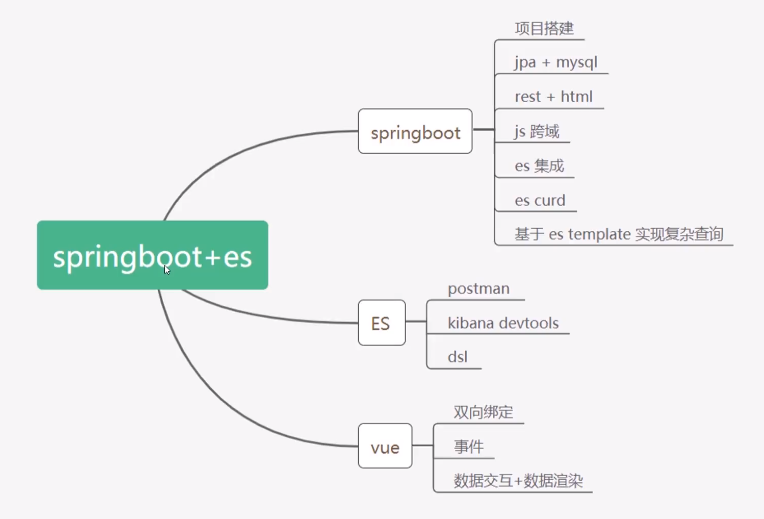
SpringBoot + ElasticSearch 构建博客检索系统

课程地址：<https://www.imooc.com/learn/1161>

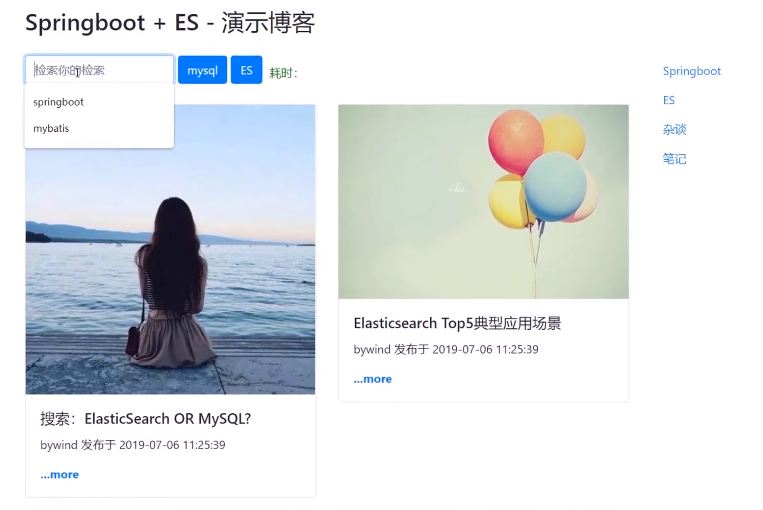
**第1章 课程介绍**

* 1. **课程导学**

#### 主要技术内容



#### 效果展示



#### 建议

* 可以反复看
* 上手做
* 学会应用

**第2章 初识 ElassticSearch**

**2-1 ElasticSearch 概念和适用场景**

**ElasticSearch是一个开源的、分布式全文检索和分析引擎，它可以进行实时快速存储、搜索分析海量数据。**



**2-2 ElasticSearch 数据类型，和关系型数据库对比**

|  |  |
| --- | --- |
| ElasticSearch VS Mysql | |
| MySql | **ElasticSearch** |
| Database | **Index** |
| Table | **Type** |
| Row | **Document** |
| Column | **Field** |
| Schema | **Mapping** |

MySql: Select \* from user.user\_info where name=‘张三’;

ElasticSearch：GET/user/user\_info/\_search?q=name:张三

**2-3 安装 ES、postman、Kibana**

分布式、RESTful 风格的搜索和分析。

ElasticSearch安装和部署

官网地址

<https://www.elastic.co/cn/downloads/>

历史版本

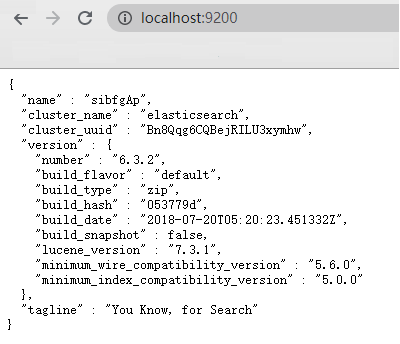
<https://www.elastic.co/cn/downloads/past-releases#elasticsearch>

课程所用版本 6.3.2，也可根据自己选择合适的版本，

1.下载后解压，双击E:\elasticsearch-6.3.2\bin目录下的 elasticsearch.bat 运行

没有警告信息表示没有问题

2.浏览器输入 <http://localhost:9200/> 出现json数据表示成功



PostMan

<https://www.postman.com/>

<https://www.postman.com/downloads/canary/>

需要注册登录



实现数据可视化。在 Elastic Stack 中进行导航。

官网地址

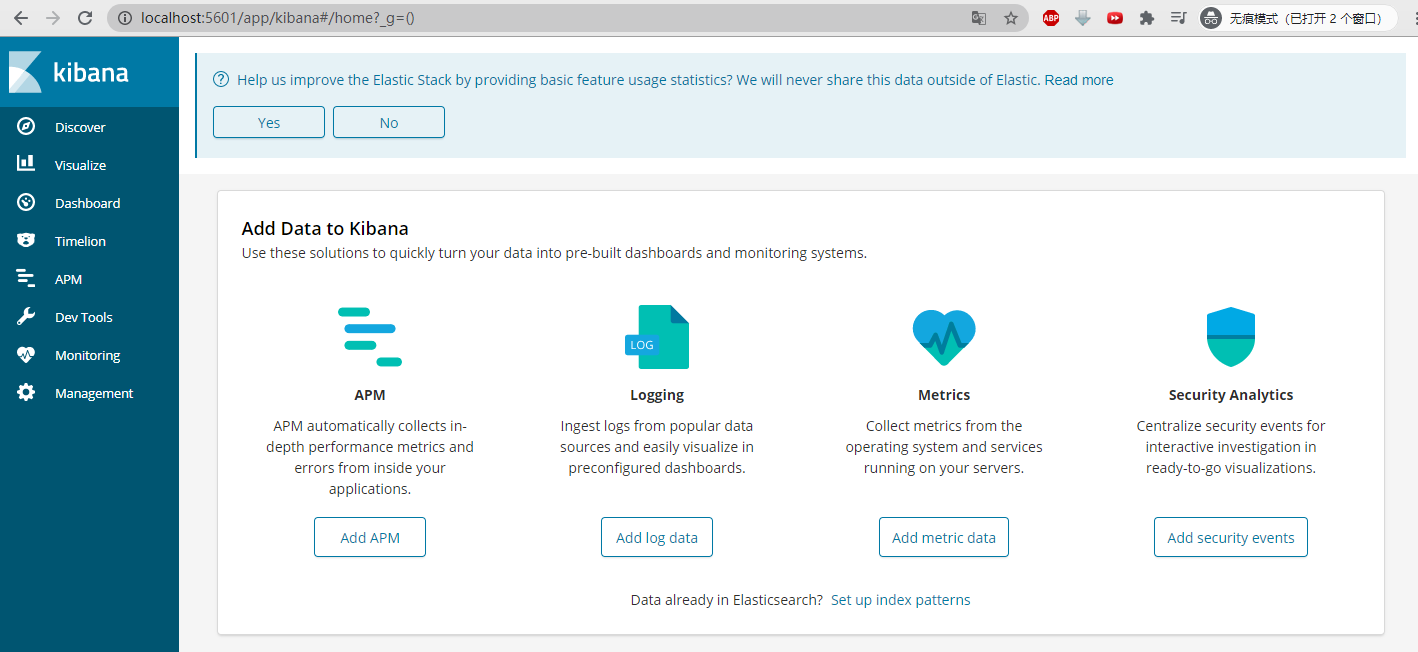
<https://www.elastic.co/cn/downloads/kibana>

<https://www.elastic.co/cn/downloads/past-releases/>

1.下载后解压，双击E:\kibana-6.3.2-windows-x86\_64\bin目录下的 kibana.bat 运行

等待一段时间，出现Server ruuning at http://localhost:5601

2.浏览器输入 <http://localhost:5601/> 出现面板表示成功



**2-4 演示 postman、kibana对ES的交互**

**PostMan**

新建Collection 命名为ES，创建request

Get 查看所有索引

localhost:9200/\_all

PUT 创建索引-test

localhost:9200/test

DEL 删除索引-test

localhost:9200/test

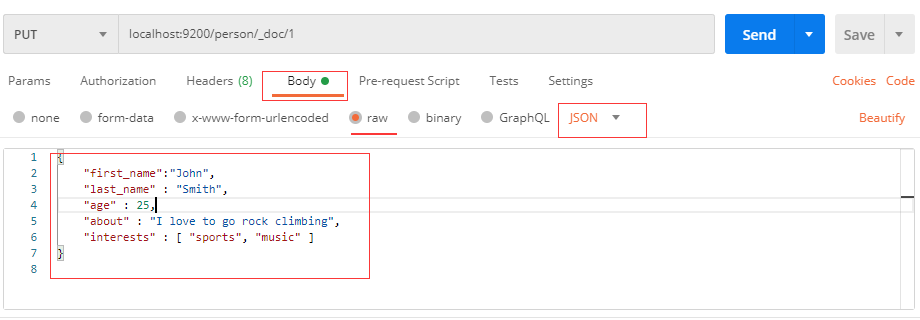
PUT 创建索引-person-1

localhost:9200/person

PUT 新增数据-person-1

localhost:9200/person/\_doc/1

点击body 选择raw JSON格式，输入内容 如下图：



{

"first\_name":"John",

"last\_name" : "Smith",

"age" : 25,

"about" : "I love to go rock climbing",

"interests" : [ "sports", "music" ]

}

PUT 新增数据-person-2

localhost:9200/person/\_doc/2

{

"first\_name" : "Eric",

"last\_name" : "Smith",

"age" : 23,

"about" : "I love basketball",

"interests" : [ "sports", "reading" ]

}

GET 搜索数据 根据id-person-id

localhost:9200/person/\_doc/1

GET 搜索数据 根据关键字-person-name

注：请求时需要删除body里面的内容，否则会出错

localhost:9200/person/\_doc/\_search?q=first\_name:john

查询结果：



**Kibana**

<http://localhost:5601/app/kibana>

点击Dev Tools

GET \_search

{

"query": {

"match\_all": {}

}

}

查看索引

GET \_all

搜索person

GET /person/\_doc/1

POST /person/\_search

{

"query": {

"bool": {

"should": [

{"match": {

"first\_name": "Eric"

}

}

]

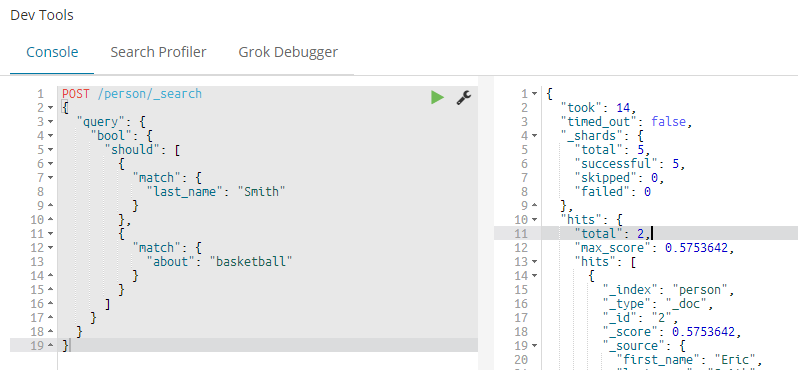
}

}

Bool 查询 使用关键字 should 应该匹配哪些字段

Should 相当于 mysql 的or 满足一个即可，返回多个结果

Must 相当于MySQL的and 必须都满足



POST /person/\_search

{

"query": {

"bool": {

"should": [

{

"match": {

"last\_name": "Smith"

}

},

{

"match": {

"about": "basketball"

}

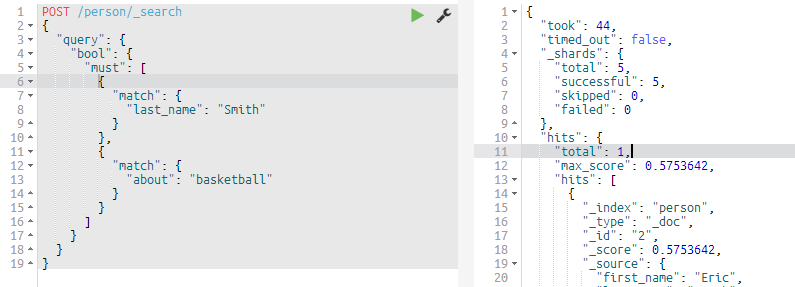
}

]

}

}

}



POST /person/\_search

{

"query": {

"bool": {

"must": [

{

"match": {

"last\_name": "Smith"

}

},

{

"match": {

"about": "basketball"

}

}

]

}

}

}

**第3章 博客网站全文检索**

**3-1 基于 Mysql 实现**

此处需要在MySQL中创建，后面会用到

CREATE DATABASE blog;

USE blog;

CREATE TABLE `t\_blog` (

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '自增id',

`title` varchar(60) DEFAULT NULL COMMENT '博客标题',

`author` varchar(60) DEFAULT NULL COMMENT '博客作者',

`content` mediumtext COMMENT '博客内容',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb4

# 自己造的几条数据

INSERT INTO `blog`.`t\_blog`(`id`, `title`, `author`, `content`, `create\_time`, `update\_time`) VALUES (1, 'Springboot 为什么这', 'bywind', '没错 Springboot ', '2019-12-08 01:44:29', '2019-12-08 01:44:34');

INSERT INTO `blog`.`t\_blog`(`id`, `title`, `author`, `content`, `create\_time`, `update\_time`) VALUES (3, 'Springboot 中 Redis', 'bywind', 'Spring Boot', '2019-12-08 01:44:29', '2019-12-08 01:44:29');

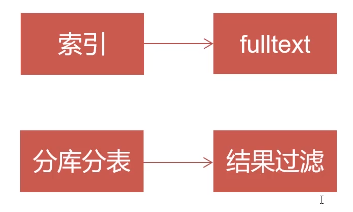
INSERT INTO `blog`.`t\_blog`(`id`, `title`, `author`, `content`, `create\_time`, `update\_time`) VALUES (4, 'Springboot 中如何优化', 'bywind', NULL, '2019-12-08 01:44:29', '2019-12-08 01:44:29');

INSERT INTO `blog`.`t\_blog`(`id`, `title`, `author`, `content`, `create\_time`, `update\_time`) VALUES (5, 'Springboot 消息队列', 'bywind', NULL, '2019-12-08 01:44:29', '2019-12-08 01:44:29');

INSERT INTO `blog`.`t\_blog`(`id`, `title`, `author`, `content`, `create\_time`, `update\_time`) VALUES (6, 'Docker Compose + Springboot', 'bywind', NULL, '2019-12-08 01:44:29', '2019-12-08 01:44:29');

检索

SELECT \* FROM t\_blog WHERE title LIKE "%spring%" or content LIKE "%spring%"



Mysql 不适合做全文检索的场景

**3-2 基于ES实现**

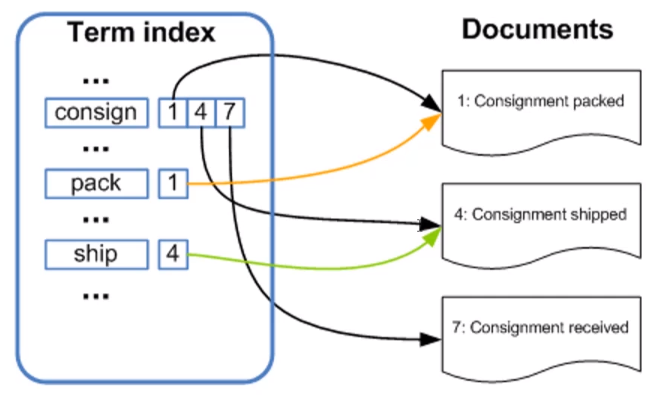
倒排索引工作原理

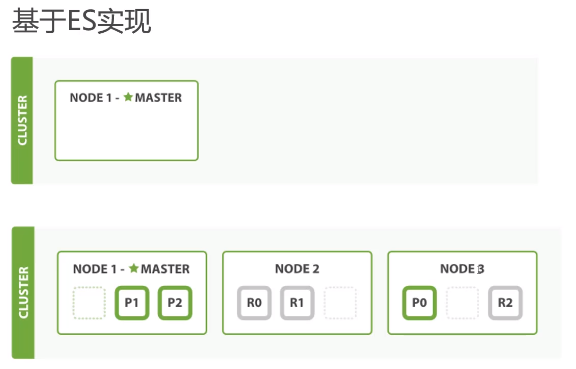
1.存入数据时使用分词器拆分数据

2.不同的词源指向不同的Documents（等同于MySQL中的Row）

3.ES会维护最小词源到DocumentID的映射（一个词源可以对应多个）

4.检索关键词时，ES会拆分最小词源，由于数据会存在不同的Document，根据不同的词源，可以共同定位到包含所有关键词的Document，根据权重，返回（例如：查询ABC，根据A可知存有A数据的ID有1.5.7，B数据的ID有2.5.8，D数据的ID有5.9.8，可知同时包含ABC的ID为5）





ES拥有更好的分布式、水平扩展能力

当有节点加入或移除时，整个集群会重新平均分布所有的数据

Master主要负责集群中索引的创建、删除以及数据的Rebalance等操作

对于节点的管理是透明的

**第4章 Mysql、ES 数据同步**

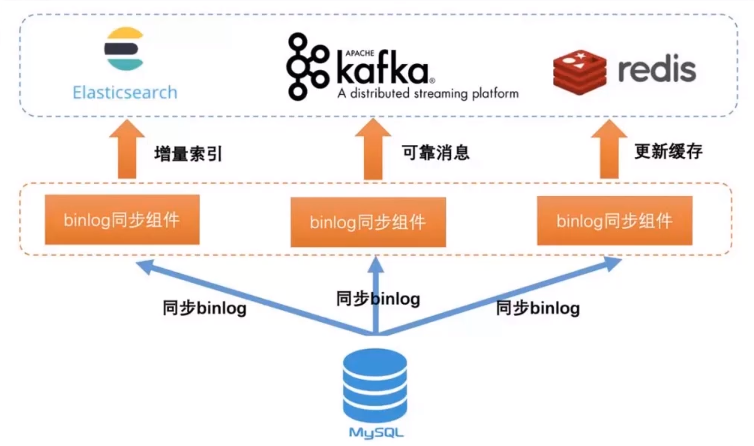
**4-1 数据同步中间件**

全量：第一次建立es索引后，把MySQL数据一次性全部打包同步

增量：MySQL产生新的数据（新增、更新、删除），对应同步到ES

#### 开源中间件

基于MySQL的binlog订阅



[Alibaba/canal](https://github.com/alibaba/canal)阿里巴巴 MySQL binlog 增量订阅&消费组件

[Siddontang/go-mysql-elasticsearch](https://github.com/siddontang/go-mysql-elasticsearch)

不足：不支持 ES6.X 以上、Mysql 8.X 以上



采集、转换、充实，然后输出。

官网地址

<https://www.elastic.co/cn/downloads/logstash>

<https://www.elastic.co/cn/downloads/past-releases/>



id MySQL和es之间映射关系

time 标识最大时间

**4-2 logstash全量、增量同步解决方案**

<https://www.elastic.co/cn/downloads/logstash>

jar 包下载地址

<https://mvnrepository.com/artifact/mysql/mysql-connector-java/5.1.31>

下载后放入logstash-6.3.2 文件夹下

在E:\logstash-6.3.2\config 中创建

**mysql.conf**

具体内容如下:

结合电脑环境进行配置驱动位置、用户名、密码等

input{

jdbc{

# jdbc驱动包位置

jdbc\_driver\_library => " E:\\logstash-6.3.2\\mysql-connector-java-5.1.31.jar"

# 要使用的驱动包类

jdbc\_driver\_class => "com.mysql.jdbc.Driver"

# mysql数据库的连接信息

jdbc\_connection\_string => "jdbc:mysql://127.0.0.1:3306/blog"

# mysql用户

jdbc\_user => "root"

# mysql密码

jdbc\_password => "root"

# 定时任务，多久执行一次查询，默认一分钟，如果想要没有延迟，可以使用 schedule => "\* \* \* \* \* \*"

schedule => "\* \* \* \* \*"

# 清空上传的sql\_last\_value记录

clean\_run => true

# 你要执行的语句

statement => "select \* FROM t\_blog WHERE update\_time > :sql\_last\_value AND update\_time < NOW() ORDER BY update\_time desc"

}

}

output {

elasticsearch{

# es host : port

hosts => ["127.0.0.1:9200"]

# 索引

index => "blog"

# \_id

document\_id => "%{id}"

}

}

在E:\logstash-6.3.2\bin 目录下打开命令窗口

1.按住Shift键，鼠标右键快捷方式，先打开Powershell窗口，输入 start cmd 回车，打开命令窗口

2.在命令窗口输入

logstash -f ../config/mysql.conf

（logstash -f config/mysql.conf）此条命令未测试

错误: 找不到或无法加载主类 Files\Java\jdk1.8.0\_221\lib;C:\Program

在55行左右 给%CLASSPATH%添加双引号 源文件内容

%JAVA% %JAVA\_OPTS% -cp %CLASSPATH% org.logstash.Logstash %\*

修改成

%JAVA% %JAVA\_OPTS% -cp "%CLASSPATH%" org.logstash.Logstash %\*

3.等待一段时间，输出配置文件中的如下SQL命令，表示成功

select \* FROM t\_blog WHERE update\_time > '2020-08-01 17:16:00' AND update\_time < NOW() ORDER BY update\_time desc

7.5.0版本**报错，错误信息：**

Unable to find driver class via URLClassLoader in given driver jars: com.mysql.jdbc.Driver and com.mysql.jdbc.Driver

logstash7.5.0

这里只是一个解决方法。只需将驱动程序Jar文件复制到<logstash\_install\_dir>/logstash-core/lib/jars/目录。

然后删除conf里的jdbc驱动包配置

**mysql.conf**

input{

jdbc{

# 要使用的驱动包类

jdbc\_driver\_class => "com.mysql.jdbc.Driver"

# mysql数据库的连接信息

jdbc\_connection\_string => "jdbc:mysql://127.0.0.1:3306/blog"

# mysql用户

jdbc\_user => "root"

# mysql密码

jdbc\_password => "root"

# 定时任务，多久执行一次查询，默认一分钟，如果想要没有延迟，可以使用 schedule => "\* \* \* \* \* \*"

schedule => "\* \* \* \* \*"

# 清空上传的sql\_last\_value记录

clean\_run => true

# 你要执行的语句

statement => "select \* FROM t\_blog WHERE update\_time > :sql\_last\_value AND update\_time < NOW() ORDER BY update\_time desc"

}

}

output {

elasticsearch{

# es host : port

hosts => ["127.0.0.1:9200"]

# 索引

index => "blog"

# \_id

document\_id => "%{id}"

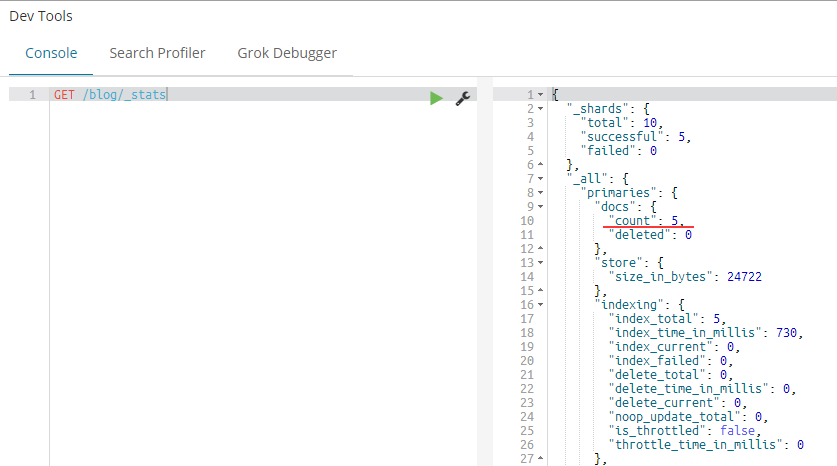
}

}

**成功**

在Kibana输入命令查看是否同步成功

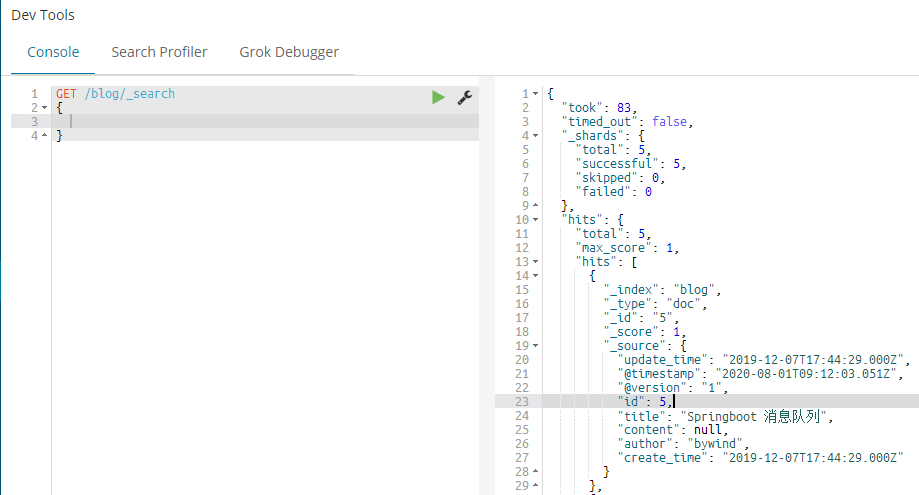
GET /blog/\_stats

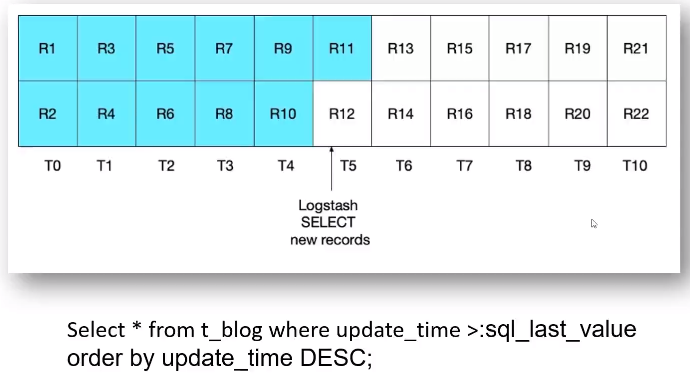


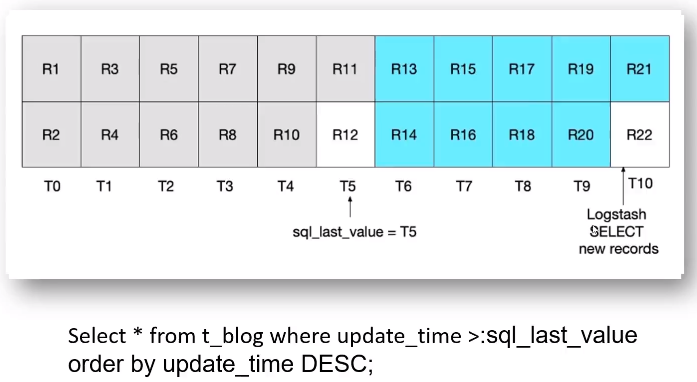
查看全部内容

GET /blog/\_search

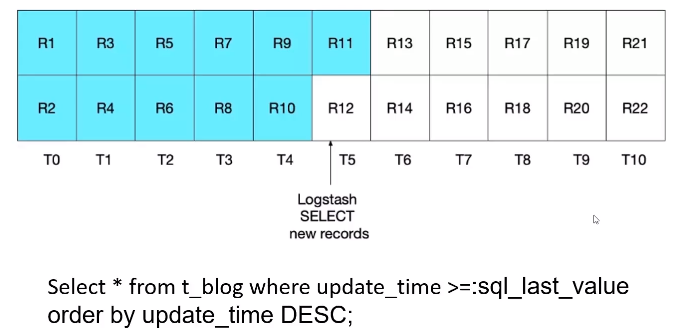
{}

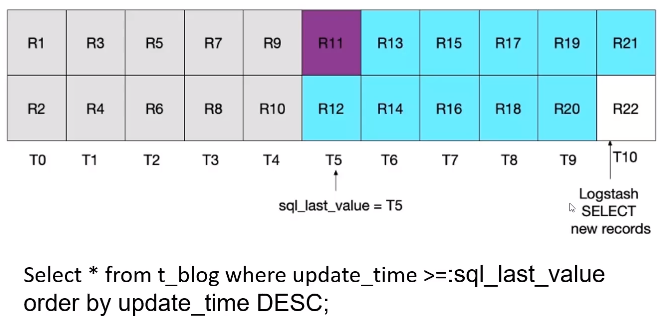




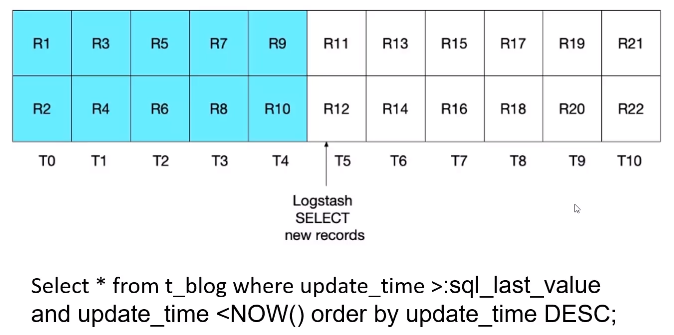


临界点数据不会被扫描到，不会同步到ES





临界点老数据太多，造成不必要的开销



解决临界点数据无法扫描和临界点老数据重复扫描的问题

**第 5 章**

**5-1 分词器介绍**

standard：ES默认，将词汇单元转换为小写形式，并且去除停用词、标点符号，支持中文（单字切分）。

simple：通过非字母字符来分割文本信息，然后将词汇单元统一转换为小写形式，会去除掉数字类型的字符。

whitespace：仅仅是去除空格、小写，不支持中文，对分割的词汇单元不做标准化的处理。

language：特定语言的分词器，不支持中文。

**5-2 IK分词器的安装和使用**

使用kibanna

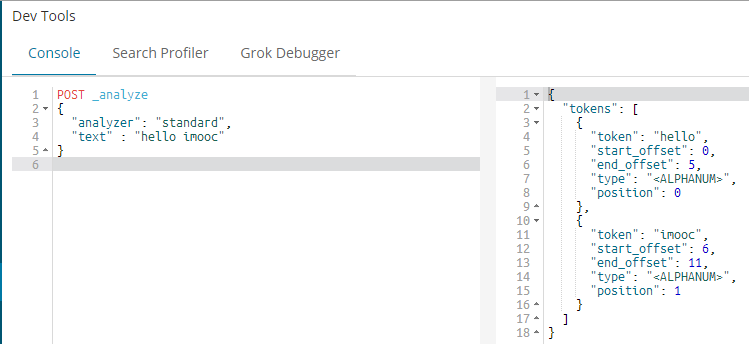
POST \_analyze

{

"analyzer": "standard",

"text" : "hello imooc"

}



按单字 分割

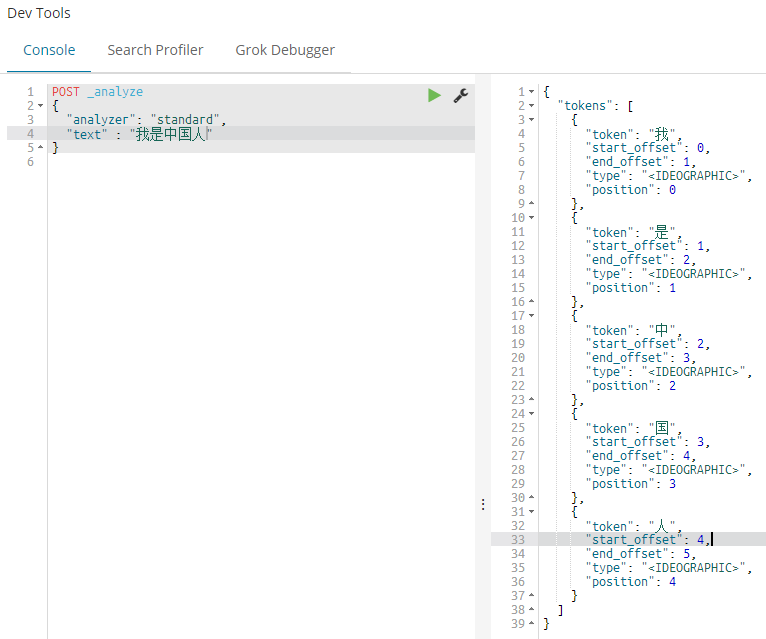
POST \_analyze

{

"analyzer": "standard",

"text" : "我是中国人"

}



**ik分词器下载地址：（注意版本一致）**

<https://github.com/medcl/elasticsearch-analysis-ik/releases>

<https://github.com/medcl/elasticsearch-analysis-ik>

主要原理：ik分词器的config目录下，存储由中国词语构成的词典

**下载对应版本的分词器,**

**在E:\elasticsearch-6.3.2\plugins\ 目录下， 新建ik文件夹，（尽量起英文不加空格）**

**把解压内容拷贝到ik文件夹下**

**重启ES，**双击E:\elasticsearch-6.3.2\bin目录下的 elasticsearch.bat 运行

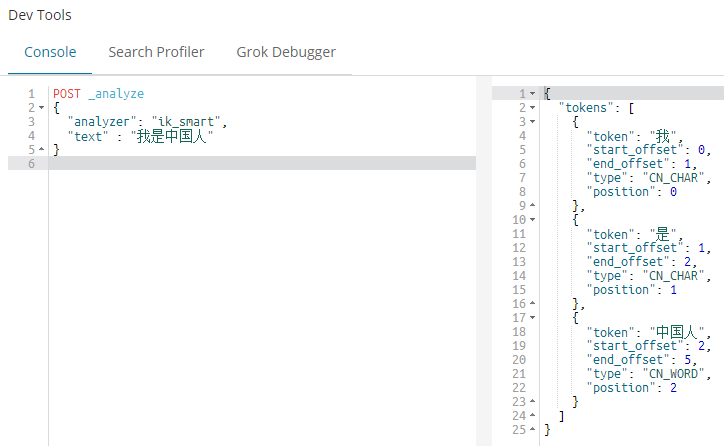
POST \_analyze

{

"analyzer": "ik\_smart",

"text" : "我是中国人"

}



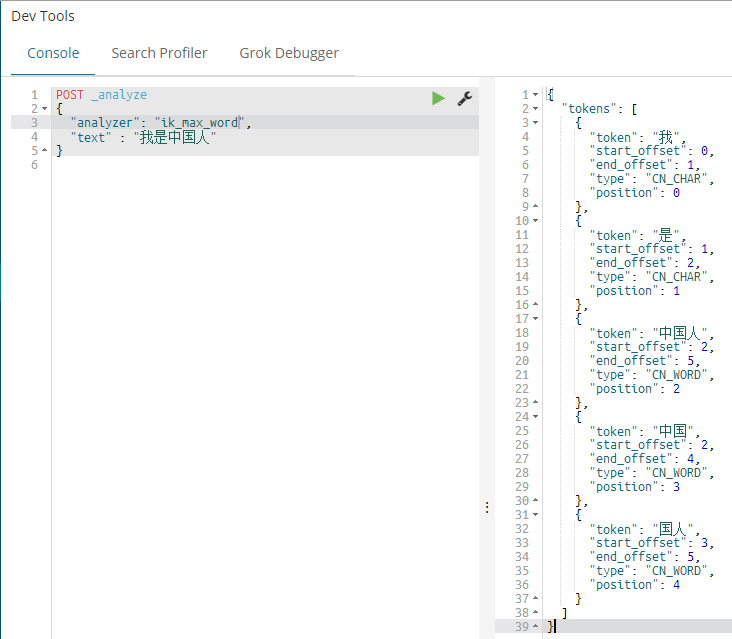
POST \_analyze

{

"analyzer": "ik\_max\_word",

"text" : "我是中国人"

}



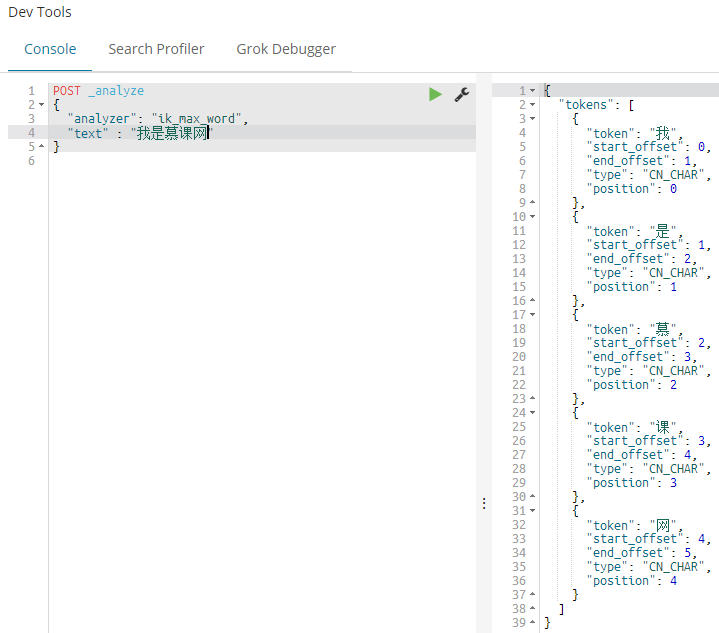
POST \_analyze

{

"analyzer": "ik\_max\_word",

"text" : "我是慕课网"

}



在E:\elasticsearch-6.3.2\plugins\ik\config目录下的

Main.dic添加慕课网后，

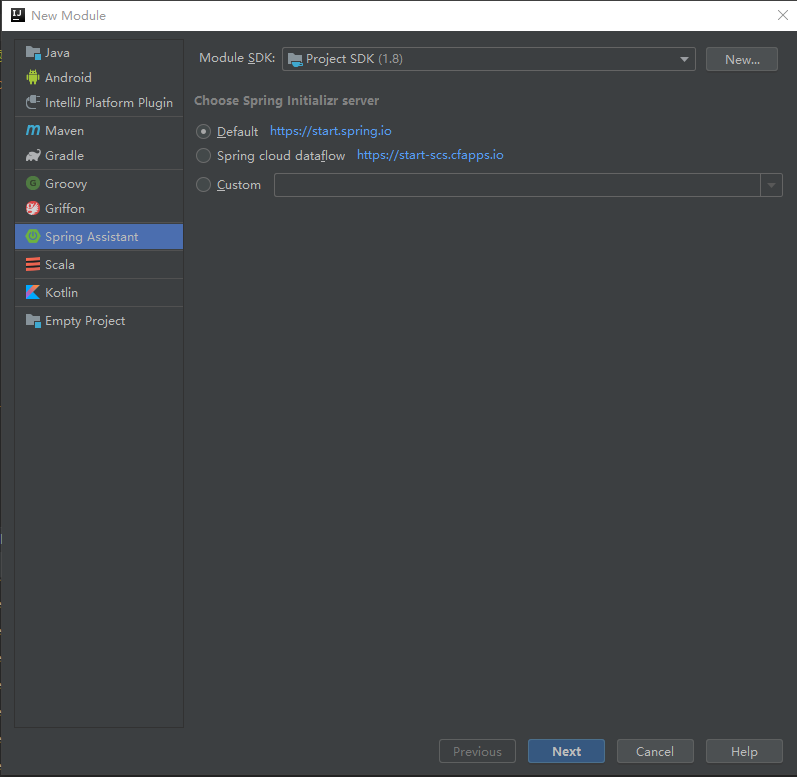
**重启ES，**双击E:\elasticsearch-6.3.2\bin目录下的 elasticsearch.bat 运行

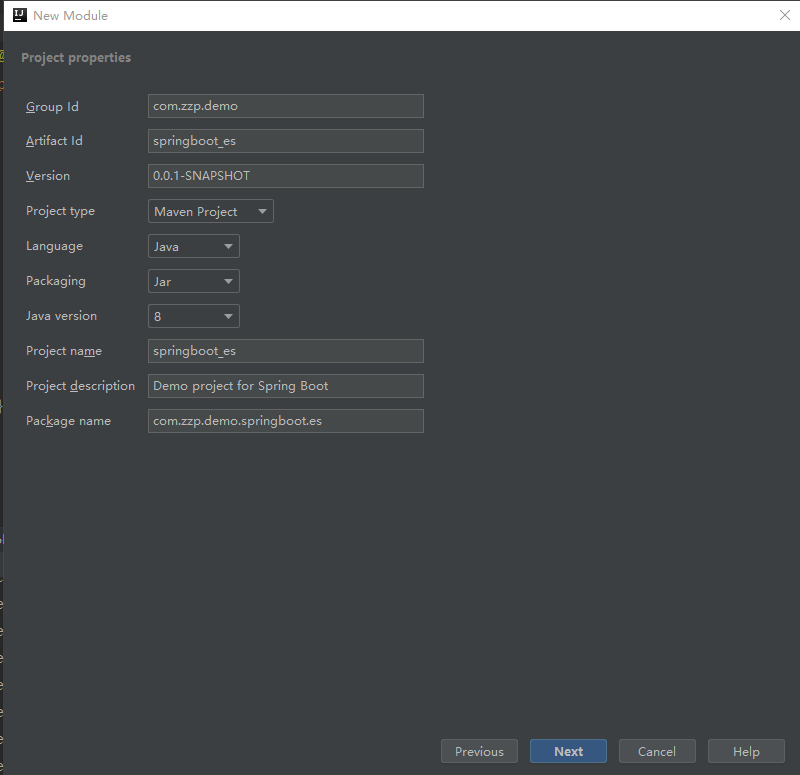


**5-3 springboot 项目搭建**

IDE： idea

步骤：File→New Module





选择依赖

Developer Tools

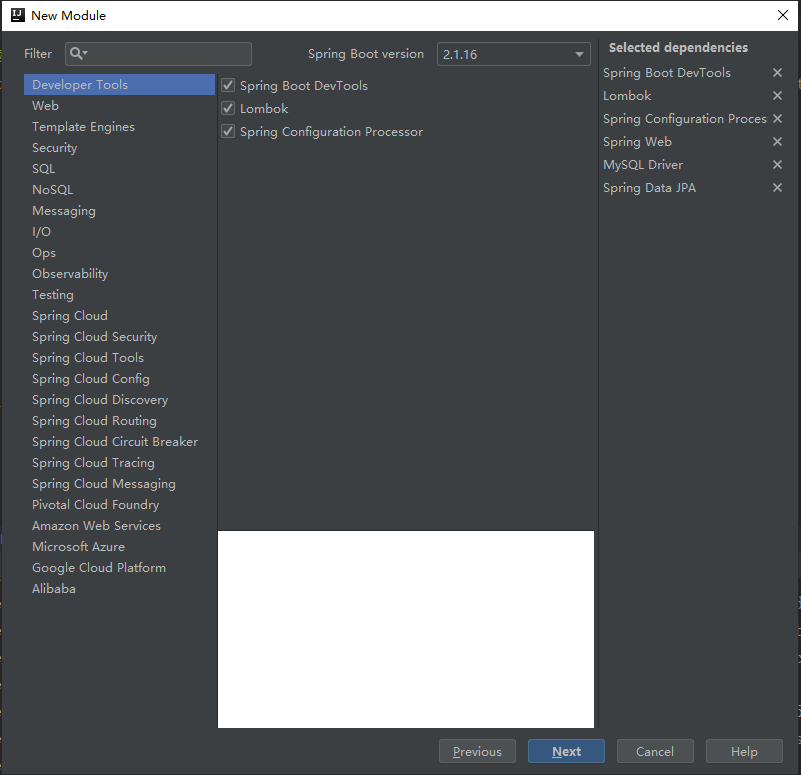
* Spring Boot DevTools
* Lombok
* Spring Configuration Processor

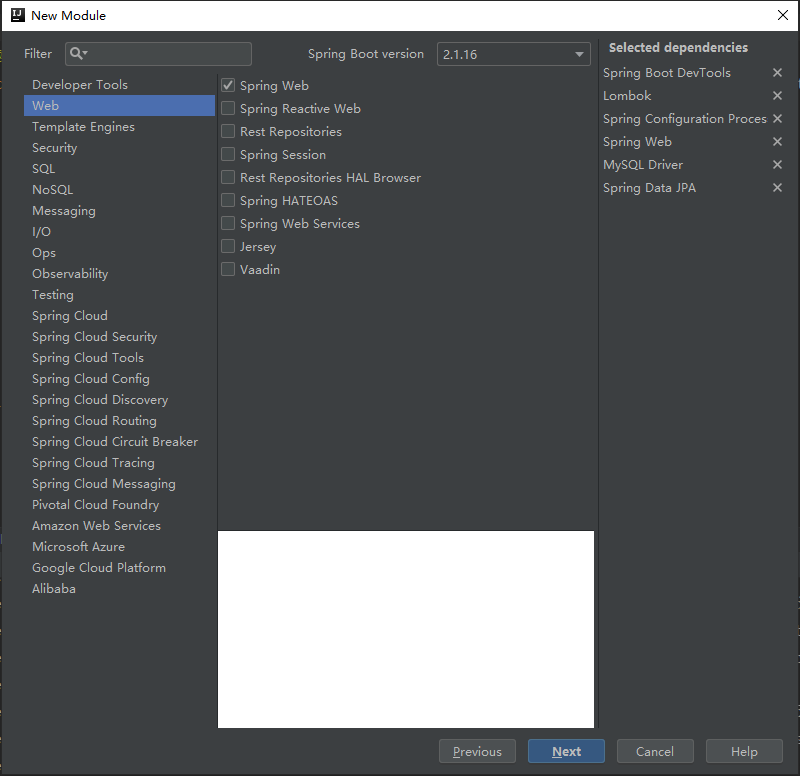
Web

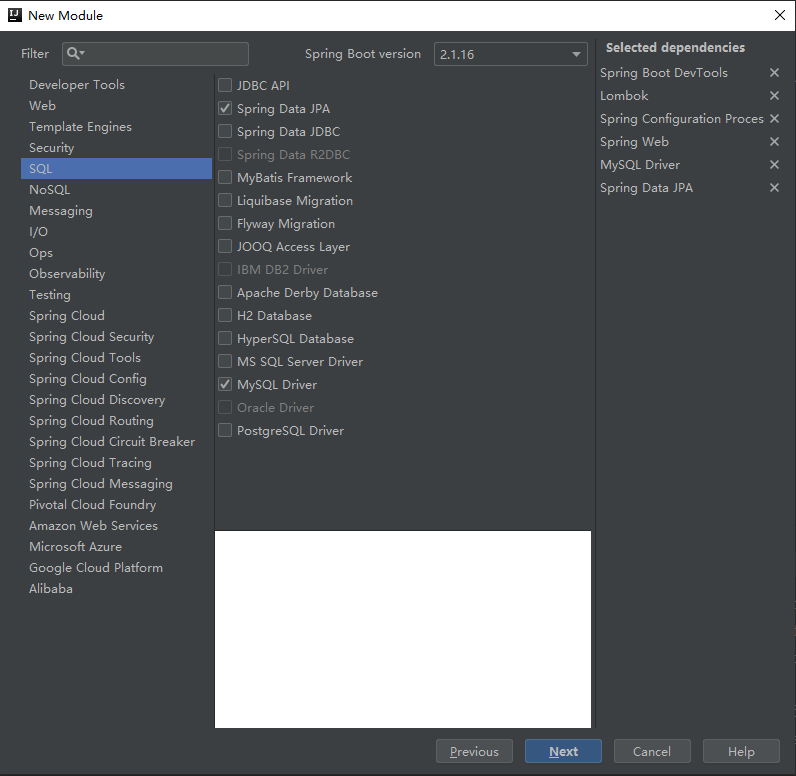
* Spring Web

SQL

* Spring Data JPA
* MySQL Driver







选择位置

等待加载完成

对application.properties 进行配置，内容如下：

#通用数据源配置

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/blog?allowPublicKeyRetrieval=true&useSSL=false&useUnicode=true&characterEncoding=utf8

spring.datasource.username=root

spring.datasource.password=root

#Hikari数据源专用配置

spring.datasource.hikari.maximum-pool-size=20

spring.datasource.hikari.minimum-idle=5

#JPA相关配置

spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect

#es

spring.data.elasticsearch.cluster-nodes=127.0.0.1:9300

spring.data.elasticsearch.cluster-name=elasticsearch

#mvc

spring.mvc.static-path-pattern=/\*\*

spring.devtools.livereload.enabled=true

spring.devtools.restart.additional-paths=static/\*\*

#日期格式化4

spring.jackson.date-format=yyyy-MM-dd HH:mm:ss

或者对application.properties重命名为application.yml， 配置内容如下：

server:

port: 8081

spring:

#数据库配置

datasource:

driver-class-name: com.mysql.cj.jdbc.Driver

url: jdbc:mysql://localhost:3306/blog?useUnicode=true&characterEncoding=utf-8

username: root

password: root

# hikari 数据源专用配置

hikari:

maximum-pool-size: 20

minimum-idle: 5

# jpa 相关配置

jpa:

hibernate:

ddl-auto: update

show-sql: true

# 数据库方言

database-platform: org.hibernate.dialect.MySQLDialect

# es 配置

data:

elasticsearch:

cluster-nodes: localhost:9300

cluster-name: elasticsearch # es 集群的名称

# mvc 静态资源映射

mvc:

static-path-pattern: /\*\*

# 静态资源热部署

devtools:

livereload:

enabled: true

restart:

additional-paths: static/\*\*

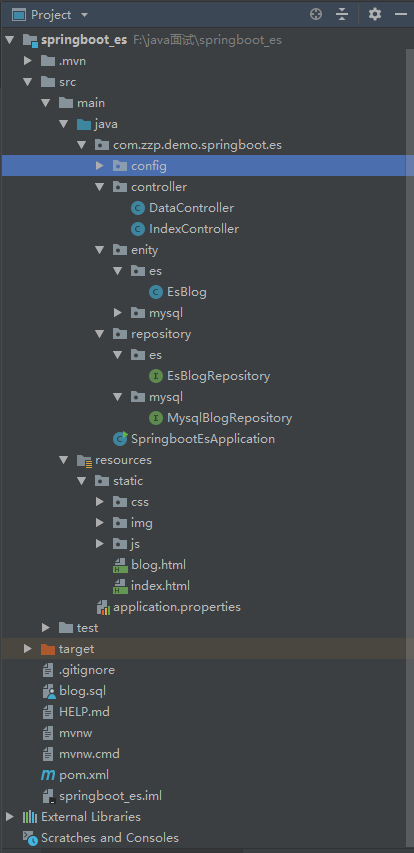
# 日期格式化

jackson:

date-format: yyyy-MM-dd HH:mm:ss

**5-4 项目结构和JPA演示**

Debug 运行、F8 进行下一步



**5-5 springboot 集成ES**

**Pom.xml 中添加依赖**

<!--spring data 操作es-->  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-elasticsearch</artifactId>  
</dependency>

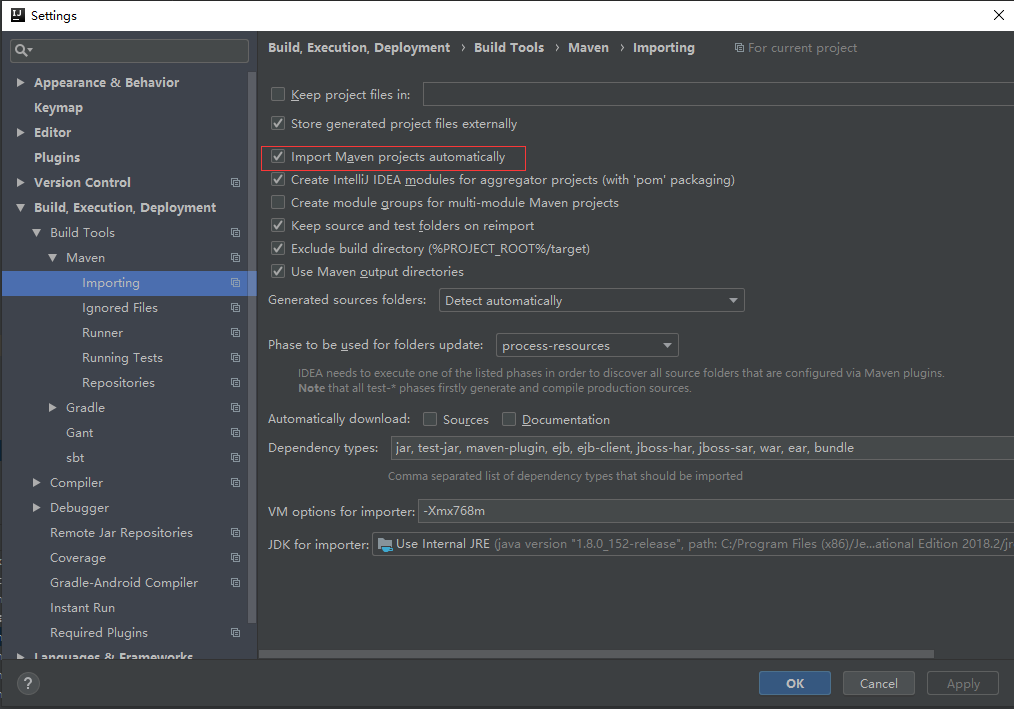
出现红色，表示资源没加载，尝试用以下方法解决：

1.右键pom.xml点击Maven→Download Sources

2.File→Setting→Build→Build Tools→Maven→Importing 勾选

Import Maven projects automatically

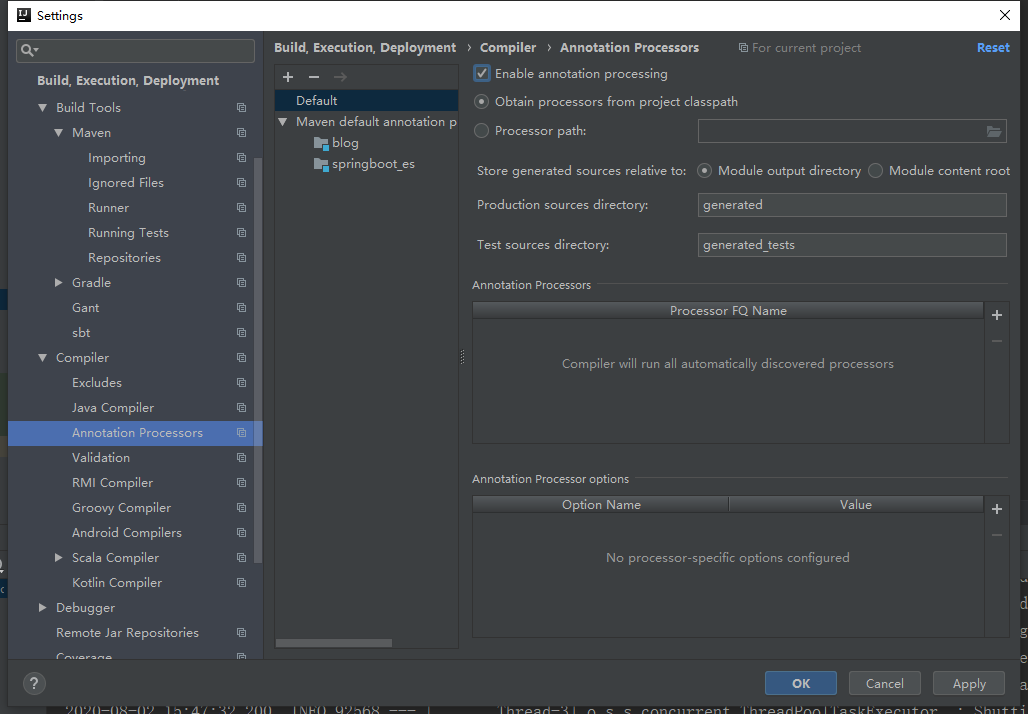
如下图所示：



@Data注解，不起作用

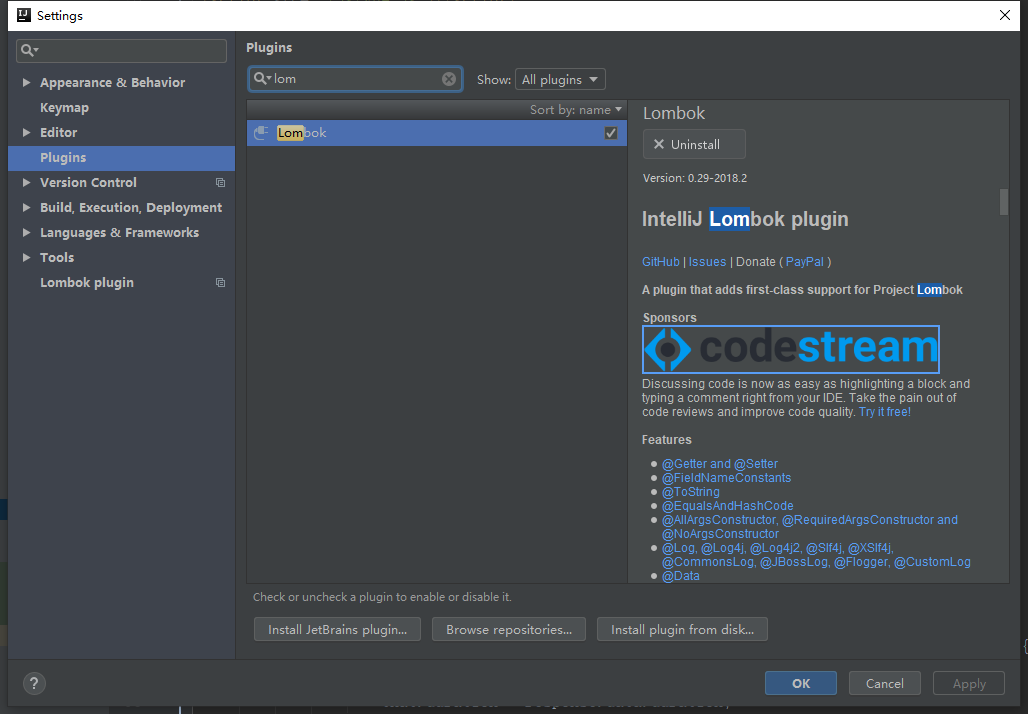
1.Setting→Build→Compiler→Annotation Processors

勾选 Enable annotation processing



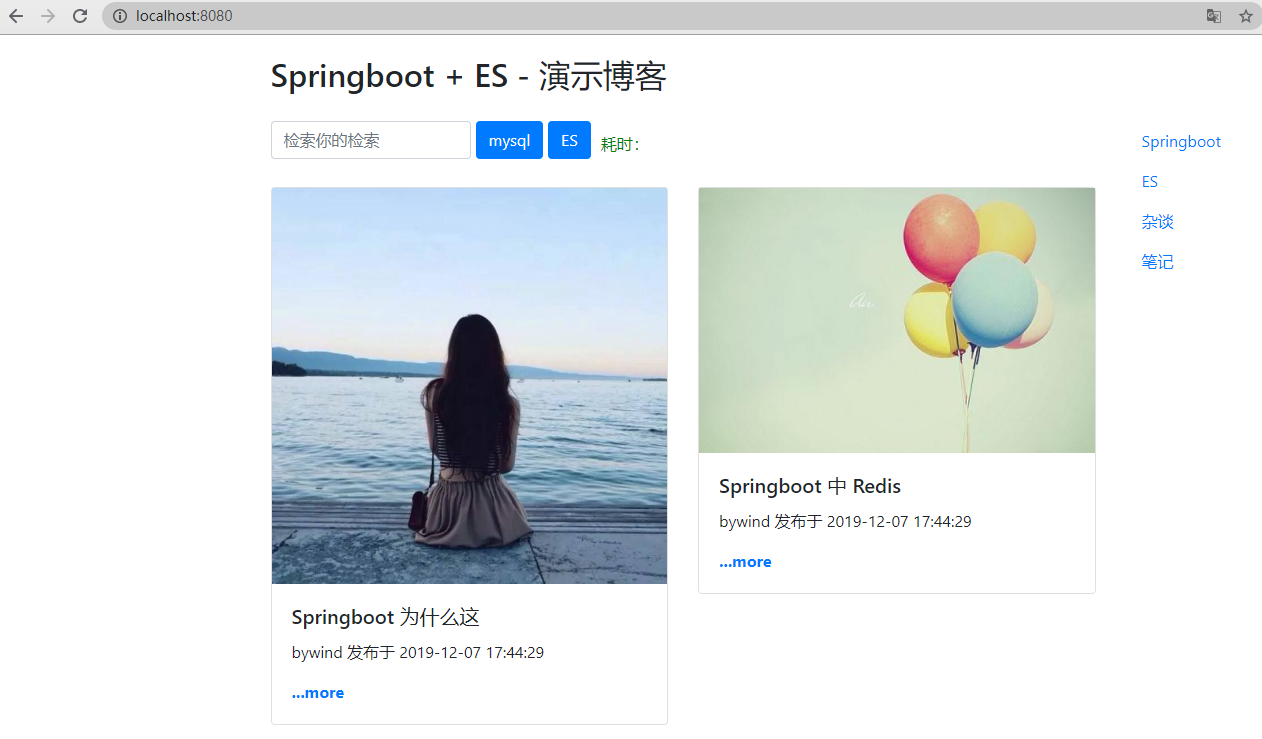
2.Setting→plugin→安装lombok 插件

重启



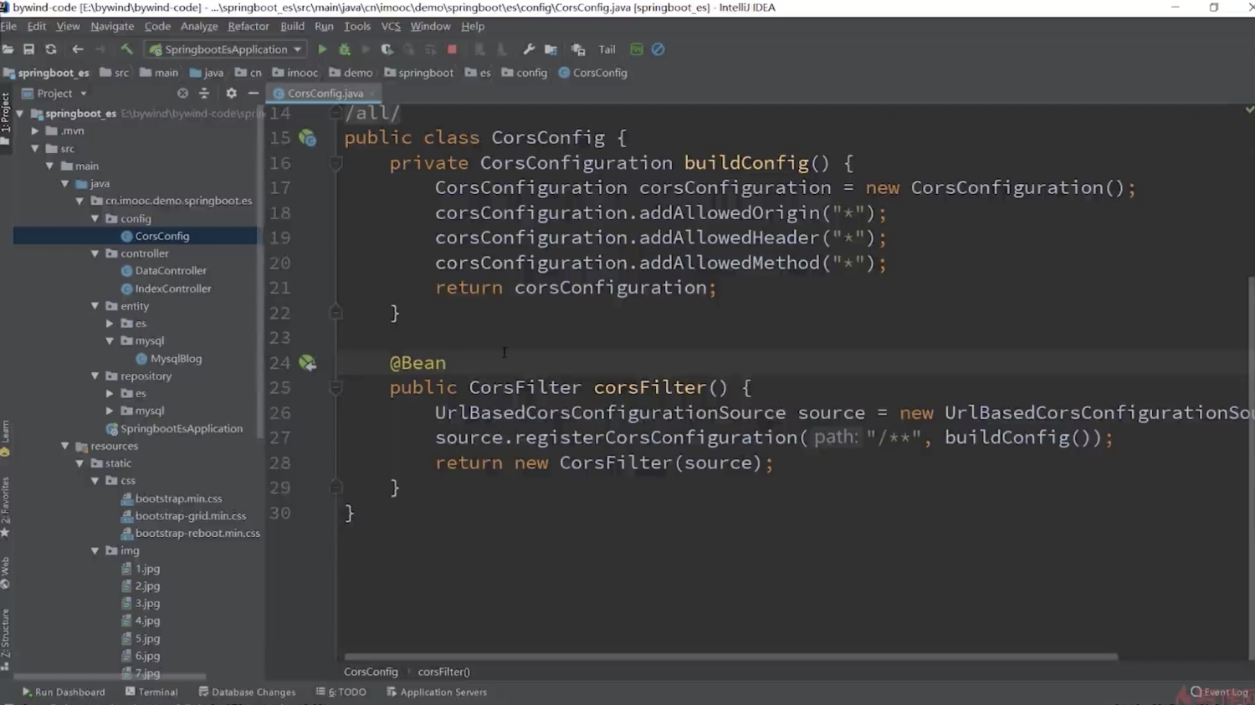
**5-6 项目后端 REST API 实现**

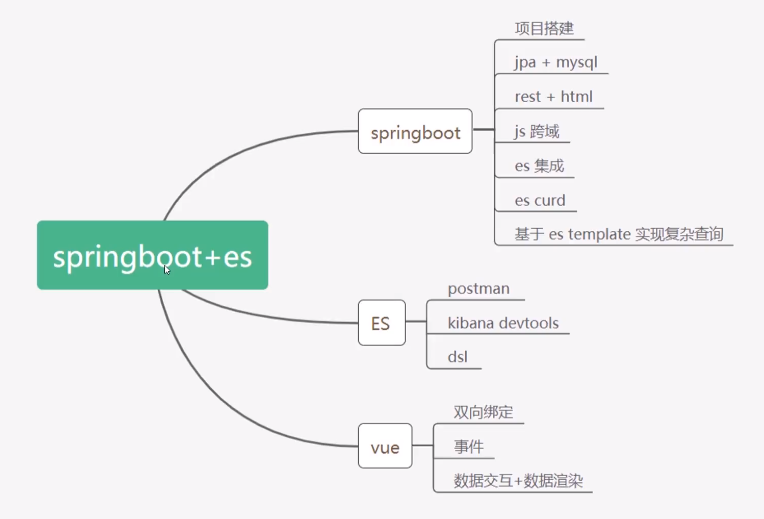
注：端口号保持一致



**第 6 章**

**6-1 课程回顾与总结**





**最后**

首先感谢大佬的课程分享，通过课程学到了很多东西