**[分布式服务框架 dubbo/dubbox 入门示例](http://www.cnblogs.com/yjmyzz/p/dubbox-demo.html)**

dubbo是一个分布式的服务架构，可直接用于生产环境作为SOA服务框架。

官网首页：http://dubbo.io/ ，官方用户指南 http://dubbo.io/User+Guide-zh.htm上面的几张图画得不错，完全可以当做SOA架构的学习资料

淘宝将这个项目开源出来以后，得到了不少同行的支持，包括：

当当网的扩展版本dubbox ：https://github.com/dangdangdotcom/dubbox

京东的扩展版本jd-hydra: http://www.oschina.net/p/jd-hydra

不过，略有遗憾的是，据说在淘宝内部，dubbo由于跟淘宝另一个类似的框架HSF（非开源）有竞争关系，导致dubbo团队已经解散（参见http://www.oschina.net/news/55059/druid-1-0-9 中的评论），反到是当当网的扩展版本仍在持续发展，墙内开花墙外香。

不管如何，能在阿里、当当、京东这些大型网站正式使用的框架，总不至于差到哪里去。

本文下面的示例均基于当当的dubbox版本，由于dubbox并没向maven提交编译后的jar包，所以只能从github clone代码到本地编译得到jar包。

编译及测试步骤：(以下步骤全在windows环境中完成)

1. 本机先安装github on Windows的客户端，将在path路径中，把git.exe加进去

2. 命令行下 git clone https://github.com/dangdangdotcom/dubbox 把代码拉到本地

3. mvn install -Dmaven.test.skip=true 跳过测试编译

4. 在本机[安装一个zookeeper](http://www.cnblogs.com/yjmyzz/p/4587663.html" \t "_blank)，参考zoo.cfg如下：

tickTime=2000  
initLimit=10  
syncLimit=5  
dataDir=D:/java/zookeeper-3.4.6/data  
dataLogDir=D:/java/zookeeper-3.4.6/log  
clientPort=2181  
server.1=localhost:2287:3387

然后输入 bin/zkServer.cmd 启用zookeeper

5. intellij Idea中导入源码

6. 运行 \dubbox\dubbo-demo\dubbo-demo-provider\src\test\java\com\alibaba\dubbo\demo\provider\DemoProvider.java

把服务提供方跑起来，成功后，可以在ZK里，用 ls / 看下，会发现zk里多出了一个dubbo的节点，所有服务全注册在这里了

7. 运行\dubbox\dubbo-demo\dubbo-demo-consumer\src\test\java\com\alibaba\dubbo\demo\consumer\DemoConsumer.java

服务销售方调用测试，可以看console里的输出

8. 运行\dubbox\dubbo-demo\dubbo-demo-consumer\src\test\java\com\alibaba\dubbo\demo\consumer\RestClient.java

跑一下rest调用

9. 浏览器访问 http://localhost:8888/services/users/100.xml 或 http://localhost:8888/services/users/100.json

dubbox官方的示例，虽然已经很简单了，但是对于初次接触的人来讲，仍然略显复杂，下面的代码在其基础上简化了一下：

一、先定义服务接口及传输对象DTO

项目结构如下



代码：

User.java

[+ View Code?](http://www.cnblogs.com/yjmyzz/p/dubbox-demo.html)

|  |  |
| --- | --- |
| 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  51  52  53  54  55  56  57 | package yjmyzz.dubbo.demo.api;  import org.codehaus.jackson.annotate.JsonProperty;  import javax.validation.constraints.Min;  import javax.validation.constraints.NotNull;  import javax.validation.constraints.Size;  import javax.xml.bind.annotation.XmlAccessType;  import javax.xml.bind.annotation.XmlAccessorType;  import javax.xml.bind.annotation.XmlElement;  import javax.xml.bind.annotation.XmlRootElement;  import java.io.Serializable;    @XmlRootElement  @XmlAccessorType(XmlAccessType.FIELD)  public class User implements Serializable {        @NotNull      @Min(1L)      private Long id;        @JsonProperty("username")      @XmlElement(name = "username")      @NotNull      @Size(min = 6, max = 50)      private String name;        public User() {      }        public User(Long id, String name) {          this.id = id;          this.name = name;      }        public Long getId() {          return id;      }        public void setId(Long id) {          this.id = id;      }        public String getName() {          return name;      }        public void setName(String name) {          this.name = name;      }        @Override      public String toString() {          return "User (" +                  "id=" + id +                  ", name='" + name + '\'' +                  ')';      }  } |

UserService.java

[+ View Code?](http://www.cnblogs.com/yjmyzz/p/dubbox-demo.html)

|  |  |
| --- | --- |
| 1  2  3  4  5 | package yjmyzz.dubbo.demo.api;    public interface UserService {      User getUser(Long id);  } |

UserRestService.java

[+ View Code?](http://www.cnblogs.com/yjmyzz/p/dubbox-demo.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | package yjmyzz.dubbo.demo.api;    import javax.validation.constraints.Min;    public interface UserRestService {      User getUser(@Min(value = 1L, message = "User ID must be greater than 1") Long id);  } |

pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cnblogs.yjmyzz</groupId>

<artifactId>dubbo-hello-api</artifactId>

<version>0.1</version>

<dependencies>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>dubbo</artifactId>

<version>2.8.4</version>

</dependency>

<dependency>

<groupId>javax.validation</groupId>

<artifactId>validation-api</artifactId>

<version>1.0.0.GA</version>

</dependency>

<dependency>

<groupId>javax.annotation</groupId>

<artifactId>javax.annotation-api</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-mapper-asl</artifactId>

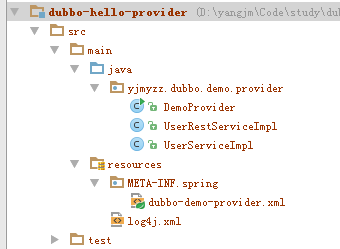
<version>1.9.12</version>

</dependency>

</dependencies>

</project>

二、定义服务生产者（即：服务接口的实现方）



UserServiceImpl.java

[+ View Code?](http://www.cnblogs.com/yjmyzz/p/dubbox-demo.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | package yjmyzz.dubbo.demo.provider;    import yjmyzz.dubbo.demo.api.User;  import yjmyzz.dubbo.demo.api.UserService;    public class UserServiceImpl implements UserService {        public User getUser(Long id) {          return new User(id, "username" + id);      }  } |

UserRestServiceImpl.java

[+ View Code?](http://www.cnblogs.com/yjmyzz/p/dubbox-demo.html)

|  |  |
| --- | --- |
| 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 | package yjmyzz.dubbo.demo.provider;    import com.alibaba.dubbo.rpc.RpcContext;  import com.alibaba.dubbo.rpc.protocol.rest.support.ContentType;  import yjmyzz.dubbo.demo.api.User;  import yjmyzz.dubbo.demo.api.UserRestService;  import yjmyzz.dubbo.demo.api.UserService;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import javax.ws.rs.\*;  import javax.ws.rs.core.MediaType;    @Path("users")  @Consumes({MediaType.APPLICATION\_JSON, MediaType.TEXT\_XML})  @Produces({ContentType.APPLICATION\_JSON\_UTF\_8, ContentType.TEXT\_XML\_UTF\_8})  public class UserRestServiceImpl implements UserRestService {        private UserService userService;        public void setUserService(UserService userService) {          this.userService = userService;      }        @GET      @Path("{id : \\d+}")      public User getUser(@PathParam("id") Long id) {          if (RpcContext.getContext().getRequest(HttpServletRequest.class) != null) {              System.out.println("Client IP address from RpcContext: " + RpcContext.getContext().getRequest(HttpServletRequest.class).getRemoteAddr());          }          if (RpcContext.getContext().getResponse(HttpServletResponse.class) != null) {              System.out.println("Response object from RpcContext: " + RpcContext.getContext().getResponse(HttpServletResponse.class));          }          return userService.getUser(id);      }  } |

DemoProvider.java

[+ View Code?](http://www.cnblogs.com/yjmyzz/p/dubbox-demo.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | package yjmyzz.dubbo.demo.provider;  import org.springframework.context.support.ClassPathXmlApplicationContext;  import java.io.IOException;    public class DemoProvider {      public static void main(String[] args) throws IOException {          ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("classpath\*:META-INF/spring/\*.xml");          context.start();          System.out.println("服务已经启动...");          System.in.read();      }  } |

配置文件：resources\META-INF\spring\dubbo-demo-provider.xml

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gifhttp://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2

3 <beans xmlns="http://www.springframework.org/schema/beans"

4 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

5 xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"

6 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd

7 http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd">

8

9 <dubbo:application name="demo-provider" owner="programmer" organization="dubbox"/>

10

11 <dubbo:registry address="zookeeper://127.0.0.1:2181"/>

12

13 <dubbo:protocol name="dubbo" serialization="kryo" optimizer="yjmyzz.dubbo.demo.api.SerializationOptimizerImpl"/>

14

15 <!-- use tomcat server -->

16 <dubbo:protocol name="rest" port="8888" threads="500" contextpath="services" server="tomcat" accepts="500"

17 extension="com.alibaba.dubbo.rpc.protocol.rest.support.LoggingFilter"/>

18

19

20 <dubbo:service interface="yjmyzz.dubbo.demo.api.UserService" ref="userService" protocol="dubbo" />

21

22 <dubbo:service interface="yjmyzz.dubbo.demo.api.UserRestService" ref="userRestService" protocol="rest" validation="true"/>

23

24 <bean id="userService" class="yjmyzz.dubbo.demo.provider.UserServiceImpl"/>

25

26 <bean id="userRestService" class="yjmyzz.dubbo.demo.provider.UserRestServiceImpl">

27 <property name="userService" ref="userService"/>

28 </bean>

29

30

31 </beans>

[复制代码](javascript:void(0);)

View Code

pom.xml

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gifhttp://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <project xmlns="http://maven.apache.org/POM/4.0.0"

3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

4 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

5

6 <modelVersion>4.0.0</modelVersion>

7

8 <groupId>com.cnblogs.yjmyzz</groupId>

9 <artifactId>dubbo-hello-provider</artifactId>

10 <version>0.1</version>

11

12 <dependencies>

13

14 <!--公用的服务接口-->

15 <dependency>

16 <groupId>com.cnblogs.yjmyzz</groupId>

17 <artifactId>dubbo-hello-api</artifactId>

18 <version>0.1</version>

19 </dependency>

20

21 <dependency>

22 <groupId>com.alibaba</groupId>

23 <artifactId>dubbo</artifactId>

24 <version>2.8.4</version>

25 </dependency>

26

27 <dependency>

28 <groupId>org.javassist</groupId>

29 <artifactId>javassist</artifactId>

30 <version>3.15.0-GA</version>

31 </dependency>

32

33 <dependency>

34 <groupId>org.apache.mina</groupId>

35 <artifactId>mina-core</artifactId>

36 <version>1.1.7</version>

37 </dependency>

38

39 <dependency>

40 <groupId>org.glassfish.grizzly</groupId>

41 <artifactId>grizzly-core</artifactId>

42 <version>2.1.4</version>

43 </dependency>

44

45 <dependency>

46 <groupId>org.apache.httpcomponents</groupId>

47 <artifactId>httpclient</artifactId>

48 <version>4.2.1</version>

49 </dependency>

50

51 <dependency>

52 <groupId>com.alibaba</groupId>

53 <artifactId>fastjson</artifactId>

54 <version>1.1.39</version>

55 </dependency>

56

57 <dependency>

58 <groupId>com.thoughtworks.xstream</groupId>

59 <artifactId>xstream</artifactId>

60 <version>1.4.1</version>

61 </dependency>

62

63 <dependency>

64 <groupId>org.apache.bsf</groupId>

65 <artifactId>bsf-api</artifactId>

66 <version>3.1</version>

67 </dependency>

68

69 <dependency>

70 <groupId>org.apache.zookeeper</groupId>

71 <artifactId>zookeeper</artifactId>

72 <version>3.4.6</version>

73 </dependency>

74

75 <dependency>

76 <groupId>com.github.sgroschupf</groupId>

77 <artifactId>zkclient</artifactId>

78 <version>0.1</version>

79 </dependency>

80

81 <dependency>

82 <groupId>org.apache.curator</groupId>

83 <artifactId>curator-framework</artifactId>

84 <version>2.5.0</version>

85 </dependency>

86

87 <dependency>

88 <groupId>com.googlecode.xmemcached</groupId>

89 <artifactId>xmemcached</artifactId>

90 <version>1.3.6</version>

91 </dependency>

92

93 <dependency>

94 <groupId>org.apache.cxf</groupId>

95 <artifactId>cxf-rt-frontend-simple</artifactId>

96 <version>2.6.1</version>

97 </dependency>

98

99 <dependency>

100 <groupId>org.apache.cxf</groupId>

101 <artifactId>cxf-rt-transports-http</artifactId>

102 <version>2.6.1</version>

103 </dependency>

104

105 <dependency>

106 <groupId>org.apache.thrift</groupId>

107 <artifactId>libthrift</artifactId>

108 <version>0.8.0</version>

109 </dependency>

110

111 <dependency>

112 <groupId>com.caucho</groupId>

113 <artifactId>hessian</artifactId>

114 <version>4.0.7</version>

115 </dependency>

116

117 <dependency>

118 <groupId>javax.servlet</groupId>

119 <artifactId>javax.servlet-api</artifactId>

120 <version>3.1.0</version>

121 </dependency>

122

123 <dependency>

124 <groupId>org.mortbay.jetty</groupId>

125 <artifactId>jetty</artifactId>

126 <version>6.1.26</version>

127 <exclusions>

128 <exclusion>

129 <groupId>org.mortbay.jetty</groupId>

130 <artifactId>servlet-api</artifactId>

131 </exclusion>

132 </exclusions>

133 </dependency>

134

135 <dependency>

136 <groupId>log4j</groupId>

137 <artifactId>log4j</artifactId>

138 <version>1.2.16</version>

139 </dependency>

140

141 <dependency>

142 <groupId>org.slf4j</groupId>

143 <artifactId>slf4j-api</artifactId>

144 <version>1.6.2</version>

145 </dependency>

146

147 <dependency>

148 <groupId>redis.clients</groupId>

149 <artifactId>jedis</artifactId>

150 <version>2.1.0</version>

151 </dependency>

152

153 <dependency>

154 <groupId>javax.validation</groupId>

155 <artifactId>validation-api</artifactId>

156 <version>1.0.0.GA</version>

157 </dependency>

158

159 <dependency>

160 <groupId>org.hibernate</groupId>

161 <artifactId>hibernate-validator</artifactId>

162 <version>4.2.0.Final</version>

163 </dependency>

164

165 <dependency>

166 <groupId>javax.cache</groupId>

167 <artifactId>cache-api</artifactId>

168 <version>0.4</version>

169 </dependency>

170

171 <dependency>

172 <groupId>org.jboss.resteasy</groupId>

173 <artifactId>resteasy-jaxrs</artifactId>

174 <version>3.0.7.Final</version>

175 </dependency>

176

177 <dependency>

178 <groupId>org.jboss.resteasy</groupId>

179 <artifactId>resteasy-client</artifactId>

180 <version>3.0.7.Final</version>

181 </dependency>

182

183 <dependency>

184 <groupId>org.jboss.resteasy</groupId>

185 <artifactId>resteasy-netty</artifactId>

186 <version>3.0.7.Final</version>

187 </dependency>

188

189 <dependency>

190 <groupId>org.jboss.resteasy</groupId>

191 <artifactId>resteasy-jdk-http</artifactId>

192 <version>3.0.7.Final</version>

193 </dependency>

194

195 <dependency>

196 <groupId>org.jboss.resteasy</groupId>

197 <artifactId>resteasy-jackson-provider</artifactId>

198 <version>3.0.7.Final</version>

199 </dependency>

200

201 <dependency>

202 <groupId>org.jboss.resteasy</groupId>

203 <artifactId>resteasy-jaxb-provider</artifactId>

204 <version>3.0.7.Final</version>

205 </dependency>

206

207 <dependency>

208 <groupId>org.apache.tomcat.embed</groupId>

209 <artifactId>tomcat-embed-core</artifactId>

210 <version>8.0.11</version>

211 </dependency>

212

213 <dependency>

214 <groupId>org.apache.tomcat.embed</groupId>

215 <artifactId>tomcat-embed-logging-juli</artifactId>

216 <version>8.0.11</version>

217 </dependency>

218

219 <dependency>

220 <groupId>com.esotericsoftware.kryo</groupId>

221 <artifactId>kryo</artifactId>

222 <version>2.24.0</version>

223 </dependency>

224

225 <dependency>

226 <groupId>de.javakaffee</groupId>

227 <artifactId>kryo-serializers</artifactId>

228 <version>0.26</version>

229 </dependency>

230

231 <dependency>

232 <groupId>de.ruedigermoeller</groupId>

233 <artifactId>fst</artifactId>

234 <version>1.55</version>

235 </dependency>

236

237 </dependencies>

238

239

240 </project>

[复制代码](javascript:void(0);)

View Code

测试时，运行DemoProvider中的main方法即可启动服务，所有服务注册在ZooKeeper，层次结构类似下面这样：

[复制代码](javascript:void(0);)

/dubbo

/dubbo/yjmyzz.dubbo.demo.api.UserRestService

/dubbo/yjmyzz.dubbo.demo.api.UserRestService/providers

/dubbo/yjmyzz.dubbo.demo.api.UserRestService/configurators

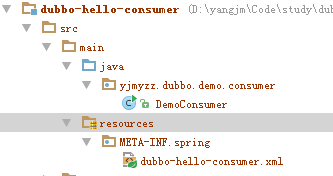
/dubbo/yjmyzz.dubbo.demo.api.UserService

/dubbo/yjmyzz.dubbo.demo.api.UserService/providers

/dubbo/yjmyzz.dubbo.demo.api.UserService/configurators

[复制代码](javascript:void(0);)

三、服务消费方



DemoConsumer.java

[+ View Code?](http://www.cnblogs.com/yjmyzz/p/dubbox-demo.html)

|  |  |
| --- | --- |
| 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 | package yjmyzz.dubbo.demo.consumer;    import org.springframework.context.support.ClassPathXmlApplicationContext;  import yjmyzz.dubbo.demo.api.UserService;  import javax.ws.rs.client.Client;  import javax.ws.rs.client.ClientBuilder;  import javax.ws.rs.client.WebTarget;  import javax.ws.rs.core.Response;    public class DemoConsumer {        public static void main(String[] args) {          final String port = "8888";            //测试Rest服务          getUser("[http://localhost:](http://localhost/)" + port + "/services/users/1.json");          getUser("[http://localhost:](http://localhost/)" + port + "/services/users/1.xml");            //测试常规服务          ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("classpath\*:META-INF/spring/\*.xml");          context.start();          UserService userService = context.getBean(UserService.class);          System.out.println(userService.getUser(1L));      }          private static void getUser(String url) {          System.out.println("Getting user via " + url);          Client client = ClientBuilder.newClient();          WebTarget target = client.target(url);          Response response = target.request().get();          try {              if (response.getStatus() != 200) {                  throw new RuntimeException("Failed with HTTP error code : " + response.getStatus());              }              System.out.println("Successfully got result: " + response.readEntity(String.class));          } finally {              response.close();              client.close();          }      }  } |

配置文件：resources\META-INF\spring\dubbo-hello-consumer.xml

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gifhttp://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

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"

4 xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"

5 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd

6 http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd">

7

8 <dubbo:application name="demo-consumer" owner="programmer" organization="dubbox"/>

9

10 <dubbo:registry address="zookeeper://127.0.0.1:2181"/>

11

12 <dubbo:reference id="userRestService" interface="yjmyzz.dubbo.demo.api.UserRestService"/>

13

14 <dubbo:reference id="userService" interface="yjmyzz.dubbo.demo.api.UserService"/>

15

16 </beans>

[复制代码](javascript:void(0);)

View Code

pom.xml

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gifhttp://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <project xmlns="http://maven.apache.org/POM/4.0.0"

3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

4 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

5

6 <modelVersion>4.0.0</modelVersion>

7

8 <groupId>com.cnblogs.yjmyzz</groupId>

9 <artifactId>dubbo-hello-consumer</artifactId>

10 <version>0.1</version>

11

12

13 <dependencies>

14 <!--公用的服务接口-->

15 <dependency>

16 <groupId>com.cnblogs.yjmyzz</groupId>

17 <artifactId>dubbo-hello-api</artifactId>

18 <version>0.1</version>

19 </dependency>

20

21

22 <dependency>

23 <groupId>com.alibaba</groupId>

24 <artifactId>dubbo</artifactId>

25 <version>2.8.4</version>

26 </dependency>

27

28 <dependency>

29 <groupId>org.javassist</groupId>

30 <artifactId>javassist</artifactId>

31 <version>3.15.0-GA</version>

32 </dependency>

33

34 <dependency>

35 <groupId>org.apache.mina</groupId>

36 <artifactId>mina-core</artifactId>

37 <version>1.1.7</version>

38 </dependency>

39

40 <dependency>

41 <groupId>org.glassfish.grizzly</groupId>

42 <artifactId>grizzly-core</artifactId>

43 <version>2.1.4</version>

44 </dependency>

45

46 <dependency>

47 <groupId>org.apache.httpcomponents</groupId>

48 <artifactId>httpclient</artifactId>

49 <version>4.2.1</version>

50 </dependency>

51

52 <dependency>

53 <groupId>com.alibaba</groupId>

54 <artifactId>fastjson</artifactId>

55 <version>1.1.39</version>

56 </dependency>

57

58 <dependency>

59 <groupId>com.thoughtworks.xstream</groupId>

60 <artifactId>xstream</artifactId>

61 <version>1.4.1</version>

62 </dependency>

63

64 <dependency>

65 <groupId>org.apache.bsf</groupId>

66 <artifactId>bsf-api</artifactId>

67 <version>3.1</version>

68 </dependency>

69

70 <dependency>

71 <groupId>org.apache.zookeeper</groupId>

72 <artifactId>zookeeper</artifactId>

73 <version>3.4.6</version>

74 </dependency>

75

76 <dependency>

77 <groupId>com.github.sgroschupf</groupId>

78 <artifactId>zkclient</artifactId>

79 <version>0.1</version>

80 </dependency>

81

82 <dependency>

83 <groupId>org.apache.curator</groupId>

84 <artifactId>curator-framework</artifactId>

85 <version>2.5.0</version>

86 </dependency>

87

88 <dependency>

89 <groupId>com.googlecode.xmemcached</groupId>

90 <artifactId>xmemcached</artifactId>

91 <version>1.3.6</version>

92 </dependency>

93

94 <dependency>

95 <groupId>org.apache.cxf</groupId>

96 <artifactId>cxf-rt-frontend-simple</artifactId>

97 <version>2.6.1</version>

98 </dependency>

99

100 <dependency>

101 <groupId>org.apache.cxf</groupId>

102 <artifactId>cxf-rt-transports-http</artifactId>

103 <version>2.6.1</version>

104 </dependency>

105

106 <dependency>

107 <groupId>org.apache.thrift</groupId>

108 <artifactId>libthrift</artifactId>

109 <version>0.8.0</version>

110 </dependency>

111

112 <dependency>

113 <groupId>com.caucho</groupId>

114 <artifactId>hessian</artifactId>

115 <version>4.0.7</version>

116 </dependency>

117

118 <dependency>

119 <groupId>javax.servlet</groupId>

120 <artifactId>javax.servlet-api</artifactId>

121 <version>3.1.0</version>

122 </dependency>

123

124 <dependency>

125 <groupId>org.mortbay.jetty</groupId>

126 <artifactId>jetty</artifactId>

127 <version>6.1.26</version>

128 <exclusions>

129 <exclusion>

130 <groupId>org.mortbay.jetty</groupId>

131 <artifactId>servlet-api</artifactId>

132 </exclusion>

133 </exclusions>

134 </dependency>

135

136 <dependency>

137 <groupId>log4j</groupId>

138 <artifactId>log4j</artifactId>

139 <version>1.2.16</version>

140 </dependency>

141

142 <dependency>

143 <groupId>org.slf4j</groupId>

144 <artifactId>slf4j-api</artifactId>

145 <version>1.6.2</version>

146 </dependency>

147

148 <dependency>

149 <groupId>redis.clients</groupId>

150 <artifactId>jedis</artifactId>

151 <version>2.1.0</version>

152 </dependency>

153

154 <dependency>

155 <groupId>javax.validation</groupId>

156 <artifactId>validation-api</artifactId>

157 <version>1.0.0.GA</version>

158 </dependency>

159

160 <dependency>

161 <groupId>org.hibernate</groupId>

162 <artifactId>hibernate-validator</artifactId>

163 <version>4.2.0.Final</version>

164 </dependency>

165

166 <dependency>

167 <groupId>javax.cache</groupId>

168 <artifactId>cache-api</artifactId>

169 <version>0.4</version>

170 </dependency>

171

172 <dependency>

173 <groupId>org.jboss.resteasy</groupId>

174 <artifactId>resteasy-jaxrs</artifactId>

175 <version>3.0.7.Final</version>

176 </dependency>

177

178 <dependency>

179 <groupId>org.jboss.resteasy</groupId>

180 <artifactId>resteasy-client</artifactId>

181 <version>3.0.7.Final</version>

182 </dependency>

183

184 <dependency>

185 <groupId>org.jboss.resteasy</groupId>

186 <artifactId>resteasy-netty</artifactId>

187 <version>3.0.7.Final</version>

188 </dependency>

189

190 <dependency>

191 <groupId>org.jboss.resteasy</groupId>

192 <artifactId>resteasy-jdk-http</artifactId>

193 <version>3.0.7.Final</version>

194 </dependency>

195

196 <dependency>

197 <groupId>org.jboss.resteasy</groupId>

198 <artifactId>resteasy-jackson-provider</artifactId>

199 <version>3.0.7.Final</version>

200 </dependency>

201

202 <dependency>

203 <groupId>org.jboss.resteasy</groupId>

204 <artifactId>resteasy-jaxb-provider</artifactId>

205 <version>3.0.7.Final</version>

206 </dependency>

207

208 <dependency>

209 <groupId>org.apache.tomcat.embed</groupId>

210 <artifactId>tomcat-embed-core</artifactId>

211 <version>8.0.11</version>

212 </dependency>

213

214 <dependency>

215 <groupId>org.apache.tomcat.embed</groupId>

216 <artifactId>tomcat-embed-logging-juli</artifactId>

217 <version>8.0.11</version>

218 </dependency>

219

220 <dependency>

221 <groupId>com.esotericsoftware.kryo</groupId>

222 <artifactId>kryo</artifactId>

223 <version>2.24.0</version>

224 </dependency>

225

226 <dependency>

227 <groupId>de.javakaffee</groupId>

228 <artifactId>kryo-serializers</artifactId>

229 <version>0.26</version>

230 </dependency>

231

232 <dependency>

233 <groupId>de.ruedigermoeller</groupId>

234 <artifactId>fst</artifactId>

235 <version>1.55</version>

236 </dependency>

237 </dependencies>

238

239

240 </project>

[复制代码](javascript:void(0);)

View Code

其它注意事项：

dubbo构架中，zk充着“服务注册中心”的角色，所以生产者与消费者的xml配置文件中，都要配置zk地址，如果zk采用集群部署时，配置写法参考下面这样：

<dubbo:registry address="zookeeper://172.28.\*.102:2181?backup=172.28.\*.102:2182,172.28.\*.102:2183"/>

dubbo还有一个管理界面，用于服务治理，包括启用/禁用服务，设置服务的路由规则（即：A地址的Consumer直接调用B机器的Provider，而不是由负载均衡算法分配）等等。

使用方法：将dubbo-admin这个项目编译成war包后，部署到jetty或其它兼容web server即可（当然要修改\WEB-INF\dubbo.properties里zk的地址）

部署完成后，访问管理界面时，默认用户名，密码均是root。

另外dubbo-monitor项目用于性能监控，可以用图表方式直观展示主要性能指标。