


# GraphQL

Getting Started with GraphQL in Spring Boot Applications



 [github.com/sonallux](https://github.com/sonallux)

# REST API

- `GET /person/{id}: Person`
- `GET /person/{id}/friends: Person[]`
- `GET /person/{id}/posts: Post[]`

```
interface Person {  
  id: number  
  name: string  
  age: number  
  aboutMe: string  
}
```

```
interface Post {  
  id: number  
  author: number  
  title: string  
  content: string  
  likes: number  
}
```

ts

## Profil einer Person

- Name
- Alter
- Über mich
- Freunde: (Name)
- Posts: (Titel, Likes)

## Probleme

- overfetching: zu viele Daten werden geladen
- underfetching: 3 Requests nötig

# REST API

- `GET /person/{id}: Person`
- `GET /person/{id}/friends: Person[]`
- `GET /person/{id}/posts: Post[]`

```
interface Person {  
  id: number  
  name: string  
  age: number  
  aboutMe: string  
}
```

```
interface Post {  
  id: number  
  author: number  
  title: string  
  content: string  
  likes: number  
}
```

ts

## Newsfeed für eine Person aufbauen

Liste mit Posts der Freunde

- Autor
- Titel
- Likes

## Probleme

- overfetching: zu viele Daten werden geladen
- underfetching:  $N + 1$  Requests

# GraphQL

- Datenabfrage- und Manipulationssprache
- Intern bei Facebook entwickelt, 2015 veröffentlicht
- Schema basiert
- Stark typisiert
- Transportprotokoll unabhängig
- Sprachunabhängige Spezifikation<sup>[1]</sup>

1. <https://spec.graphql.org>

# Schema

```
type Person {  
  id: ID!  
  name: String!  
  age: Int  
  friends(limit: Int = 10, offset: Int = 0): [Person!]!  
}  
  
type Query {  
  person(id: ID!): Person  
}  
  
type Mutation {  
  addFriendship(personId: ID): Int  
}  
  
type Subscription {  
  friendshipAdded: FriendshipEvent  
}  
  
type FriendshipEvent {  
  person1Id: ID!  
  person2Id: ID!  
}
```

Demo

# GraphQL over HTTP <sup>[1]</sup>

- Working draft / proposal
- De facto Standard
- `POST /graphql` mit JSON Body, der die Query oder Mutation als String unter dem Key `query` enthält
- `GET /graphql` mit Queryparameter `query`, der die Query als String enthält
- Antwort jeweils ein JSON mit `data` und `errors` Key

1. <https://github.com/graphql/graphql-over-http>

# GraphQL in der Java Welt

- **GraphQL Java**<sup>[1]</sup>: reine Implementierung der Spezifikation
- **Spring for GraphQL**<sup>[2]</sup>: Offizielle Integration von GraphQL in das Spring Ökosystem
- **Netflix DGS**<sup>[3]</sup>: GraphQL Server für Spring Boot mit GraphQL Federation<sup>[4]</sup> Unterstützung
- **GraphQL Java Kickstart**<sup>[5]</sup>: Community Integration von GraphQL in Spring Boot

1. <https://www.graphql-java.com>

2. <https://spring.io/projects/spring-graphql>

3. <https://netflix.github.io/dgs>

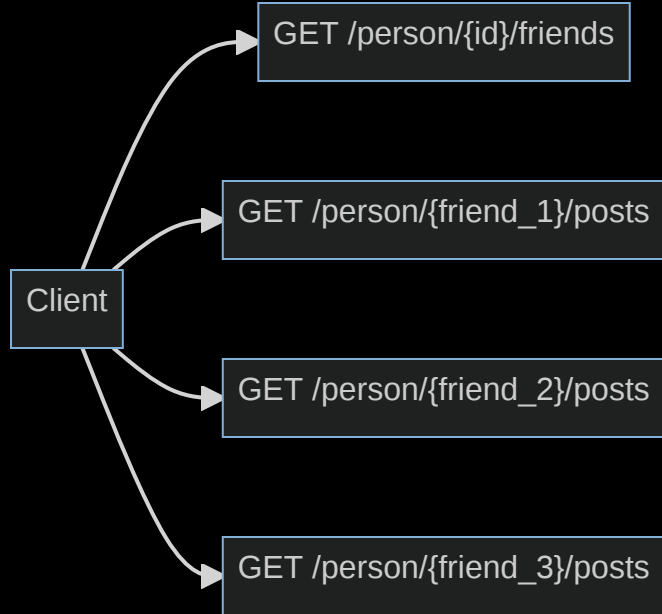
4. <https://www.apollographql.com/docs/federation/federation-spec>

5. <https://github.com/graphql-java-kickstart/graphql-spring-boot>

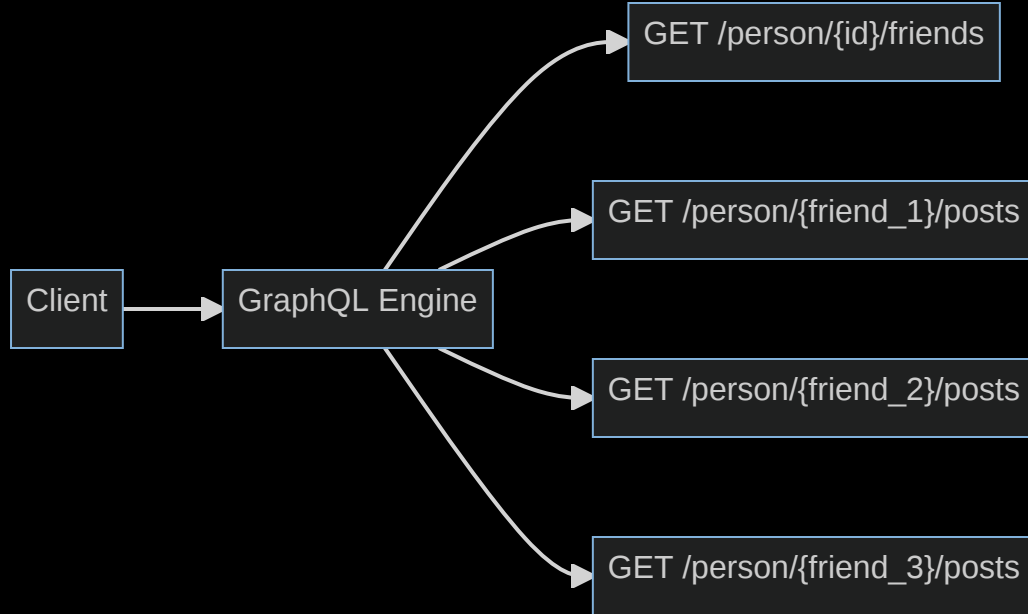


Live Coding

# N+1 Problem



# N+1 Problem



# Lösung: DataLoader<sup>[1]</sup>

- Utility für den Applikation Data Fetching Layer
- Batching
- Caching

1. <https://github.com/graphql/dataloader> bzw. <https://github.com/graphql-java/java-dataloader>

# Vorteile

- Bei graphartige Daten
- viele (unbekannte) Clients, die jeweils anderen Sichten auf die Daten benötigen
- Schema Validierung
- Starke Typisierung
- Built-in Subscriptions

# Nachteile

- Overfetching und Underfetching oft nur verschoben
- HTTP Status Codes und Caching Mechanismen können nicht verwendet werden

# Fragen ?

- [!\[\]\(849840539e55921a3851a4ff96d7400d\_img.jpg\) Spring Tips: Spring GraphQL](#)
- [!\[\]\(c176e0b06f6c5dd85a4598b214d1ebba\_img.jpg\) Learn Spring for GraphQL with Josh Long](#)
- [!\[\]\(66a18e26647fc145bd9198dd182dd107\_img.jpg\) Spring for GraphQL Documentation](#)
- [!\[\]\(572bcf30fdd4de64673b94584b7c6eca\_img.jpg\) Spring Boot Documentation: Spring for GraphQL](#)
- [!\[\]\(ba6dc7fecffbf82e7fd414c1c97a1ece\_img.jpg\) github.com/sonallux/social-network-graphql-example](https://github.com/sonallux/social-network-graphql-example)
- [!\[\]\(7b0c59a8d567ae8f4c94e1b0dfc0504e\_img.jpg\) github.com/sonallux/graphql-getting-started-presentation](https://github.com/sonallux/graphql-getting-started-presentation)