

# 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](http://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 Elasticsearch core classes. The GitHub interface includes a sidebar with file navigation, a search bar, and a toolbar with various icons.

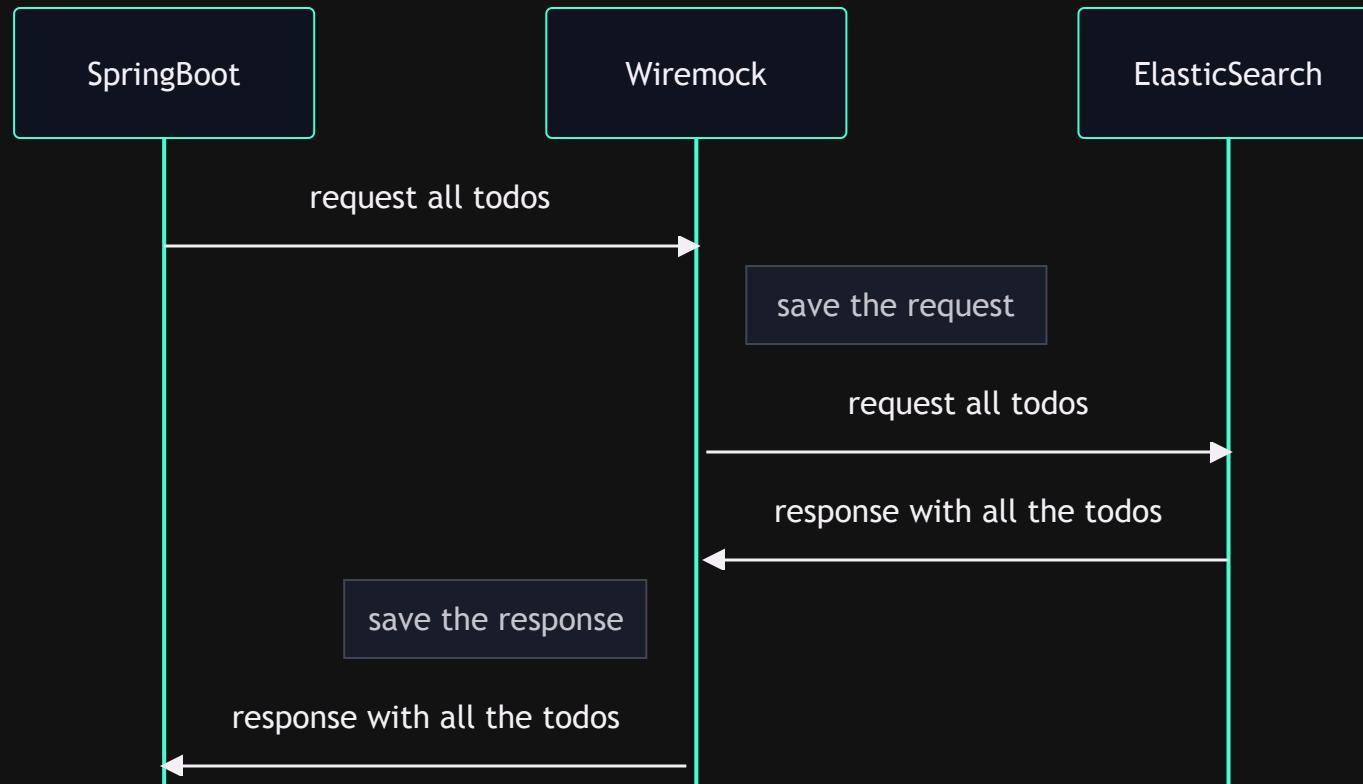
```
package co.elastic.clients.elasticsearch;

import co.elastic.clients.ApiClient;
import co.elastic.clients.elasticsearch._types.ElasticsearchException;
import co.elastic.clients.elasticsearch._types.ErrorResponse;
import co.elastic.clients.elasticsearch.async_search.ElasticsearchAsyncSearchClient;
import co.elastic.clients.elasticsearch.autoscaling.ElasticsearchAutoscalingClient;
import co.elastic.clients.elasticsearch.cat.ElasticsearchCatClient;
import co.elastic.clients.elasticsearch.ccr.ElasticsearchCcrClient;
import co.elastic.clients.elasticsearch.cluster.ElasticsearchClusterClient;
import co.elastic.clients.elasticsearch.core.BulkRequest;
import co.elastic.clients.elasticsearch.core.BulkResponse;
import co.elastic.clients.elasticsearch.core.ClearScrollRequest;
import co.elastic.clients.elasticsearch.core.ClearScrollResponse;
import co.elastic.clients.elasticsearch.core.ClosePointInTimeRequest;
import co.elastic.clients.elasticsearch.core.ClosePointInTimeResponse;
import co.elastic.clients.elasticsearch.core.CountRequest;
import co.elastic.clients.elasticsearch.core.CountResponse;
import co.elastic.clients.elasticsearch.core.CreateRequest;
import co.elastic.clients.elasticsearch.core.CreateResponse;
import co.elastic.clients.elasticsearch.core.DeleteByQueryRequest;
import co.elastic.clients.elasticsearch.core.DeleteByQueryResponse;
import co.elastic.clients.elasticsearch.core.DeleteByQueryRethrottleRequest;
import co.elastic.clients.elasticsearch.core.DeleteByQueryRethrottleResponse;
import co.elastic.clients.elasticsearch.core.DeleteRequest;
import co.elastic.clients.elasticsearch.core.DeleteResponse;
import co.elastic.clients.elasticsearch.core.DeleteScriptRequest;
import co.elastic.clients.elasticsearch.core.DeleteScriptResponse;
import co.elastic.clients.elasticsearch.core.ExistsRequest;
import co.elastic.clients.elasticsearch.core.ExistsSourceRequest;
import co.elastic.clients.elasticsearch.core.ExplainRequest;
import co.elastic.clients.elasticsearch.core.ExplainResponse;
import co.elastic.clients.elasticsearch.core.FieldCapsRequest;
import co.elastic.clients.elasticsearch.core.FieldCapsResponse;
import co.elastic.clients.elasticsearch.core.GetRequest;
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;compatibl"
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
- 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'
```

Thanks!