

spring boot：习惯大于约定，约定优于配置。

1.spring boot内置了servlet Container (tomcat , Jetty, Undertow),spring-webmvc,spring-web,jackson(json, Map等的转换),validation,starter。

2.spring boot最简单配置：<https://github.com/ManagerZhang/java-combing-for-myself>

3.spring boot依赖：

1. spring-boot-starter-actuator依赖会引入一组基本的spring项目。从而实现项目的快速配置和即时即用。
2. spring-boot-starter-web依赖会为你提供启动嵌入式Tomcat容器的自动化配置，并且提供对微服务有应用价值的端点信息，如服务器信息，应用指标以及环境详情。
3. 引入spring-boot-starter-security依赖，actuator会自动配置spring security，从而为应用提供基本的认证以及其他高级的安全特性。
4. 他还会为应用结构引入一个内部的审计框架，这个框架可以用来生成报告或者其他的用途，比如开发认证失败的锁定策略。

4.xxxApplication.java为springboot的入口，需要使用注解
@SpringBootApplication，次注解包含了@Configuration，
@EnableAutoConfiguration，@ComponentScan。

1. @Configuration：说到@Configuration就要提到@Bean，使用这两个注解就可以创建一个简单的spring配置类，可以用来代替相应的xml配置文件。

```
[html]    
<beans>  
  <bean id = "car" class="com.test.Car">  
    <property name="wheel" ref = "wheel"></property>  
  </bean>  
  <bean id = "wheel" class="com.test.Wheel"></bean>  
</beans>  
  
[java]    
@Configuration  
public class Conf {  
  @Bean  
  public Car car() {  
    Car car = new Car();  
    car.setWheel(wheel());  
    return car;  
  }  
  @Bean  
  public Wheel wheel() {  
    return new Wheel();  
  }  
}
```

@Configuration的注解类标识这个类可以使用Spring IoC容器作为bean定义的来源，@Bean注解告诉Spring，一个带有@Bean的注解方法将返回一个对象，该对象应该被注册为在Spring应用程序的上下文中的Bean。

2. @EnableAutoConfiguration：能够自动配置Spring的上下文，试图猜测和配置你想要的Bean类，通常会自动根据你的类路径和你的Bean定义自动配置。
3. @ComponentScan：会自动扫描指定包下的全部标有@Component的类，并注册成Bean，当然包括@Component的子注解@Service，@Controller，@Repository。

本类中存在一个main方法，main方法调用SpringApplication.run来启动容器，还没了解run做了哪些操作，后续查看。

5.将springboot文件打成jar包执行时，其所依赖的项目中需要添加<build>

```
<plugins>
  <plugin>                                <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-maven-plugin</artifactId>
    <configuration>
      <classifier>exec</classifier>
    </configuration>
  </plugin>
</plugins>
</build>
```

表示依赖包可被依赖。还需要在maven的setting文件中添加

```
<profile>
  <id>jdk-1.7</id>
  <activation>
    <activeByDefault>true</activeByDefault>
    <jdk>1.7</jdk>
  </activation>
  <properties>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>

    <maven.compiler.compilerVersion>1.7</maven.compiler.compilerVersi

  </properties>
</profile>

<profile>
```

```
<repositories>
<repository>
    <id>spring-releases</id>
    <url>https://repo.spring.io/libs-release</url>
</repository>
</repositories>
<pluginRepositories>
    <pluginRepository>
        <id>spring-releases</id>
        <url>https://repo.spring.io/libs-release</url>
    </pluginRepository>
</pluginRepositories>
</profile>
```

6.通过注解@Value("\${...}")将配置文件中定义的参数注入到bean中。