



链滴

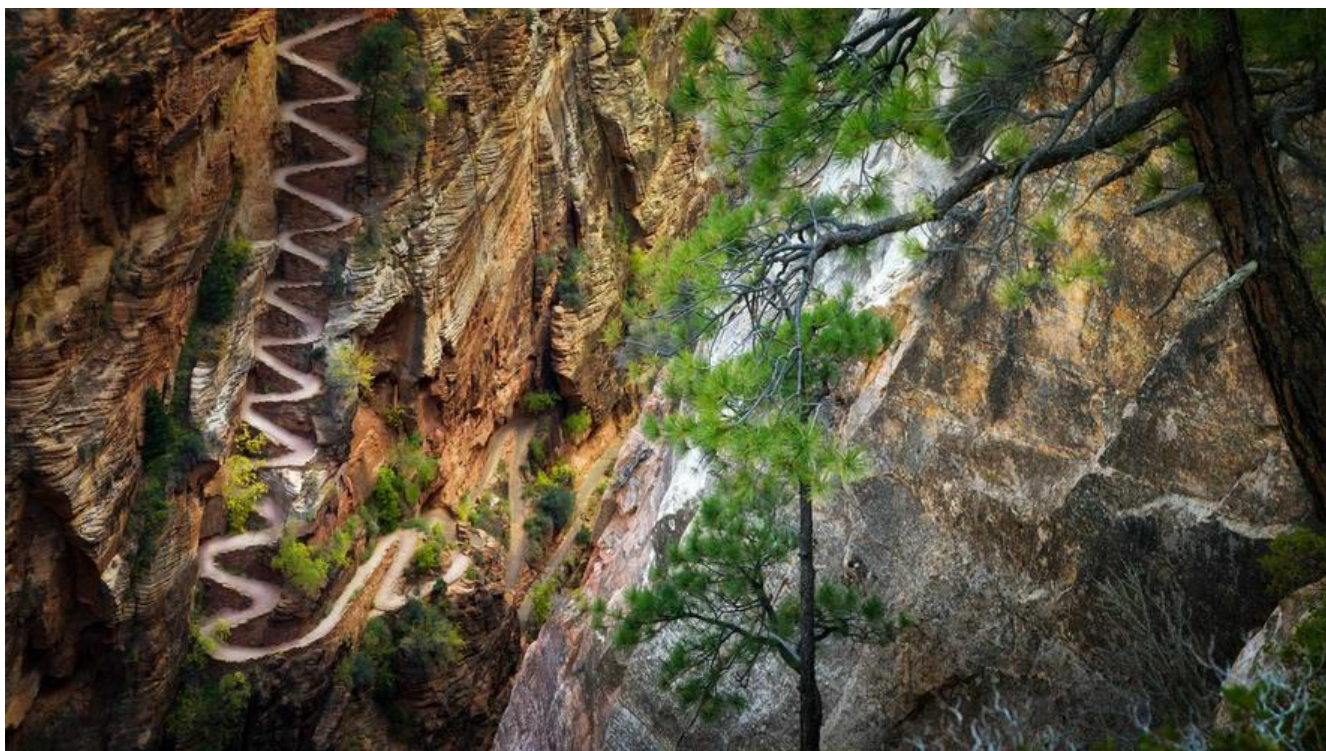
ES 集成 SpringBoot (基本 API 使用)

作者: [yscxy](#)

原文链接: <https://ld246.com/article/1605184679511>

来源网站: [链滴](#)

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



注入

```
@Autowired
private RestHighLevelClient restHighLevelClient;
```

测试创建索引

```
@Test
void testCreate() throws IOException {
    //创建索引请求
    CreateIndexRequest request = new CreateIndexRequest("yscxy_index");
    //客户端执行请求
    CreateIndexResponse createIndexResponse = restHighLevelClient.indices().create(request,
RequestOptions.DEFAULT);
    //打印返回信息!
    System.out.println(createIndexResponse);
}
```

测试获取索引

```
@Test
void testExistIndex() throws IOException {
    GetIndexRequest request = new GetIndexRequest("yscxy_index");
    boolean exists = restHighLevelClient.indices().exists(request, RequestOptions.DEFAULT);
    System.out.println(exists);
}
```

测试删除索引

```

@Test
void testDeleteIndex() throws IOException {
    DeleteIndexRequest request = new DeleteIndexRequest("yscxy_index");
    AcknowledgedResponse delete = restHighLevelClient.indices().delete(request, RequestOptions.DEFAULT);
    System.out.println(delete.isAcknowledged());
}

```

测试添加文档

```

@Test
void testAddDocument() throws IOException {
    //创建对象
    User user = new User("野生程序员", 3);
    //创建请求
    IndexRequest request = new IndexRequest("yscxy_index");
    //规则 put/yscxy_index/_doc/1
    request.id("1");
    request.timeout(TimeValue.timeValueSeconds(1));
    //或者这样设置
    request.timeout("1s");
    //将我们的数据放入请求 json
    IndexRequest source = request.source(JSON.toJSONString(user), XContentType.JSON);
    //客户端发送请求,获取响应的结果
    IndexResponse indexResponse = restHighLevelClient.index(request, RequestOptions.DEFAULT);
    System.out.println(indexResponse.toString());
    System.out.println(indexResponse.status());
}

```

获取文档

```

@Test
void testExists() throws IOException {
    GetRequest getRequest = new GetRequest("yscxy_index", "1");
    //不获取返回的_source的上下文
    getRequest.fetchSourceContext(new FetchSourceContext(false));
    getRequest.storedFields("_none_");
    boolean exists = restHighLevelClient.exists(getRequest, RequestOptions.DEFAULT);
    System.out.println(exists);
}

```

获取文档信息

```

@Test
void testGetDocument() throws IOException {
    GetRequest getRequest = new GetRequest("yscxy_index", "1");
    GetResponse getResponse = restHighLevelClient.get(getRequest, RequestOptions.DEFAULT);
    System.out.println(getResponse.getSourceAsString());
    System.out.println(getResponse);
}

```

更新文档信息

```
@Test
void testUpdate() throws IOException {
    UpdateRequest updateRequest = new UpdateRequest("yscxy_index", "1");
    updateRequest.timeout("1s");
    User user = new User("野生程序员JAVA", 18);
    updateRequest.doc(JSON.toJSONString(user), XContentType.JSON);
    UpdateResponse updateResponse = restHighLevelClient.update(updateRequest, RequestOptions.DEFAULT);
    System.out.println(updateResponse.status());
}
```

删除文档记录

```
@Test
void testDeleteRequest() throws IOException {
    DeleteRequest deleteRequest = new DeleteRequest("yscxy_index", "1");
    deleteRequest.timeout("1s");
    DeleteResponse deleteResponse = restHighLevelClient.delete(deleteRequest, RequestOptions.DEFAULT);
    //打印状态
    System.out.println(deleteResponse.status());
}
```

大批量的插入，一般是数据库导入

```
@Test
void testBulkRequest() throws IOException {
    BulkRequest bulkRequest = new BulkRequest();
    bulkRequest.timeout("10s");
    ArrayList<User> list = new ArrayList<>();
    list.add(new User("大神", 1));
    list.add(new User("小弟", 5));
    list.add(new User("二哈", 1));
    list.add(new User("龙猫", 15));
    list.add(new User("小老鼠", 21));
    list.add(new User("老大", 1));
    list.add(new User("牛二", 1));
    list.add(new User("小傻子", 11));
    list.add(new User("大马猴", 13));
    //批量处理请求（批量更新修改等等都可以这样操作！）
    for (int i = 0; i < list.size(); i++) {
        bulkRequest.add(
            new IndexRequest("yscxy_index")
                .id("" + (i + 1)) //如果没有写id可以自动生成
                .source(JSON.toJSONString(list.get(i)), XContentType.JSON)
        );
    }
    BulkResponse bulkItemResponses = restHighLevelClient.bulk(bulkRequest, RequestOptions.DEFAULT);
    System.out.println(bulkItemResponses.hasFailures()); //是否执行错误
}
```

查询

```
//查询
//SearchRequest 搜索请求
//SearchSourceBuilder 条件构造

@Test
void testSearch() throws IOException {
    SearchRequest searchRequest = new SearchRequest("yscxy_index");
    //构建搜索条件
    SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
    //查询条件, 我们可以使用QueryBuilders工具来实现
    //QueryBuilders.termQuery 精确匹配
    TermQueryBuilder queryBuilders = QueryBuilders.termQuery("name", "牛二");
    searchSourceBuilder.query(queryBuilders);
    //时间限制
    searchSourceBuilder.timeout(new TimeValue(1, TimeUnit.SECONDS));
    searchRequest.source(searchSourceBuilder);

    SearchResponse searchResponse = restHighLevelClient.search(searchRequest, RequestOptions.DEFAULT);
    System.out.println(JSON.toJSONString(searchResponse.getHits()));
    for (SearchHit s:searchResponse.getHits().getHits()) {
        System.out.println(JSON.toJSONString(s));
    }
}
```