
Best Api Documentation

Docs Like Code

Documenting APIs

How to Become a Technical Writer

Practical API Design

Irresistible APIs

API Design for C++

RESTful Web Services

Documenting Software Architectures

Developer Marketing Does Not Exist

Django for APIs

Building REST APIs with Flask

Microservice APIs

REST API Design Rulebook

Principles of Web API Design

Docs for Developers

Microservices Patterns

RESTful Web APIs

Undisturbed REST

Hacking APIs

Pure functional HTTP APIs in Scala

Natural Language Processing with Python

REST in Practice

Continuous API Management

"Dear Evil Tester"

Build APIs You Won't Hate

Designing APIs with Swagger and OpenAPI

Redmine Plugin Extension and Development

Design and Build Great Web APIs
A Philosophy of Software Design
Designing Web APIs
API Economy 101
API Management
Trends and Advances in Information Systems and
Technologies
Pro Git
R Markdown
APIs: A Strategy Guide
The Google Story (2018 Updated Edition)
Higher-Order Perl
Windows NT/2000 Native API Reference
The Product is Docs

*Downloaded
from
Best Api archive.imba.com
Documentation by guest*

ARELY GWENDOLYN

Docs Like Code Simon
and Schuster
Completely updated for
Django 4.0 & Django
REST Framework 3.13!
Django for APIs is a
project-based guide to
building modern web
APIs with Django &
Django REST
Framework. It is

suitable for beginners
who have never built
an API before as well
as professional
programmers looking
for a fast-paced
introduction to Django
fundamentals and best
practices. Over the
course of 200+ pages
you'll learn how to set
up a new project
properly, how web APIs
work under the hood,
and advanced testing
and deployment
techniques. Three

separate projects are built from scratch with progressively more advanced features including a Library API, Todo API, and Blog API. User authentication, permissions, documentation, viewsets, and routers are all covered thoroughly. Django for APIs is a best-practices guide to building powerful Python-based web APIs with a minimal amount of code.

Documenting APIs

Apress

Most Perl programmers were originally trained as C and Unix programmers, so the Perl programs that they write bear a strong resemblance to C programs. However, Perl incorporates many features that have their roots in other languages such as

Lisp. These advanced features are not well understood and are rarely used by most Perl programmers, but they are very powerful. They can automate tasks in everyday programming that are difficult to solve in any other way. One of the most powerful of these techniques is writing functions that manufacture or modify other functions. For example, instead of writing ten similar functions, a programmer can write a general pattern or framework that can then create the functions as needed according to the pattern. For several years Mark Jason Dominus has worked to apply functional programming techniques to Perl. Now Mark brings these

flexible programming methods that he has successfully taught in numerous tutorials and training sessions to a wider audience.*

Introduces powerful programming methods new to most Perl programmers that were previously the domain of computer scientists* Gradually builds up confidence by describing techniques of progressive sophistication* Shows how to improve everyday programs and includes numerous engaging code examples to illustrate the methods

[How to Become a Technical Writer](#) Sams Publishing

Believe it or not, building an API is the easy part. What is far more challenging is to put together a design

that will stand the test of time, while also meeting your developers' needs. After all, no matter how well written your code may be, without a strong foundation, you will find your API quickly failing.

Undisturbed REST works to tackle this issue through the use of modern design techniques and technology, showing how to carefully design your API with your users and longevity in mind, taking advantage of a design-first approach- while incorporating best practices and hard lessons learned. After reading Undisturbed REST, you'll have a strong understanding of APIs, best practices, and available tooling for designing, prototyping, sharing,

documenting, and generating tooling (such as SDKs) around your API. More importantly, you'll be equipped to design and build an API not just for today, but one that can stand the test of time and lead your application into tomorrow.

Practical API Design

WelcomeToCode

Windows NT/2000

Native API Reference is absolutely unique.

Currently, documentation on Windows NT's native APIs can only be found through access to the source code or occasionally Web sites where people have chosen to share bits of insight gained through reverse engineering.

This book provides the first complete reference to the API functions native to

Windows NT and covers the set of services that are offered by Windows NT to both kernel- and user-mode programs. Ideal for the intermediate and advanced level user- and kernel-mode developers of Windows systems, this books is devoted to the NT native API and consists of documentation of the 210 routines included in the API. Also included are all the functions added in Windows 2000.

Irresistible APIs Apress

If you can write clear, concise instructions, then you can be a technical writer. Learn, step-by-step, how to turn your creative writing talent into a highly lucrative career, where you get paid big money consistently to use your writing skills.

API Design for C++

Lulu.com

"This book addresses the topic of software design: how to decompose complex software systems into modules (such as classes and methods) that can be implemented relatively independently. The book first introduces the fundamental problem in software design, which is managing complexity. It then discusses philosophical issues about how to approach the software design process and it presents a collection of design principles to apply during software design. The book also introduces a set of red flags that identify design problems. You can apply the ideas in this book to minimize the complexity of large

software systems, so that you can write software more quickly and cheaply."--

Amazon.

RESTful Web Services

"O'Reilly Media, Inc."

Strategies, best practices, and patterns that will help you design resilient microservices architecture and streamline your API integrations. In *Microservice APIs*, you'll discover: Service decomposition strategies for microservices
 Documentation-driven development for APIs
 Best practices for designing REST and GraphQL APIs
 Documenting REST APIs with the OpenAPI specification (formerly Swagger)
 Documenting GraphQL APIs using the Schema Definition Language Building

microservices APIs with Flask, FastAPI, Ariadne, and other frameworks
Service implementation patterns for loosely coupled services
Property-based testing to validate your APIs, and using automated API testing frameworks like schemathesis and Dredd
Adding authentication and authorization to your microservice APIs using OAuth and OpenID Connect (OIDC)
Deploying and operating microservices in AWS with Docker and Kubernetes
Microservice APIs teaches you practical techniques for designing robust microservices with APIs that are easy to understand, consume, and maintain. You'll benefit from author

José Haro Peralta's years of experience experimenting with microservices architecture, dodging pitfalls and learning from mistakes he's made. Inside you'll find strategies for delivering successful API integrations, implementing services with clear boundaries, managing cloud deployments, and handling microservices security. Written in a framework-agnostic manner, its universal principles can easily be applied to your favorite stack and toolset. About the technology
Clean, clear APIs are essential to the success of microservice applications. Well-designed APIs enable reliable integrations between services and help simplify

maintenance, scaling, and redesigns. This book teaches you the patterns, protocols, and strategies you need to design, build, and deploy effective REST and GraphQL microservices APIs. About the book *Microservice APIs* gathers proven techniques for creating and building easy-to-consume APIs for microservices applications. Rich with proven advice and Python-based examples, this practical book focuses on implementation over philosophy. You'll learn how to build robust microservice APIs, test and protect them, and deploy them to the cloud following principles and patterns that work in any language. What's inside *Service*

decomposition strategies for microservices Best practices for designing and building REST and GraphQL APIs Service implementation patterns for loosely coupled components API authorization with OAuth and OIDC Deployments with AWS and Kubernetes About the reader For developers familiar with the basics of web development. Examples are in Python. About the author José Haro Peralta is a consultant, author, and instructor. He's also the founder of microapis.io. Table of Contents PART 1 INTRODUCING MICROSERVICE APIS 1 What are microservice APIs? 2 A basic API implementation 3 Designing microservices PART 2

DESIGNING AND BUILDING REST APIS 4
Principles of REST API design 5 Documenting REST APIs with OpenAPI 6 Building REST APIs with Python 7 Service implementation patterns for microservices PART 3 DESIGNING AND BUILDING GRAPHQL APIS 8 Designing GraphQL APIs 9 Consuming GraphQL APIs 10 Building GraphQL APIs with Python PART 4 SECURING, TESTING, AND DEPLOYING MICROSERVICE APIS 11 API authorization and authentication 12 Testing and validating APIs 13 Dockerizing microservice APIs 14 Deploying microservice APIs with Kubernetes *Documenting Software Architectures* "O'Reilly Media, Inc."

REST continues to gain momentum as the best method for building Web services, and this down-to-earth book delivers techniques and examples that show how to design and implement integration solutions using the REST architectural style. **Developer Marketing Does Not Exist** "O'Reilly Media, Inc." API Design for C++ provides a comprehensive discussion of Application Programming Interface (API) development, from initial design through implementation, testing, documentation, release, versioning, maintenance, and deprecation. It is the only book that teaches the strategies of C++

API development, including interface design, versioning, scripting, and plug-in extensibility. Drawing from the author's experience on large scale, collaborative software projects, the text offers practical techniques of API design that produce robust code for the long term. It presents patterns and practices that provide real value to individual developers as well as organizations. API Design for C++ explores often overlooked issues, both technical and non-technical, contributing to successful design decisions that produce high quality, robust, and long-lived APIs. It focuses on various API styles and patterns that will allow you to produce elegant and

durable libraries. A discussion on testing strategies concentrates on automated API testing techniques rather than attempting to include end-user application testing techniques such as GUI testing, system testing, or manual testing. Each concept is illustrated with extensive C++ code examples, and fully functional examples and working source code for experimentation are available online. This book will be helpful to new programmers who understand the fundamentals of C++ and who want to advance their design skills, as well as to senior engineers and software architects seeking to gain new expertise to complement their

existing talents. Three specific groups of readers are targeted: practicing software engineers and architects, technical managers, and students and educators. - The only book that teaches the strategies of C++ API development, including design, versioning, documentation, testing, scripting, and extensibility - Extensive code examples illustrate each concept, with fully functional examples and working source code for experimentation available online - Covers various API styles and patterns with a focus on practical and efficient designs for large-scale long-term projects [Django for APIs](#) Pearson Education

Develop RESTful web services using the Flask micro-framework and integrate them using MySQL. Use Flask to develop, deploy, and manage REST APIs with easy-to-read and understand Python code. Solve your problem from a choice of libraries. Learn to use MySQL as the web services database for your Flask API using SQLAlchemy ORM. Building REST APIs with Flask provides a primer on Flask, RESTful services, and working with pip to set up your virtual environment. The key differences between NoSQL and SQL are covered, and you are taught how to connect MySQL and Flask using SQLAlchemy. Author Kunal Relan presents best practices for creating REST APIs and

guides you in structuring your app and testing REST endpoints. He teaches you how to set up authentication and render HTML using views. You learn how to write unit tests for your REST APIs, and understand mocks, assertions, and integration testing. You will know how to document your REST APIs, deploy your Flask application on all of the major cloud platforms, and debug and monitor your Flask application. What You'll Learn Use MySQL to create Flask REST APIs Test REST endpoints Create CRUD endpoints with Flask and MySQL Deploy Flask on all of the major cloud platforms Monitor your Flask application Who This Book Is For Python developers interested

in REST API development using Flask and web developers with basic programming knowledge who want to learn how Python and REST APIs work together. Readers should be familiar with Python (command line, or at least pip) and MySQL.

Building REST APIs with Flask Elsevier

This book offers a highly accessible introduction to natural language processing, the field that supports a variety of language technologies, from predictive text and email filtering to automatic summarization and translation. With it, you'll learn how to write Python programs that work with large collections of unstructured text.

You'll access richly annotated datasets using a comprehensive range of linguistic data structures, and you'll understand the main algorithms for analyzing the content and structure of written communication. Packed with examples and exercises, *Natural Language Processing with Python* will help you: Extract information from unstructured text, either to guess the topic or identify "named entities" Analyze linguistic structure in text, including parsing and semantic analysis Access popular linguistic databases, including WordNet and treebanks Integrate techniques drawn from fields as diverse as linguistics and artificial

intelligence This book will help you gain practical skills in natural language processing using the Python programming language and the Natural Language Toolkit (NLTK) open source library. If you're interested in developing web applications, analyzing multilingual news sources, or documenting endangered languages -- or if you're simply curious to have a programmer's perspective on how human language works -- you'll find *Natural Language Processing with Python* both fascinating and immensely useful. **Microservice APIs** Simon and Schuster This book includes a selection of papers from the 2018 World

Conference on Information Systems and Technologies (WorldCIST'18), held in Naples, Italy on March 27-29, 2018. WorldCIST is a global forum for researchers and practitioners to present and discuss recent results and innovations, current trends, professional experiences and the challenges of modern information systems and technologies research together with their technological development and applications. The main topics covered are: A) Information and Knowledge Management; B) Organizational Models and Information Systems; C) Software and Systems Modeling; D) Software Systems, Architectures, Applications and Tools;

E) Multimedia Systems and Applications; F) Computer Networks, Mobility and Pervasive Systems; G) Intelligent and Decision Support Systems; H) Big Data Analytics and Applications; I) Human-Computer Interaction; J) Ethics, Computers & Security; K) Health Informatics; L) Information Technologies in Education; M) Information Technologies in Radiocommunications; N) Technologies for Biomedical Applications. REST API Design Rulebook BoD - Books on Demand Software architecture—the conceptual glue that holds every phase of a project together for its many stakeholders—is widely recognized as a

critical element in modern software development. Practitioners have increasingly discovered that close attention to a software system's architecture pays valuable dividends. Without an architecture that is appropriate for the problem being solved, a project will stumble along or, most likely, fail. Even with a superb architecture, if that architecture is not well understood or well communicated the project is unlikely to succeed. Documenting Software Architectures, Second Edition, provides the most complete and current guidance, independent of language or notation, on how to capture an architecture in a commonly understandable form.

Drawing on their extensive experience, the authors first help you decide what information to document, and then, with guidelines and examples (in various notations, including UML), show you how to express an architecture so that others can successfully build, use, and maintain a system from it. The book features rules for sound documentation, the goals and strategies of documentation, architectural views and styles, documentation for software interfaces and software behavior, and templates for capturing and organizing information to generate a coherent package. New and improved in this second edition:

Coverage of architectural styles such as service-oriented architectures, multi-tier architectures, and data models
 Guidance for documentation in an Agile development environment
 Deeper treatment of documentation of rationale, reflecting best industrial practices
 Improved templates, reflecting years of use and feedback, and more documentation layout options
 A new, comprehensive example (available online), featuring documentation of a Web-based service-oriented system
 Reference guides for three important architecture documentation languages: UML, AADL, and SysML

Principles of Web API Design Booklocker.com
 You might think more than enough design books exist in the programming world already. In fact, there are so many that it makes sense to ask why you would read yet another. Is there really a need for yet another design book? In fact, there is a greater need than ever before, and *Practical API Design: Confessions of a Java Framework Architect* fills that need! Teaches you how to write an API that will stand the test of time
 Written by the designer of the NetBeans API at Sun Technologies
 Based on best practices, scalability, and API design patterns
[Docs for Developers](#)
 Packt Publishing Ltd
 Learn to integrate

programming with good documentation. This book teaches you the craft of documentation for each step in the software development lifecycle, from understanding your users' needs to publishing, measuring, and maintaining useful developer documentation. Well-documented projects save time for both developers on the project and users of the software. Projects without adequate documentation suffer from poor developer productivity, project scalability, user adoption, and accessibility. In short: bad documentation kills projects. Docs for Developers demystifies the process of creating great developer documentation,

following a team of software developers as they work to launch a new product. At each step along the way, you learn through examples, templates, and principles how to create, measure, and maintain documentation—tools you can adapt to the needs of your own organization. What You'll Learn Create friction logs and perform user research to understand your users' frustrations Research, draft, and write different kinds of documentation, including READMEs, API documentation, tutorials, conceptual content, and release notes Publish and maintain documentation alongside regular code releases Measure the success of the content

you create through analytics and user feedback Organize larger sets of documentation to help users find the right information at the right time Who This Book Is For Ideal for software developers who need to create documentation alongside code, or for technical writers, developer advocates, product managers, and other technical roles that create and contribute to documentation for their products and services.

Microservices Patterns

"O'Reilly Media, Inc."

This book provides a thorough look at the tools available within Redmine to facilitate the development of plugins and extensions. Full of tips and tricks, a recurring case study is

also used to help guide you through the various content area extensions and plugin helpers and to provide context and real-world examples of the concepts being covered. This book is intended for those who want to build custom functionality into the Redmine project management platform. If you're already comfortable with Ruby, Rails, and Redmine, but you aren't quite sure what to do next, this book will walk you through the basics of getting a plugin written and integrated into various key areas of Redmine.

RESTful Web APIs

Apress

The definitive, bestselling account of the company that changed the way we work and live, updated

for the twentieth anniversary of Google's founding with analysis of its most recent bold moves to redefine the world—and its even more ambitious plans for the future. Moscow-born Sergey Brin and Midwest-born Larry Page dropped out of graduate school at Stanford University to, as they said, “change the world” through a powerful search engine that would organize every bit of information on the Web for free. The Google Story takes you deep inside the company's wild ride from an idea that struggled for funding in 1998 to a firm that today rakes in billions in profits. Based on scrupulous research and extraordinary access to Google, this fast-moving narrative reveals how an

unorthodox management style and a culture of innovation enabled a search-engine giant to shake up Madison Avenue, clash with governments that accuse it of being a monopoly, deploy self-driving cars to forever change how we travel, and launch high-flying Internet balloons. Unafraid of controversy, Google is surging ahead with artificial intelligence that could cure diseases but also displace millions of people from their jobs, testing the founders' guiding mantra: DON'T BE EVIL. Praise for The Google Story “[The authors] do a fine job of recounting Google's rapid rise and explaining its search business.”—The New York Times “An

intriguing insider view of the Google culture.”—Harvard Business Review “An interesting read on a powerhouse company . . . If you haven’t read anything about one of today’s most influential companies, you should. If you don’t read The Google Story, you’re missing a few extra treats.”—USA Today “Fascinating . . . meticulous . . . never bogs down.”—Houston Chronicle

Undisturbed REST
Pragmatic Bookshelf

Are you in charge of your own testing? Do you have the advice you need to advance your test approach? "Dear Evil Tester" contains advice about testing that you won't hear anywhere else. "Dear Evil Tester" is a three pronged publication designed

to: -provoke not placate, -make you react rather than relax, -help you laugh not languish. Starting gently with the laugh out loud Agony Uncle answers originally published in 'The Testing Planet'. "Dear Evil Tester" then provides new answers, to never before published questions, that will hit your beliefs where they change. Before presenting you with essays that will help you unleash your own inner Evil Tester. With advice on automating, communication, talking at conferences, psychotherapy for testers, exploratory testing, tools, technical testing, and more. Dear Evil Tester randomly samples the Software Testing stomping ground

before walking all over it. "Dear Evil Tester" is a revolutionary testing book for the mind which shows you an alternative approach to testing built on responsibility, control and laughter. Read what our early reviewers had to say: "Wonderful stuff there. Real deep." Rob Sabourin, @RobertASabourin Author of "I Am a Bug" "The more you know about software testing, the more you will find to amuse you." Dot Graham, @dorothygraham Author of "Experiences of Test Automation" "laugh-out-loud episodes" Paul Gerrard, @paul_gerrard Author of "The Tester's Pocketbook" "A great read for every Tester." Andy Glover, @cartoontester Author

of "Cartoon Tester" *Hacking APIs* Bantam "Every developer working with the Web needs to read this book." -- David Heinemeier Hansson, creator of the Rails framework "RESTful Web Services finally provides a practical roadmap for constructing services that embrace the Web, instead of trying to route around it." -- Adam Trachtenberg, PHP author and EBay Web Services Evangelist You've built web sites that can be used by humans. But can you also build web sites that are usable by machines? That's where the future lies, and that's what RESTful Web Services shows you how to do. The World Wide Web is the most popular distributed application

in history, and Web services and mashups have turned it into a powerful distributed computing platform. But today's web service technologies have lost sight of the simplicity that made the Web successful. They don't work like the Web, and they're missing out on its advantages. This book puts the "Web" back into web services. It shows how you can connect to the programmable web with the technologies you already use every day. The key is REST, the architectural style that drives the Web. This book: Emphasizes the power of basic Web technologies -- the HTTP application protocol, the URI naming standard, and the XML markup language Introduces

the Resource-Oriented Architecture (ROA), a common-sense set of rules for designing RESTful web services Shows how a RESTful design is simpler, more versatile, and more scalable than a design based on Remote Procedure Calls (RPC) Includes real-world examples of RESTful web services, like Amazon's Simple Storage Service and the Atom Publishing Protocol Discusses web service clients for popular programming languages Shows how to implement RESTful services in three popular frameworks -- Ruby on Rails, Restlet (for Java), and Django (for Python) Focuses on practical issues: how to design and implement RESTful web services and clients This is the first book that applies

the REST design philosophy to real web services. It sets down the best practices you need to make your design a success, and the techniques you need to turn your design into working code. You can harness the power of the Web for programmable applications: you just have to work with the Web instead of against it. This book shows you how.

Pure functional HTTP APIs in Scala

Apress
Pro Git (Second Edition) is your fully-updated guide to Git and its usage in the modern world. Git has come a long way since it was first developed by Linus Torvalds for Linux kernel development. It has taken the open source

world by storm since its inception in 2005, and this book teaches you how to use it like a pro. Effective and well-implemented version control is a necessity for successful web projects, whether large or small. With this book you'll learn how to master the world of distributed version workflow, use the distributed features of Git to the full, and extend Git to meet your every need.

Written by Git pros Scott Chacon and Ben Straub, Pro Git (Second Edition) builds on the hugely successful first edition, and is now fully updated for Git version 2.0, as well as including an indispensable chapter on GitHub. It's the best book for all your Git needs.

Related with Best Api Documentation:

- Vita Taxslayer Pro Practice Lab : [click here](#)