Springboot框架自带

@SpringBootApplication：启动springboot应用

属性

exclude=[ ]class 排除指定类的自动加载

excludeName=[ ]str 排除指定全路径类名自动加载

scanbasePackages=[ ]str 扫描指定包，默认情况仅扫描application所在包及子包，指定后不仅可以扫描子包，还可以扫描其他包

scanbasePackagesClasses=[]class 扫描类所在包下的所有组件

proxyBeanMethods=bool 容器中bean是否使用代理，默认true，调用同一对象，false不使用，每次调用都是new新对象

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@RestController：Controlle层，@Responst+@Controller，开放地址

属性

value="str" Controller在bean容器中的名称，不写的情况下默认为class名小写首字母

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Controller：Controlle层，开放地址

属性

value="str" Controller在bean容器中的名称，不写的情况下默认为class名小写首字母

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Service：service层，业务逻辑处理

属性

value="str" Service在bean容器中的名称，不写的情况下默认为class名小写首字母

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Component：组件，组件不好归类controller，service还是responsitory的时候使用。

属性

value=str 组件在Bean容器中的组件名

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@RequestMapping：处理请求地址映射，用在controller层的类和方法上

属性

value=str 路径

method RequestMethod.GET get请求，也可以写成@GetMapping(value=str)

RequestMethod.POST post请求，或写成@PostMapping(value=str)

RequestMethod.DELETE 从post封装的delete请求，或写成@DeleteMapping(value = str)

RequestMethod.PUT 从post封装的put请求，或写成@PutMapping(value = str)

RequestMethod.PATCH

RequestMethod.HEAD

RequestMethod.OPTIONS

RequestMethod.TRACE

headers 指定request中必须包含某些指定的header值，才能让该方法处理请求

consumes 指定处理请求的提交内容类型（Content-Type），如application/json,text/html;

produces 指定返回的内容类型，仅当request请求头中的(Accept)类型中包含该指定类型才返回

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Resource：默认按name注入bean

属性

name = "str" Resource在Bean容器中的组件名

type=xxx.class 组件类型

description= "str" 描述

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Autowired：默认按type注入bean

属性

required = boolean 引入的bean必须在容器中存在，否则抛异常，默认true

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Qualifier：按名称注入bean，可与@Autowired结合使用，解决按type找到多个bean的问题

属性

value="str" bean名称，必需！

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Import：springboot专供，将类加入bean容器，bean名称为全类名，写在congfig类上

属性

Value={[]class} 引入的bean的class。

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@ImportResource：将xml形式的bean配置加入java配置中，从而加入bean容器，写在config类上

属性

value=str 引入的classpath下的xml文件。

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@ResponseBody：返回值，用于controller层，结果直接写入HTTP response body中

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Repository：DAO层，数据处理层。

属性

value="str" repository在bean容器中的名称，不写的情况下默认为class名小写首字母

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Configuration: Bean配置类

属性

value="str" config在bean容器中的名称，不写的情况下默认为class名小写首字母

proxyBeanMethods=boolean 容器中bean是否使用代理，默认true，调用同一对象，false不使用，每次调用都是new新对象

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Bean:将被注解的类加入bean容器中。

属性

value =str bean名称

name={str1,str2,str3}/str bean别名，和bean名称一样都可以正常使用

autowireCandidate 当前bean在被其他对象作为自动注入对象的时候，是否作为候选bean，默认值是true

initMethod=str bean创建时执行方法，直接写bean中的方法名即可

destroyMethod=str bean销毁时执行方法，直接写bean中的方法名即可

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@ConditionalOnBean：按条件引入bean入容器

属性

name=str bean容器中有名为str的bean，则加载被注解的类和方法入容器

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@ConditionalOnMissingBean：按条件引入bean入容器

属性

name=str bean容器中无名为str的bean，则加载被注解的类和方法入容器

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Value:向bean中属性注入值，常常用在实体类中【不推荐，推荐使用配置绑定】

属性

value =str 属性值

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@ConfigurationProperties:按application.properties中内容配置绑定到组件中，写在@component上

属性

prefix 属性值前缀，必需

value

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@EnableConfigurationProperties:按application.properties中内容注入属性值到组件中并注入bean容器，写在component上。且被注入的类上必须写@ConfigurationProperties(prefix=...)

属性

[]xxx.class 填入xxx.class，完成对xxx.class的配置绑定并加入bean容器

Lombok插件

实现java类的get/set,constructor在编译时自动产生，某些ide需要加入插件才可正常使用此插件。Pom.xml加入依赖

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</dependency>

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Data：编译时自动生成get/set

属性

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@ToString:编译时自动生成toString()重写方法，默认输出所有属性

属性

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@AllArgsConstructor：编译时生成全参构造器

属性

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@NoArgsConstructor：编译时生成无参构造器

属性

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

@Slf4j：日志插件

使用

log.info(str)：消息日志

log.warn(str)：警告日志

log.error(str)：错误日志

属性

Spring-boot-devtools插件

实现应用热更新Ctrl+F9重启应用。Pom.xml加入依赖

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

Redis框架

Mybatis框架

数据库中间层框架

Kafka框架

消息队列框架