**SpringBoot非官方教程 | 第二十二篇： 创建含有多module的springboot工程**

2017年04月30日 17:02:39

阅读数：35305

转载请标明出处：   
<http://blog.csdn.net/forezp/article/details/71024153>   
本文出自[方志朋的博客](http://blog.csdn.net/forezp" \t "_blank)

这篇文章主要介绍如何在springboot中如何创建含有多个module的工程，栗子中含有两个 module，一个作为libarary. 工程，另外一个是主工程，调用libary .其中libary jar有一个服务，main工程调用这个服务。

**创建根工程**

创建一个maven 工程,其pom文件为：

<?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.forezp</groupId>

<artifactId>springboot-multi-module</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>pom</packaging>

<name>springboot-multi-module</name>

<description>Demo project for Spring Boot</description>

</project>

需要注意的是packaging标签为pom 属性。

**创建libary工程**

libary工程为maven工程，其pom文件的packaging标签为jar 属性。创建一个service组件,它读取配置文件的 service.message属性。

@ConfigurationProperties("service")

public class ServiceProperties {

/\*\*

\* A message for the service.

\*/

private String message;

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

}

提供一个对外暴露的方法：

@Configuration

@EnableConfigurationProperties(ServiceProperties.class)

public class ServiceConfiguration {

@Bean

public Service service(ServiceProperties properties) {

return new Service(properties.getMessage());

}

}

**创建一个springbot工程**

引入相应的依赖,创建一个web服务：

@SpringBootApplication

@Import(ServiceConfiguration.class)

@RestController

public class DemoApplication {

private final Service service;

@Autowired

public DemoApplication(Service service) {

this.service = service;

}

@GetMapping("/")

public String home() {

return service.message();

}

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}

在配置文件application.properties中加入：

service.message=Hello World

打开浏览器访问：[http://localhost:8080/](http://localhost:8080/" \t "_blank);浏览器显示：

Hello World

说明确实引用了libary中的方法。