

GRAPHQL TUTORIAL

In this series we'll be creating, from scratch, a full-stack application, including a GraphQL server on Node.js, a React front-end (with Apollo) and MongoDB to store all of our data



GraphQL

Server

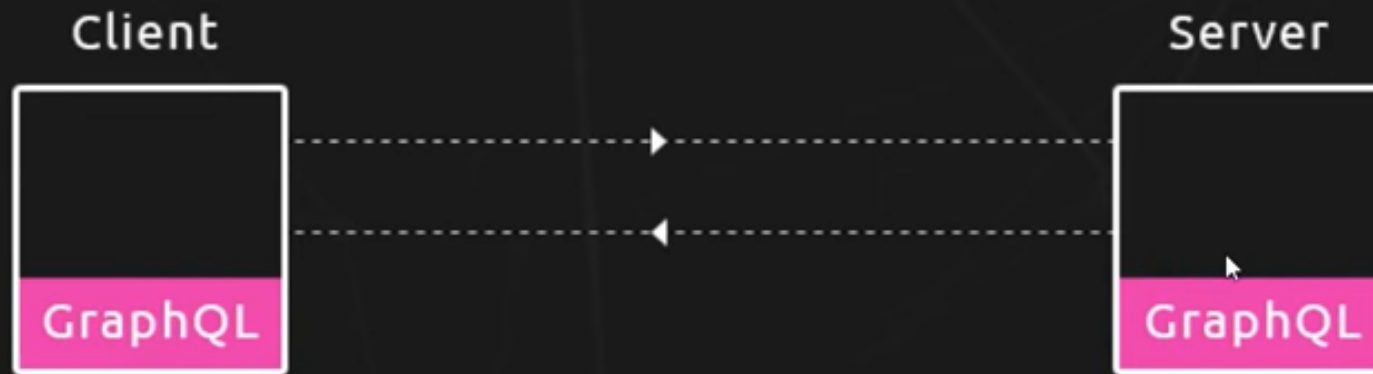


Client



What is GraphQL?

- GraphQL is a powerful query language



- Allows for a more flexible & efficient approach than REST

A RESTful Approach...

- Endpoint for getting a particular book:

`domain.com/books/:id`

title, genre, reviews, authorId

- Endpoint for getting the author info of that book:

`domain.com/authors/:id`

name, age, biography, bookIds

A GraphQL Approach...

- Query to get book data and it's author data (AND the other books):

```
{  
  book(id: 123) {  
    title  
    genre  
    reviews  
    author {  
      name  
      bio  
      books {  
        name  
      }  
    }  
  }  
}
```

A GraphQL Approach...

- Query to get book data and it's author data (AND the other books):

```
{  
  book(id: 123) {  
    title  
    genre  
    reviews  
    author {  
      name  
      bio  
      books {  
        name  
      }  
    }  
  }  
}
```

```
{  
  book(id: 123) {  
    title  
    author {  
      name  
      books {  
        name  
      }  
    }  
  }  
}
```

A BIRD'S EYE VIEW OF GRAPHQL



REST API Endpoints

Front-end application
(browser)



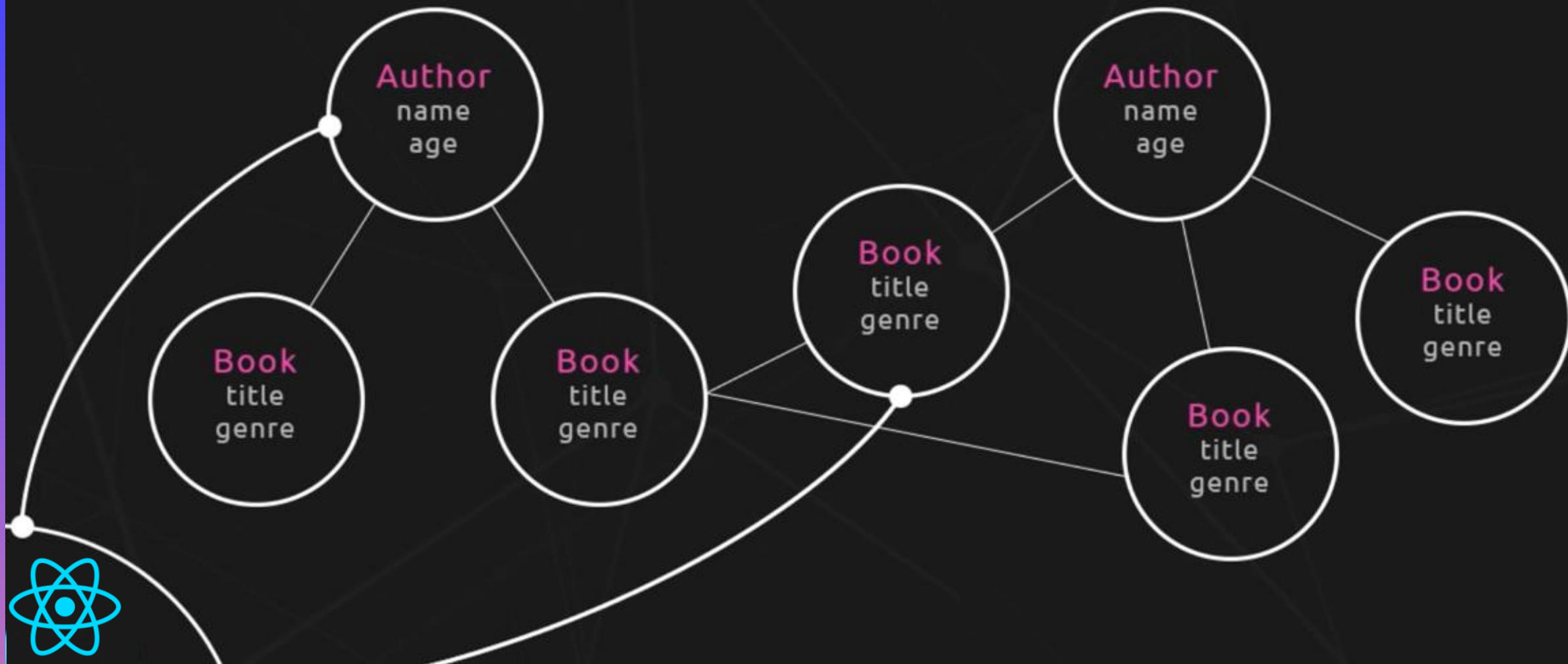
Get all books:
`domain.com/books`

Get a single book:
`domain.com/books/:id`

Get all authors:
`domain.com/authors`

Get a single author:
`domain.com/authors/:id`

Walking the Graph...



Project Overview

```
{  
  book(id: 1) {  
    title  
    genre  
    author {  
      name  
      age  
      books {  
        name  
      }  
    }  
  }  
}
```

Author(2)
name
age

Author(1)
name
age

Book(1)
title
genre

Book(3)
title
genre

Book(2)
title
genre

Book(5)
title
genre

Book(4)
title
genre



Project Overview

```
{  
  Author(id: 2) {  
    name  
    books {  
      name  
      similar {  
        name  
      }  
    }  
  }  
}
```

Author(2)
name
age

Author(1)
name
age

Book(4)
title
genre

Book(5)
title
genre

Book(1)
title
genre

Book(2)
title
genre

Book(3)
title
genre



PROJECT (STACK) OVERVIEW

We'll be using Node on the back-end to create a GraphQL server, and React on the front-end to query the server.



Project Overview

Server (Node.js)

Express App

GraphQL Server



Project Overview

Server (Node.js)

Express App
GraphQL Server

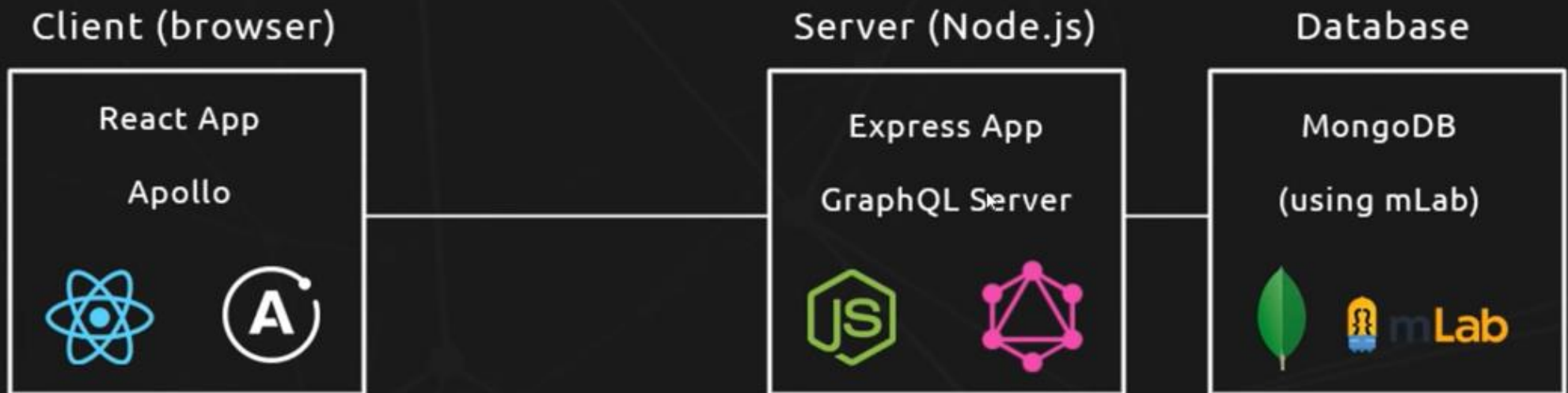


Database

MongoDB
(using mLab)



Project Overview



Project Overview

Client (browser)

React App

Apollo



Server (Node.js)

Express App

GraphQL Server



Database

MongoDB

(using mLab)



Project Overview

Client (browser)



Graphiql

Server (Node.js)



Database

