API Docs Handbook

Case Studies, Insights, and Strategies

Mark Wentowski

DOCS GEEK

Introduction

This 'handbook' offers practical insights and actionable ideas to improve API docs by delving into case studies of companies known for quality API documentation.

It focuses on essential topics to include in the API documentation, provides examples of unique approaches, analyzes their effectiveness, and presents checklists for implementation.

Rather than being a research paper, I want to inspire reflection, brainstorming, and much note-taking to help product stakeholders and documentarians formulate the best practices for documenting their API.

Use Cases

Deliver end-to-end use case workflows

Case Study

amazon web services

The Amazon Rekognition API platform lets you add image and video analysis to your application, including features like facial detection, search, and verification.

Amazon represents functionality as use cases, providing workflows from prerequisites to coding implementation, making each use case a comprehensive resource for developers.

Insight:
In-depth use cases

Unlike many APIs that only describe use cases conceptually, Amazon provides complete end-to-end documentation for each use case.

Docs Checklist

- ✓ Encompass the complete use case process
- ✓ Illustrate using architecture and sequence diagrams
- ✓ Provide sample backend configurations and setups
- ✓ Include a targeted FAQ for the use case

A compelling use case demonstrates functionality in a real-world context

The Amazon Rekognition API documentation provides a comprehensive implementation guide for each use case. It starts with a conceptual overview, illustrating practical scenarios where developers can apply the functionality in real-world applications.

The documentation then delves into user flows, showcasing the necessary inputs, API interactions, and resulting outcomes.

To enhance comprehension, Amazon includes visual aids such as architecture and sequence diagrams, allowing developers to grasp the system structure and components involved.

Sample backend configurations and setups are practical starting points for developers to kickstart their implementations.

Detailed instructions cover effectively interacting with the specific endpoints at the core of each use case, including making API requests and handling responses.

Additionally, the documentation addresses versioning management, ensuring developers are informed about any changes or updates to the API that may impact their implementations.

Finally, Amazon offers use case-specific FAQ sections.

Onboarding

Tailor the onboarding experience for each user type

Case Study

platformOS

platformOS is a platform as a service (PaaS) that allows developers to create web applications without the need to manage the underlying infrastructure.

platformOS lets non-technical users get started quickly and experience the platform while allowing more technical users to dive right in, create an app, and follow documentation for setup, deployment, and testing.

Insight: Multiple onboarding journeys

platformOS caters to non-technical users and provides tailored onboarding journeys for each user type.

Docs Checklist

- ✓ Classify users by technical proficiency
- ✓ Tailor onboarding for each user type
- ✓ Provide easy setup options
- ✓ Create a clear path for technical users

Seamless onboarding eliminates barriers to API understanding and boosts adoption rates

PlatformOS caters to users with varying technical expertise, classifying users as non-technical, semi-technical, and technical users, providing onboarding paths to meet their needs.

For non-technical users, platformOS simplifies onboarding through a 'one-click' installation to register an account and guides them through an easy setup wizard to create a demo blog site.

Semi-technical users can access a Sandbox environment and clone a demo site from GitHub, where they can follow the Hello, World! Guide to grasp the fundamentals of sending requests and reading responses.

On the other hand, technical users can dive right in by building their first app, following the comprehensive documentation to set up their development environment, deploy, and test their app.

The PlatformOS team conducted extensive developer experience (DX) research, employing various techniques such as interviews, remote usability testing, crowdsourcing exercises, surveys, analytics, and reviews at each stage of product development. This research-driven approach enabled platformOS to understand user expectations, behaviors, and needs, creating tailored onboarding journeys that provide a fast track for technical users while allowing non-technical users to experience the platform's capabilities.

Get Started

Convey the whole experience of your API in one workflow

Case Study

GitHub

The GitHub API allows developers to interact with GitHub's repositories, issues, pull requests, and other resources programmatically.

GitHub's Get Started walks developers through a complete API workflow using one endpoint, from sending an API request to interpreting the response.

Insight: Get started as a microcosm

GitHub's Get Started tutorial provides contextual understanding while covering the purpose and syntax of API elements.

Docs Checklist

- ✓ Streamline authentication and authorization
- ✓ Offer support for common programming languages
- ✓ Combine explanation with examples
- ✓ Encourage a hands-on experience

An effective Get Started provides a holistic view of the API using a simple workflow

The GitHub API's Quickstart offers a CLI that simplifies authentication and interaction with the API, eliminating the need for a separate access token. Developers can easily connect their GitHub account by running a command and using an authentication app.

In addition, a full Get Started section introduces the API, including its purpose and key features. It then proceeds to guide developers through the basics of authentication and making API requests using their preferred programming language.

GitHub's approach to getting started with its API is hands-on. Instead of presenting API elements independently, their documentation combines API elements with a tutorial. This tutorial guides users through listing and creating issues in a repository, giving them a contextual understanding of using the API.

Furthermore, the documentation offers guidance on evaluating the success of API requests and interpreting the response data, ensuring developers have a holistic comprehension of their interactions with the API.

Key Concepts

Explore the concepts and processes that inform the API design

Case Study

stripe

Stripe is a payment processor platform and payment gateway businesses use to accept and process online payments.

Stripe explores industry concepts such as online payments, including how money moves in a transaction, the roles of different parties involved, payment workflows, and setup guidelines.

Insight:
Concept guides

Stripe provides context by explaining the payments ecosystem and the underlying principles that shape the API.

Docs Checklist

- ✓ Clarify industry/domain concepts
- ✓ Outline prerequisites and assumptions
- ✓ Demonstrate usage through illustrative workflows
- ✓ Define terminology in a glossary

Key concepts shed light on the underlying processes that govern the API

Stripe offers concept guides that lay the foundation for understanding the platform's core functionalities in the context of the more extensive ecosystem surrounding it.

For example, the Payments Fundamentals guide provides a high-level overview of how payments work, the players involved, and the different stages of the online payment process.

Stripe highlights the benefits of online payment processing for various businesses, such as online retailers, SaaS companies, platforms, and marketplaces, outlining the specific advantages for each business type.

After discussing benefits, it links to dedicated concept guides targeting each business type. By tailoring the content this way, Stripe demonstrates a deep understanding of its customers' requirements and why they would use a payment processing platform like Stripe.

Finally, developers can reference the concept-specific glossary if they encounter unfamiliar terminology.

Endpoint Overviews

Treat endpoints as individual products

Case Study

PLAID

Plaid offers APIs for financial data analysis and identity verification.

Plaid provides comprehensive endpoint overviews that offer more than just a list of endpoints. They provide use case descriptions, 3-minute video demonstrations, schemas, links to sample apps, and migration guides to ensure smooth transitions.

Insight:
Productized endpoints

Plaid presents its API endpoints as products, emphasizing less its technical aspects and more on its independent capabilities and use cases.

Docs Checklist

- ✓ Highlight purpose and unique selling points (USPs)
- ✓ Illustrate through a video walkthrough
- ✓ Provide interactive examples
- ✓ Offer endpoint-level migration guidance

An endpoint overview introduces an endpoint, its unique capabilities, and its use cases

Plaid redefines endpoints as products in their API documentation, featuring dedicated introductions for each.

The endpoint overviews explain the unique capabilities of each endpoint, showcasing their functionalities and providing real-world use cases.

Plaid offers 3-minute video demonstrations for every endpoint to enhance comprehension, enabling developers to visually grasp the endpoint's functionality.

The docs break down the response object's data structure and attribute information.

Developers can refer to a comprehensive example app with sample code to effectively integrate the endpoints and kickstart their development process.

Migration guides are available for each endpoint, ensuring a smooth transition from older releases to the current version and allowing developers to leverage the latest features and improvements.

Code Tutorials

Offer pre-built apps integrating pieces of functionality

Case Study

slack

The Slack API allows developers to integrate with the Slack features, such as sending messages, managing channels, and retrieving user information.

Slack's tutorials provide a hands-on experience, covering prerequisites, setting up the development environment, and integrating functionality into apps.

Insight:
Pre-built sample apps

For each functionality, developers can choose to use a blank or a pre-built app, catering to different levels of expertise and preferences.

Docs Checklist

- ✓ Provide an index page for easy access to code samples
- ✓ Divide the process into smaller, manageable tasks
- ✓ Include commented code snippets for each step
- ✓ Offer tutorial variations catering to different skill levels

Code tutorials bridge the gap between theoretical knowledge and practical implementation

The Slack API offers comprehensive code tutorials that guide developers through integrating functionality into apps.

To facilitate easy access to code samples, the Slack API features a dedicated index page where developers can quickly explore code samples and filter based on their preferred programming language.

Each code sample focuses on a specific functionality or use case, providing detailed step-by-step instructions and well-commented code snippets. These tutorials take developers through the entire implementation process, starting from prerequisites and environment setup, guiding them through the necessary code and configurations to integrate the functionality.

Moreover, for each code sample, developers can access the corresponding GitHub repository and clone a sample application. These repositories also include a README file with instructions for configuring, running, and deploying the app.

Alternatively, more experienced developers may start from scratch with a blank app instead of using a sample app.

This approach caters to developers with varying skill levels and ensures they have the necessary guidance and resources.

Best Practices

Provide guidelines, recommendations, and industry standards

Case Study

Google

The Google Maps API provides access to mapping and location-related services, allowing developers to integrate interactive maps, geolocation, and other features.

Google includes guidelines for efficient resource management, caching strategies, authentication and access control, error handling, and user interface design principles.

Insight: UX-focused best practices

In addition to technical aspects, Google emphasizes the importance of user experience, security, and scalability in its best practices.

Docs Checklist

- ✓ Cover comprehensive API best practices
- ✓ Tailor best practices to target audiences
- ✓ Keep best practices up-to-date with evolving standards
- ✓ Provide examples for implementing best practices

Best practices optimize user experience, security, performance, compatibility, and maintainability

The Google Maps API's best practices stand out for their user-centric approach, focusing on delivering a seamless and intuitive mapping experience. By emphasizing visually appealing and interactive maps, optimizing performance, and incorporating user-friendly features like custom markers, info windows, and responsive design, developers can create applications that provide users with a smooth and engaging mapping experience.

In addition to user experience, the best practices documentation caters to various use cases, offering specific guidelines and recommendations tailored to different application domains. Whether developers are building navigation apps, geolocation services, fleet management systems, or local search functionalities, they can find detailed guidance to address their specific use case's unique requirements and considerations.

Another notable aspect is Google's commitment to security and privacy. It offers comprehensive guidelines for securing API keys, protecting user data, and implementing appropriate authentication and authorization mechanisms.

To ensure relevancy, Google Maps API keeps its best practices documentation up-to-date with the latest advancements and industry standards. As technology evolves and Google releases new versions, the documentation reflects these changes, providing accurate and timely information. Developers can rely on the best practices to stay aligned with industry standards and leverage the current capabilities of the Google Maps API.

FAQs

Group related questions together and link to extended documentation

Case Study

SendGrid

SendGrid is a cloud-based email delivery and management platform that enables businesses to send transactional and marketing emails at scale.

SendGrid groups FAQs by application area, then rather than provide answers as static content, it provides links to extended documentation.

Insight: Dynamic linking in FAQs

SendGrid's links are dynamic; the system automatically generates and groups them.

Docs Checklist

- ✓ Identify common queries from interaction points
- ✓ Group by topic or theme
- ✓ Continuously evaluate user feedback for new questions
- ✓ Link to 'deep-dive' sections

Frequently asked questions (FAQs) address common queries, provide quick solutions, and can potentially reduce support costs

In addition to addressing general inquiries, SendGrid's FAQs cover all its application areas, including transactional emails, marketing campaigns, user notifications, and subscription management. SendGrid organizes its FAQs into these application areas and then delves into specific challenges or considerations developers may encounter when integrating particular functionality.

SendGrid presents FAQs as actionable tasks and links to the extended documentation for best practices, use cases, code tutorials, and API references to help developers integrate its email delivery features.

SendGrid dynamically generates FAQs to ensure they remain in sync with documentation updates for new releases. In contrast, FAQs are often static content that requires error-prone manual updates that go out-of-date quickly.

Through its FAQs, SendGrid acknowledges the complexity of its API and the need to provide address queries for each application area.

API Reference

Go beyond syntax, show the possibilities

Case Study

elasticsearch

The Elasticsearch API is a search and analytics engine for handling and analyzing large volumes of data.

Instead of focusing on syntax alone, Elastic's code samples show how to combine query components to accomplish common tasks.

Insight: Filtering possibilities

Elastic's API reference provides examples of the multitude of ways you can filter responses for each endpoint.

Docs Checklist

- ✓ Comprehensively cover API elements
- ✓ Show how to accomplish tasks for each functionality
- ✓ Explore the breadth of ways to customize requests
- ✓ Link to related conceptual documentation

API reference shows how to combine query components to accomplish tasks

Per the norm, the Elasticsearch API reference covers all available query components like filters, sorting options, pagination parameters, and search keywords. However, it further shows how to accomplish specific tasks using combinations of query components for each endpoint.

For example, the documentation provides code snippets that show how to build a query with multiple filters, range queries, and aggregations.

It explains each query component's purpose, syntax, and how to use them to influence the API response in specific ways.