

Spring boot dubbo starter

鉴于目前后端服务以Dubbo为主，而新增的服务为Spring boot。因为Dubbo的配置采用XML形式，而Spring boot倡导的是Java Config形式。所以个人开发了这个简单插件，目的是省去XML配置形式，完全采用注解方式。

传统Spring boot与Dubbo整合方式

1.引入具体依赖

```
<!-- 具体dubbo服务 -->
<dependency>
    <groupId>demo-skyWalking-api-impl</groupId>
    <artifactId>demo-skyWalking-api-impl-test</artifactId>
    <version>1.0-SNAPSHOT</version>
</dependency>
<!-- zookeeper -->
<dependency>
    <groupId>com.github.adyliu</groupId>
    <artifactId>zkclient</artifactId>
    <version>2.0</version>
</dependency>
<dependency>
    <groupId>org.apache.zookeeper</groupId>
    <artifactId>zookeeper</artifactId>
    <version>3.3.6</version>
</dependency>
<!-- dubbo -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>dubbo</artifactId>
    <version>2.5.3</version>
    <exclusions>
        <exclusion>
            <groupId>org.springframework</groupId>
            <artifactId>spring</artifactId>
        </exclusion>
    </exclusions>
</dependency>
```

2.在resources目录下增加applicationContext.xml文件，该文件中配置了需要引用后端dubbo服务的信息，如下图所示

```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
    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">

    <dubbo:application name="consumer-of-helloworld-app" />
    <dubbo:registry address="zookeeper://127.0.0.1:2181" />
    <dubbo:reference id="demoService" interface="com.demo.service.DemoService" timeout="3000" retries="0"/>
</beans>
```

3.在Spring boot工程主类上引入XML,如下图所示

```
@ImportResource({"classpath:applicationConsumer.xml"})
@RestController
@SpringBootApplication
public class BootDemoApplication {
    ...
}
```

4.调用具体服务

```
@RestController
public class DemoController {

    @Autowired
    DemoService demoService;

}
```

以上三个步骤为spring boot整合dubbo进行开发，可以发现依旧采用的是XML方式，这种与spring boot基于java config注解形式的风格不符。

基于注解Spring boot与Dubbo整合方式

1.引入具体依赖，去掉了dubbo的依赖，加入了spring-boot-dubbo-starter

```
<!-- spring-boot-dubbo-starter -->
<dependency>
    <groupId>com.answern</groupId>
    <artifactId>spring-boot-dubbo-starter</artifactId>
    <version>1.0-SNAPSHOT</version>
</dependency>

<!-- 具体dubbo服务 -->
<dependency>
    <groupId>demo-skyWalking-api-impl</groupId>
    <artifactId>demo-skyWalking-api-impl-test</artifactId>
    <version>1.0-SNAPSHOT</version>
</dependency>

<!-- zookeeper -->
<dependency>
    <groupId>com.github.adyliu</groupId>
    <artifactId>zkclient</artifactId>
    <version>2.0</version>
</dependency>
<dependency>
    <groupId>org.apache.zookeeper</groupId>
    <artifactId>zookeeper</artifactId>
    <version>3.3.6</version>
</dependency>
```

2.在Spring boot工程主类上加入@EnableDubboClient注解，该注解功能是开启Dubbo客户端功能。

```
@EnableDubboClient
@RestController
@SpringBootApplication
public class BootDemoApplication {

    ...

}
```

3.调用具体服务

```
@RestController
public class DemoController {

    @Reference(interfaceClass = com.demo.service.DemoService.class)
    DemoService demoService;

}
```

综述

该插件只是完成了基本的调用功能，后续还需要继续完善。