Spring Data Elasticsearch - 2021 edition

Alexander Reelsen

alex@elastic.co | @spinscale

Agenda

- Spring Data Introduction
- Spring Data Elasticsearch
- Spring Data Elasticsearch Reactive
- Complex Queries
- What's next?
- Q&A

Spring Data

> Spring Data's mission is to provide a familiar and consistent, Spring-based programming model for data access while still retaining the special traits of the underlying data store.

Spring Data

- Implementations: JPA, JDBC, LDAP, MongoDB, Redis, Cassandra
- Community: Aerospike, ArangoDB, Couchbase, Google Spanner, DynamoDb, Hazelcast, Neo4j, Solr, Yugabyte, `Elasticsearch`

Spring Data

- `ElasticsearchTemplate` helper class
- Object Mapping, Annotation based metadata (`@Id`, `@Field`)
- Repositories

Entity

```
@Document(indexName = "products")
public class Product {

    @Id
    private String id;

    @Field(type = FieldType.Text)
    private String name;

    @Field(type = FieldType.Text)
    private String description;

    @Field(type = FieldType.Keyword)
    private List<String> tags;

    @Field(type = FieldType.Keyword, name = "image_url")
    private String imageUrl;

    @Field(type= FieldType.Date, format = DateFormat.date_time)
    private ZonedDateTime created;

...
}
```

Repositories

- method names reflect finders
- select fields, ranges, specify sorting, i.e. `findByNameAndPrice`, `findByPriceGreaterThan`
- **Note**: Inefficient queries like `findByNameEndingWith` / `findByNameStartingWith` / `findByNameContaining`

Repositories

```
public interface UserProfileRepository extends ElasticsearchRepository<UserProfile, String> {

}
```

Reactive Repostiories

```
public interface ProductRepository extends ReactiveSortingRepository<Product, String> {

    @Query("""
{
  "bool" : {
    "must" : [ { "multi_match" : { "query": "?0", "fields": [ "description", "name", "tags" ] } } ],
    "should" : [ { "match" : { "name" : "?0" } } ]
  }
}
""")
    Flux<Product> findProducts(String q);
}
```

Reactive Repositories

```
@RestController
@RequestMapping("/products")
public class ProductController {

    @GetMapping(value = "/product/{id}")
    public Mono<Product> retrieveProduct(@PathVariable String id) {
        // this does not return a 404 when the Mono is empty... weird default mode IMO
        return repository.findById(id);
    }

    @DeleteMapping("/product/{id}")
    public Mono<Void> deleteProduct(@PathVariable String id) {
        return repository.deleteById(id);
    }

    @GetMapping(value = "/search")
    public Flux<Product> search(@RequestParam String q) {
        return repository.findProducts(q);
    }
}
```

ElasticsearchTemplate

```
final BoolQueryBuilder qb = QueryBuilders.boolQuery()
        .must(QueryBuilders.termQuery("url", contribution.getUrl()))
        .must(QueryBuilders.termQuery("submitted_by.email", login(principal)))
        .must(QueryBuilders.termQuery("category", contribution.getCategory()));
final SearchHit<Contribution> hit = elasticsearchRestTemplate.searchOne(new NativeSearchQuery(qb), Contribution.class);
```

Complex queries

```
QueryBuilder qb = QueryBuilders.boolQuery()
        .mustNot(QueryBuilders.termQuery("category", Contribution.Category.CONTRIBUTION_RATING))
        .filter(QueryBuilders.termQuery("region", region))
        .filter(QueryBuilders.termQuery("state", Contribution.State.APPROVED))
        .filter(QueryBuilders.termQuery("cycle", Cycle.current()));

final NativeSearchQuery query = new NativeSearchQuery(qb);
query.setPageable(Pageable.unpaged());
// ensure uniqueness by email, then retrieve latest fullname
query.addAggregation(AggregationBuilders.terms("by_user").field("submitted_by.email").size(100)
        .subAggregation(AggregationBuilders.topHits("by_name").size(1).sort(SortBuilders.fieldSort("created_at")
                .order(SortOrder.DESC)).fetchSource("submitted_by.full_name", "")));

final SearchHits<Contribution> hits = elasticsearchTemplate.search(query, Contribution.class);
```

Client configuration

```java
@Configuration
public class ReactiveRestClientConfig extends AbstractReactiveElasticsearchConfiguration {
    @Override
    @Bean
    public ReactiveElasticsearchClient reactiveElasticsearchClient() {

        final String elasticsearchUrl = System.getenv("ELASTICSEARCH_URL");
        if (elasticsearchUrl != null) {
            final URI uri = URI.create(elasticsearchUrl);
            final int port = uri.getPort();
            final String host = uri.getHost();

            final InetSocketAddress addr = new InetSocketAddress(host, port);
            final ClientConfiguration.MaybeSecureClientConfigurationBuilder builder = ClientConfiguration.builder().connectedTo(addr);
            if (uri.getScheme().equals("https")) {
                builder.usingSsl();
            }

            if (uri.getUserInfo() != null) {
                final String[] data = uri.getUserInfo().split(":", 2);
                builder.withBasicAuth(data[0], data[1]);
            }

            return ReactiveRestClients.create(builder.build());
        } else {
            // fallback to localhost:9200 as default, with SSL
            final InetSocketAddress addr = new InetSocketAddress("localhost", 9200);
            final ClientConfiguration.MaybeSecureClientConfigurationBuilder builder = ClientConfiguration.builder().connectedTo(addr);
            return ReactiveRestClients.create(builder.build());
        }
    }
}
```

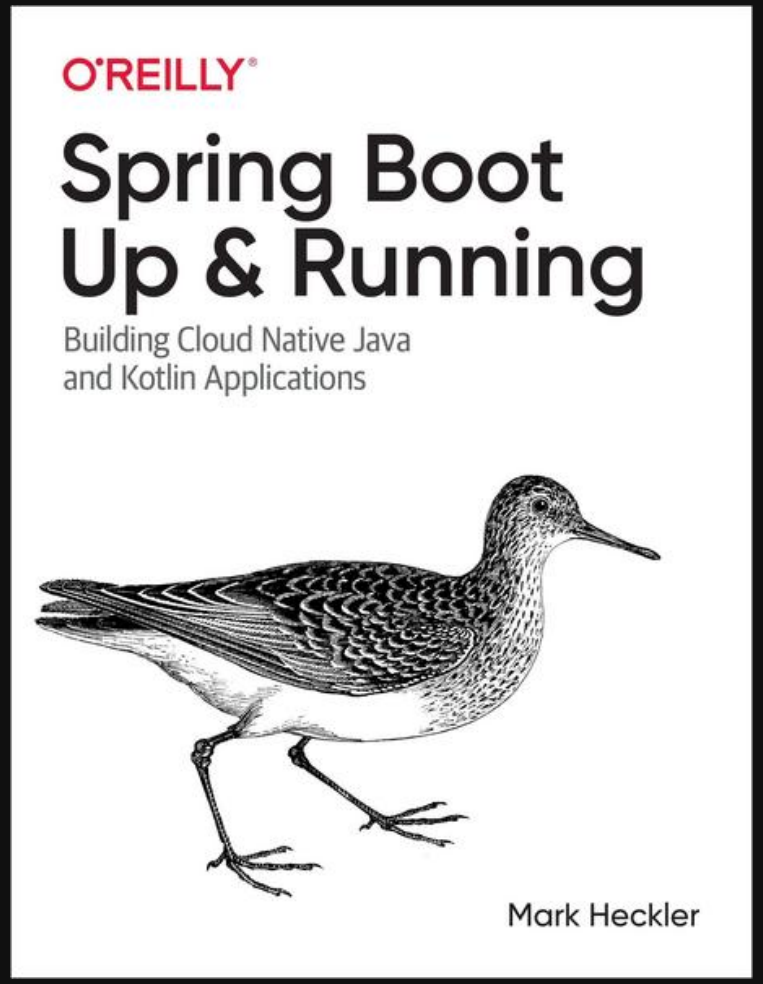
What's next

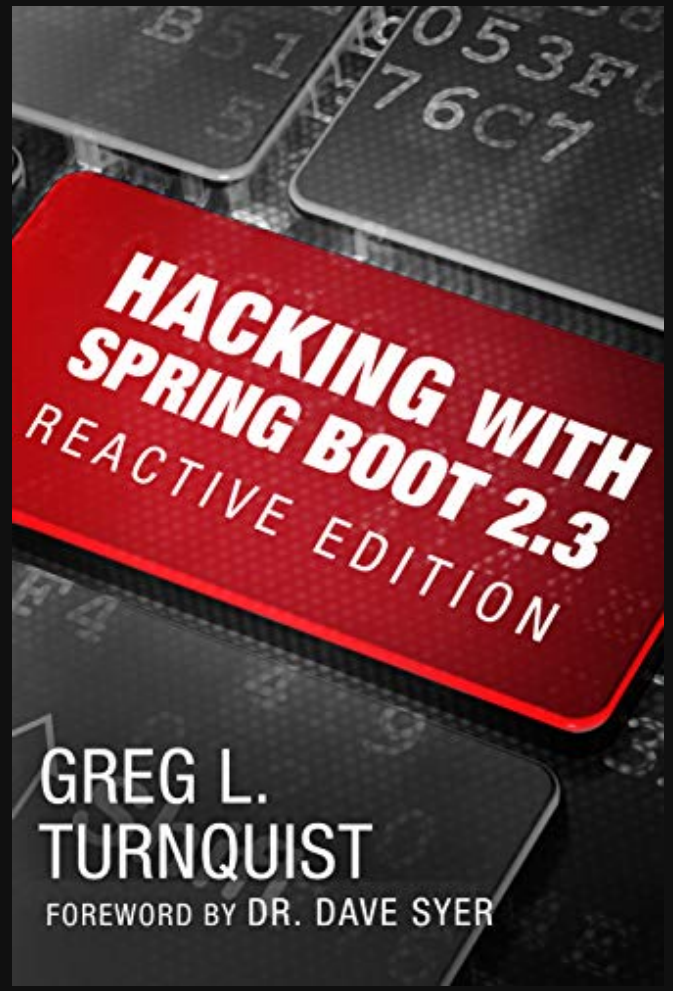
- **New** Elasticsearch Client without Elasticsearch dependency
- Client gets generated from specification
- Spring Data Elasticsearch will move to the client at some point, still in development
- New client has the concept of a Transport, which could be reused when talking to Elastic solutions

Resources

- https://github.com/spinscale/spring-boot-reactive-observability-demo
- https://github.com/spinscale/link-rating
- https://docs.spring.io/spring-data/elasticsearch/docs/current/reference/html/
- https://github.com/elastic/ecs-logging-java
- https://spinscale.de/posts/2020-04-15-introduction-into-the-elasticsearch-java-rest-client.html
- https://spinscale.de/posts/2020-08-06-introduction-into-spring-data-elasticsearch.html

Books


O'REILLY
Spring Boot
Up & Running
Building Cloud Native Java
and Kotlin Applications
Mark Heckler


HACKING WITH
SPRING BOOT 2.3
REACTIVE EDITION
GREG L.
TURNQUIST
FOREWORD BY DR. DAVE SYER

Thanks for listening

Q & A

Alexander Reelsen
alex@elastic.co | @spinscale