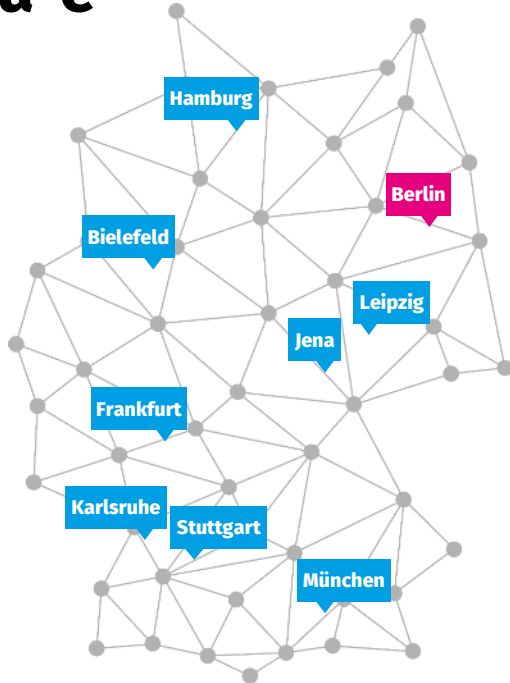


GraphQL mit Spring Boot und Spring Security

Workshop mit Christian Kumpé und Thorben Hischke, Entwicklertag Karlsruhe 5. Juni 2019

diva-e



9 Standorte

Wir halten eine umfangreiche Vor-Ort-Präsenz bei unseren Kunden als einen wichtigen Erfolgsfaktor. Deshalb sind unsere Standorte über die gesamte Bundesrepublik verteilt.

500+ Mitarbeiter

Bei uns arbeiten derzeit über 500 Mitarbeiter.

100% Know-How

Von E-Commerce über Content- und Digital-Marketing-Services bis hin zur Retail-Kompetenz: Bei uns erhalten Sie Leistungen aus allen wichtigen E-Business-Disziplinen, vernetzt unter einem Dach.

Referenten

Christian Kumpe

Expert Developer

- Informatikstudium am KIT (Universität Karlsruhe)
- Freelancer im Bereich Web und Java
- Seit Mai 2011 bei diva-e in Karlsruhe
- Seit 2002 in der Java-Welt unterwegs



Referenten

Thorben Hischke

Senior Architekt

- Dipl.-Inf. (FH), seit 2007 bei Netpioneer GmbH / diva-e
- Schwerpunkt CMS (Imperia, FirstSpirit)
- Integration und Kundenzufriedenheit



Überblick REST



Vorteile

- Dissertation in 2000
- Architekturkonzept
- HTTP-Direktive
 - Verbs
 - Status-Code
 - Content-Type
 - Caching
- 4 Maturity Levels
- Swagger



Nachteile

- In der realen Welt wird meist Maturity Level 2 und kein *HATEOS* eingesetzt.
- *HATEOS* erzeugt viele Requests
- Applikation braucht meist mehrere Zugriffe, um alle Daten zu erfragen
 - Kein Subset
 - Netzwerk-Overhead
 - Längere Laufzeit

Überblick GraphQL



Vorteile

- Facebook veröffentlicht 2016
- Abfragesprache (GraphQL Schema)
- Spezifikation
- Einen Endpunkt via HTTP
- Applikation kann Subsets erfragen
- Applikation braucht nur einen Zugriff



Nachteile

- „Umdenken“ in der Modellierung
- Query of death

Operationen in GraphQL

- **Queries**

Mit Queries können Daten abgefragt werden.

- **Mutations**

Mit Mutations können Daten verändert werden.

- **Subscriptions**

Mit Subscriptions kann der Client auf Events warten.

Unser Beispiel

Bei unserer Versicherung kann der Kunde (Person) Versicherungen (Verträge) abschließen.

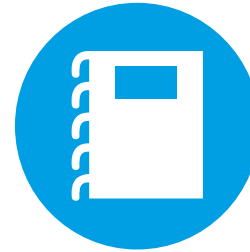


Person

- Anrede
- Vor- und Nachname
- Email
- Anschrift
- Telefonnummer
- Newsletter bestellt?
- Verträge

1

0..*



Vertrag

- Name

Projektsetup

- **Repository klonen**

```
git clone https://github.com/diva-e/graphql-spring-boot.git  
git checkout tutorial
```

- **Projekt in IDE einrichten**

Projekt bauen und GraphQLSpringBootApplication starten. Im Browser ist die Applikation unter <http://localhost:8080> erreichbar.

- **GraphQL Kickstarter Abhängigkeiten einfügen**

```
com.graphql-java-kickstart:graphql-spring-boot-starter Version 5.9.0  
com.graphql-java-kickstart:graphql-spring-boot-starter Version 5.0.6  
com.graphql-java-kickstart:graphql-spring-boot-starter-test Version 5.9.0
```

Person modellieren mit entsprechenden Abfragen

- **Schema für Person erstellen**

Vorlage ist die `Person.java` Klasse im Projekt

- **Queries für Personen erstellen**

`persons` liefert alle gespeicherten Personen zurück

`person(id)` liefert eine Person anhand ihrer ID zurück

Testen

GraphiQL

- Lokal erreichbar unter <http://localhost:8080/graphiql>
- Electron based App (<https://github.com/skevy/graphiql-app>)

Unit Tests

- Testklasse mit `@GraphQLTest` annotieren
- Schnittstelle über `GraphQLTestTemplate` ansteuern

Security

- **Spring Security einbinden**

Einfügen von `org.springframework.boot:spring-boot-starter-security` in die Projektabhängigkeiten

Dafür sorgen dass `/graphql`, `/subscriptions`, `/graphql` und `/vendor/...` erreichbar sind.

- **Method Based Security aktivieren**

Über die entsprechende Annotation: `@EnableGlobalMethodSecurity(prePostEnabled = true)`

- **Absichern der GraphQLResolver-Methoden**

Bspw. mit `@PreAuthorize("hasRole('USER')")`

Bearbeiten von Personen und deren Verträgen ermöglichen

- **Mutation zum Ändern und Erstellen von Personen**

Input Type erstellen und Mutation definieren

- **Schema für Verträge erstellen**

Vorlage ist die `Contract.java` Klasse im Projekt

- **Verträge zu Personen hinzufügen**

`addContractToPerson(personId, name)` fügt der Personen einen Vertrag hinzu

Fortgeschrittene Queries

- **Variablen**

Dynamische Anteile von Queries können mit Variablen abgebildet werden.

- **Direktiven**

Dynamische Änderungen der Struktur können mit Direktiven abgebildet werden.

- **Fragmente**

Wiederverwertbare „Abfrage-Schnipsel“ können mit Fragmenten gebaut werden.

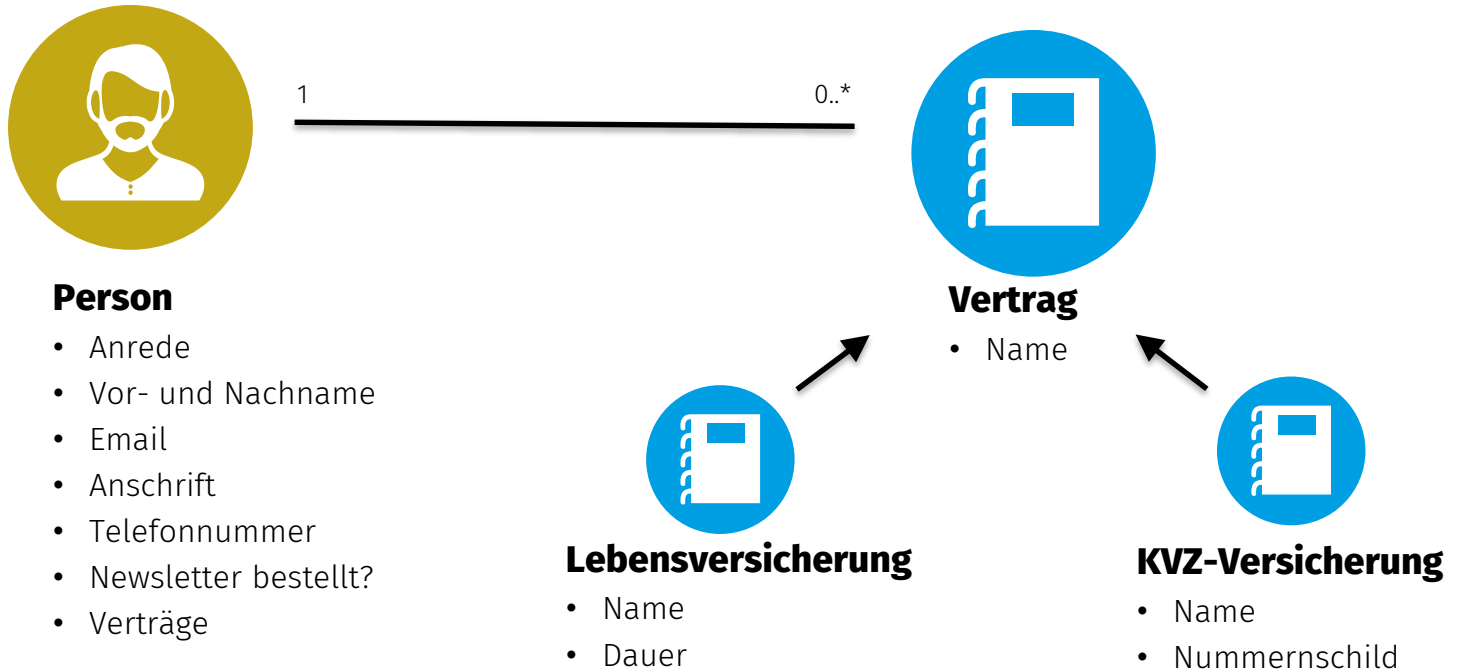
Benachrichtigungen über Ereignisse

- **Subscriptions für Änderungen an Personen und Verträgen erstellen**

`watchPersons` liefert geänderte oder neu erstellte Personen

`watchContracts` liefert geänderte oder neu erstellte Verträge

Bei unserer Versicherung kann der Kunde (Person) 2 konkrete Versicherungen (Verträge) abschließen, eine KFZ- und eine Lebensversicherung.



Weitere Möglichkeiten im Schema

- **Interfaces**

Die Klasse `Contract.java` als Interface refaktorisieren und im GraphQL-Schema ändern.

Zwei konkrete Implementierungen:

- `ContractCarInsurance.java` mit dem zusätzlichen Attribut `String numberPlate`
- `ContractLifeInsurance.java` mit dem zusätzlichen Attribut `Integer duration`

Die konkreten Implementierungen finden ihre Entsprechung im GraphQL-Schema.

Jeweils eine Mutation zur Erzeugung eines konkreten Vertrages.

Eine Mutation `assignContractToPerson(personId, contractId)` zur Verknüpfung Kunde <-> Vertrag.

- **Deprecations**

Mit `@deprecated` werden Felder als veraltet markiert.

**diva-e. You can't buy it. You can't make it.
And you sure can't fake it.**

diva^e
Digital Value Excellence

Danke

Bitte geben sie uns Feedback!



Copyright © diva-e

Alle Angaben basieren auf dem derzeitigen Kenntnisstand. Änderungen vorbehalten. Dieses Dokument der diva-e Digital Value Excellence GmbH ist ausschließlich für den Adressaten bzw. Auftraggeber bestimmt. Es bleibt bis zur einer ausdrücklichen Übertragung von Nutzungsrechten Eigentum der diva-e. Jede Bearbeitung, Verwertung, Vervielfältigung und/oder gewerbsmäßige Verbreitung des Werkes ist nur mit Einverständnis von diva-e zulässig.

All content is based on the current state of communication. Subject to change. This document of diva-e Digital Value Excellence GmbH is only intended for the client. It belongs to diva-e until its explicit transfer of usage rights. Any adaptation, utilization, copy and/or professional spreading has to be approved by diva-e.

Bitte geben Sie uns jetzt Ihr Feedback!

GraphQL mit Spring Boot und Spring
Security

Christian Kumpe, Thorben Hischke

