# 缓存(红线是重中之重)

1. 需要导入cache和redis的相关mvn依赖；
2. 在application.properties文件中配置相关的redis连接信息；
3. 在主启动类上加上@EnableCache注解开启缓存功能；
4. 在需缓存的方法上添加@Cacheable注解，需清除缓存的方法上加上@CacheEvict注解，需更新缓存的方法上添加@CachePut注解，还可以使用@Caching注解编写组合缓存处理。
5. 注意：
   1. javabean必须实现序列化机制
   2. cacheNames和values只能出现一个
   3. key和keyGenerator只能出现一个
   4. 若想指定cacheManager，则需要先创建RedisCacheConfiguration对象，然后可以设置其的序列化对象，并返回创建的RedisCacheConfiguration对象，最后再使用RedisCacheManager创建RedisCacheManager对象并注入到ioc容器里
   5. 参数可以使用#p0，#a0，#root.args[0]，#参数名获取参数值
   6. condition表示条件满足就缓存，unless表示条件满足就不缓存
   7. @cacheEvict注解清除缓存时，需注意缓存的key和清除的key必须一致beforeInvocation为true时，表示清除缓存动作在方法执行之前，为false时，表示清除缓存动作在方法执行之后，key=”#result.×××”和beforeInvocation =true不能同时使用
   8. @Cacheable注解的key不能使用#result，key中若想使用字符串需加单引号引起来
   9. @Caching注解可以组合@Cacheable，@CachePut，@CacheEvict三种注解使用，可以自己配置复杂的缓存清除规则，注：相应的注解标签，只能放在相应的属性里面使用，如：cacheable里面只能放@Cacheable注解，put里面只能放@CachePut注解，evict里面只能放@CacheEvict注解，不能乱放。
   10. 所有注解方法上共有的属性可以单抽出来，在当前类上配置@CacheConfig中配置，则可以达到一处配置，处处使用的效果，但若方法和类上同时配置，方法上的属性优先。
   11. 多个javabean需编写多个RedisCacheManager类，但必须给其中一个加上@Primary注解使其成为默认的缓存管理器，若方法和类上均未指定cacheManager属性，则默认为该缓存管理器管理缓存。

# 消息(RabbitMq)

1. RabbitMq的暴露端口是5672,web页面的访问端口是15672。
2. 交换器的类型有：
3. ．direct：根据漏油键进行点对点的消息发送
4. . topic：可以漏油键进行模糊匹配发送到所有能匹配上的队列
5. . fanout：给所有绑定在交换器的队列都发消息
6. . header：根据消息头进行发送消息到相应的队列

3. 整合：

* 1. 导入rabbitmq的相关依赖
  2. 配置相关的服务连接信息
  3. 编写一个转换器对象MessageConnverter对象，这是将对象序列化为json串。
  4. 操作
     1. . 使用AmqpAdmin对象创建交换器，队列，绑定关系，也可以自己登陆 rabbitmq页面进行相关配置。
     2. . 使用rabbitmq对象给队列发送消息内容，或者自己定义一个Message对象，直接发送消息。
     3. . 在springboot的主启动类上加上@EnableRabbit注解，并在相应的需要获取监听消息的服务上加上@RabbitListener(queues = "zpzy.nwes")注解监听相应的消息队列的内容，而消息内容会自动注入到方法的参数中。

注：Message的消息体为byte数组，在监听获取到Message的消息体后需将其转为相应的对象。

# ElasticSearch

1. elasticsearch的服务连接端口为9300，web管理端口为9200。
2. elasticsearch是将数据保存为一个json串(一个文档)进而存放到数据库，存放的动作叫做索引(index)。同时，索引也相当于关系型数据库的数据库(database)或者模式，类型相当于关系型数据库的表，一条数据就是一个文档。一个集群(cluster)可以有多个索引，一个索引可以有多个类型。
3. elasticsearch可以通过web页面对数据库进行数据操作：put请求+json串，是存储数据到服务器，get请求/(post请求+查询表达式)是获取数据，delete请求是删除数据。例如：<http://localhost:9200/study/students/1(put>请求+json串)---------->把json串保存到数据库中，study索引students类型并且主键为1。

http://localhost:9200/study/students/1(get请求)---------->查询study索引的students类型下主键为1的数据。

http://localhost:9200/study/students/1(delete请求)---------->删除study索引的students类型下主键为1的数据。

http://localhost:9200/study/students/\_search(post请求+查询json串)--------->查询符合要求的所有数据。查询json串如：

{

"query":{

"match":{

"sname":"wangwu"

}

}

}

<http://localhost:9200/study/students/_search?q>=列名:数据(post请求)-------->查询列名为该数据的json数据。

1. 整合
   1. ．导入相关elasticsearch的starter依赖包。
   2. . 配置elasticsearch的cluster-name和cluster-nodes
   3. . 操作elasticsearch

方法一：导入jest依赖，并注入jestClient对象操作elasticsearch数据库。如Students students = new Students("222","xiaozhang",10,99.0);

Index index = new Index.Builder(students).index("zyzp").type("students").build();

jestClient.execute(index);

方法二：编写一个继承ElasticSearch<保存数据的类型,主键类型>接口的接口，并注入到ioc容器中，跟jpa操作数据库一样操作elasticsearch数据库。

方法三：注入elaseticsearchTemplate对象操作elasticsearch数据库。

注：elasticsearch和相应的starter版本之间有着对应的关系，需注意版本必须匹配上才可以。

# 任务

1. 定时任务
   * 1. [在springboot启动类上开启@EnableScheduing](mailto:a.在springboot启动类上开启@EnableScheduing)注解并在相应的定时任务方法上加上@Scheduled(cron = "0-4 07 21 \* \* \*")注解。cron格式为秒 分 时 日 月 周几
     2. 时间格式允许的字符(,枚举)、(-区间)、(\*任意)、(/步长)、(?日/星期冲突匹配)、(L最后)、(W工作日)、(C和calendar联系后计算过的值)、(#星期,4#2,,第二个星期四)
2. 异步任务

如发生线程阻塞，可使用异步任务方法跳过阻塞时间直接进行后续操作。

[在springboot启动类上开启@Enable](mailto:a.在springboot启动类上开启@EnableScheduing)Async注解并在相应的异步任务方法上加上@Async注解。

1. 邮件任务
   1. .在qq邮箱设置中，开启pop3/smtp服务，并记录返回的密码。
   2. .配置相应的连接qq邮箱的数据。包括:

spring.mail.protocol=smtp

spring.mail.username=192583306@qq.com

spring.mail.password=adsucegxwhghbieg

spring.mail.host=smtp.qq.com

spring.mail.properties.mail.smtp.ssl.enable=true

③.操作

需要注入一个JavaMailSenderImpl对象mailSender。

1. 简单邮件

创建SimpleMailMessage对象并，设置主题，内容，收件人，发件人信息，然后mailSender对象使用send方法发送邮件即可。

1. 复杂邮件(带附件的邮件)

利用mailSender的createMimeMessage()方法创建一个MimeMessage对象，利用创建的MimeMessage对象创建MimeMessageHelper对象helper，利用helper对象设置主题，内容，收件人，发件人信息，附件信息(addAttachment(文件名,对应文件名的文件对象))，然后mailSender对象使用send方法发送邮件即可。