



链滴

SpringBoot 2.0 集成 spring-data-elasticsearch

作者: [limao](#)

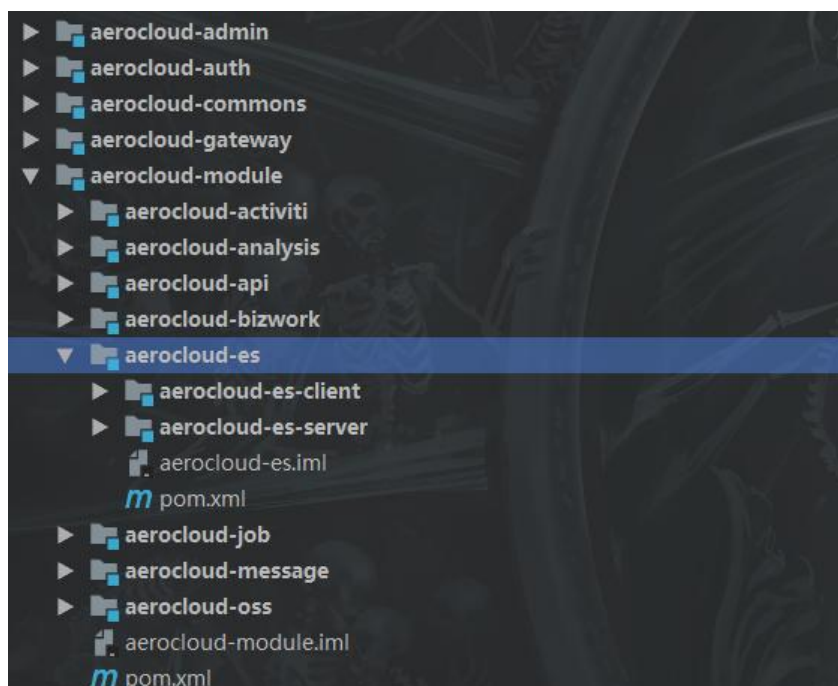
原文链接: <https://ld246.com/article/1582810782539>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

其实是spring cloud 集成 es 的,但是我想了下 还是把标题写成spring boot的吧,因为 cloud 的实现是基于 spring boot ,我这es 模块也是cloud 里的一个服务而已....

微服务目录结构...



我在module 下新建了一个spring boot 取名 es ,用来ElasticSearch独立服务使用

pom依赖...

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<parent>
```

```
<artifactId>aerocloud-es</artifactId>
```

```
<groupId>com.aerocloud</groupId>
```

```
<version>1.0.0</version>
```

```
</parent>
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<artifactId>aerocloud-es-server</artifactId>
```

```
<packaging>jar</packaging>
```

```
<version>1.0.0</version>
```

```
<dependencies>
```

```
<!-- es 依赖-->
```

```
<dependency>
```

```

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

<!--<dependency>-->
    <!--<groupId>org.springframework.data</groupId>-->
    <!--<artifactId>spring-data-elasticsearch</artifactId>-->
<!--</dependency>-->

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.aerocloud</groupId>
    <artifactId>aerocloud-commons-tools</artifactId>
    <version>1.0.0</version>
</dependency>
<!--<dependency>-->
    <!--<groupId>com.aerocloud</groupId>-->
    <!--<artifactId>aerocloud-commons-mybatis</artifactId>-->
    <!--<version>1.0.0</version>-->
<!--</dependency>-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context-support</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-stream</artifactId>
</dependency>

<!--<dependency>-->
    <!--<groupId>com.aerocloud</groupId>-->
    <!--<artifactId>aerocloud-commons-dynamic-datasource</artifactId>-->
    <!--<version>1.0.0</version>-->
<!--</dependency>-->
<dependency>
    <groupId>de.codecentric</groupId>
    <artifactId>spring-boot-admin-starter-client</artifactId>
    <version>${spring.boot.admin.version}</version>
</dependency>

```

```

<dependency>
  <groupId>io.swagger</groupId>
  <artifactId>swagger-annotations</artifactId>
  <version>1.5.20</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger2</artifactId>
</dependency>
<dependency>
  <groupId>com.aerocloud</groupId>
  <artifactId>aerocloud-es-client</artifactId>
  <version>1.0.0</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>com.aerocloud</groupId>
  <artifactId>aerocloud-analysis-client</artifactId>
  <version>1.0.0</version>
  <scope>compile</scope>
</dependency>
</dependencies>
<build>
  <finalName>${project.artifactId}</finalName>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <includeSystemScope>true</includeSystemScope>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <configuration>
        <skipTests>true</skipTests>
      </configuration>
    </plugin>
    <plugin>
      <groupId>com.spotify</groupId>
      <artifactId>docker-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

</project>
...

```

因为我是微服务架构所以会有注册中心`eureka`

yaml配置

```
spring:
  data:
    elasticsearch:
      #cluster-name: #默认为elasticsearch
      cluster-nodes: 127.0.0.1:9300 #配置es节点信息，逗号分隔，如果没有指定，则启动Client
ode (9200端口是http查询使用的。9300集群使用。这里使用9300.)
      properties:
        path:
          logs: ./elasticsearch/log #elasticsearch日志存储目录
          data: ./elasticsearch/data #elasticsearch数据存储目录
```

/**

@Document注解里面的几个属性，类比mysql的话是这样：

index -> DB

type -> Table

Document -> row

```
*/
@Data
@Document(indexName = "cpdb", type = "cptesttable", shards = 1, replicas = 0)
public class cpTest {
    @Id
    private Long id;
    @Field(type = FieldType.Text, analyzer = "ik_max_word")
    private String title; //标题
    @Field(type = FieldType.Keyword)
    private String category; // 分类

    @Field(type = FieldType.Keyword)
    private String brand; // 品牌

    @Field(type = FieldType.Double)
    private Double price; // 价格

    @Field(index = false, type = FieldType.Keyword)
    private String images; // 图片地址

}
```

加上@Id注解后，在Elasticsearch里对应的该列就是主键了，在查询时就可以直接用主键查询。其实mysql非常类似，基本就是一个数据库。

```
@Persistent
@Inherited
@Retention(RetentionPolicy.RUNTIME)
@Target({ElementType.TYPE})
```

```

public @interface Document {
    String indexName();//索引库的名称，个人建议以项目的名称命名
    String type() default "";//类型，个人建议以实体的名称命名
    short shards() default 5;//默认分区数
    short replicas() default 1;//每个分区默认的备份数
    String refreshInterval() default "1s";//刷新闻隔
    String indexStoreType() default "fs";//索引文件存储类型
}

```

加上了@Document注解之后，默认情况下这个实体中所有的属性都会被建立索引、并且分词。通过@Field注解来进行详细的指定，如果没有特殊需求，那么只需要添加@Document即可。

@Field注解的定义如下：

```

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.FIELD)
@Documented
@Inherited
public @interface Field {

    FieldType type() default FieldType.Auto;#自动检测属性的类型
    FieldIndex index() default FieldIndex.analyzed;#默认情况下分词
    DateFormat format() default DateFormat.none;
    String pattern() default "";
    boolean store() default false;#默认情况下不存储原文
    String searchAnalyzer() default "";#指定字段搜索时使用的分词器
    String indexAnalyzer() default "";#指定字段建立索引时指定的分词器
    String[] ignoreFields() default {};//如果某个字段需要被忽略
    boolean includeInParent() default false;
}

```

ElasticsearchRepository

```

/**
 * 只需要写接口 不需要写实现
 * 不需要加@Component，直接可以@Autowared
 * spring data 自动通过方法命名去实现
 * 例子 findById 这个方法名自动实现通过Id查询对象
 */
public interface cpTestRepository extends ElasticsearchRepository<cpTest, Long> {

    List<cpTest> findByBrand(String name);

}

```

ElasticsearchRepository 内置了crud 方法 方便简单功能使用

复杂的查询还是得 了解dsl 语法用 [elasticsearchTemplate](#) 来构造条件去请求