### SpringBoot

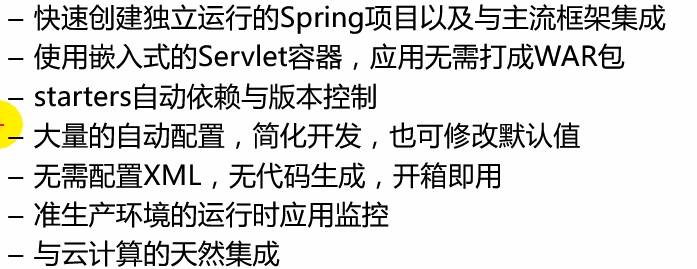
一, SpringBoot入门

1, SpringBoot来简化Spring应用开发, 约定大于配置, 去繁从简, Spring全家桶时代,

SpringBoot -- > J2EE一站式解决方案

SpringCloud -- > 分布式整体解决方案

2, 优点



3, 微服务

一个应用应该是一组小型服务, 可以通过Http的方式进行互通

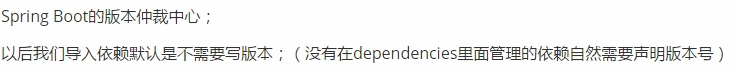
微服务: 每一个功能元素最终都是一个可独立替换和独立升级的软件单元

4, 环境约束





5, 版本仲裁

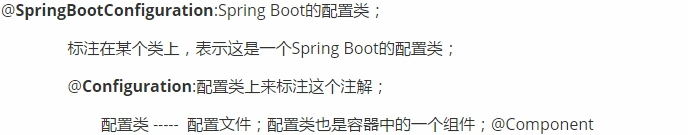


6, pom文件介绍

|  |
| --- |
| *<!--  配置SpringBoot的父类  里面管理了几乎所有项目中需要使用的jar包的版本, 默认是不需要写版本号的, 如果没有在这个父类中则需要 -->* <parent>  <artifactId>spring-boot-starter-parent</artifactId>  <groupId>org.springframework.boot</groupId>  <version>1.5.9.RELEASE</version> </parent>   <dependencies>  *<!--  导入SpringBoot依赖  启动器:  spring-boot-starter(启动器): springboot 场景启动器, 帮我们导入了web模块正常运行所依赖的组件  启动器描述:  SpringBoot将所有的功能场景都抽取出来, 做成一个个starter(启动器), 只需要在项目中引入这些starter,  相关的场景的所有依赖, 都会导入进来, 要用什么功能就导入什么场景的启动器  -->* <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency> </dependencies>  <build>  <plugins>  *<!--  简化部署:  这个插件可以在使用package打包时, 打成一个可执行的jar包  使用 java -jar [jar包] , 运行  -->* <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  </plugins> </build> |

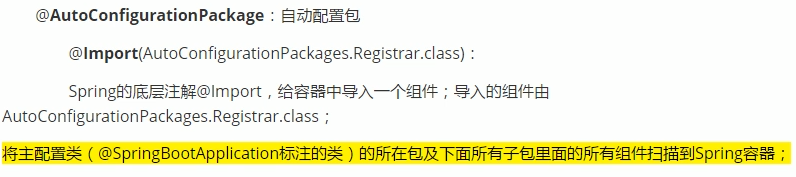
7, 注解解析

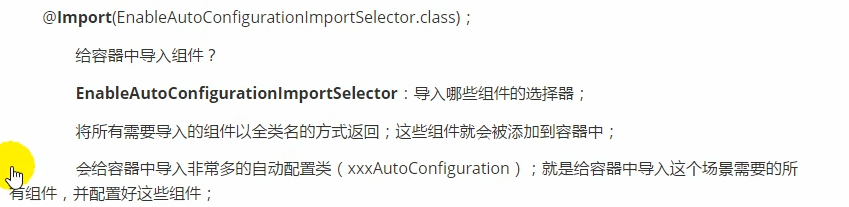
@SpringBootConfiguration



@EnableAutoConfiguration







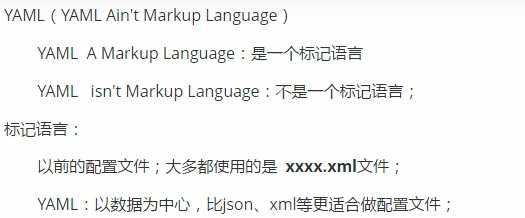


8, 快速开始

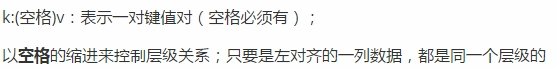


二, SpringBoot配置

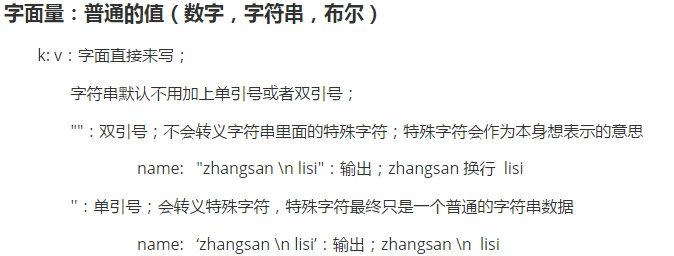
1, YAML简介



基本语法: 大小写敏感



值的写法

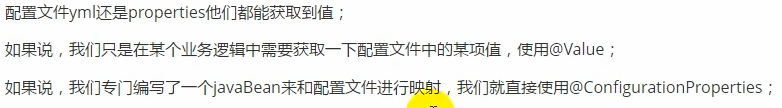






2, 获取值注解区别





3, 导入配置文件

|  |
| --- |
| */\*\*  \* 导入指定的配置文件  \*/* @PropertySource(value = {"classpath:dbconfig.properties"}) @Validated public class YmlBean {   *//@Value("${ymlbean.name}") // 从配置文件中获取  //@Email* @Value("${jdbc.username}")  private String name; |
| */\*\* 启动类上增加  \* @ImportResource: 指定外部xml配置文件,  \* 推荐使用配置类的方式! 使用@Bean进行注入  \*/* @ImportResource(locations = {"classpath:bean.xml"}) @SpringBootApplication public class DemoApplication {   public static void main(String[] args) {  SpringApplication.*run*(DemoApplication.class, args);  }  } |

4, 配置文件随机数

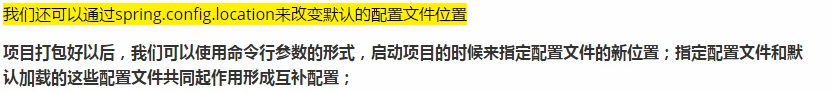


5, 激活指定profile



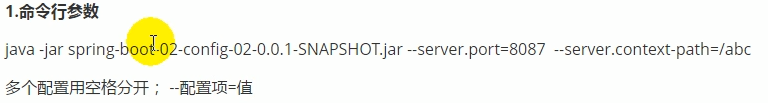
6, 文件的加载顺序, 优先级, 右高到底





7, 外部配置文件的加载顺序



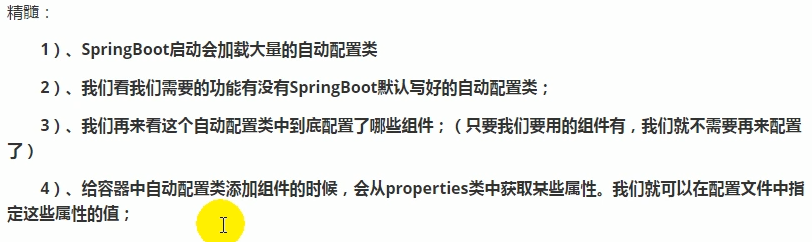


1. 官网配置文件的网址：

<https://docs.spring.io/spring-boot/docs/current/reference/html/appendix-application-properties.html#server.servlet.application-display-name>

9, 自动配置的原理





扩展

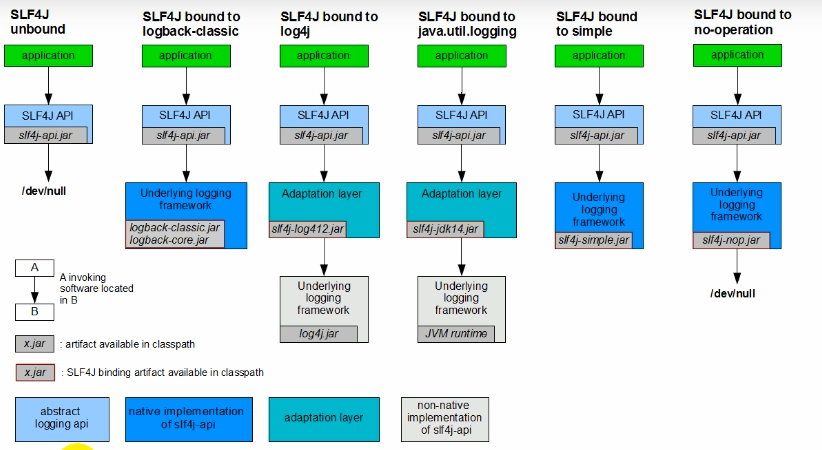


三, SpringBoot与日志

1. 简介

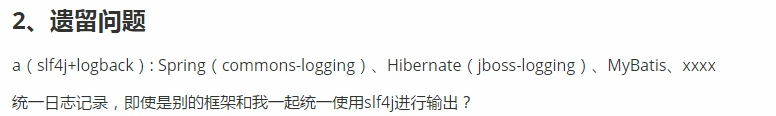


2，日志框架之间的关系

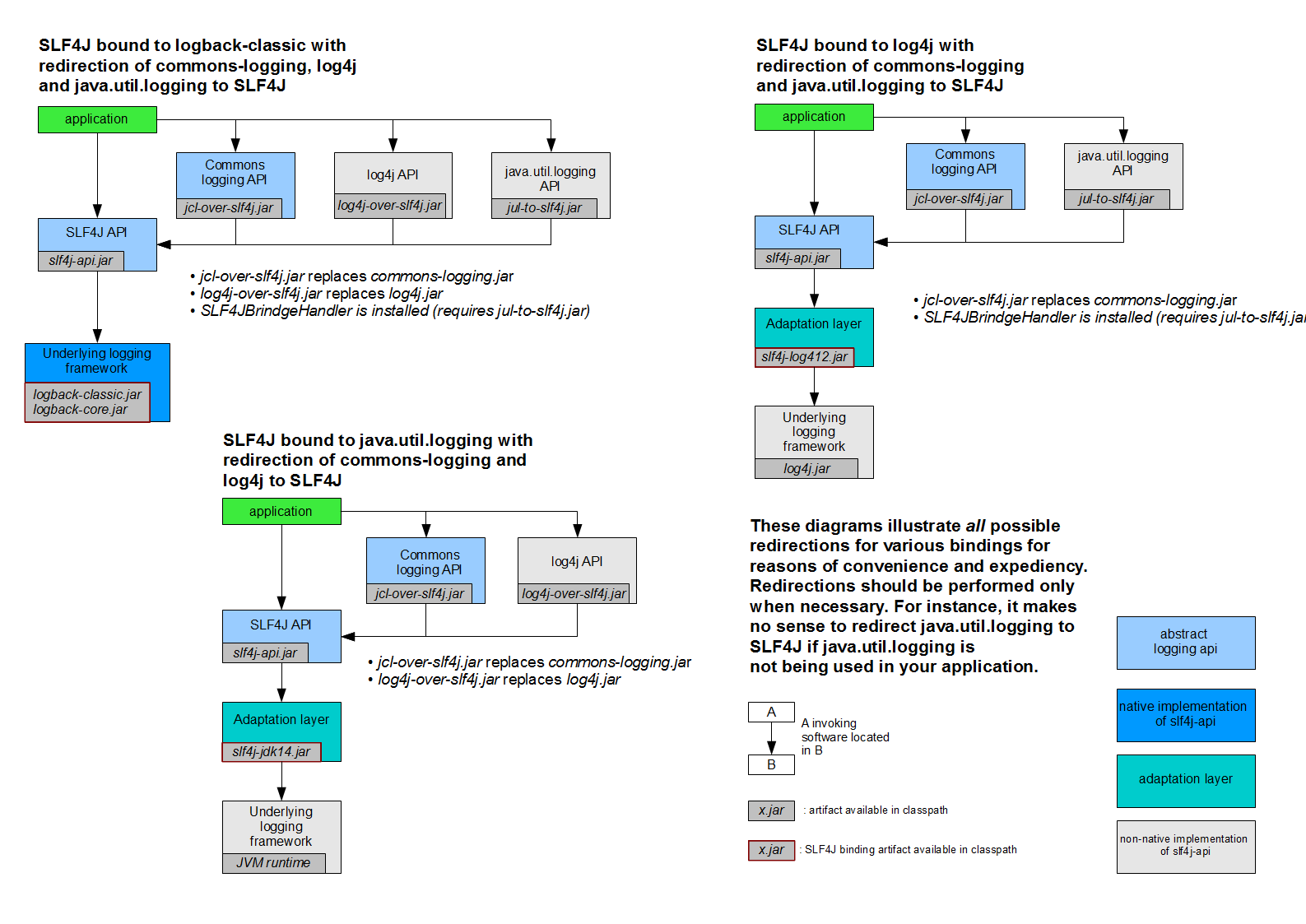


配置文件还是使用实现的日志框架api的方式进行配置,

3,遗留问题



4, 中间转换包



5, 如果要使用其他框架时





使用自定义的log配置文件

