

使用java config简化xml配置, DubboConfig.java

```
1. package cn.com.blumoon.springboot.config;
2.
3. import org.springframework.beans.factory.annotation.Value;
4. import org.springframework.context.annotation.Bean;
5. import org.springframework.context.annotation.Configuration;
6.
7. import com.alibaba.dubbo.config.ApplicationConfig;
8. import com.alibaba.dubbo.config.MonitorConfig;
9. import com.alibaba.dubbo.config.ProtocolConfig;
10. import com.alibaba.dubbo.config.ProviderConfig;
11. import com.alibaba.dubbo.config.RegistryConfig;
12. import com.alibaba.dubbo.config.spring.AnnotationBean;
13.
14. @Configuration
15. public class DubboConfig {
16.
17.     public static final String APPLICATION_NAME = "spring-boot-demo";
18.
19.     public static final String ANNOTATION_PACKAGE
20.         = "cn.com.blumoon.springboot.service";
21.
22.     @Value("${dubbo_registry_address}")
23.     private String registryAddress;
24.
25.     @Value("${dubbo_port}")
26.     private int dubboPort;
27.
28.     @Value("${dubbo_register}")
29.     private boolean dubboRegister;
30.
31.     /**
32.      * 提供方应用信息，用于计算依赖关系
33.      *
34.      * @return
35.      */
36.     @Bean
37.     public ApplicationConfig applicationConfig() {
38.         ApplicationConfig applicationConfig = new ApplicationConfig();
39.         applicationConfig.setName(APPLICATION_NAME);
40.         return applicationConfig;
41.     }
42.
43.     /**
44.      * 与<dubbo:annotation/>相当.
45.      * 提供方扫描带有@com.alibaba.dubbo.config.annotation.
46.      * Service的注解类
47.      *
48.      * @return
49.      */
```

```

50. @Bean
51. public static AnnotationBean annotationBean() {
52.     AnnotationBean annotationBean = new AnnotationBean();
53.     annotationBean.setPackage(ANNOTATION_PACKAGE);
54.     return annotationBean;
55. }
56.
57. /**
58.  * 使用zookeeper注册中心暴露服务地址
59.  *
60.  * @return
61.  */
62. @Bean
63. public RegistryConfig registryConfig() {
64.     RegistryConfig registryConfig = new RegistryConfig();
65.     registryConfig.setAddress(registryAddress);
66.     registryConfig.setRegister(dubboRegister);
67.     return registryConfig;
68. }
69.
70. /**
71.  * 用dubbo协议在指定端口暴露服务
72.  *
73.  * @return
74.  */
75. @Bean
76. public ProtocolConfig protocolConfig() {
77.     ProtocolConfig protocolConfig = new ProtocolConfig("dubbo", dubboPort);
78.     // 默认为hessian2,但不支持无参构造函数类,而这种方式效率很低
79.     protocolConfig.setSerialization("java");
80.     return protocolConfig;
81. }
82.
83. /**
84.  * 监控中心配置, protocol="registry", 表示从注册中心发现监控中心地址
85.  *
86.  * @return
87.  */
88. @Bean
89. public MonitorConfig monitorConfig() {
90.     MonitorConfig monitorConfig = new MonitorConfig();
91.     monitorConfig.setProtocol("registry");
92.     return monitorConfig;
93. }
94.
95. /**
96.  * 当ProtocolConfig和服务Config某属性没有配置时,采用此缺省值
97.  *
98.  * @return
99.  */
100. @Bean
101. public ProviderConfig providerConfig() {
102.     ProviderConfig providerConfig = new ProviderConfig();
103.     providerConfig.setTimeout(10000);

```

```
104.     providerConfig.setThreadpool("fixed");
105.     providerConfig.setThreads(100);
106.     providerConfig.setAccepts(1000);
107.     return providerConfig;
108. }
109.
110. }
```