Bean Factory.java: 291)}
                                at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(<u>AbstractBeanFactory.java:193</u>) at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(<u>DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFact</u>
                                  at org. spring framework. context. support. Abstract Application Context. finish Bean Factory Initialization (\underline{Abstract Application Context.})
                                at org.springframework.context.support.AbstractApplicationContext.refresh(<u>AbstractApplicationContext.java:472</u>) at org.springframework.web.servlet.FrameworkServlet.configureAndRefreshWebApplicationContext(<u>FrameworkServlet.java:631</u>)
                                at org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(<a href="FrameworkServlet.java:588">FrameworkServlet.java:588</a>) at org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(<a href="FrameworkServlet.java:645">FrameworkServlet.java:645</a>)
                                  at\ or \ g. spring framework. web. servlet. Framework Servlet. in it Web Application Context ( \underline{Framework Servlet.java: 508) ) and the property of the prop
                                 at org. spring framework. web. servlet. Framework Servlet. in it Servlet Bean ( \underline{Framework Servlet.} \underline{java: 449})
                                  at org.springframework.web.servlet.HttpServletBean.init(<u>HttpServletBean.java:133</u>)
tomcat7Server [Remote Java Application] D:\Program Files\Java\jdk1.6.0_43\bin\javaw.exe (2017-6-9 下午1:29:11)
```

of the service com.provider.DemoService. No provider available for the service com.provider.DemoService from the url zookeeper:,

看下关键异常: No provider available for the service

此异常是由于服务没有可以使用的提供者,就是说在zookeeper注册中心(zookeeper-url)中没有可供消费者调用的url,消费者访问提供者就失败了。

具体原因分析:由于dobbo在启动的时候会去检查各服务之间的依赖关系,由于启动的时候消费者没有检查到提供者提供的服务(此时可能提供者还没启动),所以报错

在消费者配置文件中,需要将<dubbo:registry address="zookeeper://127.0.0.1:2181"/>修改为<dubbo:registry address="zookeeper://127.0.0.1:2181" check="false"/>

由于dubbo在注册的时候是默认会检查服务的依赖关系的

分类: dubbo

标签: dubbo zookeeper springmvc整合入门级



粉丝 - 2 +加关注

» 下一篇: zookeeper环境搭建及使用
posted on 2017-06-09 15:08 ☆雪无痕☆ 阅读(34568) 评论(0) 编辑 收藏

3

2

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论,请 登录 或 注册,<u>访问</u>网站首页。



最新IT新闻:

- · 消息称美团点评将于8月23日过聆讯, 普遍接受600亿美元估值
- ·成立3年就上市,我们为你总结拼多多的超级玩法
- Facebook二季度财报令人失望 用户增速降至历史最低水平
- ·美国科技股周四将遭血洗 FANG或将带头暴跌
- · 高通高管:苹果2018年款iPhone使用英特尔调制解调器
- » 更多新闻...



最新知识库文章:

- 在腾讯的八年, 我的职业思考
- 为什么我离开了管理岗位
- ·那些让人睡不着觉的bug,你有没有遭遇过?
- 观察之道: 带你走进可观察性
- 危害程序员职业生涯的三大观念
- » 更多知识库文章...

Powered by:

博客园

Copyright © ☆雪无痕☆