

Observing Spring for GraphQL in Action



Brian Clozel

mamot.fr/@bclozel

Rossen Stoyanchev

[@rstoya05](https://twitter.com/rstoya05)





BRIAN CLOZEL

VMware

Spring Framework and Spring for GraphQL teams

Currently working with the Observability team

<https://spring.io/team/bclozel>



ROSSEN STOYANCHEV

VMware

Spring Framework and Spring for GraphQL teams

Focus on web development

Less frequently RSocket, Reactor, Spring Web Flow

<https://spring.io/team/rstoyanchev>

Agenda

Architect's view of GraphQL and Spring

What's changed over the past year

Demo

Where to next?

HTTP Protocol

Many URLs,
headers, cookies,
media types

Web Layer

@Controller
@RequestMapping methods

Application

@Service, @Repository

Exposing data with GraphQL

Schema, query language

Hierarchical, strongly typed, introspective

Client-specified response



< Docs

🔍 K

Query

Fields

```
searchForTracks(  
  keyword: String!  
  first: Int  
  last: Int  
  before: String  
  after: String  
): TrackConnection
```

```
1 query { searchForTracks(keyword: "Code") {  
2   pageInfo {  
3     startCursor,  
4     endCursor  
5   }  
6   edges {  
7     node {  
8       title  
9     }  
10  }  
11 }  
12 }
```

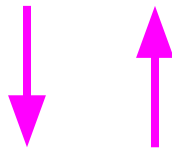


Variables

Headers

+ GraphQL

```
{  
  "data": {  
    "searchForTracks": {  
      "pageInfo": {  
        "startCursor": "T18x",  
        "endCursor": "T18xMA="  
      },  
      "edges": [  
        {  
          "node": {  
            "title": "The Code Challenger"  
          }  
        },  
        {  
          "node": {  
            "title": "The Code Unplugged"  
          }  
        },  
        {  
          "node": {  
            "title": "The Code Composer"  
          }  
        },  
        {  
          "node": {  
            "title": "The Code of the Cosmos"  
          }  
        },  
        {  
          "node": {  
            "title": "The Code Master"  
          }  
        },  
        {  
          "node": {  
            "title": "The Code of the Machine"  
          }  
        }  
      ]  
    }  
  }  
}
```



GraphQL <language> engine

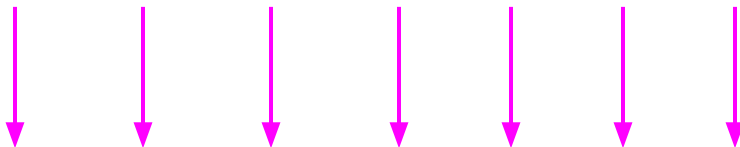
Parse requests, validate syntax,
orchestrate data fetching



GraphQL Java

Parse requests, validate syntax,
orchestrate data fetching (field by field)

DataFetcher



Data layer

@Controller, @SchemaMapping methods

Application

@Service, @Repository

Single endpoint

HTTP

WebSocket

RSocket

GraphQL Java

Parse requests, validate syntax,
orchestrate data fetching

Data layer

@Controller, @SchemaMapping methods

Application

@Service, @Repository



Spring Framework 6
Java 17, Jakarta EE migration

Version 1.1

Focus on baseline for Spring Framework 6

Java 17 and Jakarta EE

AOT/Native

Micrometer, observability

Version 1.1

Input Argument Handling

Constructor or setters, nested structure since 1.0

JSON → DataBinder → self-sufficient

ArgumentValue (null vs omitted)

Version 1.1

Controller Method Validation

Built-in support since 1.0

Startup checks if controller method has constraints

Validation groups with `@Validated`

Version 1.2

@GraphQLExceptionHandler methods

[Pagination](#) and sorting

Schema mapping [inspection](#)

DEMO TIME



<https://github.com/bclozel/graphql-music>

Where to next?

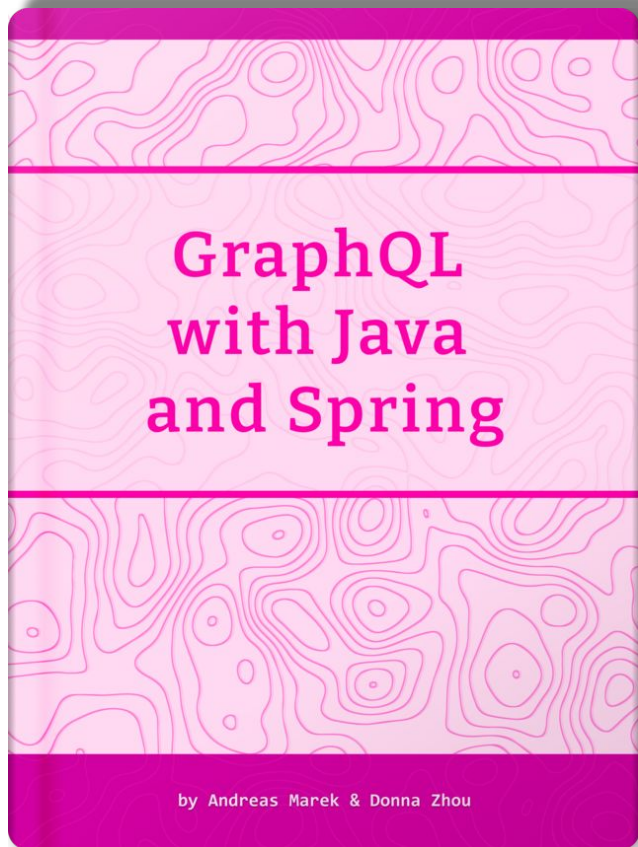
Version 1.3 in November 2023

Continue to evolve programming model

Community feedback



ANDREAS MAREK
GraphQL Java founder



DONNA ZHOU
GraphQL Java committer

THANK YOU!

Q&A

Brian Clozel

mamot.fr/@bclozel

Rossen Stoyanchev

[@rstoya05](https://twitter.com/rstoya05)

