

Restful Api Design Best Practices In Api Design With Rest Api University Series Book 3

"RESTful Web APIs allow developers to create unprecedented applications by leveraging the data on the internet. Since JavaScript is the language of the web, building APIs using Node.js provides a seamless development experience on both the front end and the back end. This video course gives you an overview of a RESTful API and goes through the logical steps of building one. It explores three different APIs, focusing on their similarities and differences to effectively implement one. We'll start off by defining APIs, showing you how they can be built on top of HTTP, and listing the properties that make an API RESTful. We will develop Twitter Notes, a web application that lets its users leave notes for their Twitter friends. We will use Twitter's API to implement a login flow and then design a web API. In addition to using Twitter's API, we will take a closer look at two other real-world APIs--Facebook API and GitHub API. Finally, we'll look at some best practices to keep the APIs secure, maintainable, and performing. By the end of this course, you will have a good grasp of APIs, HTTP, REST, OAuth 1.0a, API testing, and site reliability, performance, and security. Since the course explores three different REST APIs, you will reach a level where you will be comfortable using any RESTful API, even if it does not have an SDK."--Resource description page.

This book constitutes the refereed proceedings of the 16th International Conference on Web Engineering, ICWE 2016, held in Lugano, Switzerland, in June 2016. The 19 full research papers, 13 short papers, 3 vision papers, 11 demonstrations, 5 posters, 6 PhD Symposium and 4 tutorials presented were carefully reviewed and selected from 120 submissions. The 16th edition of ICWE accepted contributions related to different research areas revolving around Web engineering, including: Web application modelling and engineering, Human computation and crowdsourcing, Web applications composition and mashups, SocialWeb applications, SemanticWeb, and, for the first time, also the Web of Things.

Maximize the impact of your assets and business services by providing APIs for developers and other users. The journey described in this book starts with identifying business assets. As part of the API team, you then need to identify and define the requirements of traffic management, security, mediation, and orchestration. You also must define metrics for the analytics to measure the success of the overall API program. API documentation and the ease of developer onboarding also determine the success of the APIs. Finally, monetization of these APIs leads to revenue generation for the enterprise. Author De — an expert in building and managing API solutions — provides enterprise architects, designers, and technologists with insight into the world of APIs and the various technical aspects of building and managing an effective API management solution. API Management: Developing and Managing APIs for your Organization: Introduces the basics of APIs and highlights their value Provides an overview of technologies for building an API management solution and defines the requirements, including how to build a RESTful API Offers design principles for building developer-friendly APIs Explains how to secure your APIs Shows how to use API analytics to measure the success of your APIs Demonstrates how to monetize APIs Finally, API Management touches on various technical nuances of creating, distributing, and managing an API. This book will not only help you learn how to design, build, deploy, and manage an API for an enterprise scale, but also generate revenue for your organization. What You'll Learn Discover the API life cycle Design and develop APIs Implement API security Test your APIs Deploy and monitor your APIs Who This Book Is For Enterprise architects, technology enthusiasts, security architects, and operations specialists.

Build and deploy scalable cloud native microservices using the Spring framework and Kubernetes. KEY FEATURES ? Complete coverage on how to design, build, run, and deploy modern cloud native microservices. ? Includes numerous sample code exercises on microservices, Spring and Kubernetes. ? Develop a stronghold on Kubernetes, Spring, and the microservices architecture. ? Complete guide of application containerization on Kubernetes containers. ? Coverage on managing modern applications and infrastructure using observability tools. DESCRIPTION The main objective of this book is to give an overview of cloud native microservices, their architecture, design patterns, best practices, real use cases and practical coverage of modern applications. This book covers a strong understanding of the fundamentals of microservices, API first approach, Testing, observability, API Gateway, Service Mesh and Kubernetes alternatives of Spring Cloud. This book covers the implementation of various design patterns of developing cloud native microservices using Spring framework docker and Kubernetes libraries. It covers containerization concepts and hands-on lab exercises like how to build, run and manage microservices applications using Kubernetes. After reading this book, the readers will have a holistic understanding of building, running, and managing cloud native microservices applications on Kubernetes containers. WHAT YOU WILL LEARN ? Learn fundamentals of microservice and design patterns. ? Learn microservices development using Spring Boot and Kubernetes. ? Learn to develop reactive, event-driven, and batch microservices. ? Perform end-to-end microservices testing using Cucumber. ? Implement API gateway, authentication & authorization, load balancing, caching, rate limiting. ? Learn observability and monitoring techniques of microservices. WHO THIS BOOK IS FOR This book is for the Spring Developers, Microservice Developers, Cloud Engineers, DevOps Consultants, Technical Architect and Solution Architects, who have some familiarity with application development, Docker and Kubernetes containers. TABLE OF CONTENTS 1. Overview of Cloud Native microservices 2. Microservice design patterns 3. API first approach 4. Build microservices using the Spring Framework 5. Batch microservices 6. Build reactive and event-driven microservices 7. The API gateway, security, and distributed caching with Redis 8. Microservices testing and API mocking 9. Microservices observability 10. Containers and Kubernetes overview and architecture 11. Run microservices on Kubernetes 12. Service Mesh and Kubernetes alternatives of Spring Cloud

Looking for Best Practices for RESTful APIs? This book is for you! Why? Because this book is packed with practical experience on what works best for RESTful API Design. You want to design APIs like a Pro? Use API description languages to both design APIs and develop APIs efficiently. The book introduces the two most common API description languages RAML, OpenAPI, and Swagger. Your company cares about its customers? Learn API product management with a customer-centric design and development approach for APIs. Learn how to manage APIs as a product and how to follow an API-first approach. Build APIs your customers love! You want to manage the complete API lifecycle? An API development methodology is proposed to guide you through the lifecycle: API inception, API design, API development, API publication, API evolution, and maintenance. You want to build APIs right? This book shows best practices for REST design, such as the correct use of resources, URIs, representations, content types, data formats, parameters, HTTP status codes, and HTTP methods. Your APIs connect to legacy systems? The book shows best practices for connecting APIs to existing backend systems. Your APIs connect to a mesh of microservices? The

book shows the principles for designing APIs for scalable, autonomous microservices. You expect lots of traffic on your API? The book shows you how to achieve high performance, availability and maintainability. You want to build APIs that last for decades? We study API versioning, API evolution, backward- and forward-compatibility and show API design patterns for versioning. The API-University Series is a modular series of books on API-related topics. Each book focuses on a particular API topic, so you can select the topics within APIs, which are relevant for you.

Learn to rapidly build and deploy cross-platform applications from a single codebase with practical, real-world solutions using the mature Delphi 10.4 programming environment

Key Features

- Implement Delphi's modern features to build professional-grade Windows, web, mobile, and IoT applications and powerful servers
- Become a Delphi code and project guru by learning best practices and techniques for cross-platform development
- Deploy your complete end-to-end application suite anywhere

Book Description

Delphi is a strongly typed, event-driven programming language with a rich ecosystem of frameworks and support tools. It comes with an extensive set of web and database libraries for rapid application development on desktop, mobile, and internet-enabled devices. This book will help you keep up with the latest IDE features and provide a sound foundation of project management and recent language enhancements to take your productivity to the next level. You'll discover how simple it is to support popular mobile device features such as sensors, cameras, and GPS. The book will help you feel comfortable working with FireMonkey and styles and incorporating 3D user interfaces in new ways. As you advance, you'll be able to build cross-platform solutions that not only look native but also take advantage of a wide array of device capabilities. You'll also learn how to use embedded databases, such as SQLite and InterBase ToGo, synchronizing them with your own custom backend servers or modules using the powerful RAD Server engine. The book concludes by sharing tips for testing and deploying your end-to-end application suite for a smooth user experience. By the end of this book, you'll be able to deliver modern enterprise applications using Delphi confidently.

What you will learn

- Discover the latest enhancements in the Delphi IDE
- Overcome the barriers that hold you back from embracing cross-platform development
- Become fluent with FireMonkey controls, styles, LiveBindings, and 3D objects
- Build Delphi packages to extend RAD Server or modularize your applications
- Use FireDAC to get quick and direct access to any data
- Leverage IoT technologies such as Bluetooth and Beacons and learn how to put your app on a Raspberry Pi
- Enable remote apps with backend servers on Windows and Linux through REST APIs
- Develop modules for IIS and Apache web servers

Who this book is for

This book is for Delphi developers interested in expanding their skillset beyond Windows programming by creating professional-grade applications on multiple platforms, including Windows, Mac, iOS, Android, and back-office servers. You'll also find this book useful if you're a developer looking to upgrade your knowledge of Delphi to keep up with the latest changes and enhancements in this powerful toolset. Some Delphi programming experience is necessary to make the most out of this book.

Do you want to know how OpenID Connect works? This book is for you! Exploring how OpenID Connect works in detail is the subject of this book. We take a bottom-up approach and first study all the elements (actors, endpoints, and tokens) of OpenID Connect. This puts us in an excellent position for the second step: to understand the various OpenID Connect Flows - how the actors, endpoints, and tokens are put together to transmit identity claims securely. Do you wonder why there are several OpenID Connect Flows? Whether we use OpenID Connect from a mobile app, a script in a browser or from a secure backend server, there is an appropriate OpenID Connect Flow with the right tradeoffs in security, functionality, and convenience for each of these scenarios. This book helps you to choose the right one. Do you think that these OpenID Connect Flows are confusing? You are not alone; the OpenID Connect Flows tend to get confusing. However, with this book, we make it clear and easy to understand: We visualize these flows and show how to choose the flow that is appropriate for a given scenario. A picture says more than a 1000 words - that is why we explain the OpenID Connect Flows using easy to understand sequence diagrams. Do you want to understand how JWT works? This book explains what a JSON Web Token (JWT) is, how it is used in OpenID Connect, how it is constructed, what data it contains, how to read it, and how to protect its contents. Do you wonder why there are so many tokens in OpenID Connect and how to use them? There are JWT, JWS, JWE, access tokens, refresh tokens, identity tokens, and authorization codes. This book helps you to make sense of them all. Using examples, we explore how the tokens are used, constructed, signed, and encrypted. Why is OpenID Connect so popular? If used in the right way, OpenID Connect is powerful, and everyone loves it: End-users don't need to signup and remember a new password Business owners enjoy high conversion rates Developers don't get any grey hair over securely storing credentials Do you want to increase the conversion rate of your app? Signup and login to a new app become so smooth and convenient that end-users are much more likely to try a new app. It is supported, e.g. by Google, Yahoo, or Microsoft. Would you like to manage no credentials but still have authenticated users? For us developers of web and mobile apps, these signup and login features are attractive, too: we do not need to manage user credentials, and we get a higher conversion rate resulting in more new customers. In effect, this means cutting costs and increasing the number of new customers for our apps. Which programming language do you use in the book? This is not a programming book, don't expect implementations with a specific programming language or library. Instead, we focus on understanding OpenID Connect on a conceptual level, so we can design and architect apps that work with OpenID Connect. And OpenID Connect is the standard behind creating smooth login and signup experiences, increasing the customer signup rate, and creating highly converting apps.

This book is ideal for any JavaScript developer who is interested in producing well-tested code. If you have no prior experience with testing, Node.js, or any other tool, do not worry, as they will be explained from scratch.

This book constitutes the proceedings of the 13th International Conference on Service-Oriented Computing, ICSOC 2015, held in Goa, India, in November 2015. The 23 full, 9 short, and 5 demo track papers presented in this volume were carefully reviewed and selected from 132 submissions. The research track papers are organized in topical sections named: internet of services/things; data services and cloud platform management; cloud services management; service composition; business process management; cloud services; QoS and trust; service composition.

Design scalable and robust RESTful web services with JAX-RS and Jersey extension APIs

About This Book

Get to grips with the portable Java APIs used for JSON processing

Design solutions to produce, consume, and visualize RESTful web services using WADL, RAML, and Swagger

A step-by-step guide packed with many real-life use-cases to help you build efficient and secure RESTful web APIs in Java

Who This Book Is For

If you are a web developer with a basic understanding of the REST concepts but are new to the idea of designing and developing RESTful web services, this is the book for you. As all the code samples for the book are written in Java, proficiency in Java is a must.

What You Will Learn

Introduce yourself to the RESTful software architectural style and the REST API design principles

Make use of the

JSR 353 APIs and Jackson API for JSON processing Build portable RESTful web APIs, making use of the JAX-RS 2.0 API Simplify API development using the Jersey extension APIs Secure your RESTful web services with various authentication and authorization mechanisms Get to grips with the various metadata solutions to describe, produce, and consume RESTful web services Understand the design and coding guidelines to build well-performing RESTful APIs See how the role of RESTful web services changes with emerging technologies and trends In Detail REST (REpresentational State Transfer) is a simple yet powerful software architecture style to create scalable web services and allow them to be simple, lightweight, and fast. The REST API uses HTTP and JSON, so that it can be used with many programming languages such as Ruby, Java, Python, and Scala. Its use in Java seems to be the most popular though, because of the API's reusability. This book is a guide to developing RESTful web services in Java using the popular RESTful framework APIs available today. You will begin with gaining an in-depth knowledge of the RESTful software architectural style and its relevance in modern applications. Further, you will understand the APIs to parse, generate, transform, and query JSON effectively. Then, you will see how to build a simple RESTful service using the popular JAX-RS 2.0 API along with some real-world examples. This book will introduce you to the Jersey framework API, which is used to simplify your web services. You will also see how to secure your services with various authentication mechanisms. You will get to grips with various solutions to describe, produce, consume, and visualize RESTful web services. Finally, you will see how to design your web services to equip them for the future technological advances, be it Cloud or mobile computing. By the end of this book, you will be able to efficiently build robust, scalable, and secure RESTful web services, making use of the JAX-RS and Jersey framework extensions. Style and approach This book is written as a step-by-step guide to designing and developing robust RESTful web services. Each topic is explained in a simple and easy-to-understand manner with lots of real-life use-cases and their solutions.

Explore the practical side of REST to build data-centric applications with Node About This Video Work through a series of guidelines and best practices to efficiently design RESTful Web APIs with Node Understand the structure of APIs, their authentication protocols, and their implementation tools This practical guide provides the knowledge you need to delve into the endless possibilities enabled by Big Data In Detail RESTful Web APIs allow developers to create unprecedented applications by leveraging the data on the internet. Since JavaScript is the language of the web, building APIs using Node.js provides a seamless development experience on both the front end and the back end. This video course gives you an overview of a RESTful API and goes through the logical steps of building one. It explores three different APIs, focusing on their similarities and differences to effectively implement one. We'll start off by defining APIs, showing you how they can be built on top of HTTP, and listing the properties that make an API RESTful. We will develop Twitter Notes, a web application that lets its users leave notes for their Twitter friends. We will use Twitter's API to implement a login flow and then design a web API. In addition to using Twitter's API, we will take a closer look at two other real-world APIs--Facebook API and GitHub API. Finally, we'll look at some best practices to keep the APIs secure, maintainable, and performing. By the end of this course, you will have a good grasp of APIs, HTTP, REST, OAuth 1.0a, API testing, and site reliability, performance, and security. Since the course explores three different REST APIs, you will reach a level where you will be comfortable using any RESTful API, even if it does not have an SDK.

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.

Pro REST API Development with Node.js is your guide to managing and understanding the full capabilities of successful REST development. API design is a hot topic in the programming world, but not many resources exist for developers to really understand how you can leverage the advantages. This book will provide a brief background on REST and the tools it provides (well known and not so well known). Understand how there is more to REST than just JSON and URLs. You will then cover and compare the maintained modules currently available in the npm community, including Express, Restify, Vatican, and Swagger. Finally you will code an example API from start to finish, using a subset of the tools covered. The Node community is currently flooded with modules; some of them are published once and never updated again - cluttering the entire universe of packages. Pro REST API Development with Node.js shines light into that black hole of modules for the developers trying to create an API. Understand REST API development with Node.js using this book today.

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

Understand how Node-RED, the free and open-source flow-based programming tool, is used for handling IoT data and how it allows programmers of any level to interconnect I/O, APIs, and online services in new and exciting ways. This book is a comprehensive introduction to Node-RED and will get you up to speed with building web apps in no time.

Manage and understand the full capabilities of successful REST development. REST API development is a hot topic in the programming world, but not many resources exist for developers to really understand how you can leverage the advantages. This completely updated second edition provides a brief background on REST and the tools it provides (well known and not so well known), then explains how there is more to REST than just JSON and URLs. You will learn about the maintained modules currently available in the npm community, including Express, Restify, Vatican, and Swagger. Finally you will code an example API from start to finish, using a subset of the tools covered. The Node community is currently flooded with modules; some of them are published once and never updated again - cluttering the entire universe of packages. Pro REST API Development with Node.js shines light into that black hole of modules for the developers trying to create an API. Understand REST API development with Node.js using this book today. What You'll Learn Understand how REST and API development mix up with Node.js Create a scalable, technology agnostic, and uniform interface Prepare your services to be consumed by your clients Test and deploy your API Review troubleshooting techniques Who This Book Is For Any Node.js developer who wants to fully understand REST API development. Beginner and Intermediate Node.js developers who are looking to fully understand how to create RESTful microservices.

This book constitutes extended, revised and selected papers from the 7th Ith International Conference on Cloud Computing and

Service Science, CLOSER 2017, held in Porto, Portugal, in April 2017. The 16 papers presented in this volume were carefully reviewed and selected from a total of 123 submissions. CLOSER 2017 focused on the emerging area of Cloud Computing, inspired by some latest advances that concern the infrastructure, operations and available services throughout the global network. The popularity of REST in recent years has led to tremendous growth in almost-RESTful APIs that don't include many of the architecture's benefits. With this practical guide, you'll learn what it takes to design usable REST APIs that evolve over time. By focusing on solutions that cross a variety of domains, this book shows you how to create powerful and secure applications, using the tools designed for the world's most successful distributed computing system: the World Wide Web. You'll explore the concepts behind REST, learn different strategies for creating hypermedia-based APIs, and then put everything together with a step-by-step guide to designing a RESTful Web API. Examine API design strategies, including the collection pattern and pure hypermedia Understand how hypermedia ties representations together into a coherent API Discover how XMDP and ALPS profile formats can help you meet the Web API "semantic challenge" Learn close to two-dozen standardized hypermedia data formats Apply best practices for using HTTP in API implementations Create Web APIs with the JSON-LD standard and other the Linked Data approaches Understand the CoAP protocol for using REST in embedded systems

Got RESTful APIs? Great. API consumers love them. But today, such RESTful APIs are not enough for the evolving expectations of API consumers. Their apps need to be responsive, event-based and react to changes in near real-time. This results in a new set of requirements for the APIs, which power the apps. APIs now need to provide concepts such as events, notifications, triggers, and subscriptions. These concepts are not natively supported by the REST architectural style. In this book we show how to engineer RESTful APIs that support events with a webhook infrastructure. What are the alternatives to webhooks? We study several approaches for realizing events, such as Polling, Long Polling, Webhooks, HTTP Streaming, Server-Sent Events, WebSockets, WebSub and GraphQL Subscriptions. All of these approaches have their advantages and disadvantages. Can webhooks communicate in real-time? We study the non-functional requirements of a webhooks infrastructure, in areas such as security, reliability and developer experience. How do well-known API providers design webhooks? We examine the webhook infrastructure provided by GitHub, BitBucket, Stripe, Slack, and Intercom. With the best practices, case studies, and design templates provided in this book, we want to help you extend your API portfolio with a modern webhook infrastructure. So you can offer both APIs and events that developers love to use.

Microservices is an architectural style in which large, complex software applications are composed of one or more smaller services. Each of these microservices focuses on completing one task that represents a small business capability. These microservices can be developed in any programming language. This IBM® Redbooks® publication covers Microservices best practices for Java. It focuses on creating cloud native applications using the latest version of IBM WebSphere® Application Server Liberty, IBM Bluemix® and other Open Source Frameworks in the Microservices ecosystem to highlight Microservices best practices for Java.

Understand the constructs of the Python programming language and use them to build data science projects Key Features Learn the basics of developing applications with Python and deploy your first data application Take your first steps in Python programming by understanding and using data structures, variables, and loops Delve into Jupyter, NumPy, Pandas, SciPy, and sklearn to explore the data science ecosystem in Python Book Description Python is the most widely used programming language for building data science applications. Complete with step-by-step instructions, this book contains easy-to-follow tutorials to help you learn Python and develop real-world data science projects. The "secret sauce" of the book is its curated list of topics and solutions, put together using a range of real-world projects, covering initial data collection, data analysis, and production. This Python book starts by taking you through the basics of programming, right from variables and data types to classes and functions. You'll learn how to write idiomatic code and test and debug it, and discover how you can create packages or use the range of built-in ones. You'll also be introduced to the extensive ecosystem of Python data science packages, including NumPy, Pandas, scikit-learn, Altair, and Datashader. Furthermore, you'll be able to perform data analysis, train models, and interpret and communicate the results. Finally, you'll get to grips with structuring and scheduling scripts using Luigi and sharing your machine learning models with the world as a microservice. By the end of the book, you'll have learned not only how to implement Python in data science projects, but also how to maintain and design them to meet high programming standards. What you will learn Code in Python using Jupyter and VS Code Explore the basics of coding – loops, variables, functions, and classes Deploy continuous integration with Git, Bash, and DVC Get to grips with Pandas, NumPy, and scikit-learn Perform data visualization with Matplotlib, Altair, and Datashader Create a package out of your code using poetry and test it with PyTest Make your machine learning model accessible to anyone with the web API Who this book is for If you want to learn Python or data science in a fun and engaging way, this book is for you. You'll also find this book useful if you're a high school student, researcher, analyst, or anyone with little or no coding experience with an interest in the subject and courage to learn, fail, and learn from failing. A basic understanding of how computers work will be useful.

Discover the RESTful technologies, including REST, JSON, XML, JAX-RS web services, SOAP and more, for building today's microservices, big data applications, and web service applications. This book is based on a course the Oracle-based author is teaching for UC Santa Cruz Silicon Valley which covers architecture, design best practices and coding labs. Pro RESTful APIs: Design gives you all the fundamentals from the top down: from the top (architecture) through the middle (design) to the bottom (coding). This book is a must have for any microservices or web services developer building applications and services. What You'll Learn Discover the key RESTful APIs, including REST, JSON, XML, JAX, SOAP and more Use these for web services and data exchange, especially in today's big data context Harness XML, JSON, REST, and JAX-RS in examples and case studies Apply best practices to your solutions' architecture Who This Book Is For Experienced web programmers and developers.

Looking for the big picture of building APIs? This book is for you! Building APIs that consumers love should certainly be the goal of any API initiative. However, it is easier said than done. It requires getting the architecture for your APIs right. This book equips you with both foundations and best practices for API architecture. This book is for you if you want to understand the big picture of API design and development, you want to define an API architecture, establish a platform for APIs or simply want to build APIs your consumers love. This book is NOT for you, if you are looking for a step-by step guide for building APIs, focusing on every detail of the correct application of REST principles. In this case I recommend the book "API Design" of the API-University Series. What is API architecture? Architecture spans the bigger picture of APIs and can be seen from several perspectives: API architecture may refer to the architecture of the complete solution consisting not only of the API itself, but also of an API client such as a mobile app

and several other components. API solution architecture explains the components and their relations within the software solution. API architecture may refer to the technical architecture of the API platform. When building, running and exposing not only one, but several APIs, it becomes clear that certain building blocks of the API, runtime functionality and management functionality for the API need to be used over and over again. An API platform provides an infrastructure for developing, running and managing APIs. API architecture may refer to the architecture of the API portfolio. The API portfolio contains all APIs of the enterprise and needs to be managed like a product. API portfolio architecture analyzes the functionality of the API and organizes, manages and reuses the APIs. API architecture may refer to the design decisions for a particular API proxy. To document the design decisions, API description languages are used. We explain the use of API description languages (RAML and Swagger) on many examples. This book covers all of the above perspectives on API architecture. However, to become useful, the architecture needs to be put into practice. This is why this book covers an API methodology for design and development. An API methodology provides practical guidelines for putting API architecture into practice. It explains how to develop an API architecture into an API that consumers love. A lot of the information on APIs is available on the web. Most of it is published by vendors of API products. I am always a bit suspicious of technical information pushed by product vendors. This book is different. In this book, a product-independent view on API architecture is presented. The API-University Series is a modular series of books on API-related topics. Each book focuses on a particular API topic, so you can select the topics within APIs, which are relevant for you.

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.

This book constitutes the proceedings of the 19th International Conference on Service-Oriented Computing, ICSOC 2020, which is held virtually in November 2021. The 29 full, 28 short, and 3 vision papers included in this volume were carefully reviewed and selected from 189 submissions. They were organized in topical sections named: Blockchains and smart contracts, Architectures, microservices and APIs, Applications, Internet-of-Things, crowdsourced, social, and conversational services, Service composition and recommendation, Cloud computing, and Edge computing.

A step-by-step guide to building microservices using Python and Docker, along with managing and orchestrating them with Kubernetes Key Features Learn to use Docker containers to create, operate, and deploy your microservices Create workflows to manage independent deployments on coordinating services using CI and GitOps through GitHub, Travis CI, and Flux Develop a REST microservice in Python using the Flask framework and Postgres database Book Description Microservices architecture helps create complex systems with multiple, interconnected services that can be maintained by independent teams working in parallel. This book guides you on how to develop these complex systems with the help of containers. You'll start by learning to design an efficient strategy for migrating a legacy monolithic system to microservices. You'll build a RESTful microservice with Python and learn how to encapsulate the code for the services into a container using Docker. While developing the services, you'll understand how to use tools such as GitHub and Travis CI to ensure continuous delivery (CD) and continuous integration (CI). As the systems become complex and grow in size, you'll be introduced to Kubernetes and explore how to orchestrate a system of containers while managing multiple services. Next, you'll configure Kubernetes clusters for production-ready environments and secure them for reliable deployments. In the concluding chapters, you'll learn how to detect and debug critical problems with the help of logs and metrics. Finally, you'll discover a variety of strategies for working with multiple teams dealing with different microservices for effective collaboration. By the end of this book, you'll be able to build production-grade microservices as well as orchestrate a complex system of services using containers. What you will learn Discover how to design, test, and operate scalable microservices Coordinate and deploy different services using Kubernetes Use Docker to construct scalable and manageable applications with microservices Understand how to monitor a complete system to ensure early detection of problems Become well versed with migrating from an existing monolithic system to a microservice one Use load balancing to ensure seamless operation between the old monolith and the new service Who this book is for This book is for developers, engineers, or software architects who are trying to move away from traditional approaches for building complex multi-service systems by adopting microservices and containers. Although familiarity with Python programming is assumed, no prior knowledge of Docker is required.

Build maintainable websites with elegant Django design patterns and modern best practices Key Features Explore aspects of Django from Models and Views to testing and deployment Understand the nuances of web development such as browser attack and data design Walk through various asynchronous tools such as Celery and Channels Book Description Building secure and maintainable web applications requires comprehensive knowledge. The second edition of this book not only sheds light on Django, but also encapsulates years of experience in the form of design patterns and best practices. Rather than sticking to GoF design patterns, the book looks at higher-level patterns. Using the latest version of Django and Python, you'll learn about Channels and asyncio while building a solid conceptual background. The book compares design choices to help you make everyday decisions faster in a rapidly changing environment. You'll first learn about various architectural patterns, many of which are used to build Django. You'll start with building a fun superhero project by gathering the requirements, creating mockups, and setting up the project. Through project-guided examples, you'll explore the Model, View, templates, workflows, and code reusability techniques.

In addition to this, you'll learn practical Python coding techniques in Django that'll enable you to tackle problems related to complex topics such as legacy coding, data modeling, and code reusability. You'll discover API design principles and best practices, and understand the need for asynchronous workflows. During this journey, you'll study popular Python code testing techniques in Django, various web security threats and their countermeasures, and the monitoring and performance of your application. What you will learn Make use of common design patterns to help you write better code Implement best practices and idioms in this rapidly evolving framework Deal with legacy code and debugging Use asynchronous tools such as Celery, Channels, and asyncio Use patterns while designing API interfaces with the Django REST Framework Reduce the maintenance burden with well-tested, cleaner code Host, deploy, and secure your Django projects Who this book is for This book is for you whether you're new to Django or just want to learn its best practices. You do not have to be an expert in Django or Python. No prior knowledge of patterns is expected for reading this book but it would be helpful.

Using a web API to provide services to application developers is one of the more satisfying endeavors that software engineers undertake. But building a popular API with a thriving developer ecosystem is also one of the most challenging. With this practical guide, developers, architects, and tech leads will learn how to navigate complex decisions for designing, scaling, marketing, and evolving interoperable APIs. Authors Brenda Jin, Saurabh Sahni, and Amir Shevat explain API design theory and provide hands-on exercises for building your web API and managing its operation in production. You'll also learn how to build and maintain a following of app developers. This book includes expert advice, worksheets, checklists, and case studies from companies including Slack, Stripe, Facebook, Microsoft, Cloudfire, Oracle, and GitHub. Get an overview of request-response and event-driven API design paradigms Learn best practices for designing an API that meets the needs of your users Use a template to create an API design process Scale your web API to support a growing number of API calls and use cases Regularly adapt the API to reflect changes to your product or business Provide developer resources that include API documentation, samples, and tools 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.

In today's market, where rival web services compete for attention, a well-designed REST API is a must-have feature. This concise book presents a set of API design rules, drawn primarily from best practices that stick close to the Web's REST architectural style. Along with rules for URI design and HTTP use, you'll learn guidelines for media types and representational forms. REST APIs are ubiquitous, but few of them follow a consistent design methodology. Using these simple rules, you will design web service APIs that adhere to recognized web standards. To assist you, author Mark Massé introduces the Web Resource Modeling Language (WRML), a conceptual framework he created for the design and implementation of REST APIs. Learn design rules for addressing resources with URIs Apply design principles to HTTP's request methods and response status codes Work with guidelines for conveying metadata through HTTP headers and media types Get design tips to address the needs of client programs, including the special needs of browser-based JavaScript clients Understand why REST APIs should be designed and configured, not coded This book constitutes the proceedings of the 21st International Conference on Web Engineering, ICWE 2021, which was supposed to be held in Biarritz, France, in May 2021. Due to the corona pandemic the conference changed to a virtual format. The total of 22 full and 13 short contributions presented in this volume were carefully reviewed and selected from 128 submissions. The book also contains 6 demonstration, 1 poster, 3 PhD, and 3 tutorial papers. The papers were organized in topical sections named: Semantic Web; social Web; Web modeling and engineering; Web big data and data analytics; Web mining and knowledge extraction; Web of Things; Web programming; Web user interfaces; PhD symposium; posters and demonstrations; and tutorials.

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.

Design and implement scalable and maintainable RESTful solutions with Node.js 10 Key Features Create rich and scalable RESTful API solutions from scratch Explore the new features of Node.js 10, Express 4.0, and MongoDB Integrate MongoDB in your Node.js application to store and secure your data Book Description When building RESTful services, it is really important to choose the right framework. Node.js, with its asynchronous, event-driven architecture, is exactly the right choice for building RESTful APIs. This third edition of RESTful Web API Design with Node.js 10 will teach you to create scalable and rich RESTful applications based on the Node.js platform. You will be introduced to the latest NPM package handler and understand how to use it to customize your RESTful development process. You will begin by understanding the key principle that makes an HTTP application a RESTful-enabled application. After writing a simple HTTP request handler, you will create and test Node.js modules using automated tests and mock objects; explore using the NoSQL database, MongoDB, to store data; and get to grips with using self-descriptive URLs. You'll learn to set accurate HTTP status codes along with understanding how to keep your applications backward-compatible. Also, while implementing a full-fledged RESTful service, you will use Swagger to document the API and implement automation tests for a REST-enabled endpoint with Mocha. Lastly, you will explore some authentication techniques to secure your application. What you will learn Install, develop, and test your own Node.js user modules Understand the differences between HTTP and RESTful applications Use self-descriptive URLs and set accurate HTTP status codes Eliminate third-party dependencies in your tests with mocking Implement automation tests for a REST-enabled endpoint with Mocha Secure your services with NoSQL database integration within Node.js applications Integrate a simple frontend using JavaScript libraries available on a CDN server Who this book is for If you are a web developer keen to enrich your development skills to create server-side RESTful applications based on the Node.js platform, this book is for you. Some knowledge of REST would be an added advantage, but is definitely not a necessity.

Innovate at scale through well-architected API-led products that drive personalized, predictive, and adaptive customer experiences Key Features Strategize your IT investments by modeling enterprise solutions with an API-centric approach Build robust and reliable API platforms to boost business agility and omnichannel delivery Create digital value chains through the productization of your APIs Book Description API-centric architectures are foundational to delivering omnichannel experiences for an enterprise. With this book, developers will learn techniques to design loosely coupled, cloud-based, business-tier interfaces that can be consumed by a variety of client applications. Using real-world examples and case studies, the book helps you get to grips with the cloudbased design and implementation of reliable and resilient API-centric solutions. Starting with the evolution of enterprise applications, you'll learn how API-based integration architectures drive digital transformation. You'll then learn about the important principles and practices that apply to cloud-based API architectures and advance to exploring the different architecture styles and their implementation in Azure. This book is written from a practitioner's point of view, so you'll discover ideas and practices that have worked successfully in various customer scenarios. By the end of this book, you'll be able to architect, design, deploy, and monetize your API solutions in the Azure cloud while implementing best practices and industry standards. What you will learn Explore the benefits of API-led architecture in an enterprise Build highly reliable and resilient, cloud-based, API-centric solutions Plan technical initiatives based on Well-Architected Framework principles Get to grips with the productization and management of your API assets for value creation Design high-scale enterprise integration platforms on the Azure cloud Study the important principles and practices that apply to cloud-based API architectures Who this book is for This book is for solution architects, developers, engineers, DevOps professionals, and IT decision-makers who are responsible for designing and developing large distributed systems. Familiarity with enterprise solution architectures and cloud-based design will help you to comprehend the concepts covered in the book easily.

Spring REST is a practical guide for designing and developing RESTful APIs using the Spring Framework. This book walks you through the process of designing and building a REST application while taking a deep dive into design principles and best practices for versioning, security, documentation, error handling, paging, and sorting. This book provides a brief introduction to REST, HTTP, and web infrastructure. You will learn about several Spring projects such as Spring Boot, Spring MVC, Spring Data JPA, and Spring Security and the role they play in simplifying REST application development. You will learn how to build clients that consume REST services. Finally, you will learn how to use the Spring MVC test framework to unit test and integration test your REST API. After reading this book, you will come away with all the skills to build sophisticated REST applications using Spring technologies.

Master core REST concepts and create RESTful web services in Java About This Book Build efficient and secure RESTful web APIs in Java.. Design solutions to produce, consume and visualize RESTful web services using WADL, RAML, and Swagger Familiarize the role of RESTful APIs usage in emerging technology trends like Cloud, IoT, Social Media. Who This Book Is For If you are a web developer with a basic understanding of the REST concepts and envisage to get acquainted with the idea of designing and developing RESTful web services, this is the book for you. As all the code samples for the book are written in Java, proficiency in Java is a must. What You Will Learn Introduce yourself to the RESTful software architectural style and the REST API design principles Make use of the JSR 353 API, JSR 374 API, JSR 367 API and Jackson API for JSON processing Build portable RESTful web APIs, making use of the JAX-RS 2.1 API Simplify API development using the Jersey and RESTEasy extension APIs Secure your RESTful web services with various authentication and authorization mechanisms Get to grips with the various metadata solutions to describe, produce, and consume RESTful web services Understand the design and coding guidelines to build well-performing RESTful APIs See how the role of RESTful web services changes with emerging technologies and trends In Detail Representational State Transfer (REST) is a simple yet powerful software architecture style to create lightweight and scalable web services. The RESTful web services use HTTP as the transport protocol and can use any message formats, including XML, JSON(widely used), CSV, and many more, which makes it easily inter-operable across different languages and platforms. This successful book is currently in its 3rd edition and has been used by thousands of developers. It serves as an excellent guide for developing RESTful web services in Java. This book attempts to familiarize the reader with the concepts of REST. It is a pragmatic guide for designing and developing web services using Java APIs for real-life use cases following best practices and for learning to secure REST APIs using OAuth and JWT. Finally, you will learn the role of RESTful web services for future technological advances, be it cloud, IoT or social media. By the end of this book, you will be able to efficiently build robust, scalable, and secure RESTful web services using Java APIs. Style and approach Step-by-step guide to designing and developing robust RESTful web services. Each topic is explained in a simple and easy-to-understand manner with lots of real-life use-cases and their solutions.

Spring is the de facto framework for Java SE development and it has been widely used for building web and enterprise grade applications. Last year, Pivotal provided complete support for Kotlin as a language that Spring can compile to, making it an even stronger framework in terms of android development. To make things simpler, Pivotal has ...

A strategy and implementation guide for building, deploying, and managing APIs Key Features Comprehensive, end-to-end guide to business-driven enterprise APIs Distills years of experience with API and microservice strategies Provides detailed guidance on implementing API-led architectures in any business Book Description APIs are the cornerstone of modern, agile enterprise systems. They enable access to enterprise services from a wide variety of devices, act as a platform for innovation, and open completely new revenue streams. Enterprise API Management shows how to define the right architecture, implement the right patterns, and define the right organization model for

business-driven APIs. Drawing on his experience of developing API and microservice strategies for some of the world's largest companies, Luis Weir explains how APIs deliver value across an enterprise. The book explores the architectural decisions, implementation patterns, and management practices for successful enterprise APIs, as well as providing clear, actionable advice on choosing and executing the right API strategy in your enterprise. With a relentless focus on creating business value, Luis Weir reveals an effective method for planning, building, and running business products and services with APIs. What you will learn Create API strategies to deliver business value Monetize APIs, promoting them through public marketplaces and directories Develop API-led architectures, applying best practice architecture patterns Choose between REST, GraphQL, and gRPC-style API architectures Manage APIs and microservices through the complete life cycle Deploy APIs and business products, as well as Target Operating Models Lead product-based organizations to embrace DevOps and focus on delivering business capabilities Who this book is for Architects, developers, and technology executives who want to deliver successful API strategies that bring business value.

Get started with designing your serverless application using optimum design patterns and industry standard practices Key Features Learn the details of popular software patterns and how they are applied to serverless applications Understand key concepts and components in serverless designs Walk away with a thorough understanding of architecting serverless applications Book Description Serverless applications handle many problems that developers face when running systems and servers. The serverless pay-per-invocation model can also result in drastic cost savings, contributing to its popularity. While it's simple to create a basic serverless application, it's critical to structure your software correctly to ensure it continues to succeed as it grows. Serverless Design Patterns and Best Practices presents patterns that can be adapted to run in a serverless environment. You will learn how to develop applications that are scalable, fault tolerant, and well-tested. The book begins with an introduction to the different design pattern categories available for serverless applications. You will learn the trade-offs between GraphQL and REST and how they fare regarding overall application design in a serverless ecosystem. The book will also show you how to migrate an existing API to a serverless backend using AWS API Gateway. You will learn how to build event-driven applications using queuing and streaming systems, such as AWS Simple Queuing Service (SQS) and AWS Kinesis. Patterns for data-intensive serverless application are also explained, including the lambda architecture and MapReduce. This book will equip you with the knowledge and skills you need to develop scalable and resilient serverless applications confidently. What you will learn Comprehend the popular design patterns currently being used with serverless architectures Understand the various design options and corresponding implementations for serverless web application APIs Learn multiple patterns for data-intensive serverless systems and pipelines, including MapReduce and Lambda Architecture Learn how to leverage hosted databases, queues, streams, storage services, and notification services Understand error handling and system monitoring in a serverless architecture a serverless architecture Learn how to set up a serverless application for continuous integration, continuous delivery, and continuous deployment Who this book is for If you're a software architect, engineer, or someone who wants to build serverless applications, which are non-trivial in complexity and scope, then this book is for you. Basic knowledge of programming and serverless computing concepts are assumed.

Hands-On RESTful API Design Patterns and Best Practices Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs Packt Publishing Ltd

[Copyright: 4bf8ed510bca823f5c6bb38e2fdfee6a](#)