记录时间：2019.09.30
公司名称： 火鹰科技有限公司
学生姓名： 曾伟涛
记录类型： 学习笔记
记录阶段：2019.10.07-2019.10.13

spring boot注解

@SpringBootApplication

一个复合注解，包括@ComponentScan，@SpringBootConfiguration，@EnableAutoConfiguration

继承自@Configuration，二者功能也一致，标注当前类是配置类，并会将当前类内声明的一个或 多个以@Bean注解标记的方法的实例纳入到srping容器中，并且实例名就是方法名。

@EnableAutoConfiguration

启动自动的配置。

@EnableAutoConfiguration

Springboot根据你添加的jar包来配置你项目的默认配置，比如根据spring-boot-starter-web，来判断你的项目是否需要添加了webmvc和tomcat，就会自动的帮你配置web项目中所需要的默认配置。在下面博客会具体分析这个注解，快速入门的demo实际没有用到该注解。

@ComponentScan

扫描当前包及其子包下被@Component，@Controller，@Service，@Repository注解标记的类并纳入到spring容器中进行管理。是以前的<context:component-scan>（以前使用在xml中使用的标签，用来扫描包配置的平行支持）。所以本demo中的User为何会被spring容器管理

@ResponseBody

表示该方法的返回结果直接写入HTTP response body中，一般在异步获取数据时使用，用于构建RESTful的api。在使用@RequestMapping后，返回值通常解析为跳转路径，加上@esponsebody后返回结果不会被解析为跳转路径，而是直接写入HTTP response body中。比如异步获取json数据，加上@Responsebody后，会直接返回json数据。该注解一般会配合@RequestMapping一起使用

@Controller

用于定义控制器类，在spring项目中由控制器负责将用户发来的URL请求转发到对应的服务接口（service层），一般这个注解在类中，通常方法需要配合注解

@RequestMapping

@RestController

用于标注控制层组件(如struts中的action)，@ResponseBody和@Controller的合集。

@Service

一般用于修饰service层的组件

@Repository

使用@Repository注解可以确保DAO或者repositories提供异常转译，这个注解修饰的DAO或者repositories类会被ComponetScan发现并配置，同时也不需要为它们提供XML配置项。

@Bean

用@Bean标注方法等价于XML中配置的bean。

@Value

注入Spring boot application.properties配置的属性的值。

springboot中的两大核心技术

写了这么多代码相信还不知道springboot的两大核心技术是什么吧？在这里会以例子的形式来说两大技术。

1.ioc容器

```
@Controller
public class HelloWroldController{
 @RequestMapping("/sayHello.html")
 public @ResponseBody String Say(String name){
 return "hello" + name;
 }
}
```

上面的例子中， Spring Container 在容器中初始化 HelloworldController 实例后，对于客户端 发起的 / sayhello.html 请求， 会执行 say 方法，并自动将请求参数按照 say 方法声明的名称一一对 应上。

提示：Spring 通常提供一些@Contraller、@Service、@Component、@Configuration 注解，只 有使用这些注解的类才会引起 Spring 容器的注意，并根据注解含义来管理和增强对象。

Spring 可以管理和增强任意对象，如常见的@Service 注解对象，通常用来处理业务逻辑， Spring Container 往往会增强这类对象的事务控制能力。 容器管理还可以为被管理的

Bean 提供其他被管理和被增强的 Bean，如一个已经被@Service 注解的 UserService 类， 在 HelloworldController 类中，使用@Autowired 自动注入这个实例：

```
@Controller
public class HelloworldController {
 @Autowired UserService userService;
 }
```

2.AOP:面向切面编程

AOP： 上面提到的对象增强离不开 AOP 技术， AOP (Aspect Oriented Programming) 指面 向切面编程，通过预编译方式或者运行时刻对目标对象动态地添加功能。 AOP 分离了企业应用 的业务逻辑和系统级服务，比如事务服务，还有应用系统的审计、安全访问等代码。 比如要实 现用户访问控制，可以对每个 Controller 的方法使用一个自定义的注解 Function，用 SpringAOP 向 Controller 每个方法动态地添加用户权限校验功能，类似如下：

```
@RequestMapping (” / sayhello . html” )
public @ResponseBody String say (String name) {
return ” hello ” +name;
@RequestMapping （ ” / adduser . html ” ）
 @Function (” user . add” )
 public @ResponseBody String addUser (String name ) {
 .......
 }
```

注解 Function 是自定义一个注解，接受一个字符串，表示 Controller 方法对应的业务功能。 用户是否能访问“user.add” 功能， 将在数据库中配置。 使用 AOP 对所有的 Controller 进行增强：

```
@Configuration
@Aspect public class RoleAccessConf ig {
@Around (”within(@org.springframework.stereotype .Controller *) && @annotati on (function )”)
public Object functionAccessCheck(final ProceedingJoinPoint pjp, Function function ) throws Throwable {
 if (function ! =null) {
 String functionName =function.value ();
 if( !canAccess (functionName )) {
 Methods工gnature ms= (MethodSignature) pjp . getSignature ();
```

```
9  throw new RuntimeException (”Can not Acces s ” +ms . getMethod
());
10 / / 继续处理原有调用 Object o = pj p . proceed () ; return o ;
11 protected boolean canAccess (String functionName) {
12 if(functionName.length()==0) {
13 / /总是允许访问
14 return true;
15 }else{
16 / /取出当前用户对应的所有角色，从数据库中查询角色是否有访问 functior
Name 的权限
17 return false ;
18 }
```

这段代码比较复杂， 可以按照这个顺序理解：

@Configuration 注解成功引起 Spring 容器的注意：

@Aspect 1.1: Spring 容器知道，这是一个 AOP 类：

@Around 是 AOP 的一种具体方式，这里只需要知道，它 能对目标方法调用前和调用后进行处理： within(@org叩ring台amework.stereotyp已.Controller * ）可以理解为对所有使用@Controller 注解的类进行 AOP;

@annotation（句nction） 表示另外一个条件，也就是对具有 function 参数对应的注解方法 进行 AOP; functionAccessCheck 是实现 AOP 的具体代码。 这里举例只是说明 Spring 通过 AOP 来增强 Bean 的功能特性。

springboot

Spring Boot 是通过 **Starter** 来提供系统级服务， Spring Boot 已经提供了一系列 Starter， 你也可 以开发自己的 Starter。 比如需要开发一个 Web 应用 ， 只要在 pom.xml 中声明一下即可。

```
1 <dependency>
2 <groupid>org.springframework.boot</groupid>
3 <artifactid>spring-boot-starter-web</artifactid>
4 </dependency>
```

常用的Starter如下：红色为重点

名称	作用
spring-boot-starter-web	web开发支持，默认是Tomcat8
spring-boot-starter-aop	AOP开发支持，使用AspectJ

spring-boot-starter-jdbc	Spring JDBC
spring-boot-starter-data-jpa	JPA 方式访问数据库，使用 Hibernate 作为 JPA 实现
spring-boot-starter-data-elasticsearch	集成 Elasticsearch，默认访问 localhost:9200
spring-boot-starter-data-redis	集成 Redis，使用 JRedis，默认连接 localhost:6379
spring-boot-starter-cache	缓存， 支持多种缓存方式，如本地的、 Redis、 Ehcache 等
spring-boot-devtools	应用程序快速重启的工具，提升开发体验
spring-boot-starter-data-mongodb	集成 MongoDB，默认访问 mongodb://localhosvtest
spring-boot-starter-data-neo4j	集成 neo4j， 默认访问 localhost:7474
spring-boot-starter-data-gemfire	集成分布式缓存
spring-boot-starter-data-so lr	基于 Apache lucene 的搜索平台，默认访问 http://localhost:8983/solr
spring-boot-starter-data-cassandra	集成 Cassandra，默认访问 localhost:7474
spring-boot-sta此er-data-ldap	集成 ldap
spring-boot-starter-activemq	消息集成 ActiveMQ 支持
spring-boot-starter-amqp	消息集成 AMQP 协议支持，如支持 RabbitMQ
spring-boot-starter-jta-atomikos	分布式事务支持，使用 atomikos
spring-boot-starter-jta-bitronix	一个开源的分布式事务支持
spring-boot-starter-test	包含 JUnit、 Spring Test、 Hamcrest、 Mockito 等测试工具
spring-boot-starter-webservices	webservice 支持
spring-boot-starter-websocket	websocket 支持
spring-boot-starter-jersey	REST 应用和 Jersey 支持
spring-boot” starter-freemarker	Freemaker 支持

相比于 Spring, Spring Boot 具有以下优点：

- 实现约定大于配置，是一个低配置的应用系统框架。
- 不像 Spring 那样“地狱般的配置 体验”， Spring Boot 不需要配置或者极少配置，就能使用 Spring 大量的功能。

- 提供了内置的 Tomcat 或者 Jetty 容器。
- 通过依赖的 jar 包管理、自动装配技术，容易支持与其他技术体系、工具集成。
- 支持热加载，开发体验好。也支持 Spring Boot 系统监控，方便了解系统运行状况。

热部署

修改类时必须再次重新运行应用， 对于开发者来说非常不方便。 Spring Boot 提供了**spring-boot-devtools**， 它能在修改类或者配置文件的时候自动重新加载 Spring Boot 应用。

添加REST支持

对于系统并非一个单一的 Web 应用，而是由多个系统构成的。系统之间的调用方式有很多， RESTFul 也是一种很好的方式。Spring Boot 能很方便地支持 阻STFul 应用 。

在前几周的周报里的代码中，我们不少发现运用了**@RestController** 这个注解，而没有使用@Controller和@ResponseBody，其实我们可以把@RestController理解为@Controller+@ResponseBody。

> 对于多个系统互相访问，最好不要直接访问对方的数据库，而应该采用类似RESTFu1 架构，封装了逻辑的接口 。
> 这样，对方系统的数据库变更，业务逻辑变化或者版本升级， 都不会影响其他系统。

工作内容:

1.写公司项目

2.学习spring boot中mvc核心技术