1. 什么是SpringBoot？

通过Spring Boot，可以轻松地创建独立的，基于生产级别的Spring的应用程序，您可以“运行”它们。大多数Spring Boot应用程序需要最少的Spring配置。

2. SpringBoot的特征?

- 创建独立的Spring应用程序
- 直接嵌入Tomcat，Jetty或Undertow（无需部署WAR文件）
- 提供固化的“starter”依赖项，以简化构建配置
- 尽可能自动配置Spring和3rd Party库
- 提供可用于生产的功能，例如指标，运行状况检查和外部化配置
- 完全没有代码生成，也不需要XML配置

3. 如何快速构建一个SpringBoot项目?

- 通过Web界面使用。http://start.spring.io
- 通过Spring Tool Suite使用。
- 通过IntelliJ IDEA使用。
- 使用Spring Boot CLI使用。

4. SpringBoot启动类注解?它是由哪些注解组成?

@SpringBootApplication

- @SpringBootConfiguration:组合了 @Configuration 注解，实现配置文件的功能。
- @EnableAutoConfiguration:打开自动配置的功能，也可以关闭某个自动配置的选项。
- @SpringBootApplication(exclude = { DataSourceAutoConfiguration.class })
- @ComponentScan:Spring组件扫描

5. 什么是yaml?

YAML（/ˈjæməl/，尾音类似camel骆驼）是一个可读性高，用来表达数据序列化的格式。YAML参考了其他多种语言，包括：C语言、Python、Perl。更具有结构性。

6. SpringBoot支持配置文件的格式?

1.properties

```
java.xiaokaxiu.name = xiaoka
```

2.yml

```
java:
  xiaokaxiu:
     name: xiaoka
```

7. SpringBoot启动方式?

1. main方法
2. 命令行 java -jar 的方式
3. mvn/gradle

8. SpringBoot需要独立的容器运行?

不需要，内置了 Tomcat/Jetty。

9. SpringBoot配置途径?

1. 命令行参数
2. java:comp/env里的JNDI属性
3. JVM系统属性
4. 操作系统环境变量

5. 随机生成的带random.*前缀的属性(在设置其他属性时，可以引用它们，比如${random.long})
6. 应用程序以外的application.properties或者appliaction.yml文件
7. 打包在应用程序内的application.properties或者appliaction.yml文件
8. 通过@PropertySource标注的属性源
9. 默认属性

tips:这个列表按照优先级排序，也就是说，任何在高优先级属性源里设置的属性都会覆盖低优先级的相同属性。

10. application.properties和application.yml文件可放位置?优先级?

1. 外置，在相对于应用程序运行目录的/config子目录里。
2. 外置，在应用程序运行的目录里。
3. 内置，在config包内。
4. 内置，在Classpath根目录。

   这个列表按照优先级排序,优先级高的会覆盖优先级低的。

   当然我们可以自己指定文件的位置来加载配置文件。

```
java -jar xiaoka.jar ---spring.config.location=/home/application.yml
```

11. SpringBoot自动配置原理?

@EnableAutoConfiguration (开启自动配置) 该注解引入了AutoConfigurationImportSelector，该类中的方法会扫描所有存在META-INF/spring.factories的jar包。

12. SpringBoot热部署方式?

- spring-boot-devtools
- Spring Loaded
- Jrebel
- 模版热部署

13.「bootstrap.yml」 和「application.yml」?

bootstrap.yml 优先于application.yml

14. SpringBoot如何修改端口号?

yml中:

```
server :
  port : 8888
```

properties:

```
server.port = 8888
```

命令1:

```
java -jar xiaoka.jar ––– server.port=8888
```

命令2:

```
java - Dserver.port=8888 -jar xiaoka.jar
```

15. 开启SpringBoot特性的几种方式?

1. 继承spring-boot-starter-parent项目
2. 导入spring-boot-dependencies项目依赖

16. SpringBoot如何兼容Spring项目?

在启动类加:

@ImportResource(locations = {"classpath:spring.xml"})

17. SpringBoot配置监控?

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

18. 获得Bean装配报告信息访问哪个端点?

/beans 端点

19. 关闭应用程序访问哪个端点?

/shutdown

该端点默认是关闭的，如果开启，需要如下设置。

```
endpoints:
     shutdown:
       enabled: true
```

或者properties格式也是可以的。

20. 查看发布应用信息访问哪个端点?

/info

21. 针对请求访问的几个组合注解?

@PatchMapping

@PostMapping

@GetMapping

@PutMapping

@DeleteMapping

22. SpringBoot 中的starter?

可以理解成对依赖的一种合成，starter会把一个或一套功能相关依赖都包含进来，避免了自己去依赖费事，还有各种包的冲突问题。大大的提升了开发效率。

并且相关配置会有一个默认值，如果我们自己去配置，就会覆盖默认值。

23. SpringBoot集成Mybatis?

mybatis-spring-boot-starter

24. 什么是SpringProfiles?

一般来说我们从开发到生产，经过开发(dev)、测试（test）、上线(prod)。不同的时刻我们会用不同的配置。Spring Profiles 允许用户根据配置文件（dev，test，prod 等）来注册 bean。它们可以让我们自己选择什么时候用什么配置。

25. 不同的环境的配置文件?

可以是 application-{profile}.properties/yml ，但默认是启动主配置文件application.properties,一般来说我们的不同环境配置如下。

- `application.properties`：主配置文件
- `application-dev.properties`：开发环境配置文件
- `application-test.properties`：测试环境配置文件
- `application.prop-properties`：生产环境配置文件

26. 如何激活某个环境的配置?

比如我们激活开发环境。

yml：

```
spring:
 profiles:
  active: dev
```

properties:

```
spring.profiles.active=dev
```

命令行:

```
java -jar xiaoka-v1.0.jar ---spring.profiles.active=dev
```

27. 编写测试用例的注解?

@SpringBootTest

28. SpringBoot异常处理相关注解?

@ControllerAdvice

@ExceptionHandler

29. SpringBoot 1.x 和 2.x区别?·····

1. SpringBoot 2基于Spring5和JDK8，Spring 1x用的是低版本。
2. 配置变更，参数名等。
3. SpringBoot2相关的插件最低版本很多都比原来高
4. 2.x配置中的中文可以直接读取，不用转码
5. Actuator的变化
6. CacheManager 的变化

30. SpringBoot读取配置相关注解有?

- @PropertySource
- @Value
- @Environment
- @ConfigurationProperties