Create a example

pom.xml



Controller:

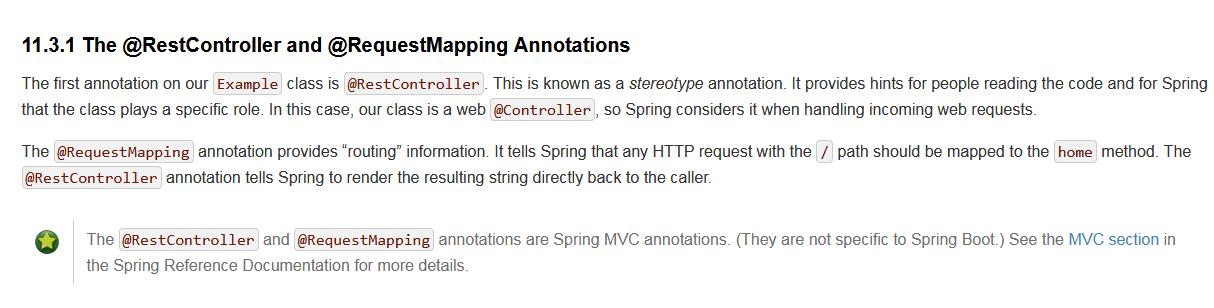


The defferent between @RestController and @Controller:

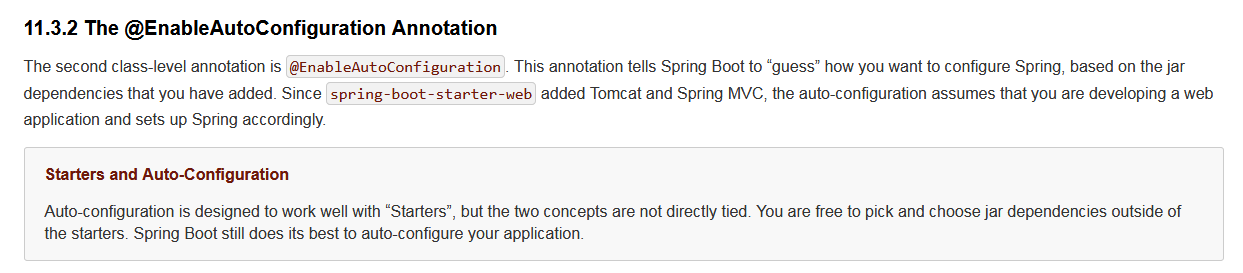
@Controller could return jsp、html or (add @ResponseBody to return json)json

@RestController just could return json.

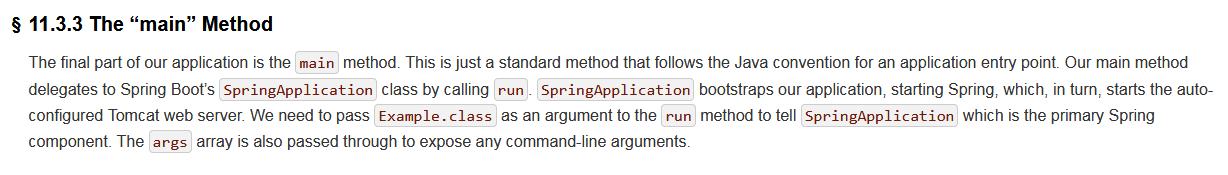
@RestController = @Controller +@ResponseBody



@EnableAutoConfiguration: aim to tell springboot how to configure spring.



Main method: to auto starting spring,auto starting the tomcat web server.



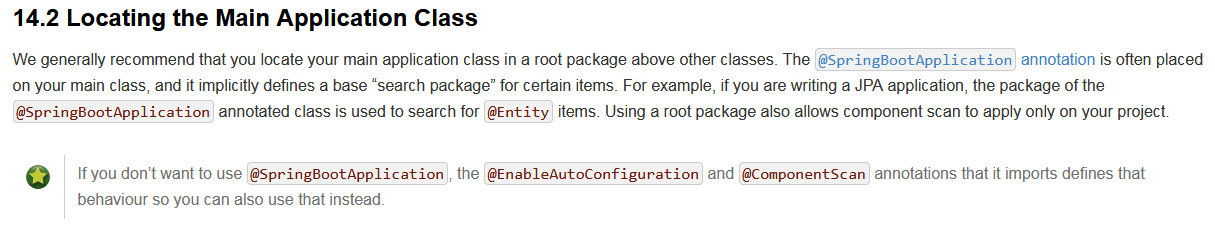
When you see that your demo is successful.



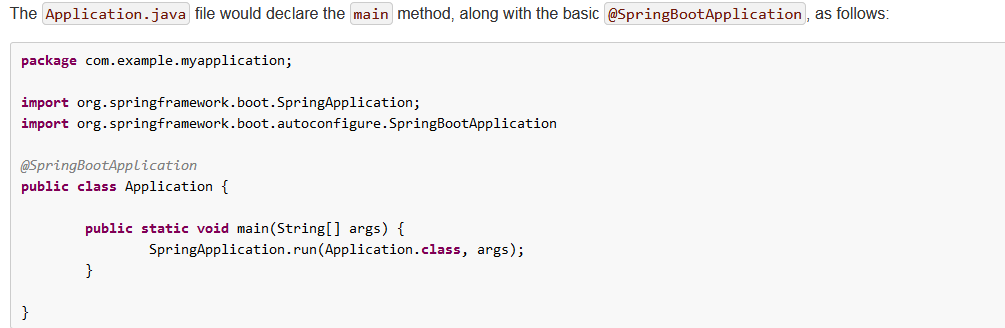
@SpringBootApplication on you main class.

Locate you main class above other clasese package.

@SpringBootApplication = @EnableAutoConfiguration +@ComponentScan



Main class demo with use the @SpringBootApplication



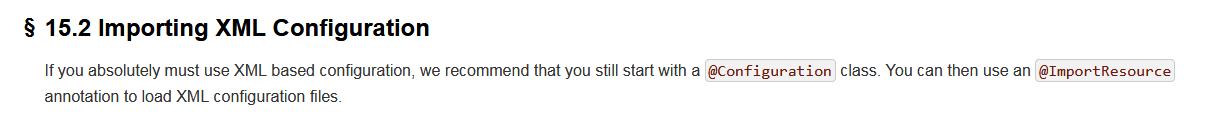
@Configuration: load one class

@Import : load one special class

@ComponentScan: load all spring conponent.

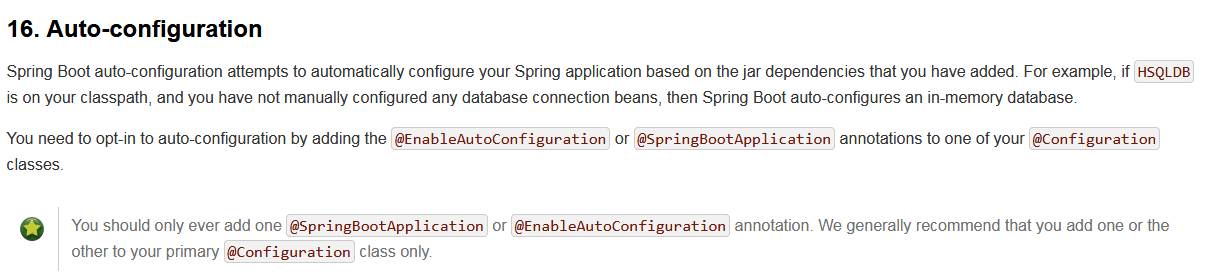


@Configuration to configure xml

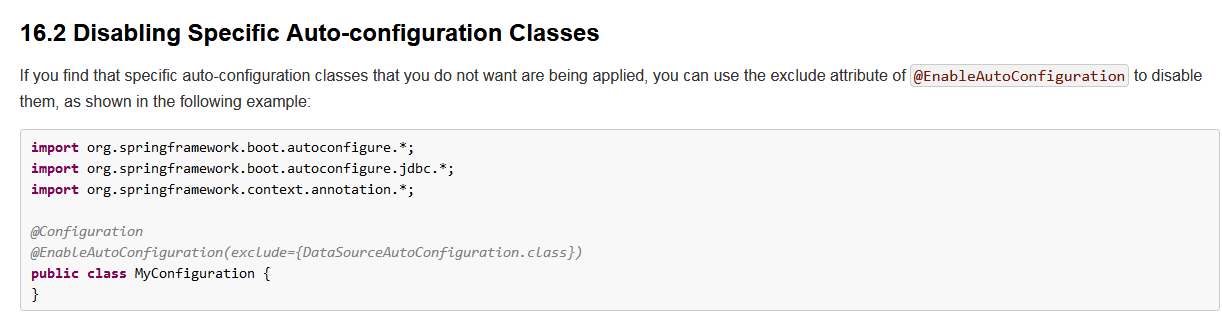


@Configuration :Recommend this annotations

@EnableAutoConfiguration ： use this could auto configuration

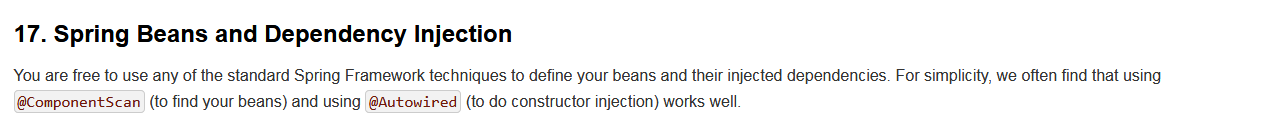


@EnableAutoConfiguration ： to Ingnore some class



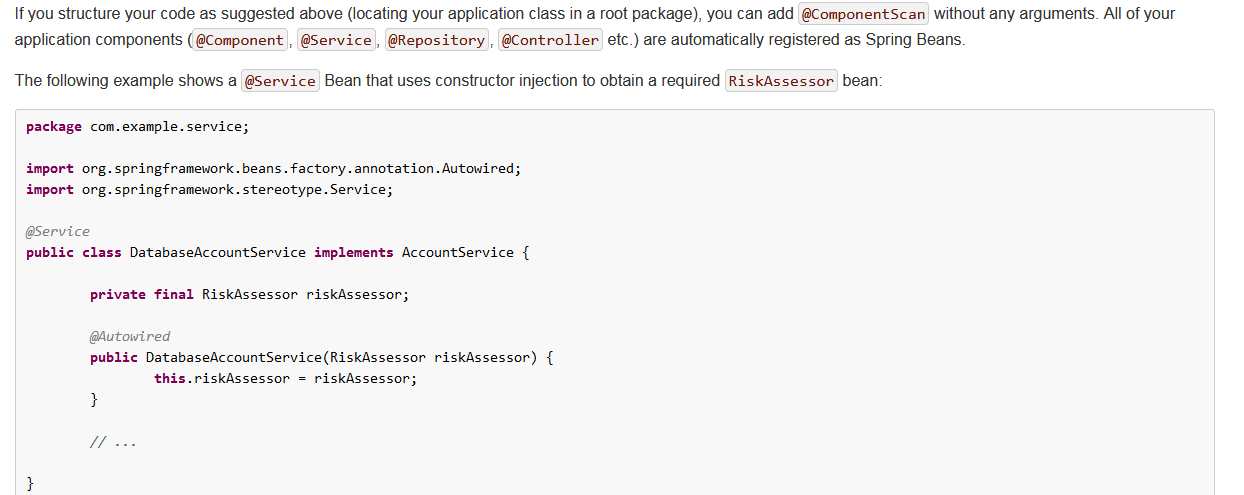
@ComponentScan ： find you beans

@Autowired： inject object by constructor.

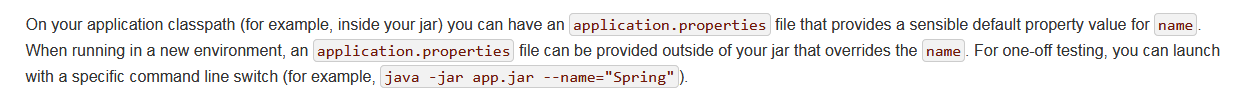


Structrue codes to use @ComponentScan:

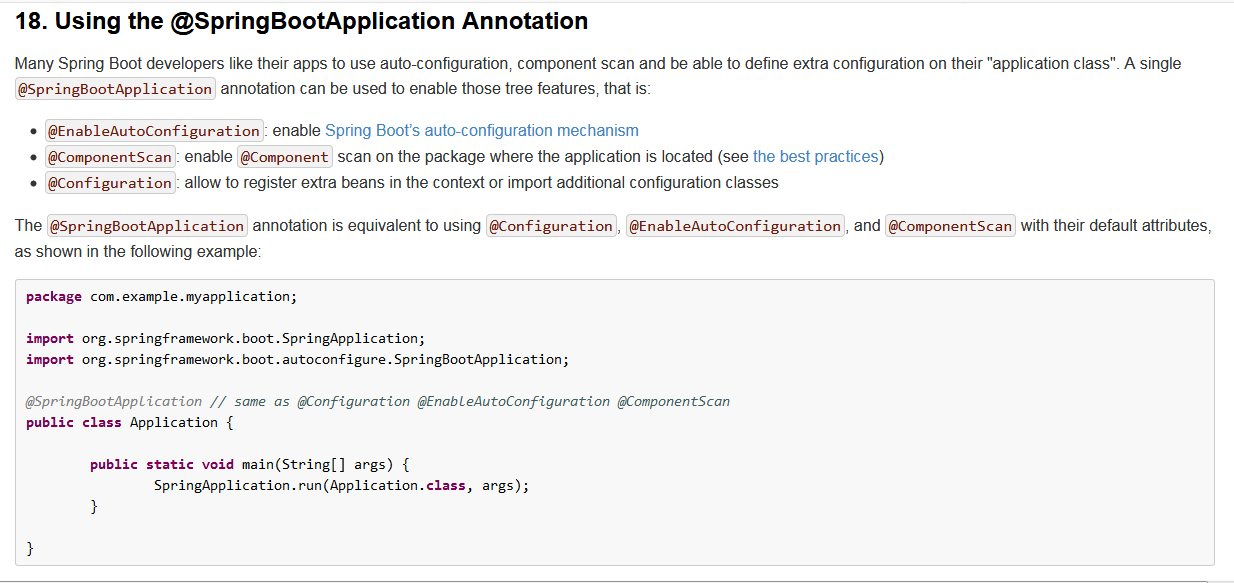
**@Repository:always use to annotation the Dao**



Default configure file name. application.properties



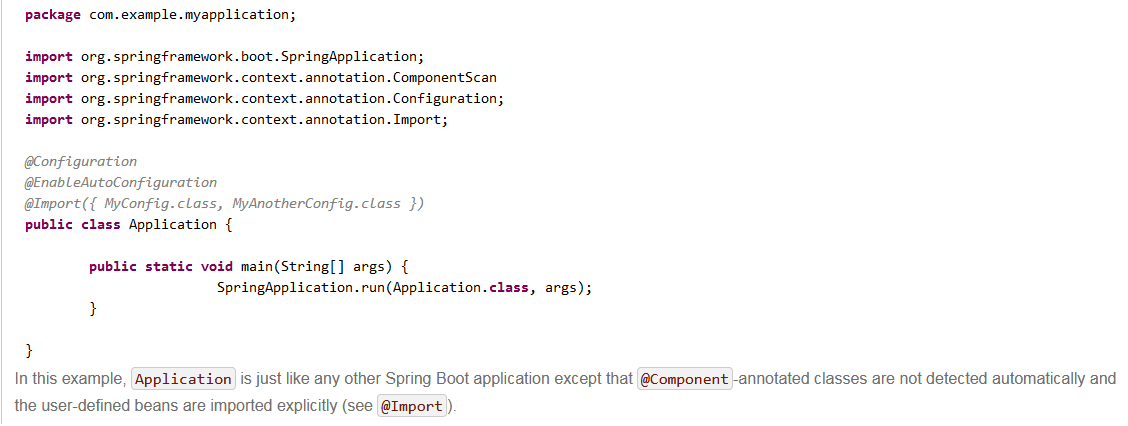
@SpringBootApplication = @EnableAutoConfiguration + @Configuration + @ComponentScan



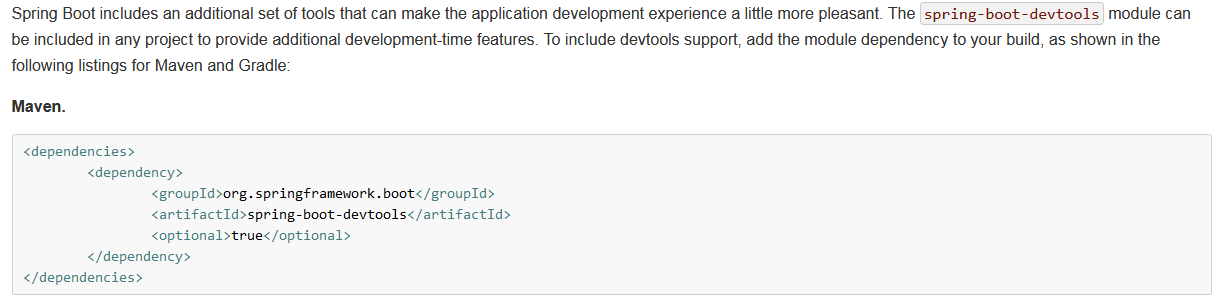
Use @EnableAutoConfiguration @Configuration @Import to annotation main class instead of @SpringBootApplication demo.

@Import : point at some specific classes.

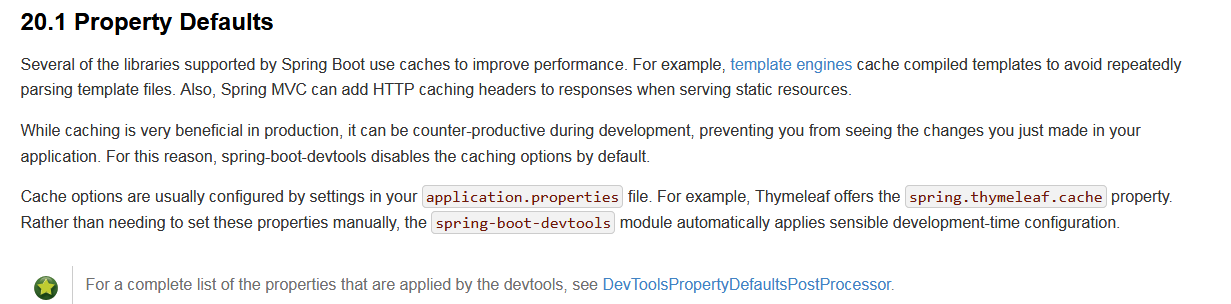
@Component: point at all class.



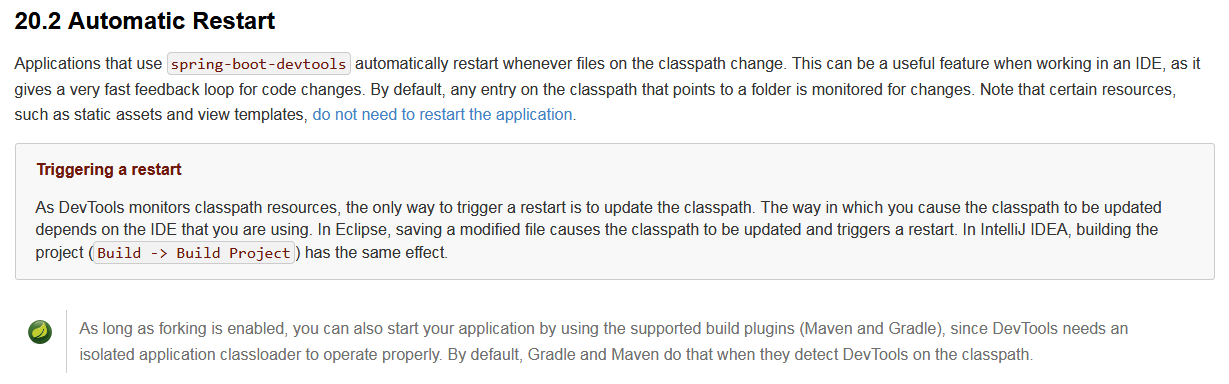
Springboot-devtools: A powerrful dependency!



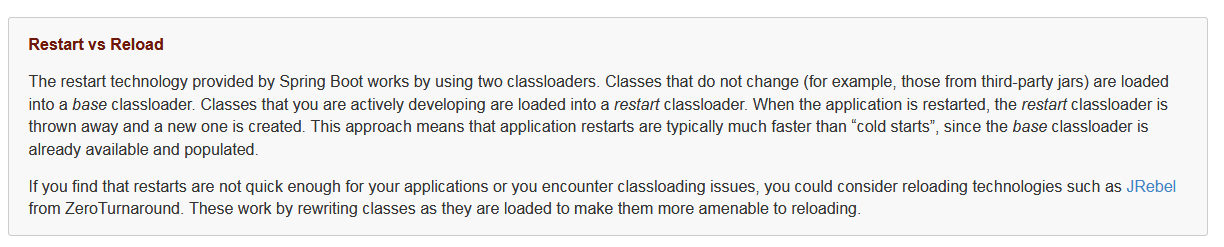
Disable to cache in default.



If some files be modified ,the springboot- devtool will restart this program auto.

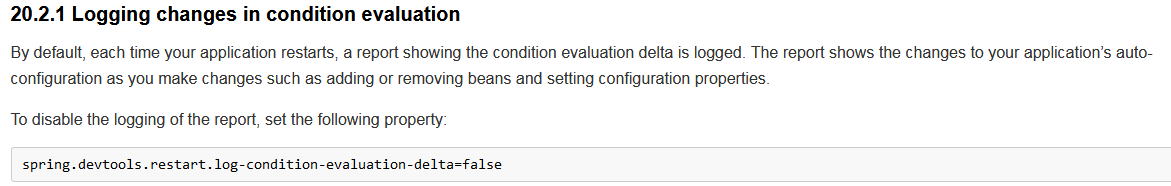


Restart detailed analysis: springboot worked with two classloaders.

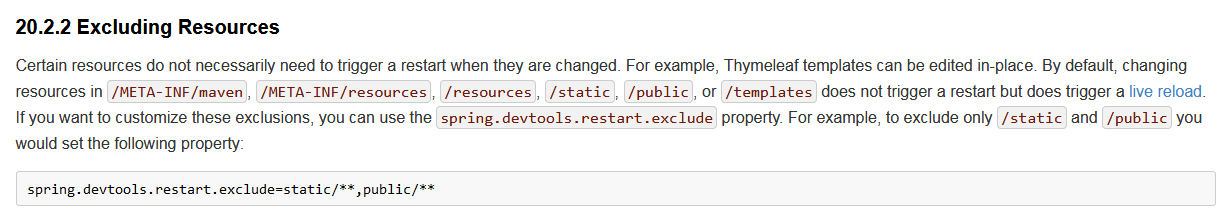


Restart logs: you can figure out it by configuring in application.properties.

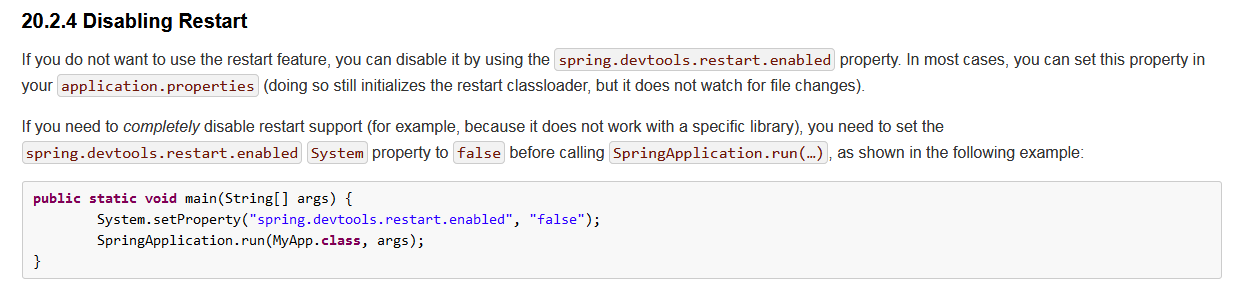
spring.devtools.restart.log-condition-evaluation-delta=false



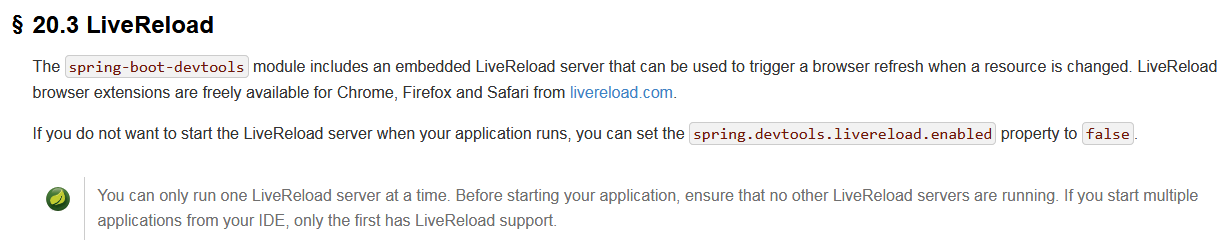
Eclude some files ,project would not to restart when they are changed



Springboot devtool default to start,you can stop it by modifying the System param.



Trigger a brower refresh. (auto)

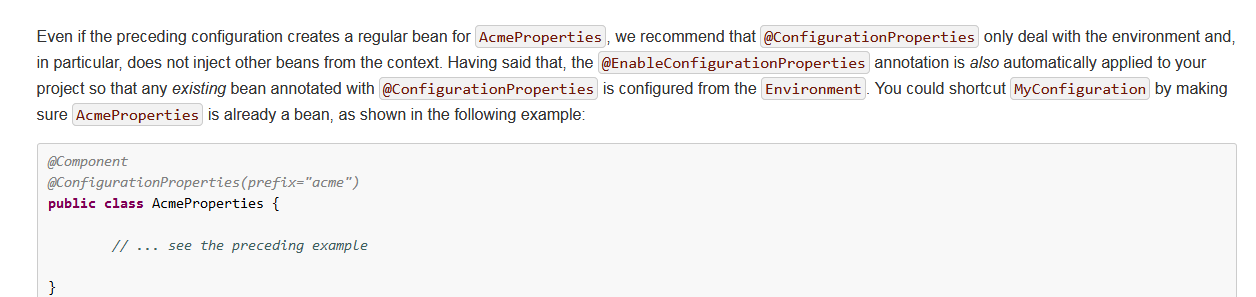


Use the configure file name definitied by self. Composed with –spring.config.name=Xxxxxx

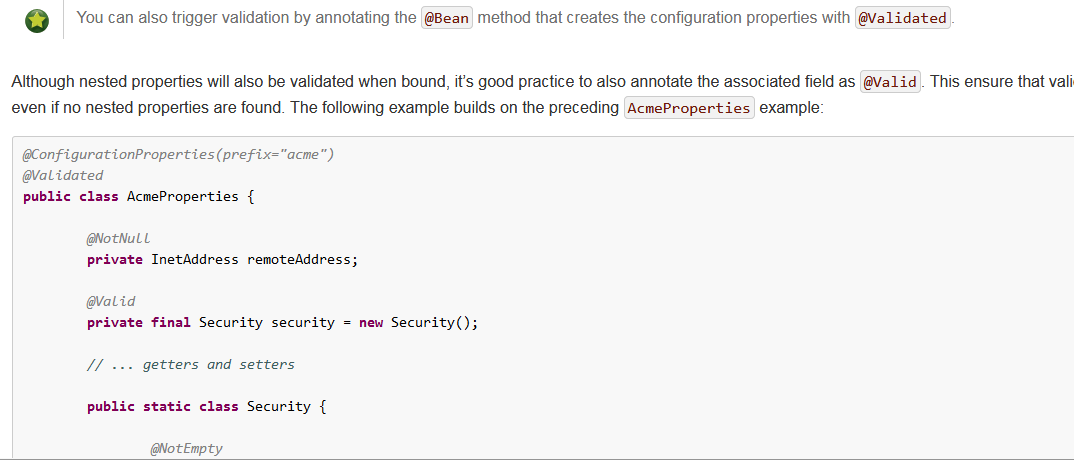


Use specific configuration

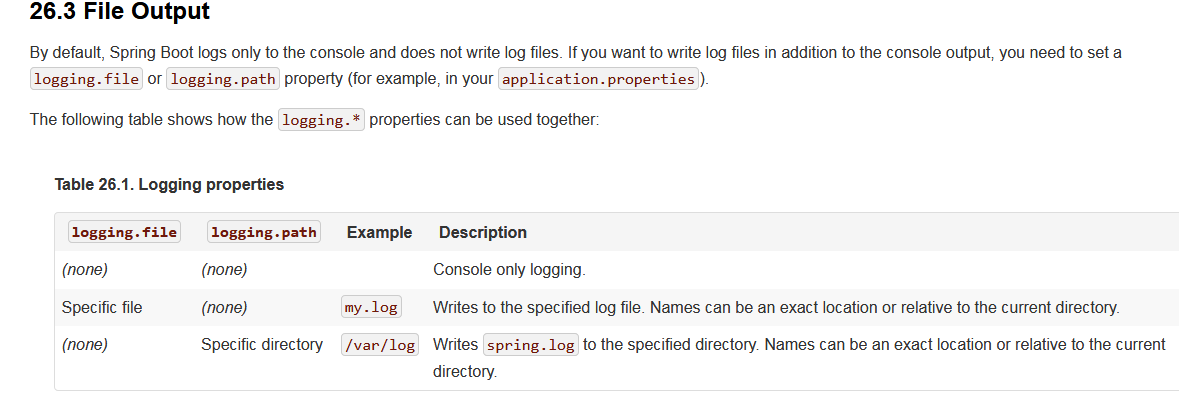
@EnableConfigurationProperties：all class be annotated by @ConfigurationProperties



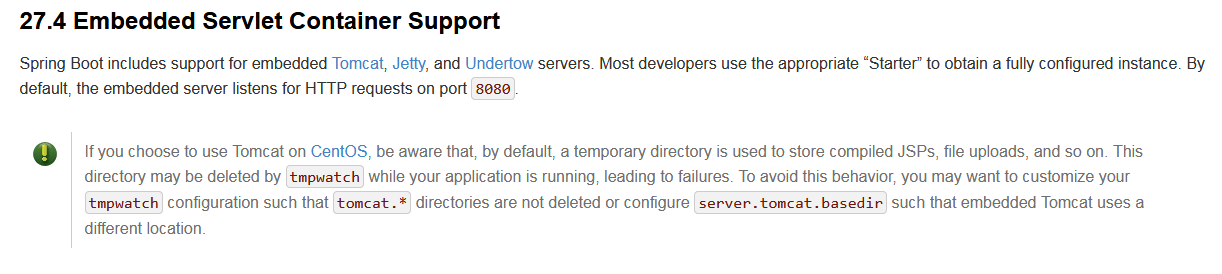
Use @Valitaded anotations to validate attributes



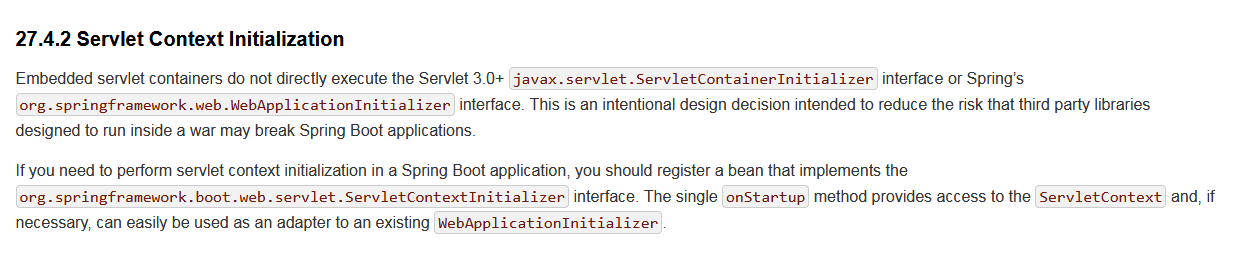
Ouput the log fiiles



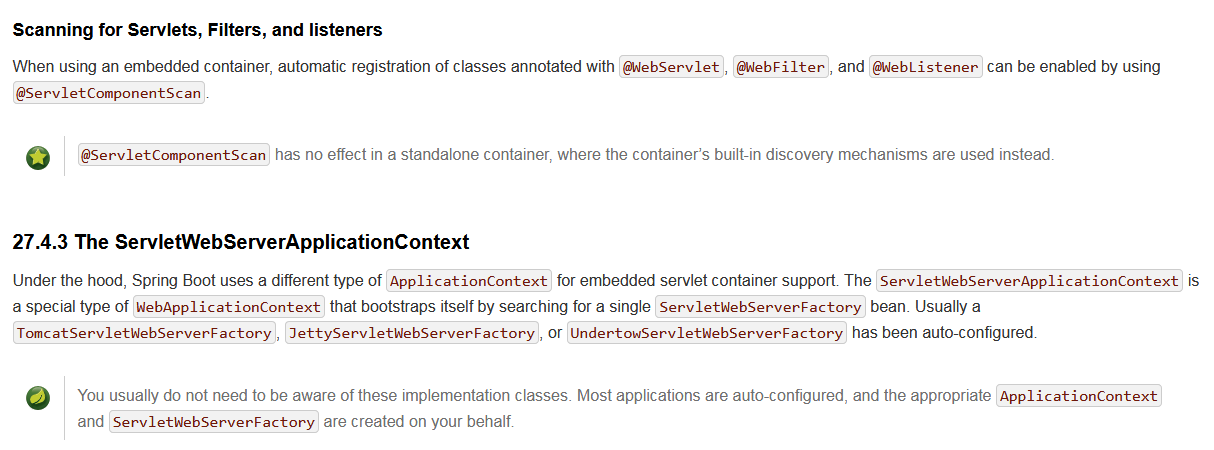
Springboot support tomcat/jetty ,the default port is 8080



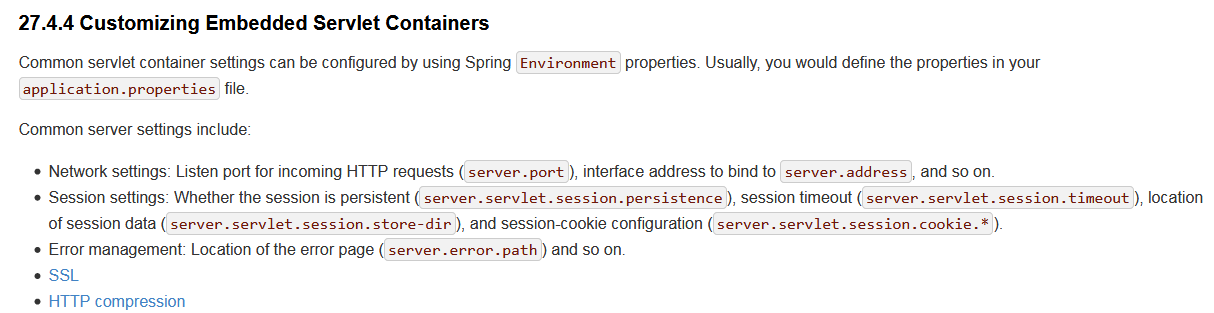
Springboot servlet



Use springboot servlet



Configure springboot servlet in application.properties



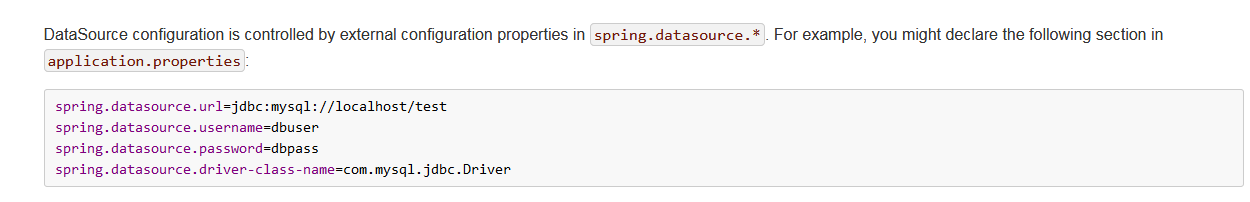
Demo in springboot servlet



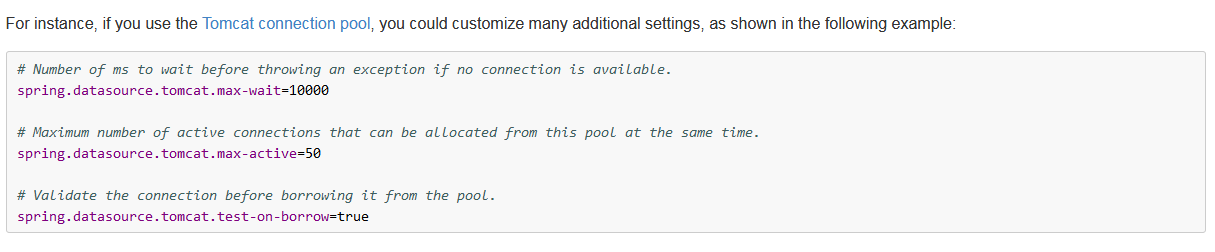
Modify the port in tomcat



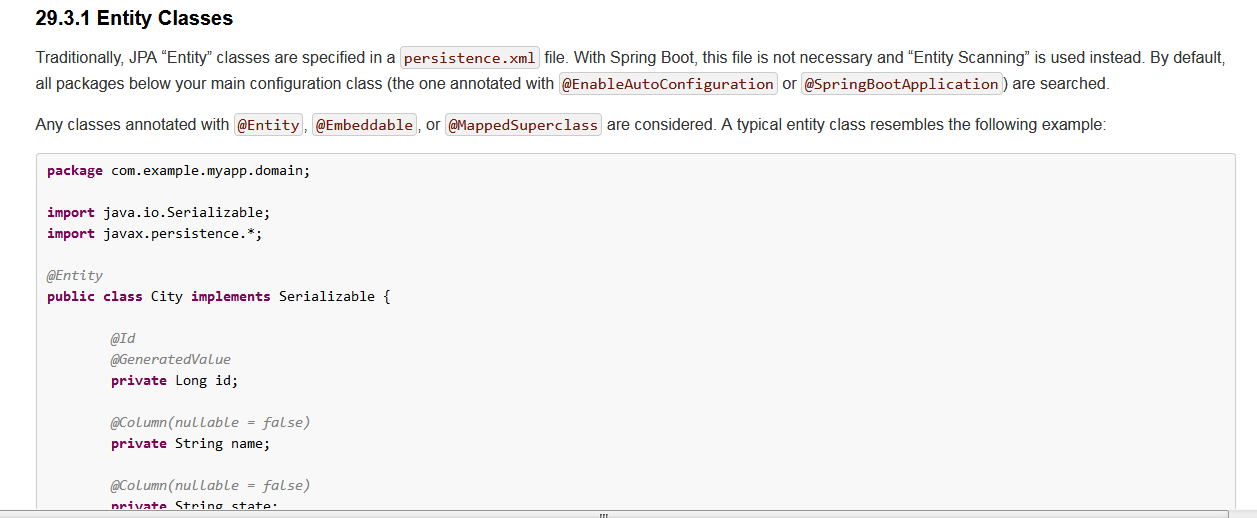
Connect to mysql database；



Tomcat connect pool

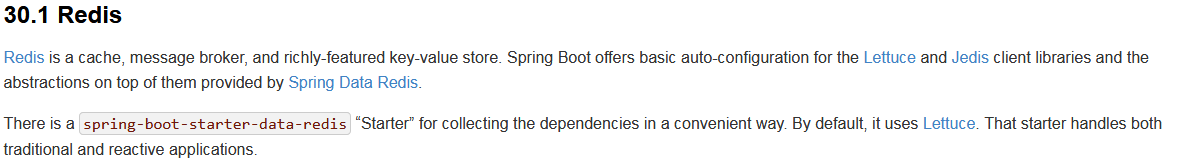


Entity Class use @Entity annotations



JOOQ: java object orentied query

Springboot user redis

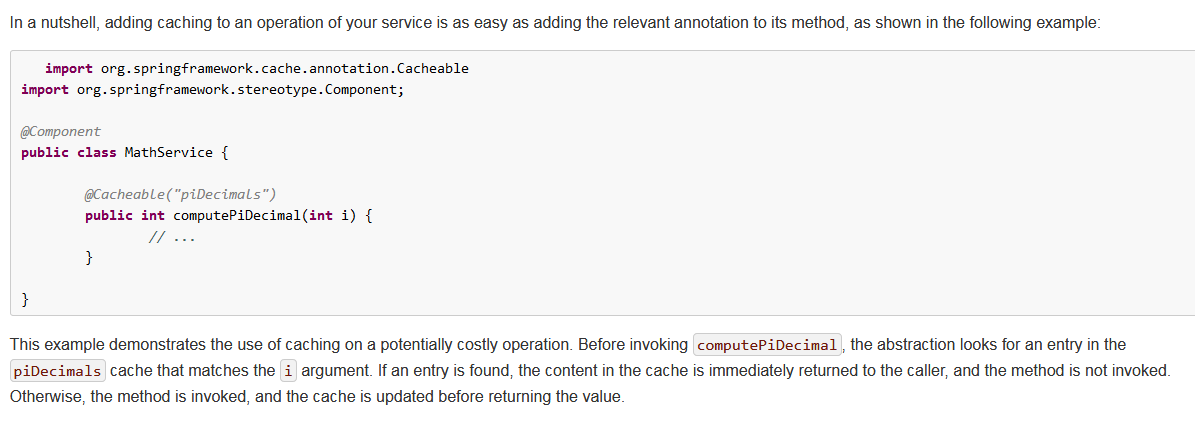


Connect redis

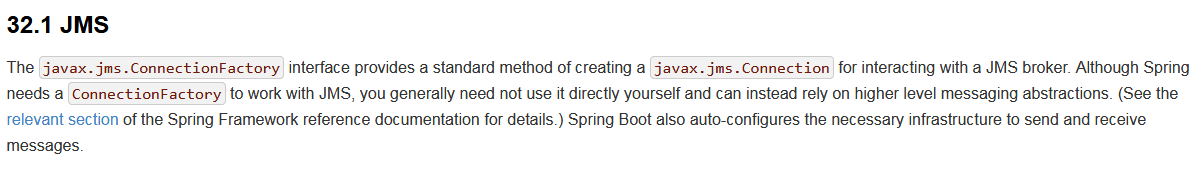


Springboot provides lots of databases,sql or nosql ,all of those will have XXXTemplate instance to operate.

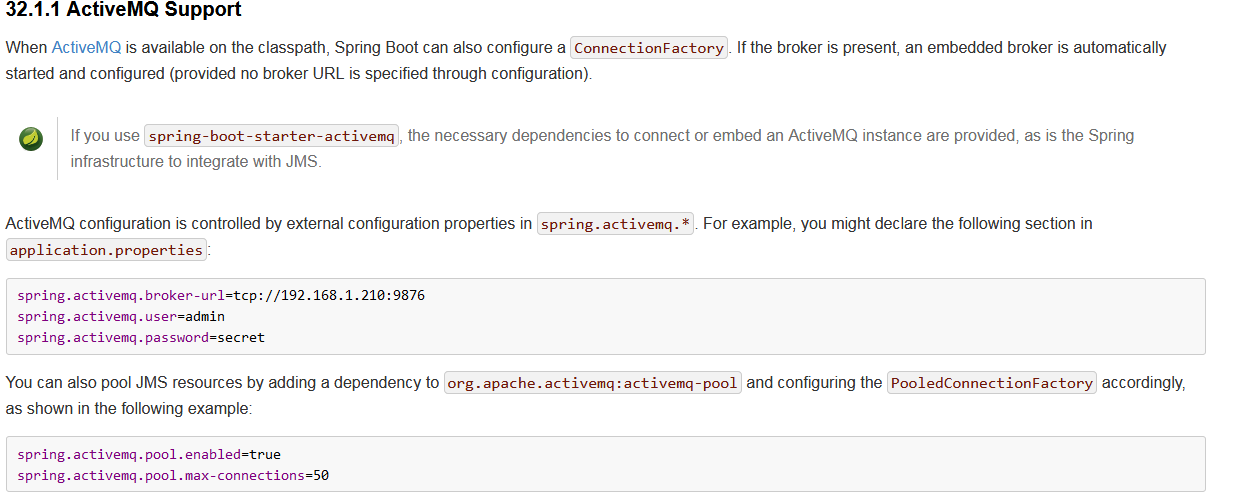
Springboot cache use demo:



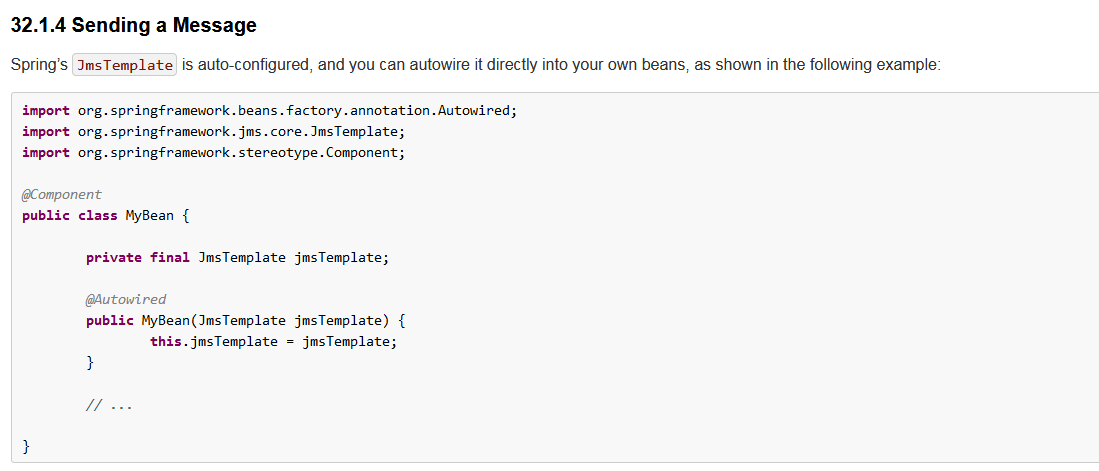
JMS: java message service



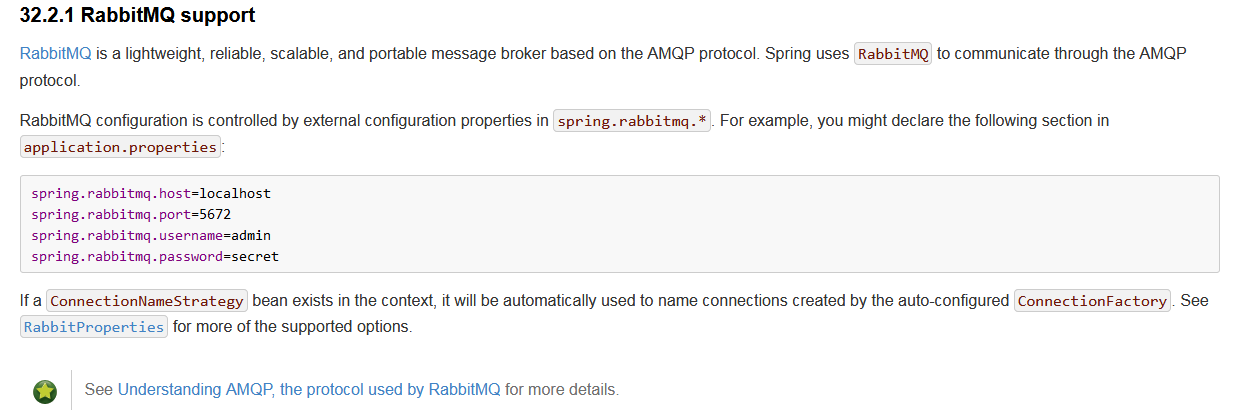
JMS-ActionMq:use demo:



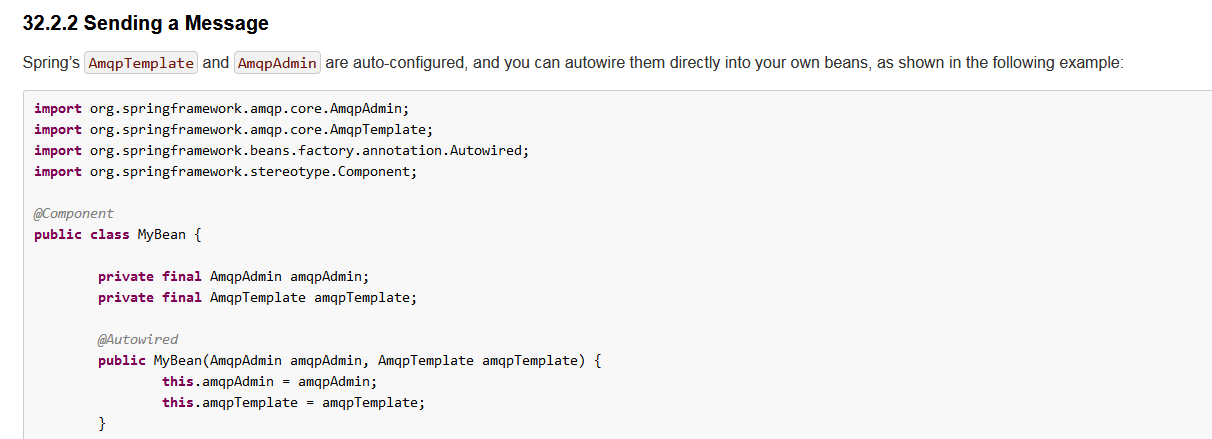
JMS send meassages:



RabbitMQ



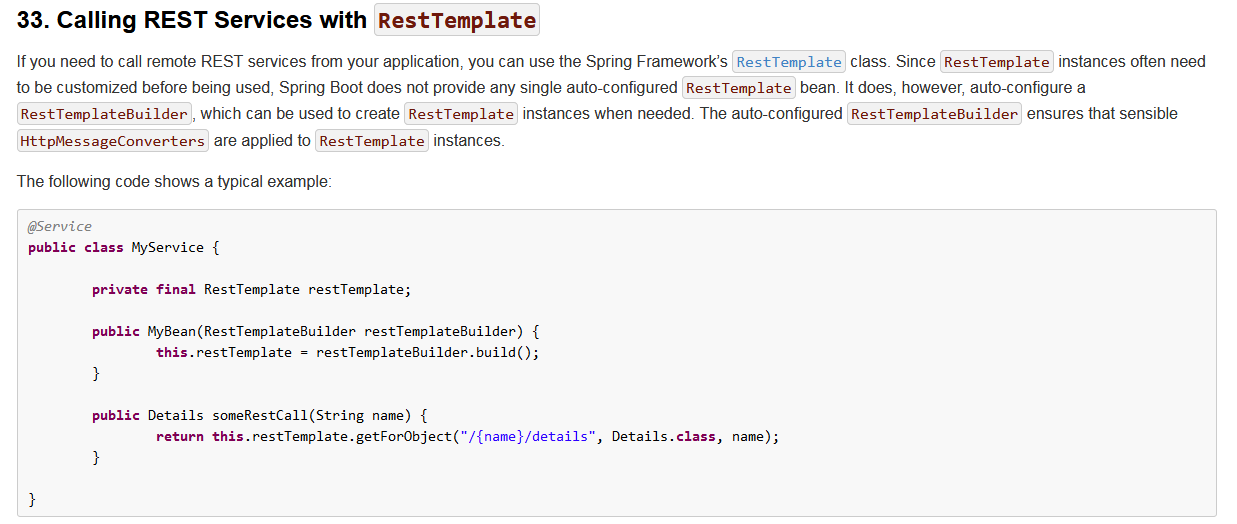
Send mesages:



Receive messages:



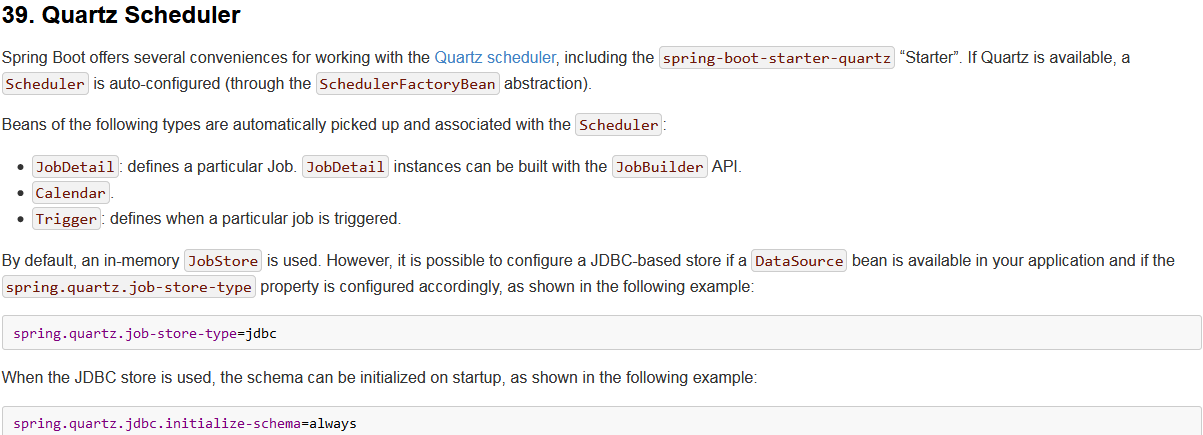
Use restTemplate by RestTemplateBuilder



Use webClient by WebClientBuilder



Springboot use quartz scheduler



50.5