spring框架中的核心：IOC 可以帮助我们管理对象的依赖关系，极大减少对象的耦合性，而 AOP 的切面编程功能可以更方面的使用动态代理来实现各种动态方法功能（如事务、缓存、日志等）。

Spring Boot应用本质上就是一个基于Spring框架的应用，它是Spring对“约定优先于配置”理念的最佳实践产物，它能够帮助开发者更快速高效地构建基于Spring生态圈的应用。

Spring Boot是 Spring 开源组织下的一个子项目，也是 Spring 组件一站式解决方案，主要是为了简化使用 Spring 框架的难度，简省繁重的配置。

Spring Boot提供了各种组件的启动器（starters），开发者只要能配置好对应组件参数，Spring Boot 就会自动配置，让开发者能快速搭建依赖于 Spring 组件的 Java 项目。

什么是springboot

        用来简化spring应用的初始搭建以及开发过程 使用特定的方式来进行配置（properties或yml文件）

                创建独立的spring引用程序 main方法运行

                嵌入的Tomcat 无需部署war文件

                简化maven配置

                自动配置spring添加对应功能starter自动化配置

springboot常用的starter有哪些

        spring-boot-starter-web 嵌入tomcat和web开发需要servlet与jsp支持

        spring-boot-starter-data-jpa 数据库支持

        spring-boot-starter-data-redis redis数据库支持

        spring-boot-starter-data-solr solr支持

        mybatis-spring-boot-starter 第三方的mybatis集成starter

springboot自动配置的原理

        在spring程序main方法中 添加@SpringBootApplication或者@EnableAutoConfiguration

        会自动去maven中读取每个starter中的spring.factories文件  该文件里配置了所有需要被创建spring容器中的bean

springboot读取配置文件的方式

        springboot默认读取配置文件为application.properties或者是application.yml

springboot集成mybatis的过程

        添加mybatis的starter maven依赖

                <dependency>

                        <groupId>org.mybatis.spring.boot</groupId>

                        <artifactId>mybatis-spring-boot-starter</artifactId>

                        <version>1.2.0</version>

                </dependency>

        在mybatis的接口中 添加@Mapper注解

        在application.yml配置数据源信息

springboot如何添加【修改代码】自动重启功能

        添加开发者工具集=====spring-boot-devtools

https://blog.csdn.net/u013605060/article/details/80255192

问：你觉得 Spring Boot 最大的优势是什么呢？

答：Spring Boot 的最大的优势是“约定优于配置“。“约定优于配置“是一种软件设计范式，开发人员按照约定的方式来进行编程，可以减少软件开发人员需做决定的数量，获得简单的好处，而又不失灵活性。

问：Spring Boot 中 “约定优于配置“的具体产品体现在哪里。

答：Spring Boot Starter、Spring Boot Jpa 都是“约定优于配置“的一种体现。都是通过“约定优于配置“的设计思路来设计的，Spring Boot Starter 在启动的过程中会根据约定的信息对资源进行初始化；Spring Boot Jpa 通过约定的方式来自动生成 Sql ，避免大量无效代码编写。具体详细可以参考：Spring Boot 为什么这么火？

问：Spring Boot Starter 的工作原理是什么？

答：Spring Boot  在启动的时候会干这几件事情：

Spring Boot 在启动时会去依赖的 Starter 包中寻找 resources/META-INF/spring.factories 文件，然后根据文件中配置的 Jar 包去扫描项目所依赖的 Jar 包。

根据 spring.factories 配置加载 AutoConfigure 类

根据 @Conditional 注解的条件，进行自动配置并将 Bean 注入 Spring Context

总结一下，其实就是 Spring Boot 在启动的时候，按照约定去读取 Spring Boot Starter 的配置信息，再根据配置信息对资源进行初始化，并注入到 Spring 容器中。这样 Spring Boot 启动完毕后，就已经准备好了一切资源，使用过程中直接注入对应 Bean 资源即可。

**微服务同时调用多个接口，怎么支持事务的啊？**

支持分布式事务，可以使用Spring Boot集成 Aatomikos来解决，但是我一般不建议这样使用，因为使用分布式事务会增加请求的响应时间，影响系统的TPS。一般在实际工作中，会利用消息的补偿机制来处理分布式的事务。

Spring Cache 三种常用的缓存注解和意义？

@Cacheable ，用来声明方法是可缓存，将结果存储到缓存中以便后续使用相同参数调用时不需执行实际的方法，直接从缓存中取值。

@CachePut，使用 @CachePut 标注的方法在执行前，不会去检查缓存中是否存在之前执行过的结果，而是每次都会执行该方法，并将执行结果以键值对的形式存入指定的缓存中。

@CacheEvict，是用来标注在需要清除缓存元素的方法或类上的，当标记在一个类上时表示其中所有的方法的执行都会触发缓存的清除操作。

Spring Cloud 是一系列框架的有序集合。它利用 Spring Boot 的开发便利性巧妙地简化了分布式系统基础设施的开发，如服务发现注册、配置中心、消息总线、负载均衡、断路器、数据监控等，都可以用 Spring Boot 的开发风格做到一键启动和部署。

Spring Cloud 是为了解决微服务架构中服务治理而提供的一系列功能的开发框架，并且 Spring Cloud 是完全基于 Spring Boot 而开发，Spring Cloud 利用 Spring Boot 特性整合了开源行业中优秀的组件，整体对外提供了一套在微服务架构中服务治理的解决方案。

用一组不太合理的包含关系来表达它们之间的关系。

Spring ioc/aop > Spring > Spring Boot > Spring Cloud

https://blog.csdn.net/yuzongtao/article/details/84295732

sql中的#和$的区别：

1、用来传入参数，sql在解析的时候会加上” “,当成字符串来解析 ，如这里 role\_id = “roleid”;

2、#{}能够很大程度上防止sql注入;

延伸：

1、用{roleId,jdbcType=INTEGER},那么sql在解析的时候值为roleId = roleId，执行时会报错;

2、${}方式无法防止sql注入;

3、$一般用入传入数据库对象，比如数据库表名;

4、能用#{}时尽量用#{};

注意：

mybaties排序时使用order by 动态参数时需要注意，使用${}而不用#{};

默认情况下，使用#{}格式的语法会导致MyBatis创建预处理语句属性并以它为背景设置安全的值（比如?）。这样做很安全，很迅速也是首选做法，有时你只是想直接在SQL语句中插入一个不改变的字符串。比如，像ORDER BY，你可以这样来使用：

ORDER BY ${columnName};  
这里MyBatis不会修改或转义字符串。

重要：接受从用户输出的内容并提供给语句中不变的字符串，这样做是不安全的。这会导致潜在的SQL注入攻击，因此你不应该允许用户输入这些字段，或者通常自行转义并检查。

Spring Boot不但能创建传统的 war 包应用，还能创建独立的不依赖于任何外部容器（如：tomcat）的独立应用，使用 java -jar 命令就能启动。同时，Spring Boot也提供了一个命令行工具来执行 Spring 的脚本。

特点：

1.独立运行：Spring Boot内嵌了各种 Servlet 容器，Tomcat、Jetty等，现在不再需要打成 war 包部署到容器中，Spring Boot 只要打成一个可执行的jar包就能独立运行，所有的依赖包都在一个 jar 包内。

2.简化maven配置:现在只要依赖 spring-boot-starter-web 启动器包，它包含所有 web 开发所有的依赖，就能拥有 Spring Web 的能力，极大简少了 maven 对依赖的配置。

使用 Maven 命令：mvn dependency:tree 也可以看到完整的依赖树

3.自动配置

无代码生成和xml配置

应用监控

缺点：

1、需要非常了解 Spring Boot 的核心技术原理，不然一旦遇到问题就很棘手。因为什么东西都集成了，自动配置化。这样，需要对配置信息非常熟悉，要找起问题来不是很容易。

2、从原始 Spring 项目很难平滑迁移至 Spring Boot 框架上来，因为有些历史老旧的 XML 配置无法通过 Java 来配置，还需要额外的 XML 文件就不是很完美。再比如，之前是独立的 Tomcat，什么参数都在线上配置好了，你改为内置的 Tomcat 就会遇到很多问题。

SpringApplication只是将一个典型的Spring应用的启动流程进行了扩展。

SpringApplication.run(.class, args)；