

Speed up writing tests with Wiremock Spring Boot

Pieter-Jan Drouillon

devoxx.be 2023

Intro

What is this about?



Hello!

Pieter-Jan Drouillon

Tech Lead @ emakina.be

@pjdrdrouillon
www.pieterjd.be

Disclaimer

Setup

Demo: spring boot + jsonplaceholder client + unittests

Quick intro to Wiremock

Search Results

```
{  
  "request": {  
    "urlPath": "/search?q=mock",  
    "method": "GET"  
  },  
  "response": {  
    "status": 200,  
    "body": "Here are the search results for 'mock'!",  
    "headers": {  
      "Content-Type": "text/plain"  
    }  
  }  
}
```

-  GET /search?q=mock
-  POST /search?q=mock
-  GET /search?q=hello

Prime Check

```
{  
  "request": {  
    "urlPath": "/isprime",  
    "method": "GET",  
    "queryParameters": {  
      "number": {  
        "matches": "[\\d]+"  
      }  
    }  
  },  
  "response": {  
    "status": 200,  
    "jsonBody": {  
      "isPrime": true  
    },  
    "headers": {  
      "Content-Type": "application/json"  
    }  
  }  
}
```

- ▶ GET /isprime?number=123
- ▢ OPTIONS /isprime?number=123
- ▢ GET /isprime?number=-7
- ▢ GET /isprime

Updated tests for jsonplaceholder client

Demo

Throw in Elastic Search

Repository + Service

Issues

- Third party library
- Rest API behind the scene
- no direct access to requests and responses

check source code

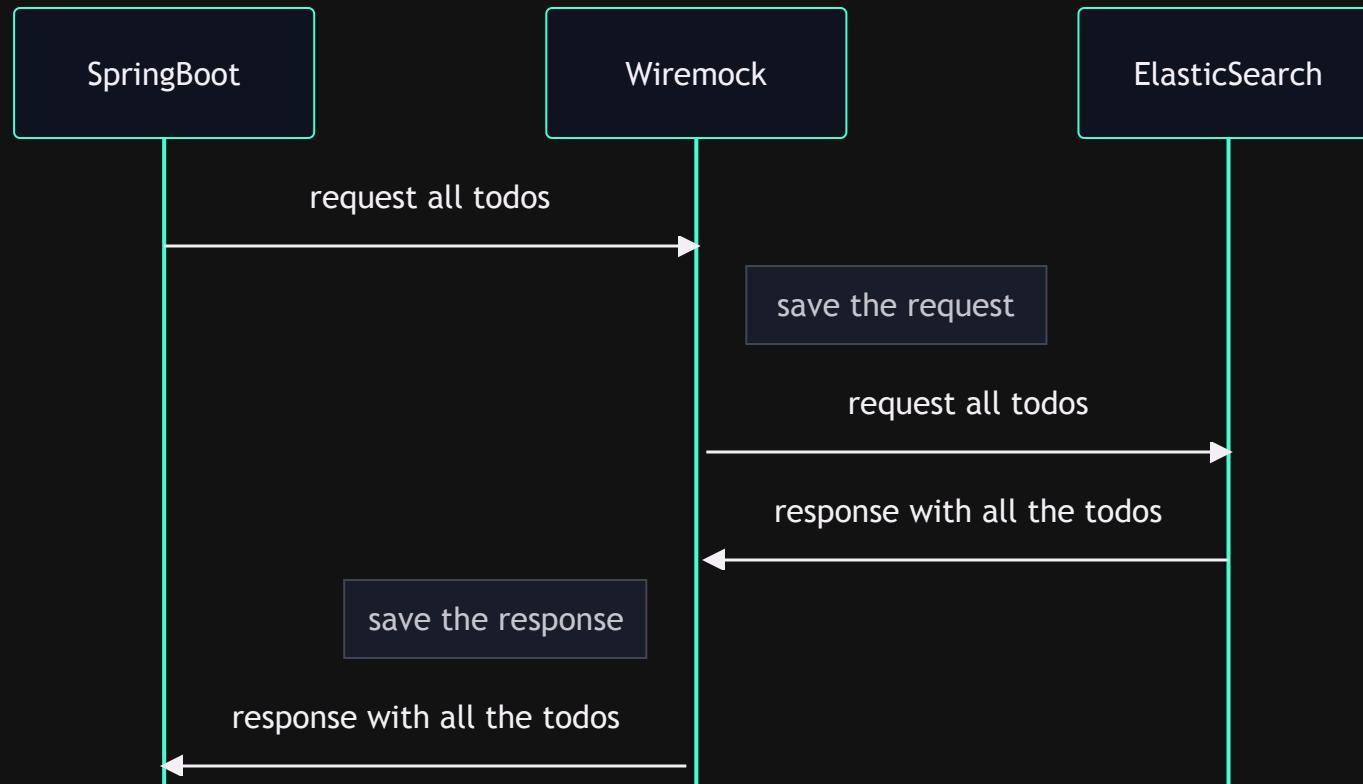
The screenshot shows a GitHub repository page for the Elasticsearch Java Client. The URL is <https://github.com/elastic/elasticsearch-java/blob/main/java-client/src/main/java/co/elastic/clients/elasticsearch/ElasticsearchClient.java>. The page displays the source code for the `ElasticsearchClient` class. The code is written in Java and imports various classes from the `co.elastic.clients.elasticsearch` package. The code handles requests like `bulk`, `clearScroll`, `closePointInTime`, `count`, `create`, `deleteByQuery`, `exists`, `existsSource`, `explain`, `fieldCaps`, and `get`. The code is annotated with line numbers and includes several import statements.

```
23
24 package co.elastic.clients.elasticsearch;
25
26 import co.elastic.clients.ApiClient;
27 import co.elastic.clients.elasticsearch._types.ElasticsearchException;
28 import co.elastic.clients.elasticsearch._types.ErrorResponse;
29 import co.elastic.clients.elasticsearch.async_search.ElasticsearchAsyncSearchClient;
30 import co.elastic.clients.elasticsearch.autoscaling.ElasticsearchAutoscalingClient;
31 import co.elastic.clients.elasticsearch.cat.ElasticsearchCatClient;
32 import co.elastic.clients.elasticsearch.ccr.ElasticsearchCcrClient;
33 import co.elastic.clients.elasticsearch.cluster.ElasticsearchClusterClient;
34 import co.elastic.clients.elasticsearch.core.BulkRequest;
35 import co.elastic.clients.elasticsearch.core.BulkResponse;
36 import co.elastic.clients.elasticsearch.core.ClearScrollRequest;
37 import co.elastic.clients.elasticsearch.core.ClearScrollResponse;
38 import co.elastic.clients.elasticsearch.core.ClosePointInTimeRequest;
39 import co.elastic.clients.elasticsearch.core.ClosePointInTimeResponse;
40 import co.elastic.clients.elasticsearch.core.CountRequest;
41 import co.elastic.clients.elasticsearch.core.CountResponse;
42 import co.elastic.clients.elasticsearch.core.CreateRequest;
43 import co.elastic.clients.elasticsearch.core.CreateResponse;
44 import co.elastic.clients.elasticsearch.core.DeleteByQueryRequest;
45 import co.elastic.clients.elasticsearch.core.DeleteByQueryResponse;
46 import co.elastic.clients.elasticsearch.core.DeleteByQueryRethrottleRequest;
47 import co.elastic.clients.elasticsearch.core.DeleteByQueryRethrottleResponse;
48 import co.elastic.clients.elasticsearch.core.DeleteRequest;
49 import co.elastic.clients.elasticsearch.core.DeleteResponse;
50 import co.elastic.clients.elasticsearch.core.DeleteScriptRequest;
51 import co.elastic.clients.elasticsearch.core.DeleteScriptResponse;
52 import co.elastic.clients.elasticsearch.core.ExistsRequest;
53 import co.elastic.clients.elasticsearch.core.ExistsSourceRequest;
54 import co.elastic.clients.elasticsearch.core.ExplainRequest;
55 import co.elastic.clients.elasticsearch.core.ExplainResponse;
56 import co.elastic.clients.elasticsearch.core.FieldCapsRequest;
57 import co.elastic.clients.elasticsearch.core.FieldCapsResponse;
58 import co.elastic.clients.elasticsearch.core.GetRequest;
59 import co.elastic.clients.elasticsearch.core.GetResponse;
```

Turn up log level httpclient

```
org.apache.http.wire          : http-outgoing-0 >> "HEAD /todo HTTP/1.1[\r][\n]"
org.apache.http.wire          : http-outgoing-0 >> "User-Agent: elastic-java/8.7.1 (Java/17.0.3)[\r][\n]"
org.apache.http.wire          : http-outgoing-0 >> "Accept: application/vnd.elasticsearch+json; compatible-w"
org.apache.http.wire          : http-outgoing-0 >> "Content-Type: application/vnd.elasticsearch+json; compatibl"
org.apache.http.wire          : http-outgoing-0 >> "X-SpringDataElasticsearch-Client: imperative[\r][\n]"
org.apache.http.wire          : http-outgoing-0 >> "X-Elastic-Client-Meta: es=8.7.1,jv=17,hl=2,t=8.7.1,hc=4,:"
org.apache.http.wire          : http-outgoing-0 >> "Host: localhost:9200[\r][\n]"
org.apache.http.wire          : http-outgoing-0 >> "Connection: Keep-Alive[\r][\n]"
org.apache.http.wire          : http-outgoing-0 >> "[\r][\n]"
o.a.h.i.nio.client.InternalIODispatch : http-outgoing-0 [ACTIVE] Request ready
h.i.n.c.ManagedNHttpClientConnectionImpl : http-outgoing-0 127.0.0.1:60398<->127.0.0.1:9200[ACTIVE][r:w]: Event cleared
h.i.n.c.ManagedNHttpClientConnectionImpl : http-outgoing-0 127.0.0.1:60398<->127.0.0.1:9200[ACTIVE][r:r]: 149 bytes read
org.apache.http.wire          : http-outgoing-0 << "HTTP/1.1 200 OK[\r][\n]"
org.apache.http.wire          : http-outgoing-0 << "X-elastic-product: Elasticsearch[\r][\n]"
org.apache.http.wire          : http-outgoing-0 << "content-type: application/vnd.elasticsearch+json;compatibil"
org.apache.http.wire          : http-outgoing-0 << "Transfer-Encoding: chunked[\r][\n]"
org.apache.http.wire          : http-outgoing-0 << "[\r][\n]"
org.apache.http.headers       : http-outgoing-0 << HTTP/1.1 200 OK
org.apache.http.headers       : http-outgoing-0 << X-elastic-product: Elasticsearch
org.apache.http.headers       : http-outgoing-0 << content-type: application/vnd.elasticsearch+json;compatibl
org.apache.http.headers       : http-outgoing-0 << Transfer-Encoding: chunked
o.a.h.i.nio.client.InternalIODispatch : http-outgoing-0 [ACTIVE] Response received
```

Man in the middle approach



Wiremock Record modus

Demo: wiremock stand-alone in record modus

Conclusion

Take Aways

- Wiremock is great
- Wiremock Spring Boot even greater
- Check Wiremock Library
- Record mode
- Check response templating
- ~~Check Wiremock Studio~~¹
- In case you do local development with Wiremock standalone

```
watchexec -N -w <path to your wiremock mappings> 'curl -X POST http://localhost:9999/__admin/reset;date'
```

¹This proprietary product has been discontinued for more than one year, and the recent security advisory announce a severe security issue that will not be fixed.

Thanks!