**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

学成在线-第11天-讲义-搜索服务



* 课程搜索需求分析



1.1 需求分析



1、根据分类搜索课程信息。

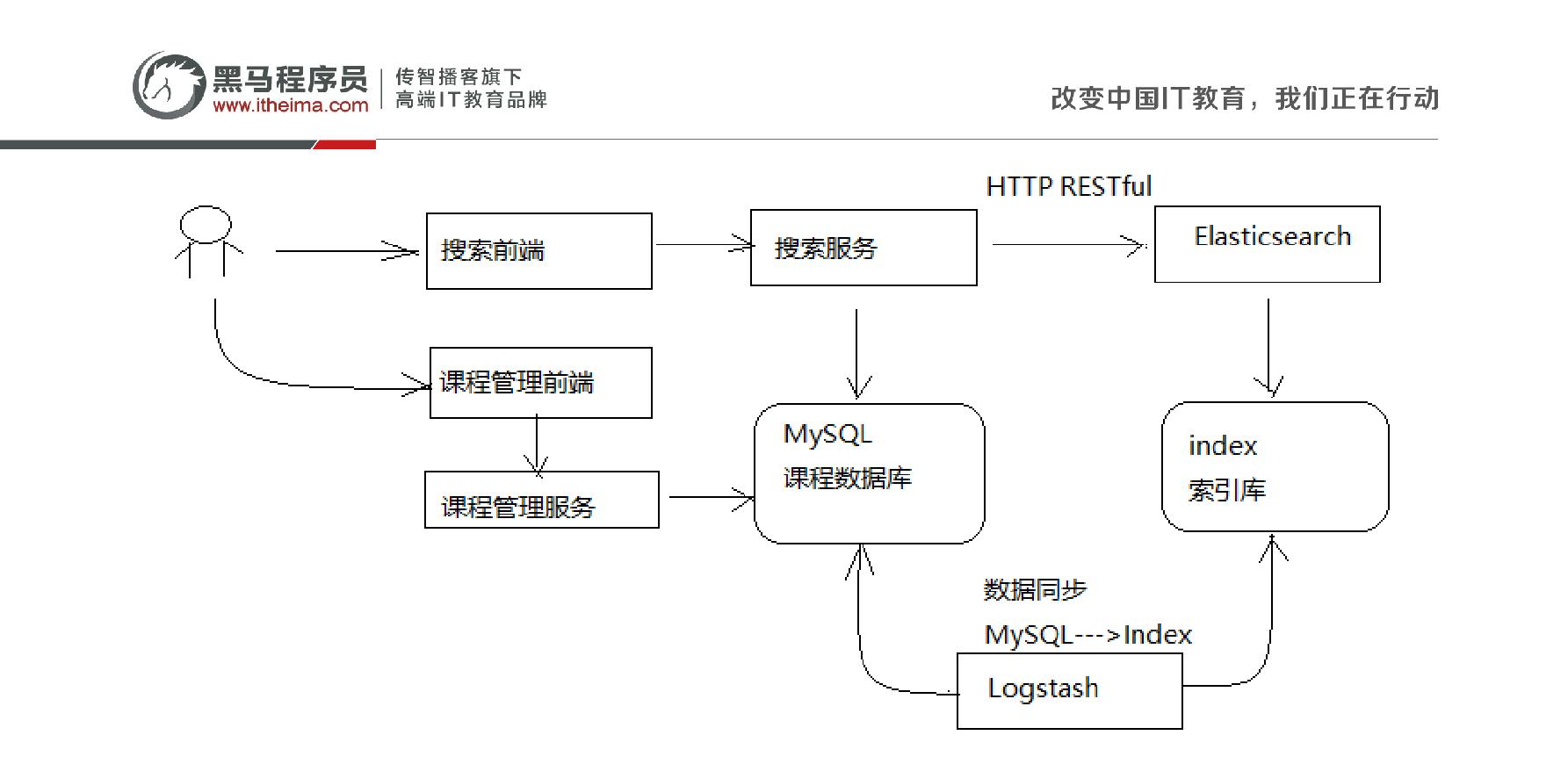
2、根据关键字搜索课程信息，搜索方式为全文检索，关键字需要匹配课程的名称、 课程内容。

3、根据难度等级搜索课程。

4、搜索结点分页显示。

1.2 搜索流程



**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

1、课程管理服务将数据写到MySQL数据库

2、使用Logstash将MySQL数据库中的数据写到ES的索引库。

3、用户在前端搜索课程信息，请求到搜索服务。

4、搜索服务请求ES搜索课程信息。

* 全文检索技术研究



参考：elasticsearch研究.md

研究ElasticSearch搜索方法。

* 课程索引



3.1 技术方案



如何维护课程索引信息？

1、当课程向MySQL添加后同时将课程信息添加到索引库。

采用Logstach实现，Logstach会从MySQL中将数据采集到ES索引库。

2、当课程在MySQL更新信息后同时更新该课程在索引库的信息。

采用Logstach实现。

3、当课程在MySQL删除后同时将该课程从索引库删除。

手工写程序实现，在删除课程后将索引库中该课程信息删除。

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

3.2 准备课程索引信息



课程发布成功在MySQL数据库存储课程发布信息，此信息作为课程索引信息。

3.2.1创建课程发布表

课程信息分布在course\_base、course\_pic等不同的表中。

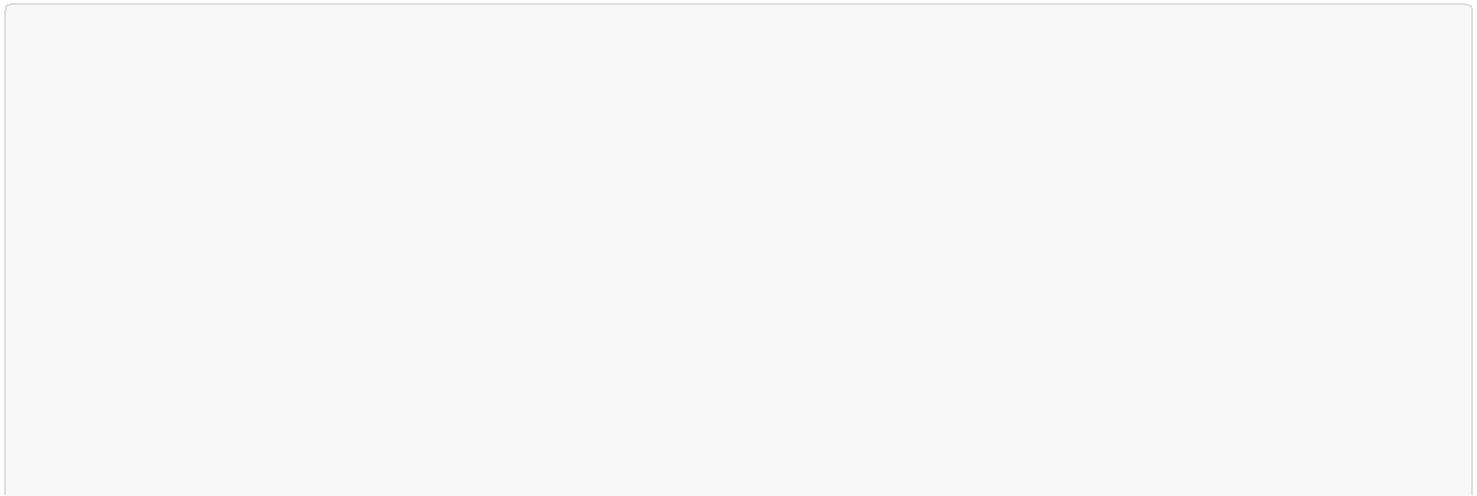
课程发布成功为了方便进行索引将这几张表的数据合并在一张表中，作为课程发布信息。

创建course\_pub表



3.2.2创建课程发布表模型

在课程管理服务创建模型：



@Data

@ToString

@Entity

@Table(name="course\_pub")

@GenericGenerator(name = "jpa‐assigned", strategy = "assigned")

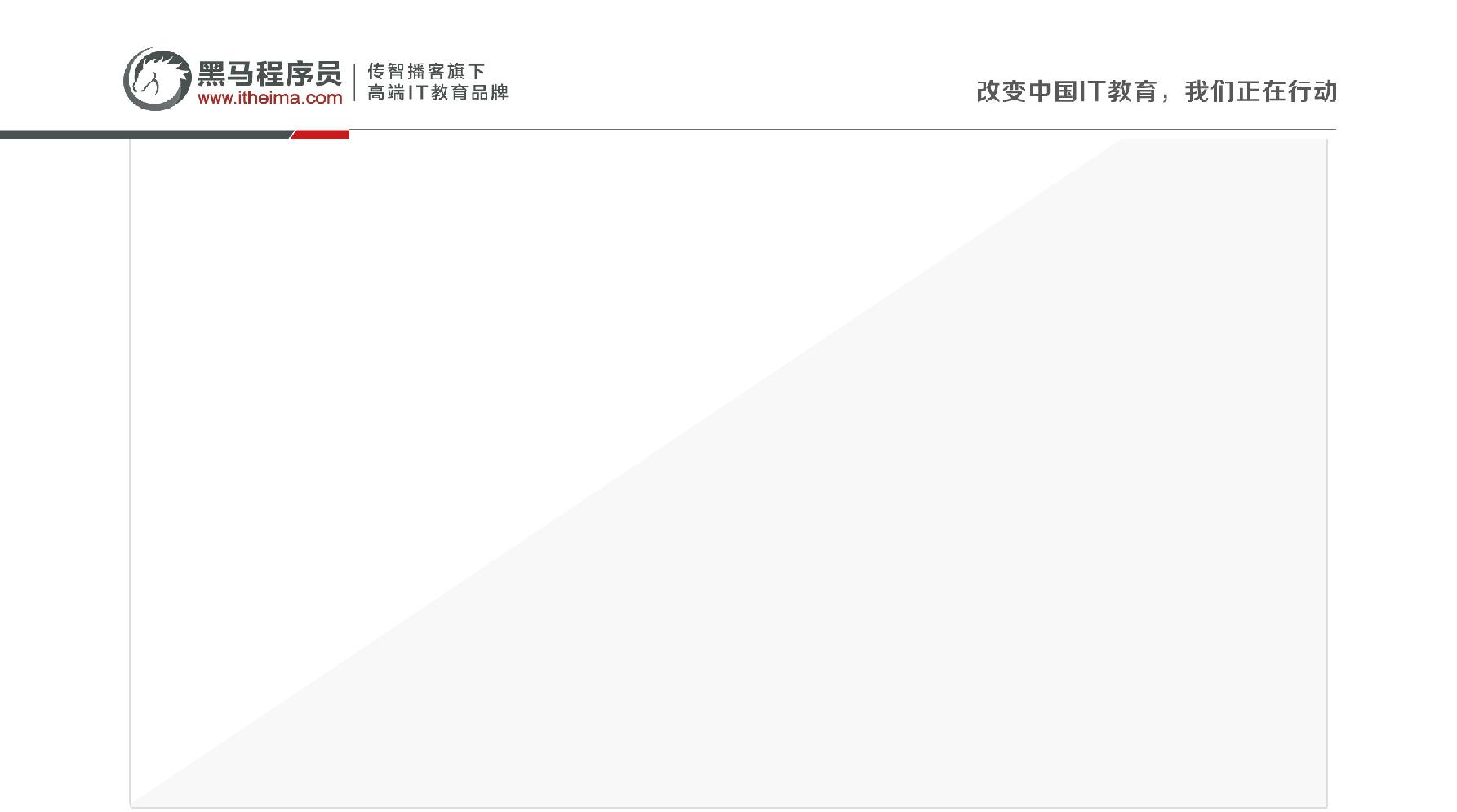
public class CoursePub implements Serializable {

private static final long serialVersionUID = ‐916357110051689487L; @Id

@GeneratedValue(generator = "jpa‐assigned")

@Column(length = 32)

private String id;

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

private String name;

private String users;

private String mt;

private String st;

private String grade;

private String studymodel;

private String teachmode;

private String description;

private String pic;//图片

private Date timestamp;//时间戳

private String charge;

private String valid;

private String qq;

private Float price;

private Float price\_old;

private String expires;

private String teachplan;//课程计划

@Column(name="pub\_time")

private String pubTime;//课程发布时间

}

3.2.3修改课程发布

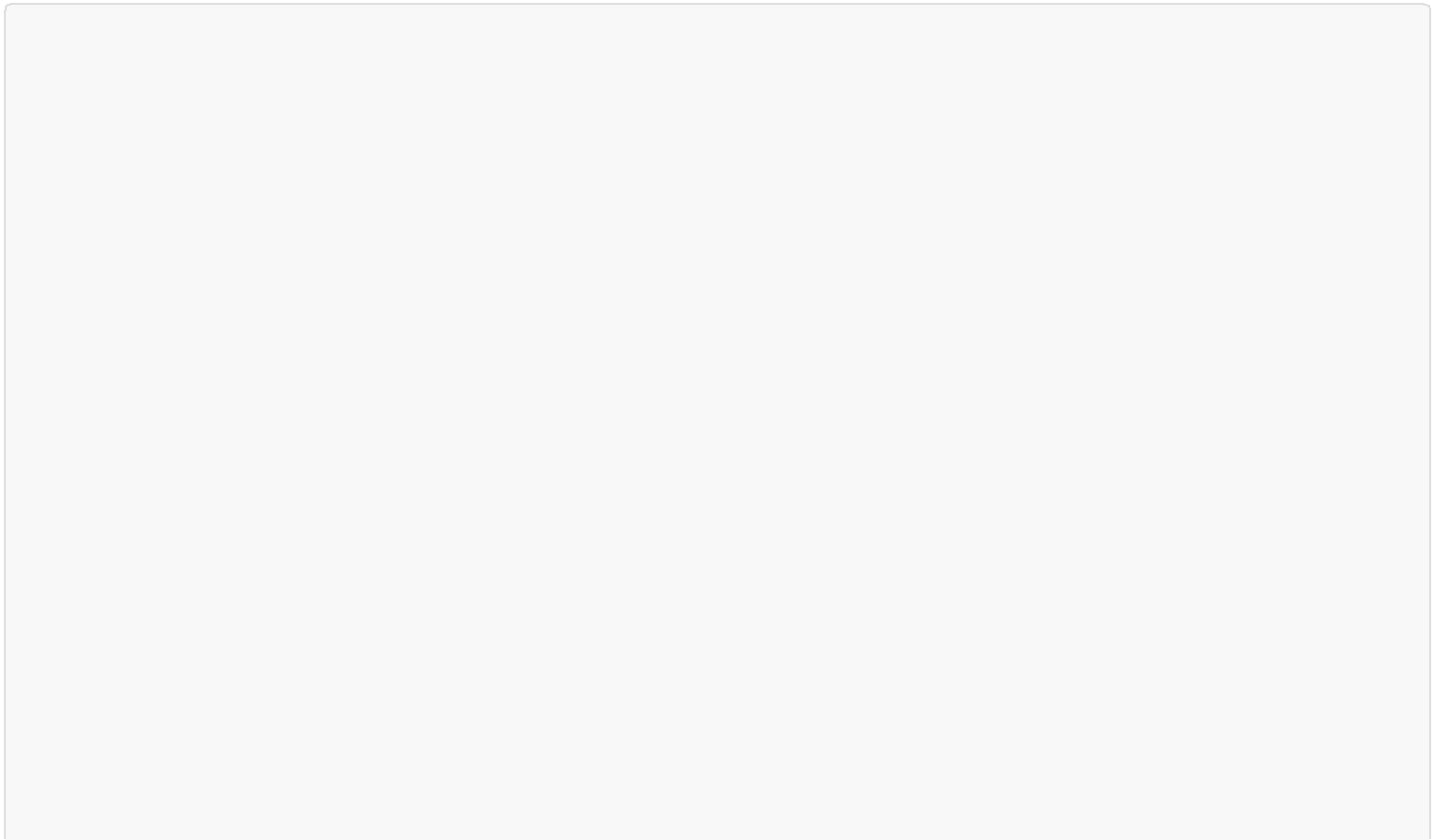
在课程管理服务定义dao：

1）创建course\_pub表的dao



public interface CoursePubRepository extends JpaRepository<CoursePub, String> { }

2) 修改课程发布service



//保存CoursePub

public CoursePub saveCoursePub(String id, CoursePub coursePub){ if(StringUtils.isNotEmpty(id)){

ExceptionCast.cast(CourseCode.COURSE\_PUBLISH\_COURSEIDISNULL);

}

CoursePub coursePubNew = null;

Optional<CoursePub> coursePubOptional = coursePubRepository.findById(id); if(coursePubOptional.isPresent()){

coursePubNew = coursePubOptional.get();

}

if(coursePubNew == null){

coursePubNew = new CoursePub();

}

BeanUtils.copyProperties(coursePub,coursePubNew);

//设置主键

coursePubNew.setId(id);

//更新时间戳为最新时间

coursePub.setTimestamp(new Date());

//发布时间

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("YYYY‐MM‐dd HH:mm:ss");

String date = simpleDateFormat.format(new Date());

coursePub.setPubTime(date);

coursePubRepository.save(coursePub);

return coursePub;

}

//创建coursePub对象

private CoursePub createCoursePub(String id){

CoursePub coursePub = new CoursePub();

coursePub.setId(id);

//基础信息

Optional<CourseBase> courseBaseOptional = courseBaseRepository.findById(id); if(courseBaseOptional == null){

CourseBase courseBase = courseBaseOptional.get();

BeanUtils.copyProperties(courseBase, coursePub);

}

//查询课程图片

Optional<CoursePic> picOptional = coursePicRepository.findById(id); if(picOptional.isPresent()){

CoursePic coursePic = picOptional.get();

BeanUtils.copyProperties(coursePic, coursePub);

}

//课程营销信息

Optional<CourseMarket> marketOptional = courseMarketRepository.findById(id); if(marketOptional.isPresent()){

CourseMarket courseMarket = marketOptional.get();

BeanUtils.copyProperties(courseMarket, coursePub);

}

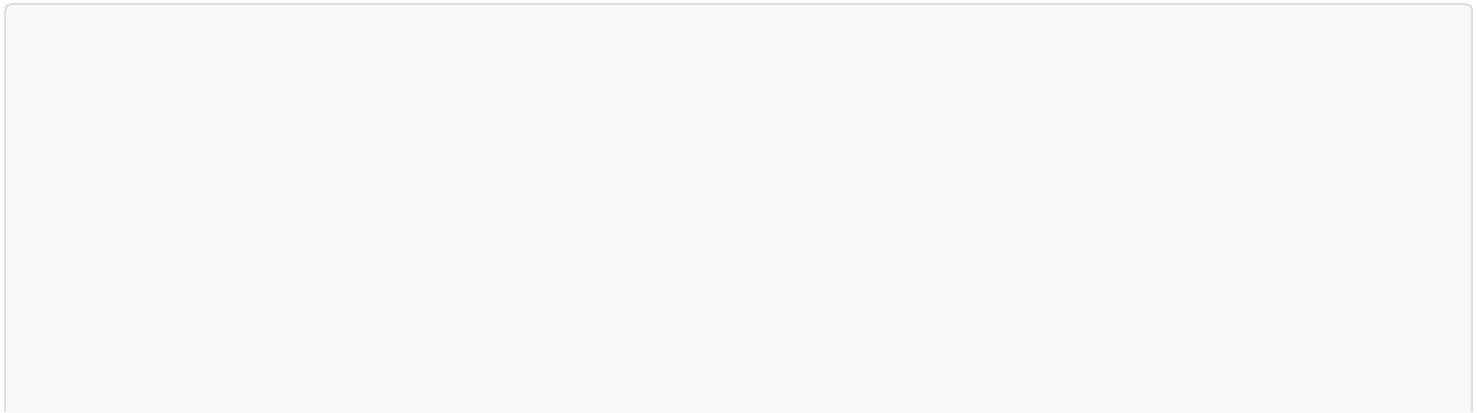
//课程计划

TeachplanNode teachplanNode = teachplanMapper.selectList(id); //将课程计划转成json

String teachplanString = JSON.toJSONString(teachplanNode); coursePub.setTeachplan(teachplanString); return coursePub;

}

修改课程发布方法，添加调用saveCoursePub方法的代码，添加部分的代码如下：



//课程发布

@Transactional

public CoursePublishResult publish(String courseId){

....

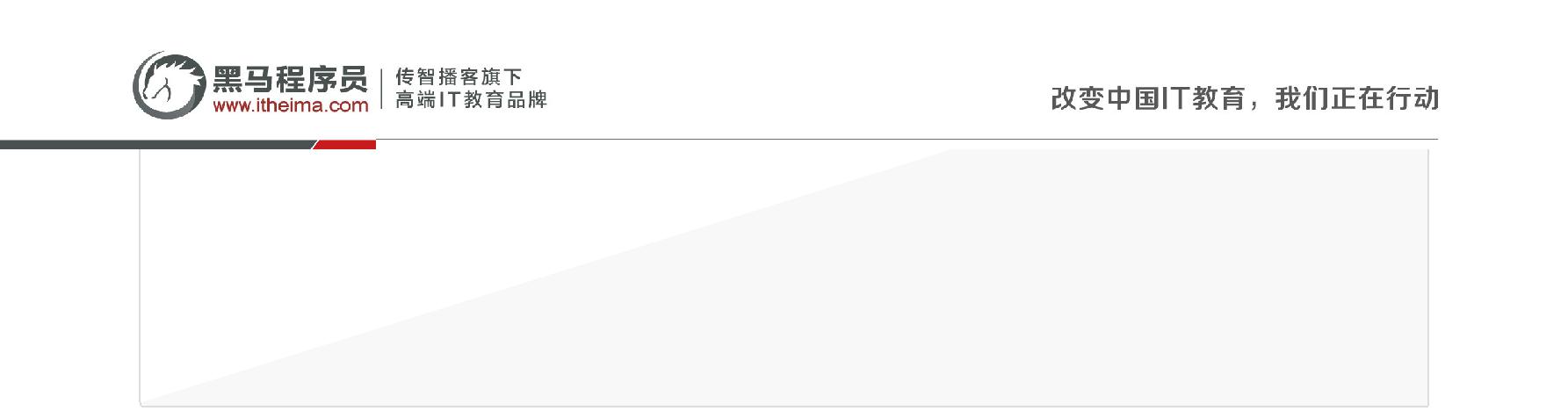
//创建课程索引

//创建课程索引信息

CoursePub coursePub = createCoursePub(courseId);

//向数据库保存课程索引信息

CoursePub newCoursePub = saveCoursePub(courseId, coursePub);

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

if(newCoursePub==null){

//创建课程索引信息失败

ExceptionCast.cast(CourseCode.COURSE\_PUBLISH\_CREATE\_INDEX\_ERROR);

}

....

}

3.3 搭建ES环境



3.3.1 ES安装

开发环境使用ES单机环境，启动ES服务端。

注意：旧的ES环境，可以删除elasticsearch-6.2.1\data\nodes目录以完全清除ES环境。

安装elasticsearch-head并启动。

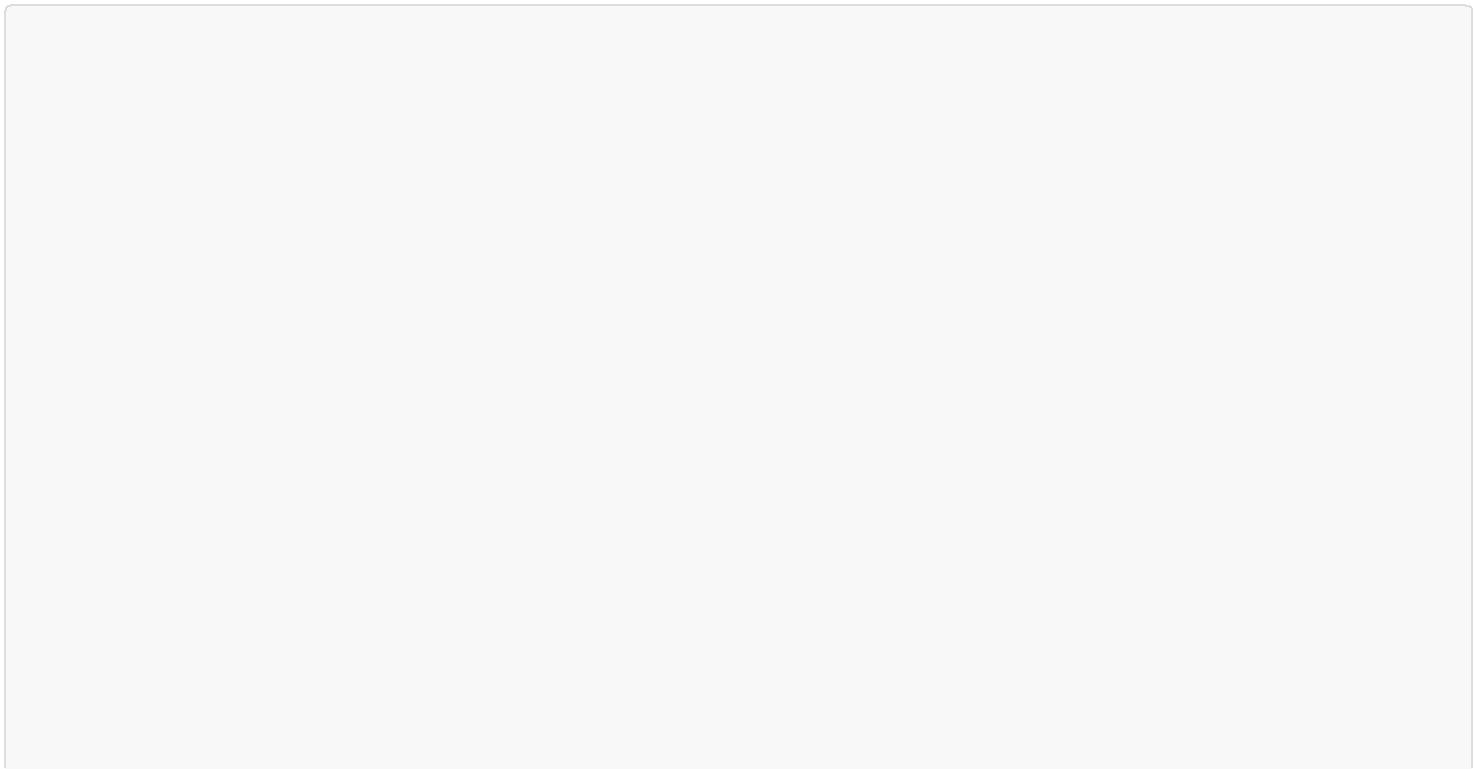
3.3.2 创建索引库

创建索引库

创建xc\_course索引库，一个分片，0个副本。

3.3.3 创建映射

Post <http://localhost:9200/xc_course/doc/_mapping>



{

"properties" : {

"description" : {

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

"search\_analyzer": "ik\_smart",

"type" : "text"

},

"grade" : {

"type" : "keyword"

},

"id" : {

"type" : "keyword"

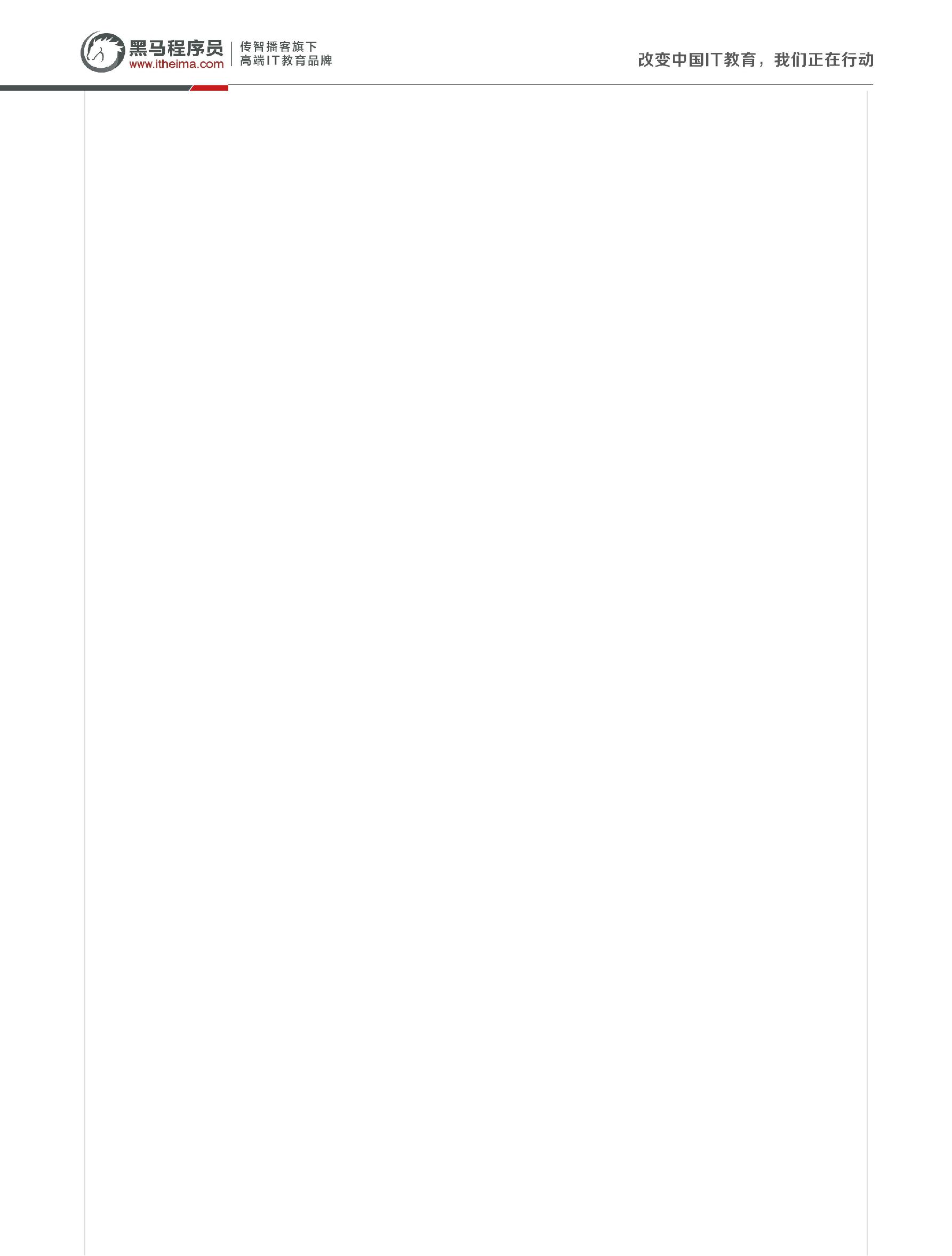
},

"mt" : {

"type" : "keyword"

},

"name" : {

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

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

"search\_analyzer": "ik\_smart",

"type" : "text"

},

"users" : {

"index" : false,

"type" : "text"

},

"charge" : {

"type" : "keyword"

},

"valid" : {

"type" : "keyword"

},

"pic" : {

"index" : false,

"type" : "keyword"

},

"qq" : {

"index" : false,

"type" : "keyword"

},

"price" : {

"type" : "float"

},

"price\_old" : {

"type" : "float"

},

"st" : {

"type" : "keyword"

},

"status" : {

"type" : "keyword"

},

"studymodel" : {

"type" : "keyword"

},

"teachmode" : {

"type" : "keyword"

},

"teachplan" : {

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

"search\_analyzer": "ik\_smart",

"type" : "text"

},

"expires" : {

"type" : "date",

"format": "yyyy‐MM‐dd HH:mm:ss"

},

"pub\_time" : {

"type" : "date",

"format": "yyyy‐MM‐dd HH:mm:ss"

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

},

"start\_time" : {

"type" : "date",

"format": "yyyy‐MM‐dd HH:mm:ss"

},

"end\_time" : {

"type" : "date",

"format": "yyyy‐MM‐dd HH:mm:ss"

}

}

}

3.4 Logstash创建索引

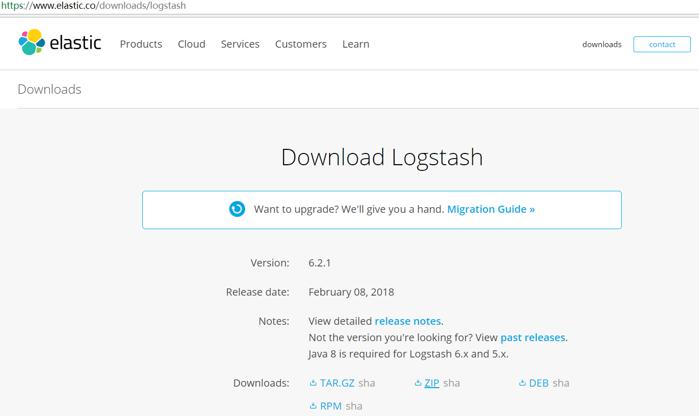


Logstash是ES下的一款开源软件，它能够同时 从多个来源采集数据、转换数据，然后将数据发送到Eleasticsearch中创建索引。

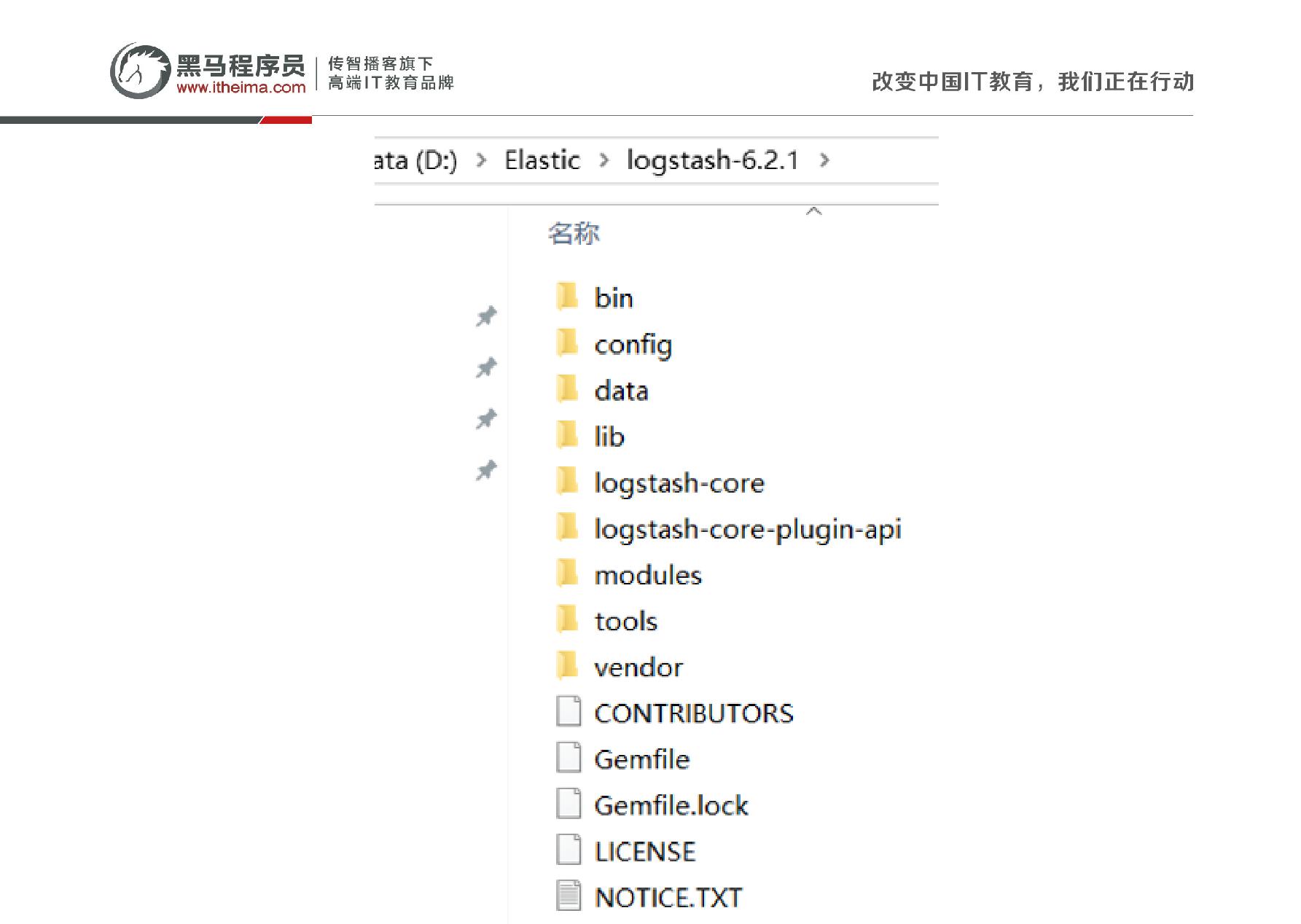
本项目使用Logstash将MySQL中的数据采用到ES索引中。

3.4.1 下载Logstash

下载Logstash6.2.1版本，和本项目使用的Elasticsearch6.2.1版本一致。



解压：

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

3.4.2 安装logstash-input-jdbc

logstash-input-jdbc 是ruby开发的，先下载ruby并安装

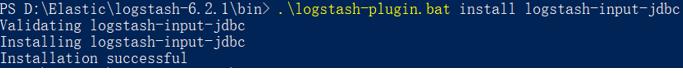
下载地址: <https://rubyinstaller.org/downloads/>

下载2.5版本即可。

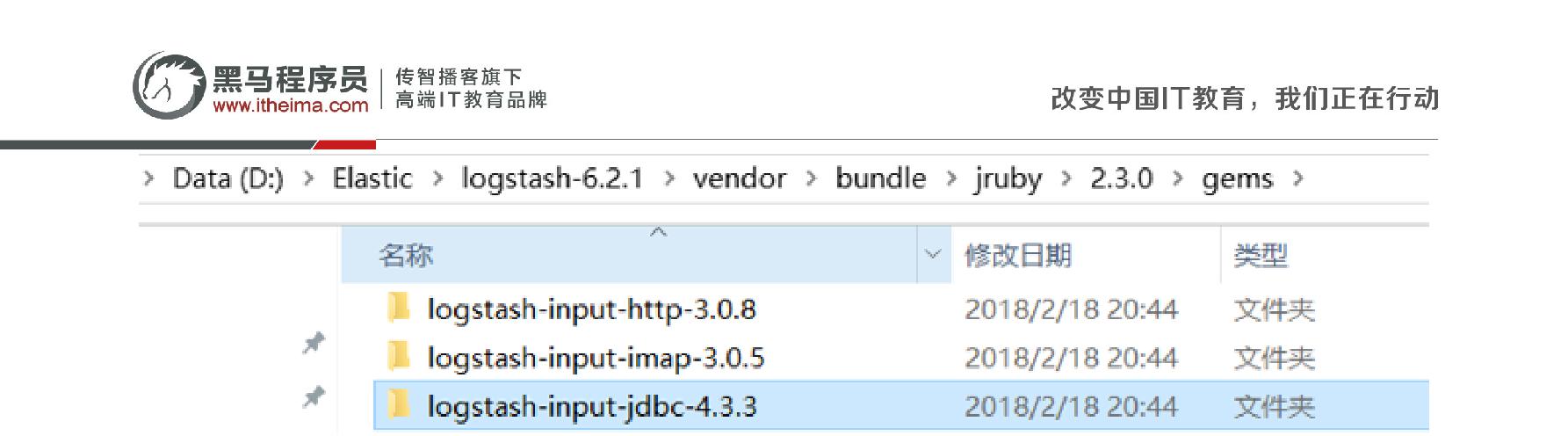
安装完成查看是否安装成功



Logstash5.x以上版本本身自带有logstash-input-jdbc，6.x版本本身不带logstash-input-jdbc插件，需要手动安装



安装成功后我们可以在logstash根目录下的以下目录查看对应的插件版本

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

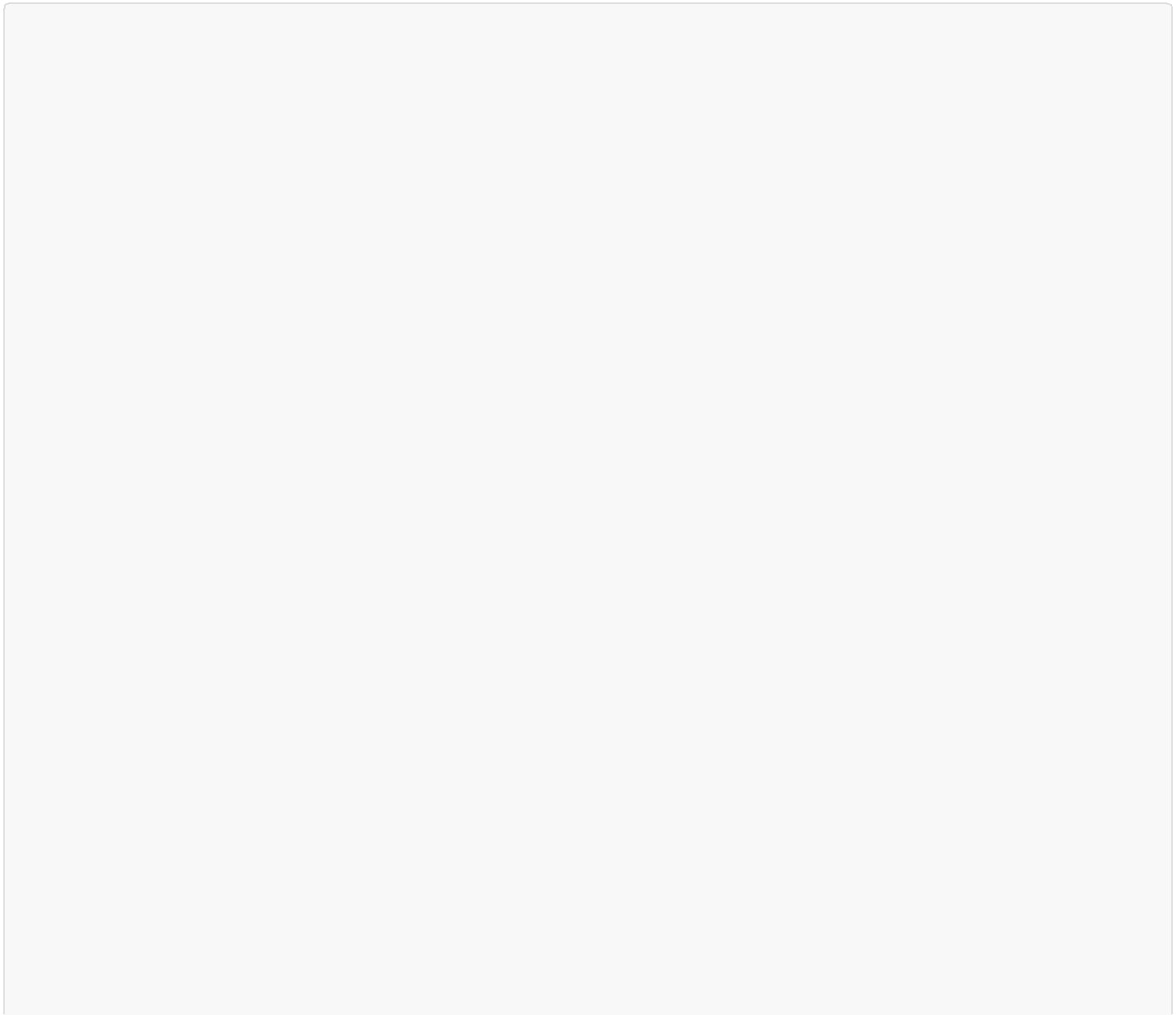
解压老师提供的logstash-6.2.1.zip,此logstash中已集成了logstash-input-jdbc插件。

3.4.3 创建模板文件

Logstash的工作是从MySQL中读取数据，向ES中创建索引，这里需要提前创建mapping的模板文件以便logstash使用。

在logstach的config目录创建xc\_course\_template.json，内容如下：

本教程的xc\_course\_template.json目录是：D:/ElasticSearch/logstash-6.2.1/config/xc\_course\_template.json



{

"mappings" : {

"doc" : {

"properties" : {

"charge" : {

"type" : "keyword"

},

"description" : {

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

"search\_analyzer" : "ik\_smart",

"type" : "text"

},

"end\_time" : {

"format" : "yyyy‐MM‐dd HH:mm:ss",

"type" : "date"

},

"expires" : {

"format" : "yyyy‐MM‐dd HH:mm:ss",

"type" : "date"

},

"grade" : {

"type" : "keyword"

},

"id" : {

"type" : "keyword"

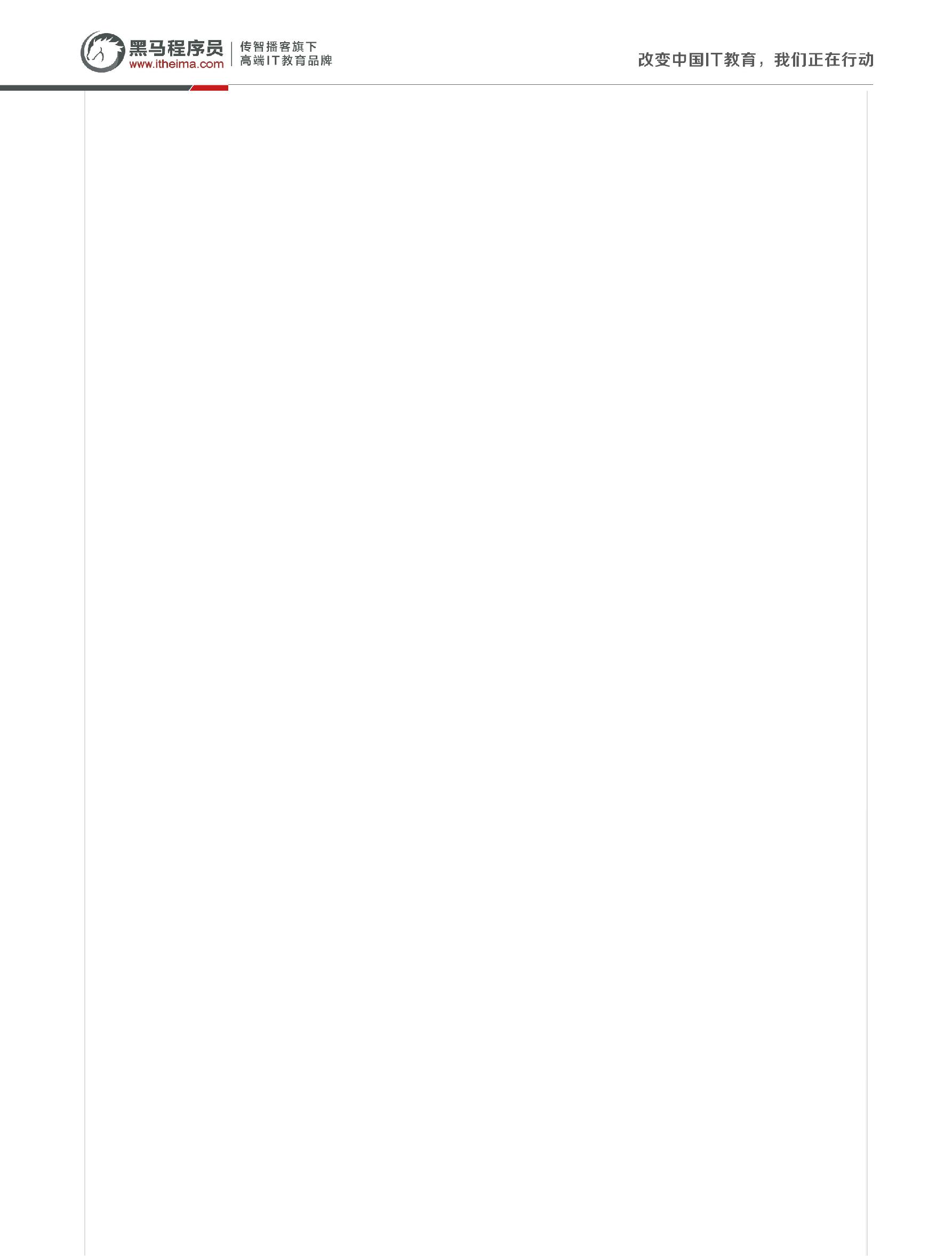
},

"mt" : {

"type" : "keyword"

},

"name" : {

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

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

"search\_analyzer" : "ik\_smart",

"type" : "text"

},

"pic" : {

"index" : false,

"type" : "keyword"

},

"price" : {

"type" : "float"

},

"price\_old" : {

"type" : "float"

},

"pub\_time" : {

"format" : "yyyy‐MM‐dd HH:mm:ss",

"type" : "date"

},

"qq" : {

"index" : false,

"type" : "keyword"

},

"st" : {

"type" : "keyword"

},

"start\_time" : {

"format" : "yyyy‐MM‐dd HH:mm:ss",

"type" : "date"

},

"status" : {

"type" : "keyword"

},

"studymodel" : {

"type" : "keyword"

},

"teachmode" : {

"type" : "keyword"

},

"teachplan" : {

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

"search\_analyzer" : "ik\_smart",

"type" : "text"

},

"users" : {

"index" : false,

"type" : "text"

},

"valid" : {

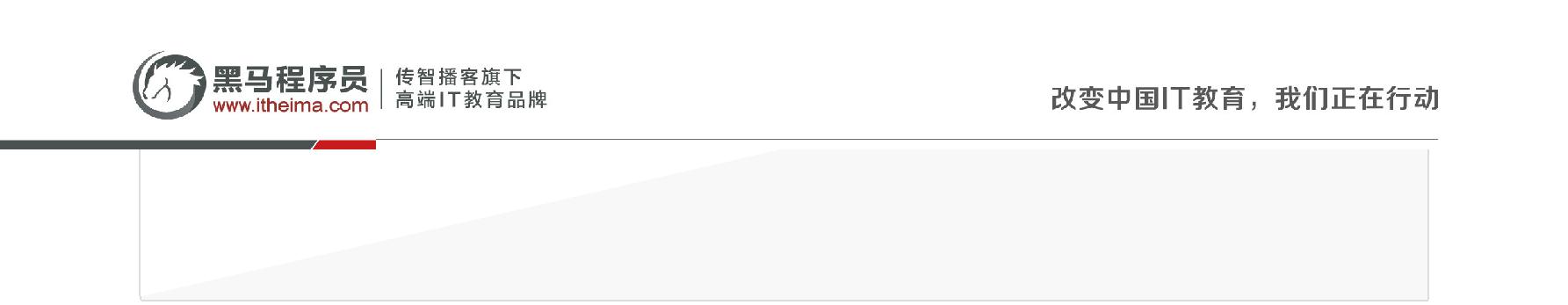
"type" : "keyword"

}

}

}

},

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

"template" : "xc\_course"

}

3.4.4 配置mysql.conf

在logstash的config目录下配置mysql.conf文件供logstash使用，logstash会根据mysql.conf文件的配置的地址从MySQL中读取数据向ES中写入索引。

参考<https://www.elastic.co/guide/en/logstash/current/plugins-inputs-jdbc.html>

配置输入数据源和输出数据源。



input {

stdin {

}

jdbc {

jdbc\_connection\_string => "jdbc:mysql://localhost:3306/xc\_course? useUnicode=true&characterEncoding=utf‐8&useSSL=true&serverTimezone=UTC"

* the user we wish to excute our statement as jdbc\_user => "root"

jdbc\_password => mysql

* the path to our downloaded jdbc driver

jdbc\_driver\_library => "F:/develop/maven/repository3/mysql/mysql‐connector‐java/5.1.41/mysql‐ connector‐java‐5.1.41.jar"

* the name of the driver class for mysql jdbc\_driver\_class => "com.mysql.jdbc.Driver" jdbc\_paging\_enabled => "true" jdbc\_page\_size => "50000"

#要执行的sql文件

#statement\_filepath => "/conf/course.sql"

statement => "select \* from course\_pub where timestamp > date\_add(:sql\_last\_value,INTERVAL 8

HOUR)"

#定时配置

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

record\_last\_run => true

last\_run\_metadata\_path => "D:/ElasticSearch/logstash‐6.2.1/config/logstash\_metadata"

}

}

output {

elasticsearch {

#ES的ip地址和端口

hosts => "localhost:9200"

#hosts => ["localhost:9200","localhost:9202","localhost:9203"] #ES索引库名称

index => "xc\_course"

document\_id => "%{id}"

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

document\_type => "doc"

template =>"D:/ElasticSearch/logstash‐6.2.1/config/xc\_course\_template.json"

template\_name =>"xc\_course"

template\_overwrite =>"true"

}

stdout {

#日志输出

codec => json\_lines

}

}

说明：

1、ES采用UTC时区问题

ES采用UTC 时区，比北京时间早8小时，所以ES读取数据时让最后更新时间加8小时 where timestamp > date\_add(:sql\_last\_value,INTERVAL 8 HOUR)

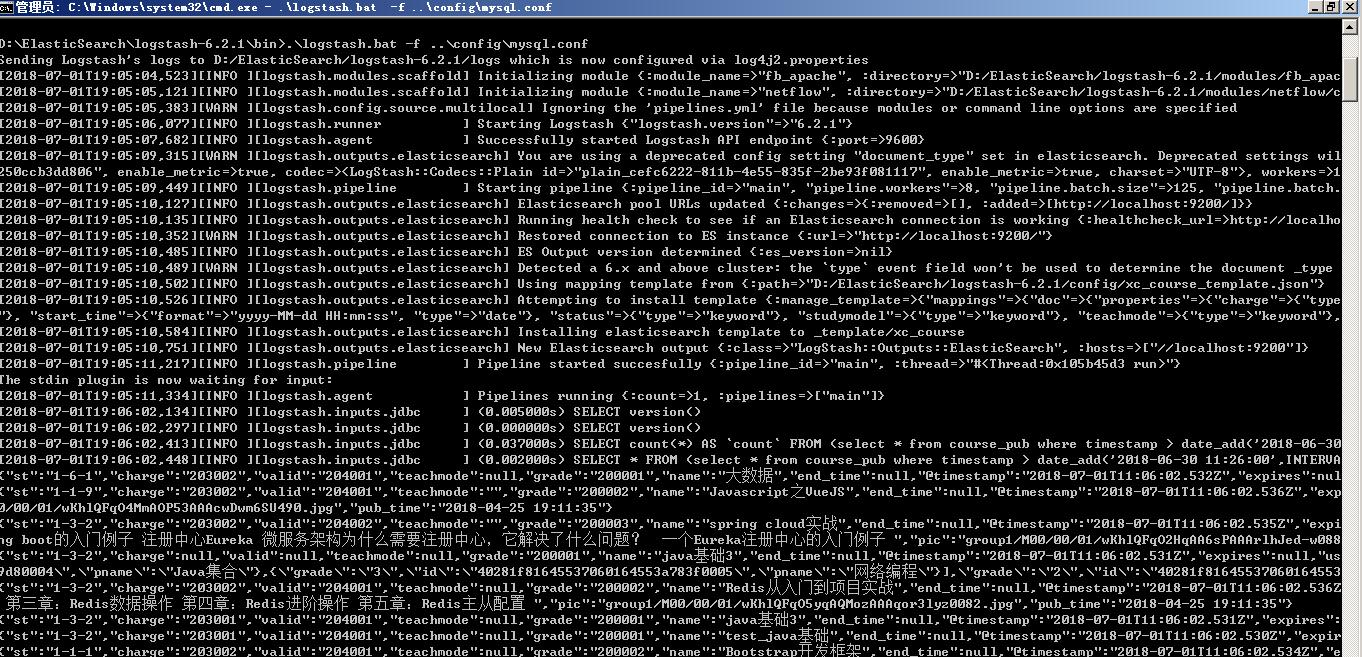
2、logstash每个执行完成会在D:/ElasticSearch/logstash-6.2.1/config/logstash\_metadata记录执行时间下次以此时间为基准进行增量同步数据到索引库。

3.4.5 测试

启动logstash.bat：



.\logstash.bat ‐f ..\config\mysql.conf



修改course\_pub中的数据，并且修改timestamp为当前时间，观察Logstash日志是否读取到要索引的数据。

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

最后用head登录ES查看索引文档内容是否修改。



* 课程搜索



4.1 需求分析



1、根据分类搜索课程信息。

2、根据关键字搜索课程信息，搜索方式为全文检索，关键字需要匹配课程的名称、 课程内容。

3、根据难度等级搜索课程。

4、搜索结点分页显示。

技术分析：

1、根据关键字搜索，采用MultiMatchQuery，搜索name、description、teachplan

2、根据分类、课程等级搜索采用过虑器实现。

3、分页查询。

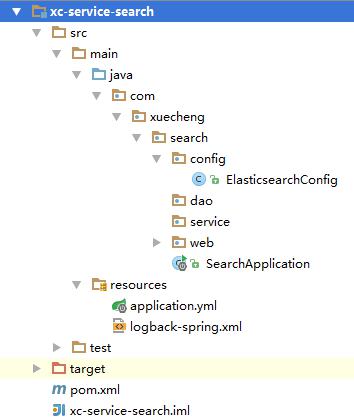
4、高亮显示。

4.2 创建搜索服务工程

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

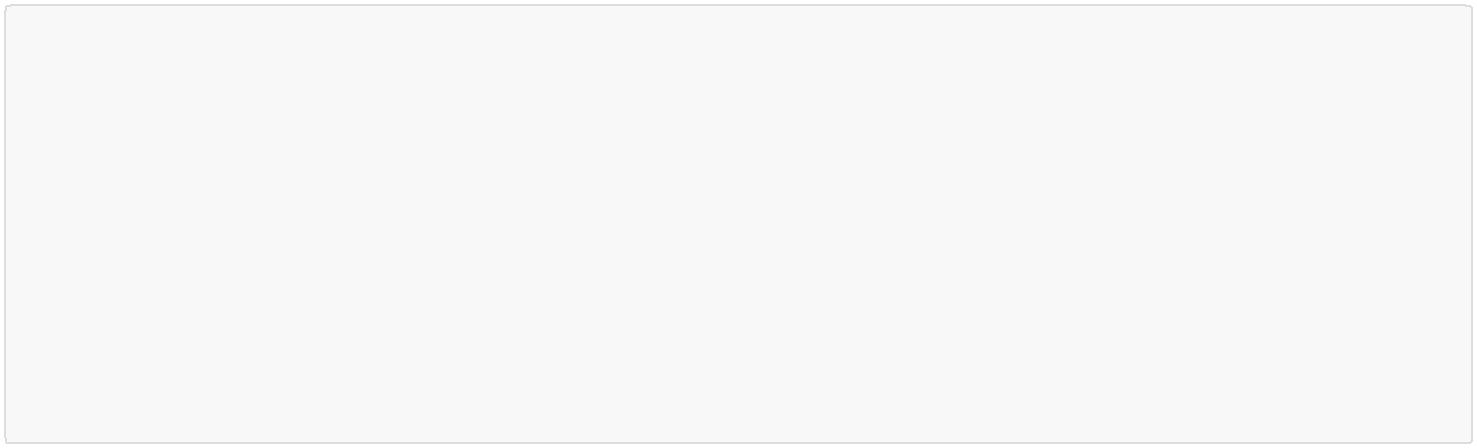


1）创建xc-service-search工程



2）配置

1、配置appliction.yml



server:

port: 40100

spring:

application:

name: xc‐search‐service

elasticsearch:

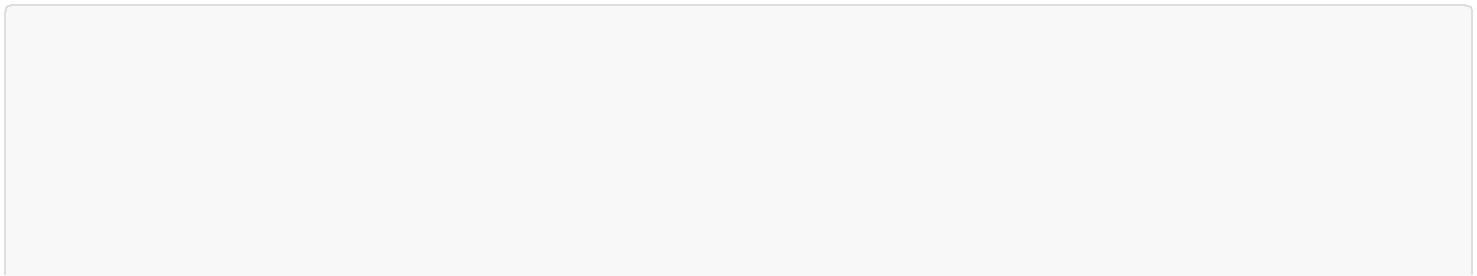
hostlist: 127.0.0.1:9200 #多个结点中间用逗号分隔

course:

index: xc\_course

type: doc

2、配置RestHighLevelClient和RestClient



package com.xuecheng.search.config;

import org.apache.http.HttpHost;

import org.elasticsearch.client.RestClient;

import org.elasticsearch.client.RestHighLevelClient;

import org.springframework.beans.factory.annotation.Value;

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

@Configuration

public class ElasticsearchConfig {

@Value("${xuecheng.elasticsearch.hostlist}")

private String hostlist;

@Bean

public RestHighLevelClient restHighLevelClient(){

//解析hostlist配置信息

String[] split = hostlist.split(",");

//创建HttpHost数组，其中存放es主机和端口的配置信息 HttpHost[] httpHostArray = new HttpHost[split.length]; for(int i=0;i<split.length;i++){

String item = split[i];

httpHostArray[i] = new HttpHost(item.split(":")[0], Integer.parseInt(item.split(":") [1]), "http");

}

//创建RestHighLevelClient客户端

return new RestHighLevelClient(RestClient.builder(httpHostArray));

}

@Bean

public RestClient restClient(){

//解析hostlist配置信息

String[] split = hostlist.split(",");

//创建HttpHost数组，其中存放es主机和端口的配置信息 HttpHost[] httpHostArray = new HttpHost[split.length]; for(int i=0;i<split.length;i++){

String item = split[i];

httpHostArray[i] = new HttpHost(item.split(":")[0], Integer.parseInt(item.split(":") [1]), "http");

}

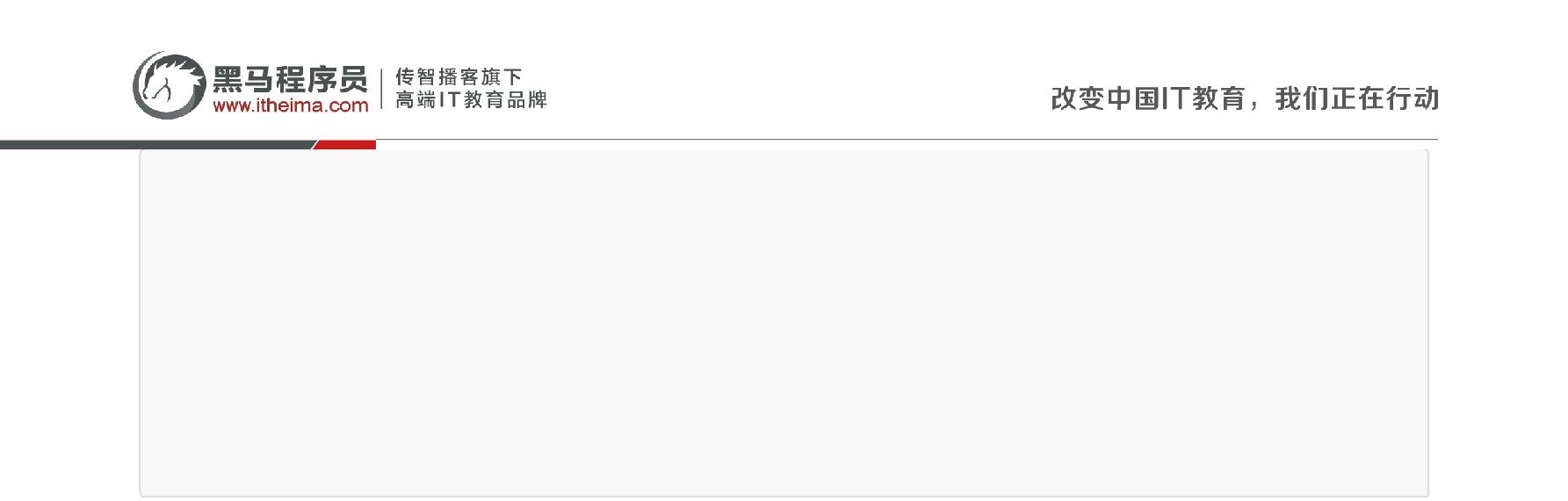
return RestClient.builder(httpHostArray).build();

}

}

4.3 API



**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

@Api(value = "课程搜索",description = "课程搜索",tags = {"课程搜索"})

public interface EsCourseControllerApi {

@ApiOperation("课程搜索")

public QueryResponseResult<CoursePub> list(int page,int size, CourseSearchParam courseSearchParam) throws IOException;

}

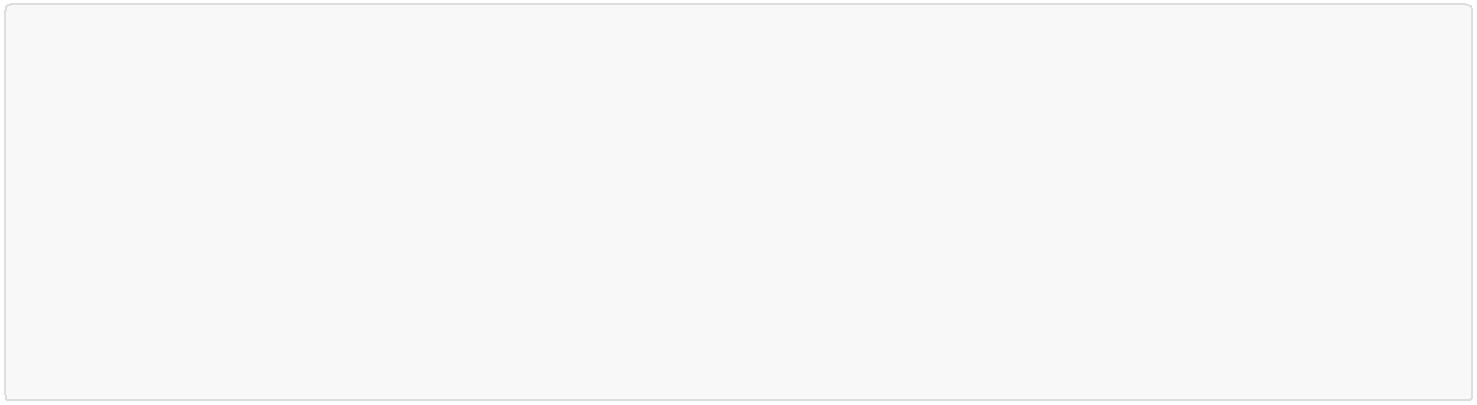
4.4 Service



Service方法代码复杂，这里分三步完成。

4.4.1 按关键字搜索

1）在appliction.yml中配置source\_field



elasticsearch:

hostlist: 127.0.0.1:9200 #多个结点中间用逗号分隔

course:

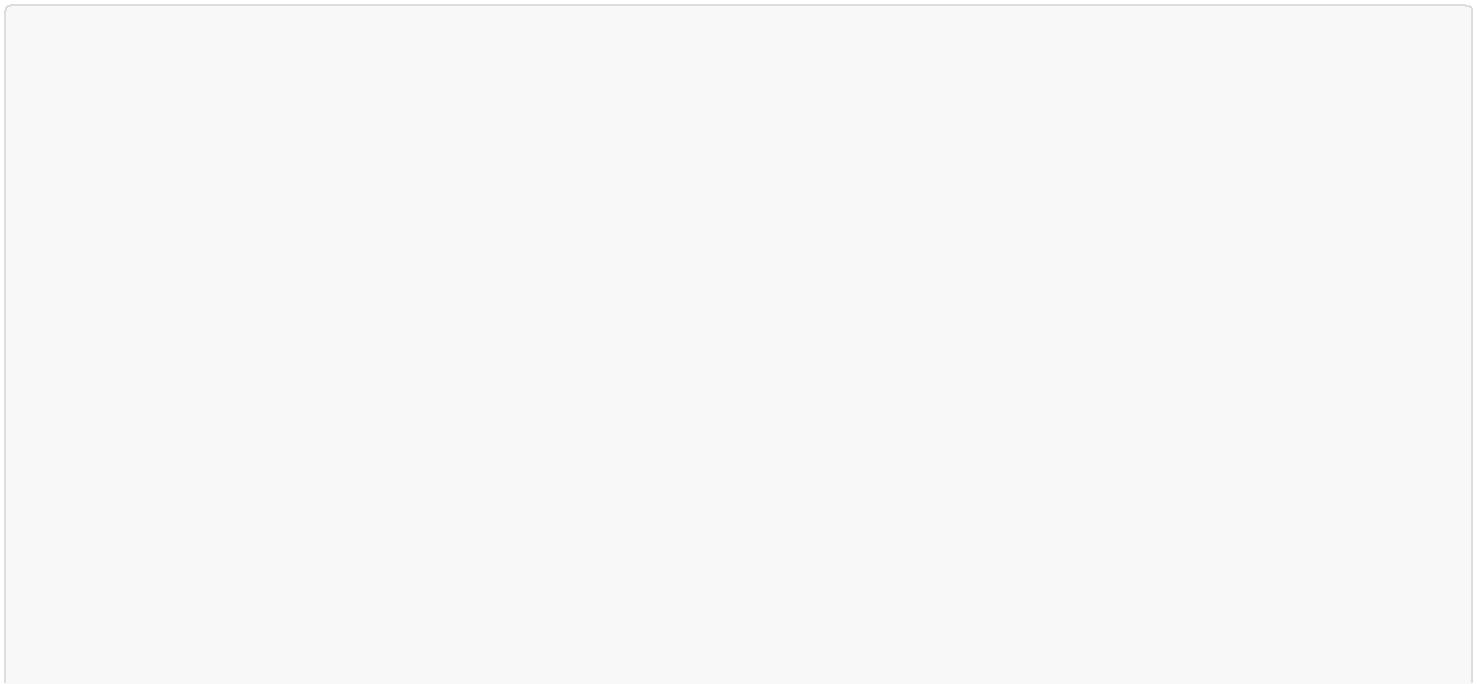
index: xc\_course

type: doc

source\_field:

id,name,grade,mt,st,charge,valid,pic,qq,price,price\_old,status,studymodel,teachmode,expires,pub\_ time,start\_time,end\_time

2）service完整代码如下



@Service

public class EsCourseService {

private static final Logger LOGGER = LoggerFactory.getLogger(EsCourseService.class);

@Value("${xuecheng.elasticsearch.course.index}")

private String es\_index;

@Value("${xuecheng.elasticsearch.course.type}")

private String es\_type;

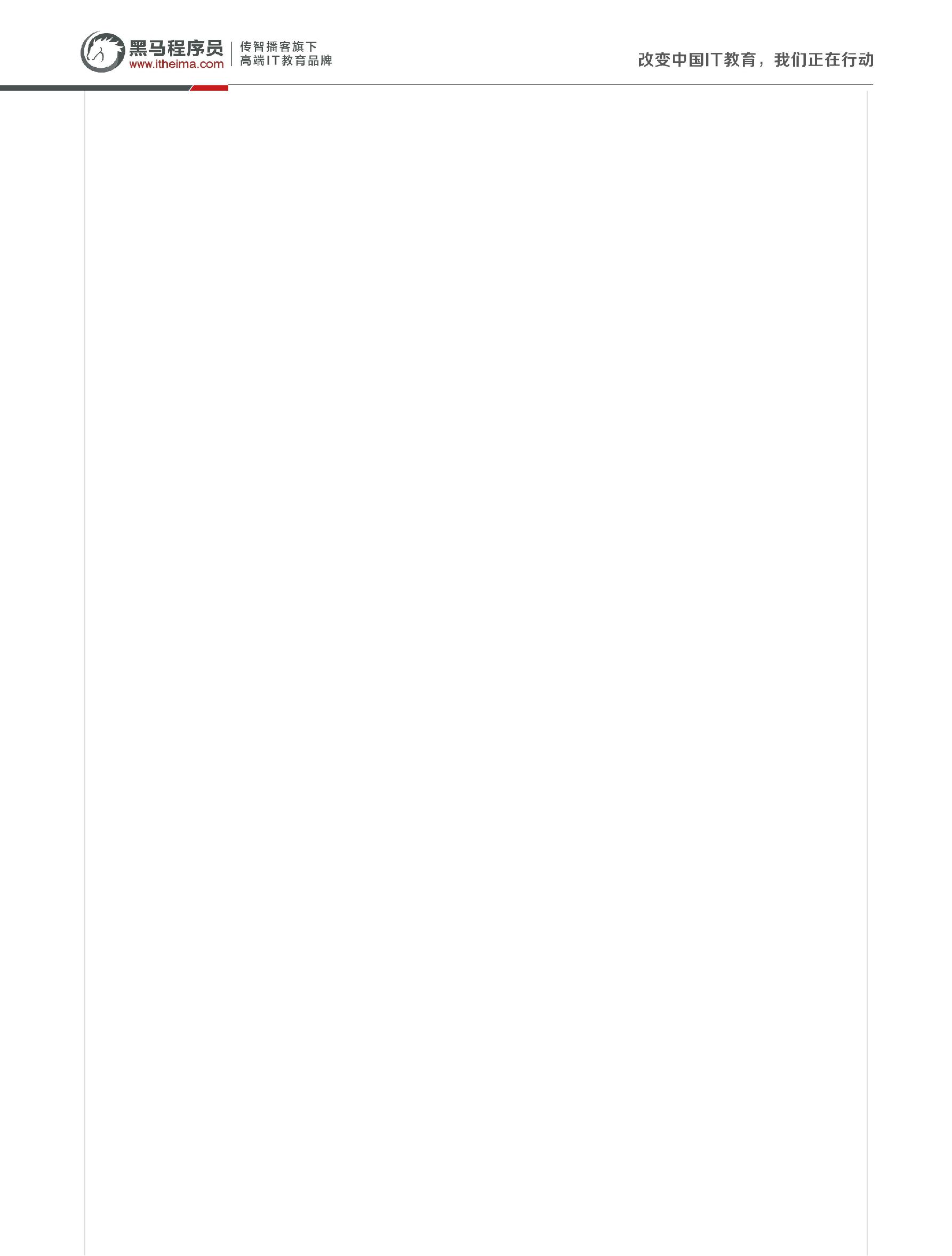
@Value("${xuecheng.elasticsearch.course.source\_field}")

private String source\_field;

@Autowired

RestHighLevelClient restHighLevelClient;

public QueryResponseResult<CoursePub> list(int page,int size,CourseSearchParam

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

courseSearchParam) {

//设置索引

SearchRequest searchRequest = new SearchRequest(es\_index); //设置类型

searchRequest.types(es\_type);

SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder(); BoolQueryBuilder boolQueryBuilder = QueryBuilders.boolQuery(); //source源字段过虑

String[] source\_fields = source\_field.split(",");

searchSourceBuilder.fetchSource(source\_fields, new String[]{});

//关键字

if(StringUtils.isNotEmpty(courseSearchParam.getKeyword())){

//匹配关键字

MultiMatchQueryBuilder multiMatchQueryBuilder = QueryBuilders.multiMatchQuery(courseSearchParam.getKeyword(), "name", "teachplan","description");

//设置匹配占比

multiMatchQueryBuilder.minimumShouldMatch("70%");

//提升另个字段的Boost值

multiMatchQueryBuilder.field("name",10);

boolQueryBuilder.must(multiMatchQueryBuilder);

}

//布尔查询

searchSourceBuilder.query(boolQueryBuilder);

//请求搜索

searchRequest.source(searchSourceBuilder);

SearchResponse searchResponse = null;

try {

searchResponse = restHighLevelClient.search(searchRequest);

} catch (IOException e) { e.printStackTrace();

LOGGER.error("xuecheng search error..{}",e.getMessage());

return new QueryResponseResult(CommonCode.SUCCESS,new QueryResult<CoursePub>());

}

//结果集处理

SearchHits hits = searchResponse.getHits();

SearchHit[] searchHits = hits.getHits();

//记录总数

long totalHits = hits.getTotalHits();

//数据列表

List<CoursePub> list = new ArrayList<>();

for (SearchHit hit : searchHits) {

CoursePub coursePub = new CoursePub();

//取出source

Map<String, Object> sourceAsMap = hit.getSourceAsMap();

//取出名称

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

String name = (String) sourceAsMap.get("name");

coursePub.setName(name);

//图片

String pic = (String) sourceAsMap.get("pic");

coursePub.setPic(pic);

//价格

Float price = null;

try {

if(sourceAsMap.get("price")!=null ){

price = Float.parseFloat((String) sourceAsMap.get("price"));

}

} catch (Exception e) {

e.printStackTrace();

}

coursePub.setPrice(price);

Float price\_old = null;

try {

if(sourceAsMap.get("price\_old")!=null ){

price\_old = Float.parseFloat((String) sourceAsMap.get("price\_old"));

}

} catch (Exception e) { e.printStackTrace();

}

coursePub.setPrice\_old(price\_old);

list.add(coursePub);

}

QueryResult<CoursePub> queryResult = new QueryResult<>();

queryResult.setList(list);

queryResult.setTotal(totalHits);

QueryResponseResult<CoursePub> coursePubQueryResponseResult = new

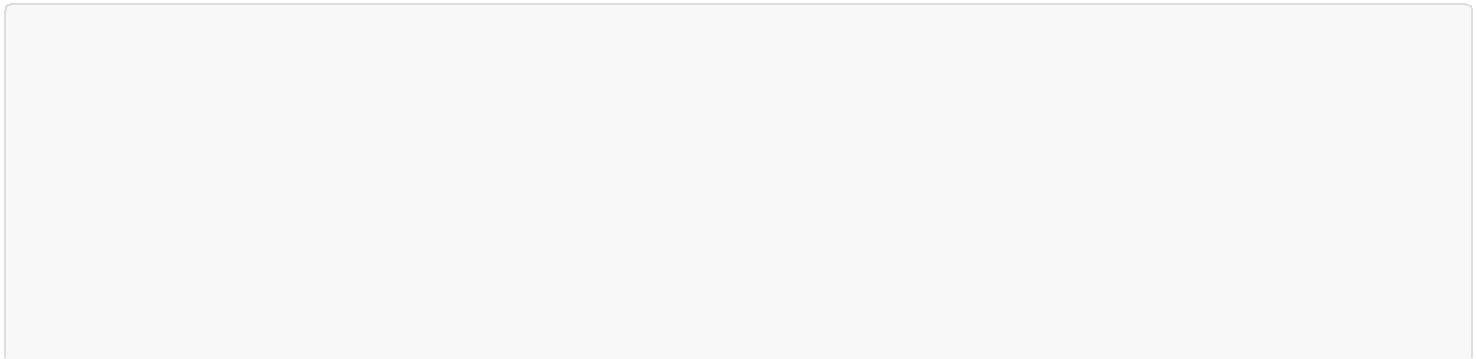
QueryResponseResult<CoursePub>(CommonCode.SUCCESS,queryResult); return coursePubQueryResponseResult;

}

}

4.4.2 按分类和难度等级搜索

按分类和难度等级搜索代码如下：



@Service

public class EsCourseService {

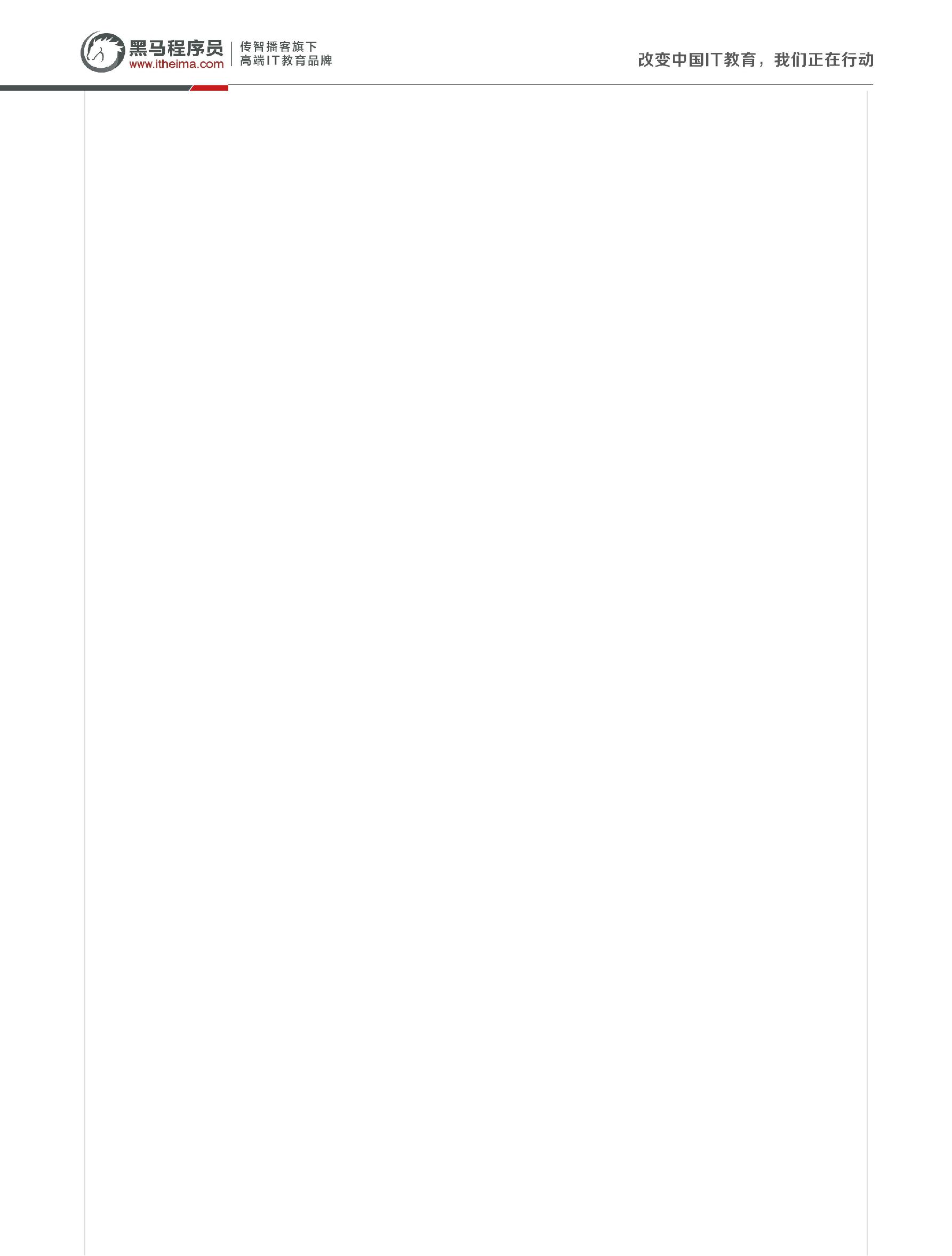
private static final Logger LOGGER = LoggerFactory.getLogger(EsCourseService.class);

@Value("${xuecheng.elasticsearch.course.index}")

private String es\_index;

@Value("${xuecheng.elasticsearch.course.type}")

private String es\_type;

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

@Value("${xuecheng.elasticsearch.course.source\_field}")

private String source\_field;

@Autowired

RestHighLevelClient restHighLevelClient;

public QueryResponseResult<CoursePub> list(int page,int size,CourseSearchParam courseSearchParam) {

//设置索引

SearchRequest searchRequest = new SearchRequest(es\_index); //设置类型

searchRequest.types(es\_type);

SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder(); BoolQueryBuilder boolQueryBuilder = QueryBuilders.boolQuery(); //source源字段过虑

String[] source\_fields = source\_field.split(",");

searchSourceBuilder.fetchSource(source\_fields, new String[]{});

//关键字

if(StringUtils.isNotEmpty(courseSearchParam.getKeyword())){

//匹配关键字

MultiMatchQueryBuilder multiMatchQueryBuilder = QueryBuilders.multiMatchQuery(courseSearchParam.getKeyword(), "name", "teachplan","description");

//设置匹配占比

multiMatchQueryBuilder.minimumShouldMatch("70%");

//提升另个字段的Boost值

multiMatchQueryBuilder.field("name",10);

boolQueryBuilder.must(multiMatchQueryBuilder);

}

//过虑

if(StringUtils.isNotEmpty(courseSearchParam.getMt())){

boolQueryBuilder.filter(QueryBuilders.termQuery("mt",courseSearchParam.getMt()));

}

if(StringUtils.isNotEmpty(courseSearchParam.getSt())){

boolQueryBuilder.filter(QueryBuilders.termQuery("st",courseSearchParam.getSt()));

}

if(StringUtils.isNotEmpty(courseSearchParam.getGrade())){

boolQueryBuilder.filter(QueryBuilders.termQuery("grade",courseSearchParam.getGrade()));

}

//布尔查询

searchSourceBuilder.query(boolQueryBuilder);

//请求搜索

searchRequest.source(searchSourceBuilder);

SearchResponse searchResponse = null;

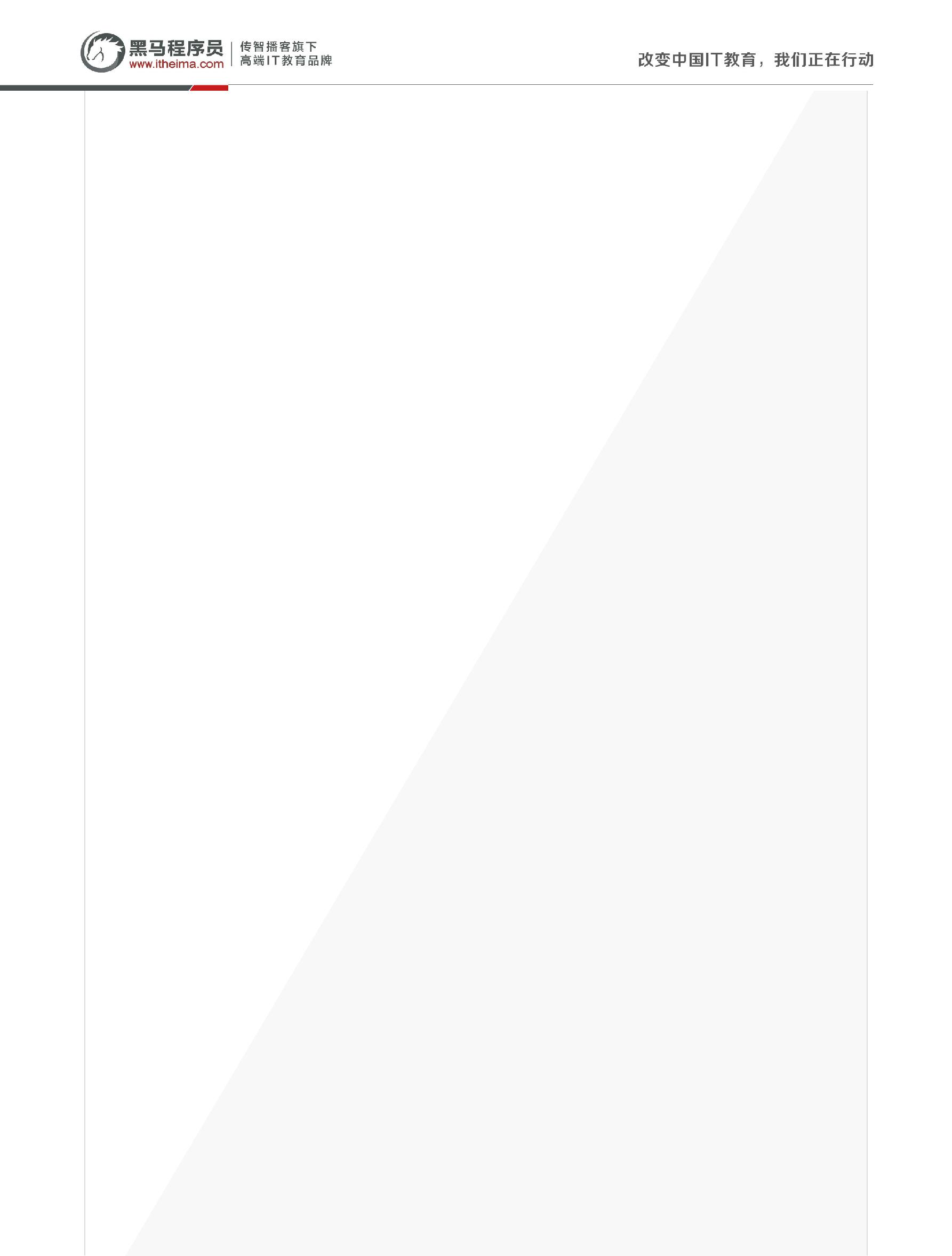
try {

searchResponse = restHighLevelClient.search(searchRequest);

} catch (IOException e) { e.printStackTrace();

LOGGER.error("xuecheng search error..{}",e.getMessage());

return new QueryResponseResult(CommonCode.SUCCESS,new QueryResult<CoursePub>());

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

}

//结果集处理

SearchHits hits = searchResponse.getHits();

SearchHit[] searchHits = hits.getHits();

//记录总数

long totalHits = hits.getTotalHits();

//数据列表

List<CoursePub> list = new ArrayList<>();

for (SearchHit hit : searchHits) {

CoursePub coursePub = new CoursePub();

//取出source

Map<String, Object> sourceAsMap = hit.getSourceAsMap();

//取出名称

String name = (String) sourceAsMap.get("name");

coursePub.setName(name);

//图片

String pic = (String) sourceAsMap.get("pic");

coursePub.setPic(pic);

//价格

Float price = null;

try {

if(sourceAsMap.get("price")!=null ){

price = Float.parseFloat((String) sourceAsMap.get("price"));

}

} catch (Exception e) { e.printStackTrace();

}

coursePub.setPrice(price);

Float price\_old = null;

try {

if(sourceAsMap.get("price\_old")!=null ){

price\_old = Float.parseFloat((String) sourceAsMap.get("price\_old"));

}

} catch (Exception e) {

e.printStackTrace();

}

coursePub.setPrice\_old(price\_old);

list.add(coursePub);

}

QueryResult<CoursePub> queryResult = new QueryResult<>();

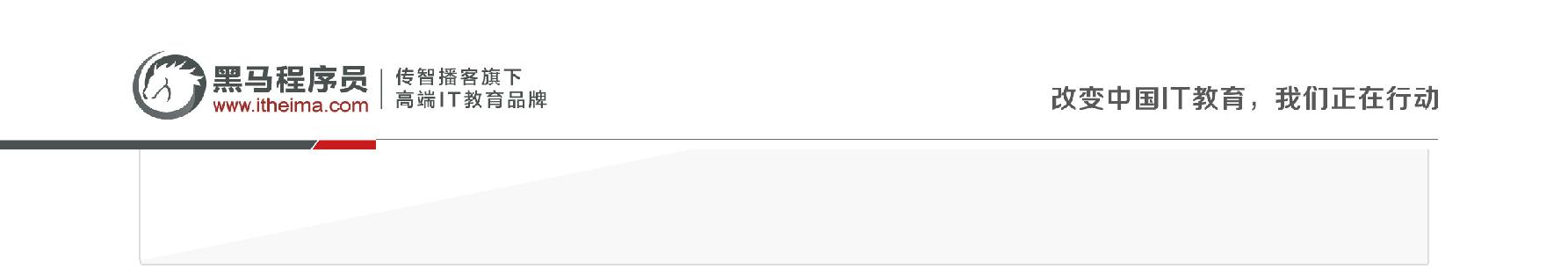
queryResult.setList(list);

queryResult.setTotal(totalHits);

QueryResponseResult<CoursePub> coursePubQueryResponseResult = new

QueryResponseResult<CoursePub>(CommonCode.SUCCESS,queryResult);

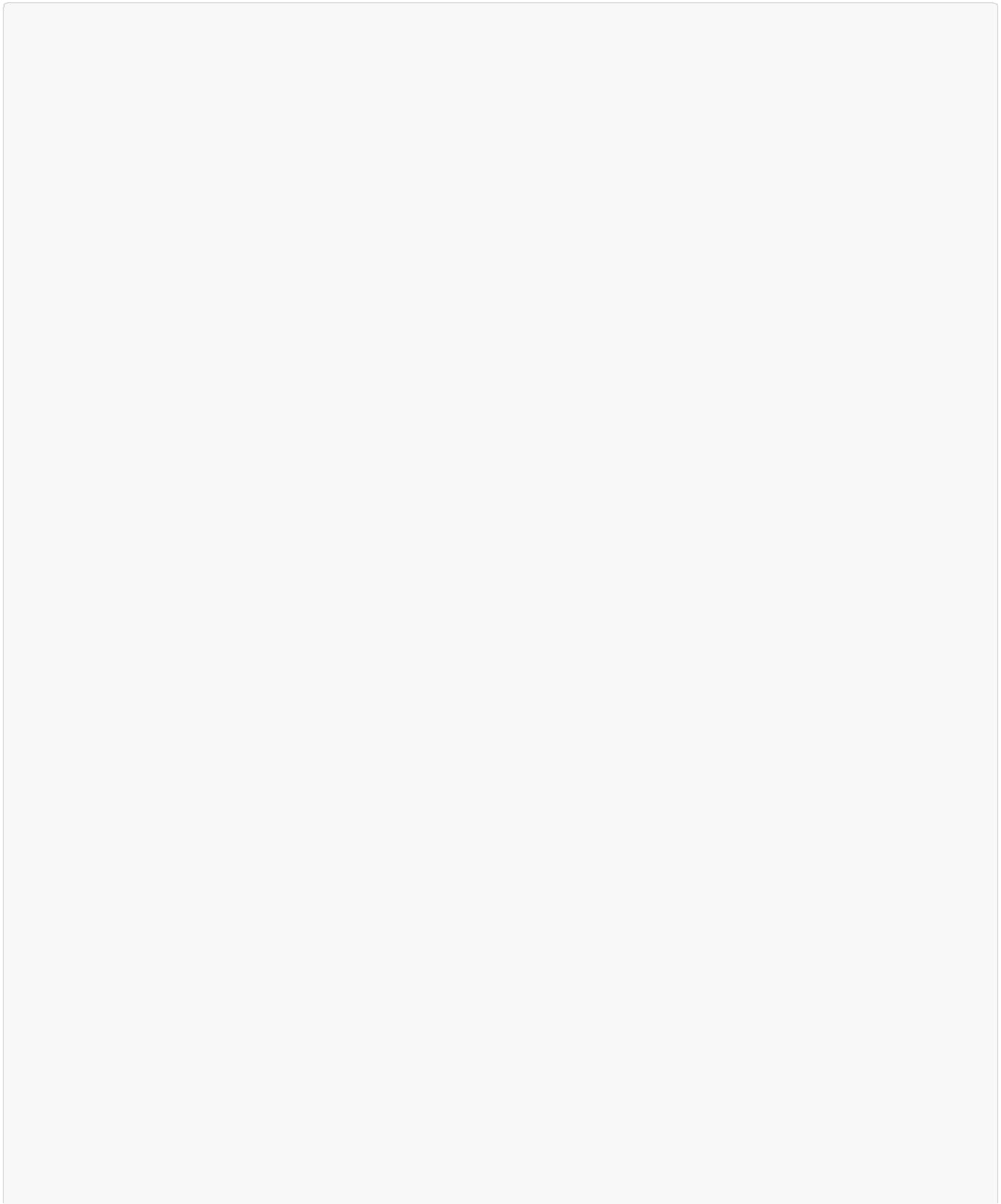
return coursePubQueryResponseResult;

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

}

}

4.4.3 分页与高亮



@Service

public class EsCourseService {

private static final Logger LOGGER = LoggerFactory.getLogger(EsCourseService.class);

@Value("${xuecheng.elasticsearch.course.index}")

private String es\_index;

@Value("${xuecheng.elasticsearch.course.type}")

private String es\_type;

@Value("${xuecheng.elasticsearch.course.source\_field}")

private String source\_field;

@Autowired

RestHighLevelClient restHighLevelClient;

public QueryResponseResult<CoursePub> list(int page,int size,CourseSearchParam courseSearchParam) {

//设置索引

SearchRequest searchRequest = new SearchRequest(es\_index); //设置类型

searchRequest.types(es\_type);

SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder(); BoolQueryBuilder boolQueryBuilder = QueryBuilders.boolQuery(); //source源字段过虑

String[] source\_fields = source\_field.split(",");

searchSourceBuilder.fetchSource(source\_fields, new String[]{});

//关键字

if(StringUtils.isNotEmpty(courseSearchParam.getKeyword())){

//匹配关键字

MultiMatchQueryBuilder multiMatchQueryBuilder = QueryBuilders.multiMatchQuery(courseSearchParam.getKeyword(), "name", "teachplan","description");

//设置匹配占比

multiMatchQueryBuilder.minimumShouldMatch("70%");

//提升另个字段的Boost值

multiMatchQueryBuilder.field("name",10);

boolQueryBuilder.must(multiMatchQueryBuilder);

}

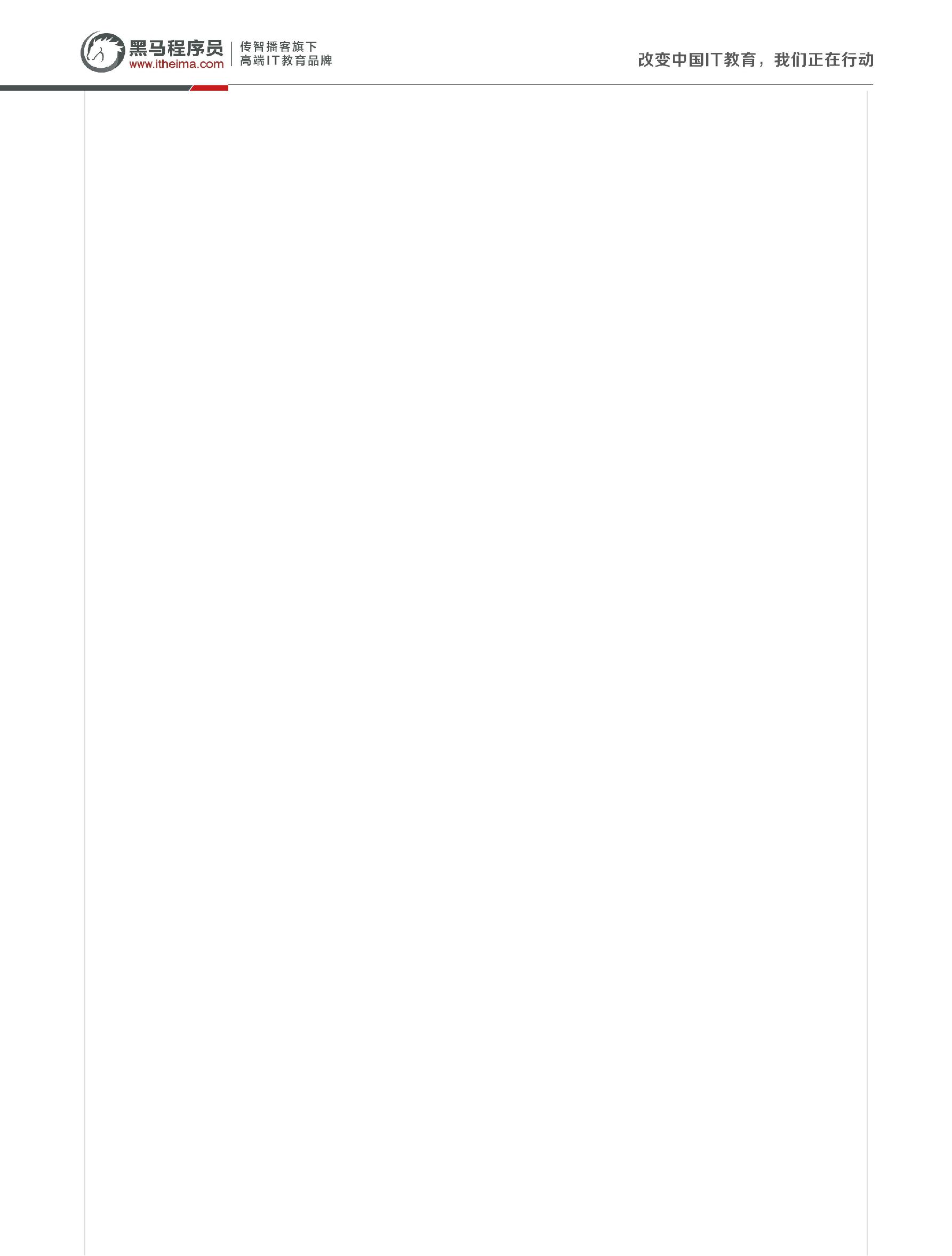
//过虑

if(StringUtils.isNotEmpty(courseSearchParam.getMt())){

boolQueryBuilder.filter(QueryBuilders.termQuery("mt",courseSearchParam.getMt()));

}

if(StringUtils.isNotEmpty(courseSearchParam.getSt())){

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

boolQueryBuilder.filter(QueryBuilders.termQuery("st",courseSearchParam.getSt()));

}

if(StringUtils.isNotEmpty(courseSearchParam.getGrade())){

boolQueryBuilder.filter(QueryBuilders.termQuery("grade",courseSearchParam.getGrade()));

}

//分页

if(page<=0){

page = 1;

}

if(size<=0){

size = 20;

}

int start = (page‐1)\*size;

searchSourceBuilder.from(start);

searchSourceBuilder.size(size);

//布尔查询

searchSourceBuilder.query(boolQueryBuilder);

//高亮设置

HighlightBuilder highlightBuilder = new HighlightBuilder(); highlightBuilder.preTags("<font class='eslight'>"); highlightBuilder.postTags("</font>");

//设置高亮字段

highlightBuilder.fields().add(new HighlightBuilder.Field("name")); searchSourceBuilder.highlighter(highlightBuilder);

//请求搜索

searchRequest.source(searchSourceBuilder);

SearchResponse searchResponse = null;

try {

searchResponse = restHighLevelClient.search(searchRequest);

} catch (IOException e) { e.printStackTrace();

LOGGER.error("xuecheng search error..{}",e.getMessage());

return new QueryResponseResult(CommonCode.SUCCESS,new QueryResult<CoursePub>());

}

//结果集处理

SearchHits hits = searchResponse.getHits();

SearchHit[] searchHits = hits.getHits();

//记录总数

long totalHits = hits.getTotalHits();

//数据列表

List<CoursePub> list = new ArrayList<>();

for (SearchHit hit : searchHits) {

CoursePub coursePub = new CoursePub();

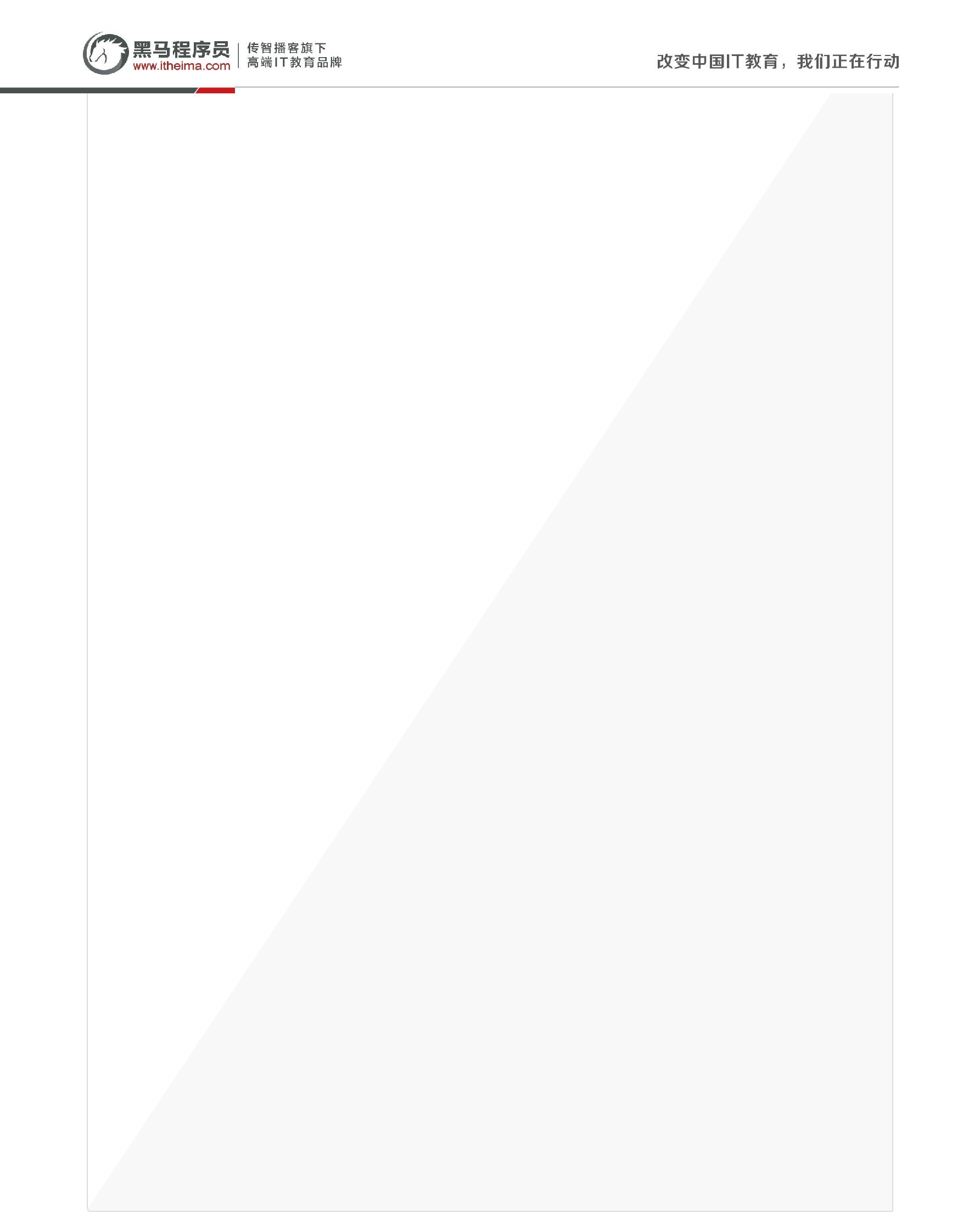
//取出source

Map<String, Object> sourceAsMap = hit.getSourceAsMap();

//取出名称

String name = (String) sourceAsMap.get("name");

//取出高亮字段内容

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

Map<String, HighlightField> highlightFields = hit.getHighlightFields(); if(highlightFields!=null){

HighlightField nameField = highlightFields.get("name"); if(nameField!=null){

Text[] fragments = nameField.getFragments();

StringBuffer stringBuffer = new StringBuffer();

for (Text str : fragments) {

stringBuffer.append(str.string());

}

name = stringBuffer.toString();

}

}

coursePub.setName(name);

//图片

String pic = (String) sourceAsMap.get("pic");

coursePub.setPic(pic);

//价格

Float price = null;

try {

if(sourceAsMap.get("price")!=null ){

price = Float.parseFloat((String) sourceAsMap.get("price"));

}

} catch (Exception e) { e.printStackTrace();

}

coursePub.setPrice(price);

Float price\_old = null;

try {

if(sourceAsMap.get("price\_old")!=null ){

price\_old = Float.parseFloat((String) sourceAsMap.get("price\_old"));

}

} catch (Exception e) { e.printStackTrace();

}

coursePub.setPrice\_old(price\_old);

list.add(coursePub);

}

QueryResult<CoursePub> queryResult = new QueryResult<>();

queryResult.setList(list);

queryResult.setTotal(totalHits);

QueryResponseResult<CoursePub> coursePubQueryResponseResult = new

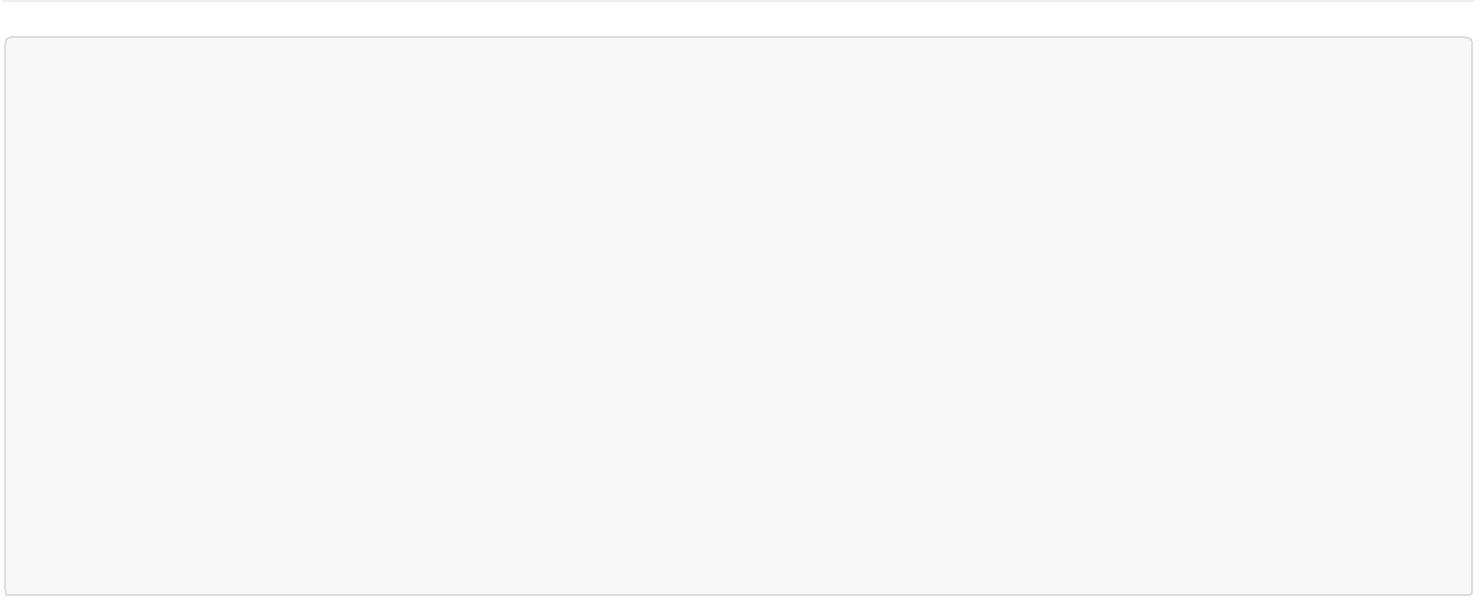
QueryResponseResult<CoursePub>(CommonCode.SUCCESS,queryResult); return coursePubQueryResponseResult;

}

}

**北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090**

4.5 Controller



@RestController

@RequestMapping("/search/course")

public class EsCourseController implements EsCourseControllerApi { @Autowired

EsCourseService esCourseService;

@Override

@GetMapping(value="/list/{page}/{size}")

public QueryResponseResult<CoursePub> list(@PathVariable("page") int page, @PathVariable("size") int size, CourseSearchParam courseSearchParam) throws IOException {

return esCourseService.list(page,size,courseSearchParam);

}

}

4.5 测试



使用postman测试/search/course

