<https://www.cnblogs.com/MaxElephant/p/8108140.html>

@SpringBootApplication的使用

之前用户使用的是3个注解注解他们的main类。分别是@Configuration,@EnableAutoConfiguration,@ComponentScan。由于这些注解一般都是一起使用，spring boot提供了一个统一的注解@SpringBootApplication。

@SpringBootApplication = (默认属性)@Configuration + @EnableAutoConfiguration + @ComponentScan。

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | @SpringBootApplication  public class ApplicationMain {      public static void main(String[] args) {          SpringApplication.run(Application.class, args);      }  } |

分开解释@Configuration,@EnableAutoConfiguration,@ComponentScan。

1、@Configuration：提到@Configuration就要提到他的搭档@Bean。使用这两个注解就可以创建一个简单的spring配置类，可以用来替代相应的xml配置文件。

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <beans>      <bean id = "car" class="com.test.Car">          <property name="wheel" ref = "wheel"></property>      </bean>      <bean id = "wheel" class="com.test.Wheel"></bean>  </beans> |

 相当于：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | @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下的子注解@Service,@Repository,@Controller。