Spring Boot自定义Starter

Spring Boot是用来简化新Spring应用的初始搭建以及开发过程的全新框架，该框架使用特定方式来进行配置，从而使开发人员不再需要定义样板化的配置。Spring Boot在启动时会在starter包中寻找：

*resources/META-INF/spring.factories*

文件，根据文件中配置的jar包扫描所依赖的JAR包(类似于Service Provider Interface机制)。

1. 创建Spring Boot项目，并添加依赖到pom.xml文件

*<dependencies>*

*<dependency>*

*<groupId>org.springframework.boot</groupId>*

*<artifactId>spring-boot-autoconfigure</artifactId>*

*<version>1.5.9.RELEASE</version>*

*</dependency>*

*</dependencies>*

其中创建的自定义Spring Boot信息如下：

*<artifactId>spring-boot-starter-fys</artifactId>*

*<version>1.0-SNAPSHOT</version>*

1. 编写Service类，Starter中要实现的功能，示例仅定义sayHelllo方法输出信息

*@Setter*

*public class HelloService {*

*private String msg;*

*private boolean show = true;*

*public String sayHello() {*

*return show ? "Hello." + msg : "Hidden";*

*}*

*}*

1. 创建属性列，prefix=“hello”代表该项目在属性文件中配置的前缀，如下：

*@Getter*

*@Setter*

*@ConfigurationProperties(prefix = "hello")*

*public class HelloProperties {*

*private String msg = "Hello FYS";*

*private boolean show = true;*

*}*

1. 创建自动化配置类，相当于Java配置类，在这里创建Bean，在该示例中获取配置文件中属性

*@Configuration*

*@EnableConfigurationProperties(HelloProperties.class)*

*@ConditionalOnClass(HelloService.class)*

*@ConditionalOnProperty(*

*prefix = "hello",*

*value = "enabled",*

*matchIfMissing = true*

*)*

*public class HelloAutoConfiguration {*

*@Autowired*

*private HelloProperties helloProperties;*

*@Bean*

*@ConditionalOnMissingBean(HelloService.class)*

*public HelloService helloService() {*

*System.out.println(">>>> The HelloService Not Found, Execute Create New Bean.");*

*HelloService helloService = new HelloService();*

*helloService.setMsg(helloProperties.getMsg());*

*helloService.setShow(helloProperties.isShow());*

*return helloService;*

*}*

*}*

1. 在META-INF目录下创建spring.factories，加载该项目中自动化配置类

*org.springframework.boot.autoconfigure.EnableAutoConfiguration=com.fys.springstarter.HelloAutoConfiguration*

1. 测试程序中，引入starter，创建spring-boot项目，添加依赖：

*<parent>*

*<groupId>org.springframework.boot</groupId>*

*<artifactId>spring-boot-starter-parent</artifactId>*

*<version>1.5.9.RELEASE</version>*

*</parent>*

*<dependencies>*

*<dependency>*

*<groupId>com.fys</groupId>*

*<artifactId>spring-boot-starter-fys</artifactId>*

*<version>1.0-SNAPSHOT</version>*

*</dependency>*

*<dependency>*

*<groupId>org.springframework.boot</groupId>*

*<artifactId>spring-boot-starter-web</artifactId>*

*<version>1.5.9.RELEASE</version>*

*</dependency>*

*</dependencies>*

1. 测试程序

*@RestController*

*@SpringBootApplication*

*public class HelloWorldApplication {*

*@Autowired*

*private HelloService helloService;*

*@RequestMapping("/")*

*public String sayHello() {*

*return helloService.sayHello();*

*}*

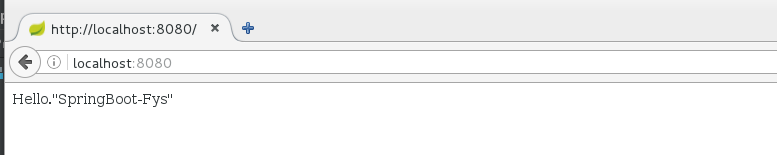
*public static void main(String[] args) {*

*SpringApplication.run(HelloWorldApplication.class, args);*

*}*

*}*

1. 浏览器中测试如下：



<http://objcoding.com/2018/02/02/Costom-SpringBoot-Starter/>