Code Graphql

Apriorit experts wrote this ebook to share their experience working with microservice architectures. This guide shows you how to speed up microservices development using code generation tools and connect gRPC-based microservices to a GraphQL client.

GraphQL in Action gives you the tools to get comfortable with the GraphQL language, build and optimize a data API service, and use it in a front-end client application. Summary Reduce bandwidth demands on your APIs by getting only the results you need—all in a single request! The GraphQL query language simplifies interactions with web servers, enabling smarter API queries that can hugely improve the efficiency of data requests. In GraphQL in Action, you'll learn how to bring those benefits to your own APIs, giving your clients the power to ask for exactly what they need from your server, no more, no less. Practical and example-driven, this book teaches everything you need to get started with GraphQL—from design principles and syntax right through to performance optimization. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology GraphQL APIs are fast, efficient, and easy to maintain. They reduce app latency and server cost while boosting developer productivity. This powerful query layer offers precise control over API requests and returns, making apps faster and less prone to error. About the book GraphQL in Action gives you the tools to get comfortable with the GraphQL language, build and optimize a data API service, and use it in a front-end client application. By working through set up, security, and error handling you'll learn to create a complete GraphQL server. You'll also unlock easy ways to incorporate GraphQL into your existing codebase so you can build simple, scalable data APIs. What's inside Define a GraphQL schema for relational and document databases Implement GraphQL types using both the schema language and object constructor methods Optimize GraphQL resolvers with data caching and batching Design GraphQL fragments that match UI components' data requirements Consume GraphQL API queries, mutations, and subscriptions with and without a GraphQL client library About the reader For web developers familiar with client-server applications. About the author Samer Buna has over 20 years of experience in software development including front-ends, back-ends, API design, and scalability. Table of Contents PART 1- EXPLORING GRAPHQL 1 Introduction to GraphQL 2 Exploring GraphQL APIs 3 Customizing and organizing GraphQL operations PART 2 - BUILDING GRAPHQL APIs 4 Designing a GraphQL schema 5 Implementing schema resolvers 6 Working with database models and relations 7 Optimizing data fetching 8 Implementing mutations PART 3 - USING GRAPHQL APIs 9 Using GraphQL APIs without a client library 10 Using GraphQL APIs with Apollo client

React helps you create and work on an app in just a few minutes. But learning how to put all the pieces together is hard. How do you validate a form? Or implement a complex multistep user action without writing messy code? How do you test your code? Make it reusable? Wire it to a backend? Keep it easy to understand? The React Cookbook delivers answers fast. Many books teach you how to get started, understand the framework, or use a component library with React, but very few provide examples to help you solve particular problems. This easy-to-use cookbook includes the example code developers need to unravel the most common problems when using React, categorized by topic area and problem. You'll learn how to: Build a single-page application in React using a rich UI Create progressive web applications that users can install and work with offline Integrate with backend services such as REST and GraphQL Automatically test for accessibility problems in your application Secure applications with fingerprints and security tokens using WebAuthn Deal with bugs and avoid common functional and performance problems

This book constitutes the proceedings of the 17th International Conference on Service-Oriented Computing, ICSOC 2019, held in Toulouse, France, in October 2019. The 28 full and 12 short papers presented together with 7 poster and 2 invited papers in this volume were carefully reviewed and selected from 181 submissions. The papers have been organized in the following topical sections: Service Engineering; Run-time Service Operations and Management; Services and Data; Services in the Cloud; Services on the Internet of Things; Services in Organizations, Business and Society; and Services at the Edge.

Jamstack is the new approach to building faster and more secure sites exploring the capabilities of JavaScript. Jumpstart Jamstack Development explores this emerging web development paradigm by building Jamstack websites, from learning Jamstack features to implementing the principles via popular frameworks such as GatsbyJS, Sanity, and Netlify.

"GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API. In this course, we will learn about GraphQL and how we can use it to create truly decoupled client and server. Writing backend with GraphQL makes our code more declarative wherein the client demands certain data in a particular format and the server responds in a predictable manner. This enables us to evolve our API without versions and iterate on features faster. For front-end and networking, we will use Facebook's React and Relay respectively. We will learn GraphQL by creating a backend for a blog platform. You will learn about GraphQL schema design, authentication, pagination, testing, query batching and more and how to connect a GraphQL backend to React and Relay client. The course will teach everything to get up and running with GraphQL and React."--Resource description page.

Nuxt.js is a progressive web framework for adding SSR capabilities to Vue.js apps. This practical guide will help you up and running with the fundamentals of Nuxt.js, how to integrate it with the latest version of Vue.js and enable you to build an entire project including authentication, testing, and deployment with

Nuxt.js and Vue.js.

JavaScript is the little scripting language that could. Once used chiefly to add interactivity to web browser windows, JavaScript is now a primary building block of powerful and robust applications. In this practical book, new and experienced JavaScript developers will learn how to use this language to create APIs as well as web, mobile, and desktop applications. Author and engineering leader Adam D. Scott covers technologies such as Node.js, GraphQL, React, React Native, and Electron. Ideal for developers who want to build full stack applications and ambitious web development beginners looking to bootstrap a startup, this book shows you how to create a single CRUD-style application that will work across several platforms. Explore GraphQL's simple process for querying data Learn about shared authentication for APIs, web apps, and native applications Build performant web applications with React and Styled Components Use React Native to write cross-platform applications for iOS and Android that compile to native code Learn how to write desktop applications with Electron

Want to build APIs like Facebook? Since Facebook's framework for building APIs, GraphQL, has become publicly available, this ambition seems to be within reach for many companies. And that is great. But first, let's learn what GraphQL really is and - maybe even more importantly - let's figure out how to apply GraphQL to build APIs that consumers love. Do you like to learn hands-on? In this book, we take a hands-on approach to learning GraphQL. We first explore the concepts of the two GraphQL languages using examples. Then we start writing some code for our first GraphQL API. We develop this API step by step, from creating a schema and resolving queries, over mocking data and connecting data sources all the way to developing mutations and setting up event subscriptions. Are your API consumers important to you? This book shows you how to apply a consumer-oriented design process for GraphQL APIs, so you can deliver what your consumers really want: an API that solves their problems and offers a great developer experience. Do you want to enable the API consumers so they can build great apps? This book explains the GraphQL query language, which allows the API consumers to retrieve data, write data and get notified when data changes. More importantly, you let them decide, which data they really need from the API. Do you want to make your API easy and intuitive to use? This book shows you how to use the GraphQL schema language to define a type system for your API, which serves as a reference documentation and helps your API consumers write queries that are syntactically correct. Do you want to profit from what has worked for others? This book provides a collection of best practices for GraphQL that have worked for other companies, e.g. regarding pagination, authentication and caching. REST vs. GraphQL: Which one is better? GraphQL and REST are competing philosophies for building APIs. It is not in the scope of this book to compare or discuss the two approaches. The focus of this book is on a hands-on approach for learning GraphQL.

A fast-paced, practical guide to helping you leverage React Testing Library to test the DOM output of components Key Features Get to grips with React Testing Library and create tests that don't break with changes in implementation Learn how to put RTL into practice by implementing it in real-world scenarios Test apps to be more accessible and ensure your tests will work with actual DOM nodes Book Description React Testing Library (RTL) is a lightweight and easy-to-use tool for testing the document object model (DOM) output of components. This book will show you how to use this modern, user-friendly tool to test React components, reducing the risk that your application will not work as expected in production. The book demonstrates code snippets that will allow you to implement RTL easily, helping you to understand the guiding principles of the DOM Testing Library to write tests from the perspective of the user. You'll explore the advantages of testing components from the perspective of individuals who will actually use your components, and use test-driven development (TDD) to drive the process of writing tests. As you advance, you'll discover how to add RTL to React projects, test components using the Context API, and also learn how to write user interface (UI) end-to-end tests using the popular Cypress library. Throughout this book, you'll work with practical examples and useful explanations to be able to confidently create tests that don't break when changes are made. By the end of this React book, you will have learned all you need to be able to test React components confidently. What you will learn Explore React Testing Library and its use cases Get to grips with the RTL ecosystem Apply jest-dom to enhance your tests using RTL Gain the confidence you need to create tests that don't break with changes using RTL Integrate Cucumber and Cypress into your test suite Use TDD to drive the process of writing tests Apply your existing React knowledge for using RTL Who this book is for This book is for software engineers, quality engineers and React developers who want to learn about modern practices used for testing React components using the latest testing tool, RTL. Basic knowledge of React development is required to get the most out of this book.

Our Architect Team has created this Book with Great care and most of the latest technologies are covered One can learn from the questions itself as they are well detailed. THESE CHALLENGES ARE NOT A COLLECTION OF REGULAR INTERVIEW QUESTIONS SCRAPPED FROM WEB Interview Questions from the below Topics. 1. BlockChain 2. Microservices 3. Docker 4. Kubernetes 5. Reactive 6. Spring Boot 7. Apachespark 8. AI-ML-DL 9. JHipster 10. Advanced JDBC 11. Mysql 12. JShell 13. Appium 14. Elastic search 15. Mockito 16. PowerMock 17. Regex 18. MongoDB 19. SQL 20. Redis 21. Generic 22. JDK 23. Scrum – Agile 24. Quantum 25. Serverless 26. Security 27. Android 28. Selenium 29. JWT 30. Hacking 31. Capacity Planning 32. Postman 33. Progressive 34. BDD 35. Swagger 36. Jmeter 37. Logging 38. Concurrency 39. Linux 40. RasperryPI 41. Arduino 42. Terms 43. Charts 44. Tomcat 45. Kotlin 46. Architectures 47. Hibernate 48. GIT 49. Web Development 50. Softwares and Libraries 51. AWS 52. AZURE Functions 53. Maven 54. HyperLedger 55.

HTTP/2 56. WireShark 57. IOT 58. ELK 59. Graffana 60. Wildfly 61. Software Design 62. Jenkins 63. SonarQube 64. Patterns AntiPatterns 65. Famous and Useful Softwares 66. FAAS 67. Quartz

This book is the introduction to Elixir for experienced programmers, completely updated for Elixir 1.6 and beyond. Explore functional programming without the academic overtones (tell me about monads just one more time). Create concurrent applications, butget them right without all the locking and consistency headaches. Meet Elixir, a modern, functional, concurrent language built on the rock-solid Erlang VM. Elixir's pragmatic syntax and built-in support for metaprogramming will make you productive and keep you interestedfor the long haul. Maybe the time is right for the Next Big Thing. Maybe it's Elixir. Functional programming techniques help you manage the complexities of today's real-world, concurrent systems; maximize uptime; and manage security. Enter Elixir, with its modern, Ruby-like, extendable syntax, compile and runtime evaluation, hygienic macro system, and more. But, just as importantly, Elixir brings a sense of enjoyment to parallel,functional programming. Your applications become fun to work with, and the language encourages you to experiment. Part 1 covers the basics of writing sequential Elixir programs. We'll look at the language, the tools, and the conventions. Part 2 uses these skills to start writing concurrent code-applications that use all the cores on your machine, or all the machines on your network! And we do it both with and without OTP. Part 3 looks at the more advanced features of the language, from DSLs and code generation to extending the syntax. This edition is fully updated with all the new features of Elixir 1.6, with a new chapter on structuring OTP applications, and new sections on the debugger, code formatter, Distillery, and protocols. What You Need: You'll need a computer, a little experience with another high-levellanguage, and a sense of adventure. No functional programmingexperience is needed.

With a new generation of services and frameworks, frontend and mobile developers can use their existing skill set to build full stack applications by leveraging the cloud. Developers can build robust applications with production-ready features such as authentication, APIs, data layers, machine learning, chatbots, and AR scenes more easily than ever by taking advantage of these new serverless and cloud technologies. This practical guide explains how. Nader Dabit, developer advocate at Amazon Web Services, shows developers how to build full stack applications using React, AWS, GraphQL, and the Amplify Framework. You’ll learn how to create and incorporate services into your client applications while exploring general best practices, deployment strategies, continuous integration and delivery, and rich media management along the way. Learn how to build applications that solve real problems Understand what is (and is not) possible when using these technologies Examine how authentication works—and learn the difference between authentication and authorization Discover how serverless functions work and why they’re important Use GraphQL in your application—and learn why it’s important Learn how to build full stack applications on AWS

Buku ini membahas tentang pengembangan API Services menggunakan GraphQL di atas framework ASP.NET Core. GraphQL merupakan sebuah bahasa untuk melakukan query atau manipulasi resource atau data terhadap sumber data seperti database. Selain itu graphql juga menyediakan runtime untuk memproses query dan object graphql lainnya. Bahasa dan runtime ini awalnya dibangun oleh facebook pada tahun 2012 untuk digunakan secara internal. Pada tahun 2015 graphql mulai dipublikasikan secara open source untuk umum. GraphQL menyediakan alternatif lain untuk pengembangan api services selain yang berbasiskan REST. Beberapa perusahaan terkemuka termasuk facebook sendiri sudah banyak yang menggunakan graphql seperti github, pinterest, airbnb, paypal, twitter, dan lainnya. Ada beberapa hal yang membedakan graphql dengan rest services diantaranya yaitu: · GraphQL menyediakan data sesuai dengan kebutuhan, tidak kurang dan tidak lebih. Output data properti atau kolom dapat ditentukan secara dinamis, berbeda halnya dengan rest services yang sudah ditentukan data dari kolom apa saja yang akan ditampilkan. · GraphQL menyediakan satu endpoint untuk semua resource atau semua data yang disediakan, berbeda halnya dengan rest api yang menyediakan resource atau data melalui beberapa endpoint atau beberapa controller. · Respon data yang ditampilkan dari graphql secara struktur akan sama dengan struktur data yang direquest oleh client. · GraphQL dapat menyediakan beberapa resource atau data sekaligus dalam satu request. Rest services harus menyediakannya dengan beberapa url. Topik-topik yang dijelaskan dalam buku ini mulai dari penjelasan arsitektur graphql & asp.net core; instalasi library-library yang diperlukan; pembuatan asp.net core web di visual studio; penggunaan ef core untuk akses database sql server & postgresql; penggunaan query dan mutation; penggunaan tools untuk akses graphql; penggunaan logging dan data loader; real-time services dengan subscription; implementasi asp.net core identity; penggunaan jwt authentication; deployment; dan masih banyak lagi yang lainnya. Pembahasan pada buku ini dilakukan secara bertahap, sehingga diharapkan setelah mempelajarinya pembaca akan memiliki skill atau kemampuan yang cukup untuk membangun layanan berbasiskan GraphQL dengan ASP.NET Core sesuai dengan kebutuhan di lapangan. Semoga buku ini dapat menjadi alternatif lainnya sebagai sumber pembelajaran latihan, tutorial, atau referensi bagi mereka yang ingin mempelajari pemrograman backend GraphQL API Services dengan ASP.NET Core yang terhubung ke database SQL Server dan PostgreSQL.

A step-by-step guide that will help you design, develop, scale, and deploy RESTful APIs with TypeScript 3 and Node.js Key Features Gain in-depth knowledge of OpenAPI and Swagger to build scalable web services Explore a variety of test frameworks and test runners such as Stryker, Mocha, and Chai Create a pipeline by Dockerizing your environment using Travis CI, Google Cloud Platform, and GitHub Book Description In the world of web development, leveraging data is the key to developing comprehensive applications, and RESTful APIs help you to achieve this systematically. This book will guide you in designing and developing web services with the power of TypeScript 3 and Node.js. You'll design REST APIs using best practices for request handling, validation, authentication, and authorization. You'll also understand how to enhance the capabilities of your APIs with ODMs, databases, models and views, as well as asynchronous callbacks. This book will guide you in securing your environment by testing your services and initiating test automation with different testing approaches. Furthermore, you'll get to grips with developing secure, testable, and more efficient code, and be able to scale and deploy TypeScript 3 and Node.js-powered RESTful APIs on cloud platforms such as the Google Cloud Platform. Finally, the book will help you explore microservices and give you an overview of what GraphQL can allow you to do. By the end of this book, you will be able to use RESTful web services to create your APIs for mobile and web apps and other platforms. What you will learn Explore various methods to plan your services in a scalable way Understand how to handle different request types and the response status code Get to grips with securing web services Delve into error handling and logging your web services for improved debugging Uncover the microservices architecture and GraphQL Create automated CI/CD pipelines for release and deployment strategies Who this book is for If you’re a developer who has a basic understanding of REST concepts and want to learn how to design and develop RESTful APIs, this book is for you. Prior knowledge of TypeScript will help you make the most out of this book.

Many companies, from startups to Fortune 500 companies alike, use Node.js to build performant backend services. And engineers love Node.js for its approachable API and familiar syntax. Backed by the world's largest package repository, Node's enterprise foothold is only expected to grow. In this hands-on guide, author Thomas Hunter II proves that Node.js is just as capable as traditional enterprise platforms for building services that are observable, scalable, and resilient. Intermediate to advanced Node.js developers will find themselves integrating application code with a breadth of tooling from each layer of a modern service stack. Learn why running redundant copies of the same Node.js service is necessary Know which protocol to choose, depending on the situation Fine-tune your application containers for use in production Track down errors in a distributed setting to determine which service is at fault Simplify app code and increase performance by offloading work to a reverse proxy Build dashboards to monitor service health and throughput Find out why so many different tools are required when operating in an enterprise environment

This book discusses the development of API Services using GraphQL on top of the ASP.NET Core framework. GraphQL is a language for querying or manipulating resources or data against data sources such as databases. In addition, graphql also provides a runtime for processing queries and other graphql objects. The language and runtime were originally built by Facebook in 2012 to be used internally. In 2015 graphql began to be published in open source to the public. GraphQL provides another alternative for developing api services besides those based on REST. Several leading companies including Facebook itself already use graphql such as github, pinterest, airbnb, paypal, twitter, and others. There are several things that distinguish graphql from rest services, including: - GraphQL provides data according to your needs, nothing less and nothing more. Output data properties or columns can be determined dynamically, in contrast to rest services that have determined data from any column to be displayed. - GraphQL provides one endpoint for all resources or all data provided, in contrast to rest api that provides resources or data through multiple endpoints or multiple controllers. - The data response displayed from graphql will be structurally the same as the data structure requested by the client. - GraphQL can provide multiple resources or data in one request. Rest services should provide it with multiple urls. The topics described in this book start from the explanation of graphql & asp.net core architecture; installation of the necessary libraries; creation of asp.net core web in visual studio; use of ef core for database access sql server & postgresql; use of queries and mutations; use of tools for graphql access; use of logging and data loaders; real-time services with subscription; implementation of asp.net core identity; use of jwt authentication; deployment; and many more. The discussion in this book is carried out in stages, so that it is hoped that after studying it, readers will have sufficient skills or abilities to build GraphQL-based services with ASP.NET Core according to the needs in the field. Hopefully this book can be another alternative as a learning resource for exercises, tutorials, or a reference for those who want to learn the backend programming of GraphQL API Services with ASP.NET Core that is connected to SQL Server and PostgreSQL databases.

While GraphQL is a technology mainly driven by the needs of clients, there is a clear lack of resources on how to build reliable GraphQL servers. Over the last few years, I helped build and maintain some of the biggest GraphQL APIs out there at both Shopify and GitHub. During those years, I also worked with various companies with their adoption of GraphQL. From my experiences with GraphQL, I’ve observed the good, the bad, and the ugly. This led to many talks and blog posts on the subject, but still found that teams and individuals willing to use GraphQL in a pragmatic way lacked the resources to do so. This is what lead me to write this book: A collection of learnings and good practices when building GraphQL schemas at scale. Every language and every GraphQL implementation does things slightly differently. This book is completely language agnostic and instead focuses on concepts and patterns that are achievable no matter how you're building a GraphQL server. Think of it as a complete journey of what goes into building a GraphQL API, from design, to architectures, to implementation, and even documentation.

Build a variety of cross-platform applications with the world's most complete full-stack JavaScript framework— Meteor About This Book Develop a set of real-world applications each exploring different features of Meteor Make your app more appealing by adding reactivity and responsiveness to it Work with the most powerful feature of Meteor—the “full stack reactivity”—through building real-time applications with many third party libraries Who This Book Is For If you are a developer who is looking forward to taking your application development skills with Meteor to next level by getting your hands-on different projects, this book is for you. What You Will Learn See how Meteor fits in the modern web application development by using its reactive data system Make your front-end behave consistently across environments by implementing a predictable state container with Redux Get familiar with React and overview of Angular 2 Add a map to your application with a real-time geolocation Plugin into Meteor social media APIs like Twitter's streaming and Facebook's Messenger Add search functionality from scratch to your existing app and data Add responsiveness with Bootstrap 4 and Google's Material Design using Less and Sass Distribute your data across machines and data centers by adding Apache Cassandra to your existing stack. Learn how to scale your microservices with the high performant language neutral framework gRPC. Learn how to query multiple data sources using GraphQL. In Detail This book starts with the basic installation and overview of the main components in Meteor. You'll get hands-on multiple versatile applications covering a wide range of topics from adding a front-end views with the hottest rendering technology React to implementing a microservices oriented architecture.All the code is written with ES6/7 which is the latest significantly improved JavaScript language. We'll also look at real-time data streaming, server to server data exchange, responsive styles on the front-end, full-text search functionality, and integration of many third-party libraries and APIs using npm. By the end of the book, you'll have the skills to quickly prototype and even launch your next app idea in a matter of days. Style and Approach This book takes an easy-to-follow project-based approach. Each project starts with the goal of what you will learn and an overview the technologies used.

The Road to GraphQL is your personal journey to master pragmatic GraphQL in JavaScript. The book is full with applications you are going to build along the way with React.js and Node.js. Afterward, you will be able to implement full-stack JavaScript applications. I wrote the The Road to GraphQL over the last year, while building several GraphQL applications for my clients and for myself. During this time, I came to understand the practical genius of GraphQL, and how it dramatically improves communication in client-server architectures. Not only does it improve the interface between the client and the server, it also enhances client-side state management by eliminating remote data management. Sophisticated GraphQL libraries like Apollo Client provide powerful features like caching, optimistic UI, and data prefetching for free. This book covers the fundamentals of GraphQL itself, as well as its ecosystem. I applied the same principles as my other books: Stay pragmatic Keep it simple Answer the why, not just the how Experience a problem, solve a problem This book is not intended to be an end-all reference for GraphQL APIs, nor an in-depth guide about the internals of the GraphQL specification. Instead, its purpose is to journey through learning GraphQL with JavaScript the pragmatic way, building client and server applications yourself. The book covers lots of facets about GraphQL in JavaScript that are important for building modern applications, without just throwing the libraries like Apollo at problems before experiencing them. It starts with the basic HTTP requests to perform GraphQL queries first, then moves on to using dedicated GraphQL libraries for it. You will even get the chance to implement your own GraphQL client library, so you understand how these libraries work under the hood. There are no hidden abstractions in this book, just plenty of fundamentals for JavaScript, React.js, Node.js, and GraphQL. Requirements To get the most out of this book, you should be familiar with the basics of web development, which includes some knowledge of HTML, CSS and JavaScript. You will also need to be familiar with the term API, because they are discussed frequently. I encourage you to join the official Slack Group for the book, help or get help from others. React On the client-side, this book uses React to teach about GraphQL in JavaScript. My other book called The Road to learn React teaches you all the fundamentals about React. It also teaches you to make the transition from JavaScript ES5 to JavaScript ES6. The book is available for free and after having read the Road to learn React, you should possess all the knowledge to implement the GraphQL client-side application with this book. Node On the server-side, this book uses Node with Express as library to teach about GraphQL in JavaScript. You don't need to know much about those technologies before using them for your first GraphQL powered applications. The book will guide you through the process of setting up a Node application with Express and shows you how to weave GraphQL into the mix. Afterward, you should be able to consume the GraphQL API provided by your server-side application in your client-side application.

A practical course to get you up to speed with the key aspects of GraphQL, including queries, mutations, scalar types, image management, authentication, and authorization About This Video Learn the key aspects of GraphQL and create sample applications Explore how to create custom scalars, how authentication and authorization work in GraphQL, and much more

In Detail Have you heard about GraphQL? Have you always wanted to understand GraphQL? If yes, this course will help you learn about GraphQL in a hands-on manner. As you progress through the course, you will acquire the skills necessary to understand and work with GraphQL's basic as well as advanced features, including but not limited to the following: Understand basic scalar types Create custom scalars Become familiar with resolvers and type definitions (Schemas) Understand the importance of a data model in GraphQL Explore how GraphQL queries work, including parameterized queries Gain working knowledge of GraphQL mutations Get to grips with how file uploads work using GraphQL Learn how to serve images in the GraphQL context Study how authentication and authorization work in GraphQL You'll create four sample applications: A basic application to run GraphQL queries An advanced application that uses React to work with GraphQL Enhance the existing application and extend it with authentication and authorization An application to understand how to serve/display images and upload files using only GraphQL In this course, you'll mostly use Apollo GraphQL services for building your apps. Familiarity with React is required to get started with this course. You'll use React to create several applications; however, it only covers the React parts that are relevant to the context. Downloading the example code for this course: You can download the example code files for this course on GitHub at the following link: https://github.com/fullstacktraining/Practical-GraphQL-Become-a-GraphQL-Ninja . If you require support please email: customercare@packt.com.

Combining React, one of the most widely used JavaScript frameworks, and GraphQL, the modern way of querying an API, two revolutionary technologies will give you a future-proof and scalable stack you can start building your business around. This book will guide you in implementing applications by using React, Apollo, Node.js and SQL.

Get the definitive guide on Gatsby, the JavaScript framework for building blazing fast websites and applications. Used by Nike, Costa Coffee, and other companies worldwide, Gatsby is emerging as one of the key technologies in the Jamstack (JavaScript, APIs, and markup) ecosystem. With this comprehensive guide, you'll learn how to architect, build, and deploy Gatsby sites independently or with CMSs, commerce systems, and other data sources. Author Preston So begins by showing you how to set up a Gatsby site from scratch. From there, youâ??ll learn ways to use Gatsbyâ??s declarative rendering and GraphQL API, build complex offline-enabled sites, and continuously deploy Gatsby sites on a variety of platforms, including Gatsby Cloud. Discover how Gatsby integrates with many data sources and plug-ins Set up, configure, and architect Gatsby sites using Gatsby's CLI, React, JSX, and GraphQL with high performance out of the box Build an independent Gatsby site based on Markdown and data- and content-driven Gatsby sites that integrate with CMSs and commerce platforms Deploy Gatsby sites with full CI/CD and test coverage on a variety of platforms, including Netlify, Vercel, and Gatsby Cloud

Gain all the essentials you need to create scalable microservices, which will help you solve real challenges when deploying services into production. This book will take you through creating a scalable data layer with polygot persistence. You’ll cover data access and query patterns in Spring and JPA in high-performance environments. As part of this topic, you’ll see the advantages of multiple persistence frameworks in Java and especially the easy persistence offered by NoSQL databases and reactive web solutions. The last few chapters present advanced concepts that are useful for very high-performance real-time applications: you’ll implement applications using Spring’s good support for Web sockets in their raw form as well as for connecting to message brokers such as RabbitMQ. This can be useful for applications such as navigation systems and gaming platforms. What You Will Learn Build end-to-end modern applications using microservices, persistence essentials, reactive web, and other high-performance concepts Master Spring’s configuration options Secure microservices efficiently Monitor your services post deployment Who This Book Is For Java developers and architects interested in microservices.

This is a book for developers, who not only want to learn how to develop software for Alexa but also want to make money with Alexa. Want to start a side business or a SaaS startup? Just as in the early days of mobile, when fortunes were made with mobile apps on the app store, it is now the perfect time to catch the opportunities offered by voice apps. Amazon Alexa, the voice platform with the broadest adoption, helps developers like you and me, to develop, distribute, market and monetize their Alexa Skills on the Amazon Alexa Store. Want to develop and program Alexa Skills? In this book, you learn step-by-step how to create your first Alexa Skill with the Alexa Developer Console, AWS Lambda, the Alexa CLI, and node.js with the Alexa SDK. Want to scale and grow your Alexa Software Startup? You get a deep-dive into the various ways of making money with Alexa. You learn about the business models for Alexa Skills, marketing and monetizing your Alexa Skill on and off the Alexa Store, opportunities for offering in-skill purchases, and about programming the various purchase and payment flows. Want to build advanced Alexa Skills that users love? The book covers many advanced features of Alexa in plain English, such as account linking, audio streaming, session management and much more. You learn how to personalize your Skill with the user's data and linking the Skill to popular cloud apps, such as Spotify, Google and many more. This will help you create unique apps that stand out on the market and improve the lives of many Alexa users.

Healthcare systems have been in a state of flux for a number of years now due to increasing digitalization. Medicine itself is also facing new challenges, and how to maximize the possibilities of artificial intelligence, whether digitalization can help to strengthen patient orientation, and dealing with the issue of data quality and completeness are all issues which require attention, creativity and research. This book presents the proceedings of the 64th annual conference of the German Association for Medical Informatics, Biometry and Epidemiology (GMDS 2019), held in Dortmund, Germany, from 8 - 11 September 2019. The theme of this year’s conference is Shaping Change – Creative Solutions for Innovative Medicine, and the papers presented here focus on active participation in shaping change while ensuring that good scientific practice, evidence and regulation are not lost as a result of innovation. The book is divided into 8 sections: biostatistics; healthcare IT; interoperability - standards, classification, terminology; knowledge engineering and decision support; medical bioinformatics and systems biology; patient centered care; research infrastructure; and sociotechnical systems / usability and evaluation of healthcare IT. The book will be of interest to all those facing the challenges posed by the ongoing revolution in medicine and healthcare.

A developer's guide to designing, testing, and securing production-ready modern APIs with the help of practical ideas to improve your application's functionality Key Features Build resilient software for your enterprises and customers by understanding the complete API development life cycle Overcome the challenges of traditional API design by adapting to a new and evolving culture of modern API development Use Spring and Spring Boot to develop future-proof scalable APIs Book Description The philosophy of API development has evolved over the years to serve the modern needs of enterprise architecture, and developers need to know how to adapt to these modern API design principles. Apps are now developed with APIs that enable ease of integration for the cloud environment and distributed systems. With this Spring book, you'll discover various kinds of production-ready API implementation using REST APIs and explore async using the reactive paradigm, gRPC, and GraphQL. You'll learn how to design evolving REST-based APIs supported by HATEOAS and ETAGs and develop reactive, async, non-blocking APIs. After that, you'll see how to secure REST APIs using Spring Security and find out how the APIs that you develop are consumed by the app's UI. The book then takes you through the process of testing, deploying, logging, and monitoring your APIs. You'll also explore API development using gRPC and GraphQL and design modern scalable architecture with microservices. The book helps you gain practical knowledge of modern API implementation using a sample e-commerce app. By the end of this Spring book, you'll be able to develop, test, and deploy highly scalable, maintainable, and developer-friendly APIs to help your customers to transform their business. What you will learn Understand RESTful API development, its design paradigm, and its best practices Become well versed in Spring's core components for implementing RESTful web services Implement reactive APIs and explore async API development Apply Spring Security for authentication using JWT and authorization of requests Develop a React-based UI to consume APIs Implement gRPC inter-service communication Design GraphQL-based APIs by understanding workflows and tooling Gain insights into how you can secure, test, monitor, and deploy your APIs Who this book is for This book is for inexperienced Java programmers, comp science, or coding boot camp graduates who have knowledge of basic programming constructs, data structures, and algorithms in Java but lack the practical web development skills necessary to start working as a developer. Professionals who've recently joined a startup or a company and are tasked with creating real-world web APIs and services will also find this book helpful. This book is also a good resource for Java developers who are looking for a career move into web development to get started with the basics of web service development.

This book gets you a running start with serverless GraphQL APIs on Amazon's AWS AppSync. Whether you are new to GraphQL, or you are an experienced GraphQL developer, this book will provide you with the knowledge needed to get started with AWS AppSync. Do you like learning by doing? After quickly covering the GraphQL foundations, you will dive into the practice of developing APIs with AWS AppSync with in-depth walkthroughs, screenshots, and code samples. Do I learn everything I need to get started? The book guides you through the step-by-step process of designing GraphQL APIs: creating a GraphQL schema, developing GraphQL APIs, connecting data sources, developing resolvers with AppSync templates, securing your API, offering real-time data, developing offline support and synchronization for your apps and much more. Why GraphQL? GraphQL is now a viable option for modern API design. And since Facebook, Yelp, and Shopify have built successful APIs with GraphQL, many companies consider following in the technological footsteps of these tech giants. Using GraphQL is great, but by itself, it is only half the rent: It requires the manual installation and maintenance of software infrastructure components. Why Serverless GraphQL with AppSync? AppSync is a cloud-based platform for GraphQL APIs. It is serverless, so you waste no time setting up infrastructure. It scales up and down dynamically depending on the load. It supports your app developers with an SDK for synchronization and offline support. You pay only what you use, so no upfront investment is needed and it may save your organizations thousands of dollars in IT costs.

This two-volume set LNCS 10924 and 10925 constitute the refereed proceedings of the 5th International Conference on Learning and Collaboration Technologies, LCT 2018, held as part of the 20th International Conference on Human-Computer Interaction, HCII 2018, in Las Vegas, NV, USA in July 2018. The 1171 papers presented at HCII 2018 conferences were carefully reviewed and selected from 4346 submissions. The papers cover the entire field of human-computer interaction, addressing major advances in knowledge and effective use of computers in a variety of applications areas. The papers in this volume are organized in the following topical sections: designing and evaluating systems and applications, technological innovation in education, learning and collaboration, learners, engagement, motification, and skills, games and gamification of learning, technology-enhanced teaching and assessment, computing and engineering education.?

Your domain is rich and interconnected, and your API should be too. Upgrade your web API to GraphQL, leveraging its flexible queries to empower your users, and its declarative structure to simplify your code. Absinthe is the GraphQL toolkit for Elixir, a functional programming language designed to enable massive concurrency atop robust application architectures. Written by the creators of Absinthe, this book will help you take full advantage of these two groundbreaking technologies. Build your own flexible, high-performance APIs using step-by-step guidance and expert advice you won't find anywhere else. GraphQL is a new way of structuring and building web services, and the result is transformational. Find out how to offer a more tailored, cohesive experience to your users, easily aggregate data from different data sources, and improve your back end's maintainability with Absinthe's declarative approach to defining how your API works. Build a GraphQL-based API from scratch using Absinthe, starting from core principles. Learn the type system and how to expand your schema to suit your application's needs. Discover a growing ecosystem of tools and utilities to understand, debug, and document your API. Take it to production, but do it safely with solid best practices in mind. Find out how complexity analysis and persisted queries can let you support your users flexibly, but responsibly too. Along the way, discover how Elixir makes all the difference for a high performance, fault-tolerant API. Use asynchronous and batching execution, or write your own custom add-ons to extend Absinthe. Go live with subscriptions, delivering data over websockets on top of Elixir (and Erlang/OTP's) famous solid performance and real-time capabilities. Transform your applications with the powerful combination of Elixir and GraphQL, using Absinthe. What You Need: To follow along with the book, you should have Erlang/OTP 19+ and Elixir 1.4+ installed. The book will guide you through setting up a new Phoenix application using Absinthe.

React is an open-source JavaScript library that is used for building user interfaces or UI components. This React book is designed to take you through the most valuable design patterns in React, helping you learn how to apply design patterns and best practices in real-world scenarios.

Get an introduction to the visual design of GraphQL data and concepts, including GraphQL structures, semantics, and schemas in this compact, pragmatic book. In it you will see simple guidelines based on lessons learned from real-life data discovery and unification, as well as useful visualization techniques. These in turn help you improve the quality of your API designs and give you the skills to produce convincing visual communications about the structure of your API designs. Finally, Visual Design of GraphQL Data shows you how to handle GraphQL with legacy data as well as with Neo4j graph databases. Spending time on schema quality means that you will work from sharper definitions, which in turn leads to greater productivity and well-structured applications. What You Will Learn Create quality GraphQL data designs Avoid structural mistakes Draw highly communicative property graph diagrams of your APIs Who This Book Is For Web developers and data architects who work with GraphQL and other APIs to build modern applications.

Build data-intensive React applications with ease using GraphQL and RelayAbout This Book* Take your React applications to the next level with GraphQl and Relay* Make your React applications speak to the back end* Explore the concepts of data fetching, data handling, and more* Build data-intensive applications* Learn to deploy your applications, which are compliant with GraphQl and RelayWho This Book Is ForThis book is perfect for those who have no prior experience or familiarity with Relay and/or GraphQL. You should be comfortable writing NodeJS applications on a MongoDB database with REST APIs as well as applications on the client-side using React and ES2015.What You Will Learn* Understand what problem GraphQL solves and how it differs from traditional REST architectures* Create a GraphQL server* Develop modular and maintainable GraphQL code* Grasp how to define GraphQL mutations* Comprehend how to define GraphQL queries* Modify a GraphQL server to be Relay-compliant* Develop a Relay client-side application with ReactJS that consumes a GraphQL endpointIn DetailThe book begins by building a solid foundation in the GraphQl specification (defining and executing data queries, and creating data sources) and then swiftly moves to how a data layer can be implemented for your web application using Relay.For years, REST has been the de facto choice for implementing APIs and while REST is a great option, it's not without its problems. Facebook solved many of these problems by developing the GraphQL specification and made it open source along with RelayJS, a React library for querying a server that implements the GraphQL specification. This book will take you from having no prior experience or just a familiarity with Relay and/or GraphQL to being able to build a production-ready application using both.One of the major challenges faced by React is data initialization, which is solved by GraphQl and Relay. While GraphQL is a data querying specification, Relay is the data fetching library that implements communicating with the GraphQl server to build data-intensive applications. We cover various concepts such as creating data containers, data masking, and more. Towards the end of the book, we will empower you to build a production-ready application.. You will also understand how you can deploy your GraphQl/Relay application on Heroku.By the end of the book, you'll have gained enough experience building an app with GraphQL and Relay to feel comfortable and to obtain mastery.

As the software industry continues to evolve, professionals are continually searching for practices that can assist with the various problems and challenges in information technology (IT). Agile development has become a popular method of research in recent years due to its focus on adapting to change. There are many factors that play into this process, so success is no guarantee. However, combining agile development with other software engineering practices could lead to a high rate of success in problems that arise during the maintenance and development of computing technologies. Software Engineering for Agile Application Development is a collection of innovative research on the methods and implementation of adaptation practices in software development that improve the quality and performance of IT products. The presented materials combine theories from current empirical research results as well as practical experiences from real projects that provide insights into incorporating agile qualities into the architecture of the software so that the product adapts to changes and is easy to maintain. While highlighting topics including continuous integration, configuration management, and business modeling, this book is ideally designed for software engineers, software developers, engineers, project managers, IT specialists, data scientists, computer science professionals, researchers, students, and academics.

Create future-proof API interfaces via GraphQL through practical React projects About This Video Build GraphQL projects from scratch using your prior React knowledge. Simplify the state management layer of your React application by removing all the boilerplate. Use the latest version of React with hooks and Apollo client(2.5) with added support for hooks. In Detail Are you a frontend developer looking for a simpler way to interact with the server side? As SPA applications become increasingly more complex, it's critical to be able to ease the load on your end user devices without sacrificing user experience for other clients of your API. This course will teach you GraphQL design principles and what makes it different from REST while creating real-world projects. You will build a job board application using Apollo Client for React and know how to use it. Create a chat application and dive deep into its advanced topics such as Authentication/Authorization and Real-time Subscriptions. In the last application we will look at how you can use Apollo client in combination with REST APIs. By the end of the course, you will know how to create real world GraphQL effectively and apply those skills at your job. Downloading the example code for this course: You can download the example code files for this course on GitHub at the following link: https://github.com/PacktPublishing/GraphQL-Projects . If you require support please email: customercare@packt.com.

REST architecture (style) is a pivot of distributed systems, simplify data integration amongst modern and legacy applications leverages through the RESTful paradigm. This book is fully loaded with many RESTful API patterns, samples, hands-on implementations and also discuss the capabilities of many REST API frameworks for Java, Scala, Python and Go

This book constitutes the refereed proceedings of the 18th International Conference on Web Engineering, ICWE 2018, held in Cáceres, Spain, in June 2018. The 18 full research papers and 17 short papers presented together with 2 practice papers, 6 demonstration papers, and 5 tutorials were carefully reviewed and selected from 103 submissions. The papers cover research areas such as Web application modeling and engineering; Web infrastructures and architectures; execution models; human computation and crowdsourcing applications; Web application composition and mashups; Social Web applications; Semantic Web applications; Web of Things applications; big data and data analytics; and security, privacy, and identity.
TypeScript is among the fastest-growing languages, helping developers build full-stack apps by integrating with powerful frameworks such as React and Node.js. With this book, you'll get started with TypeScript and build an SPA with React and middleware using Node.js and Express. Finally, you'll be able to package your web app and deploy it on AWS.
"How to Get a Job in Web Development" is designed for junior web developers. Whether you're coming from a coding bootcamp, are completely self-taught, or graduated from college with a tech-related degree, this book is for you. Written by RealToughCandy. In this book, you will learn how to: • Expertly craft the 'holy clover' of application materials: your resume, cover letter, GitHub page, and portfolio. • Leverage the power of LinkedIn, Meetups, and social media. • Handle follow-up emails and phone calls. • Prepare for the multiple types of interviews you will encounter, whether via phone, video conference, or in person. • Strategically apply to jobs so you can maximize your salary demands during negotiation. • Efficiently organize and prioritize the jobs you've applied to. • Craft results-driven email check-ins with your potential employer. • Reduce your vulnerabilities for discrimination. • And much, much more! No awkward whiteboard interviews. No hour-long explanation of Big O notation. Just practical, actionable steps that will put you far ahead of the pack when it comes to getting a job in web development. Now let's go get that job! "Just finished reading your book and all I can say is WOW! Mind you since May of 2016 I have taken about 6 online courses specifically looking for employment and around three of them were specifically for either how to get an IT or Web Developer job. These courses cannot hold a candle to the majority of the information you put in this book!" -George M., Web Developer WHY I WROTE THIS BOOK: When I started my web development journey, I was a lost hiker in the digital woods. I knew I wanted to build web apps, but didn't know what those people called themselves. Were they website builders? Programmers? The term 'software engineer' floated around a lot online – was that my aspiration? Since I didn't know exactly what I was looking for, I spent a lot of time reading and watching materials that were nothing but discouraging: mock Google coding interviews with whiteboards and markers. Lots of articles and videos that name-dropped things like binary trees, Big O notation, and time complexity. Forum post upon forum post that gave away actual coding interview questions from the biggest tech companies in the world like Facebook, Google, and Microsoft. Making things worse, some web developers I had discovered on YouTube were talking about a really good, popular book for coding interviews. I checked it out and once again my stomach sank. "I'm never going to make it in this field," I said to myself. "I've been studying and practicing and building projects for months, and I still have no idea what these people are talking about." What they didn't tell me was that the book is geared towards senior software engineers trying to get a job with Amazon and Google. I wanted to quit my coding journey. In fact, I did quit. The difference was, I didn't stay quit. Something told me to keep pushing forward, keep building projects to put in my portfolio and Github, keep reaching out and trying to find clients who needed websites. I kept pushing until I got a job as a fullstack web developer at a data company. As it turns out, the internet isn't very generous to our career field. Beginners are especially marginalized. There aren't any quality one-stop resources for discovering one of the most important questions – if not the most important question – web developers have. "How do I get a job in this field?" I wanted to change the

junior web developer tech landscape with this book. My goal is for every junior developer who reads this to find a job. And if you take the recommended actions in this book, you can do it.