

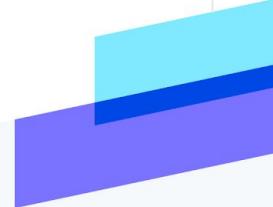
stripe

All Things Open 2022

Building Markdoc

A powerful, flexible, Open Source Markdown
framework

@irreverentmike



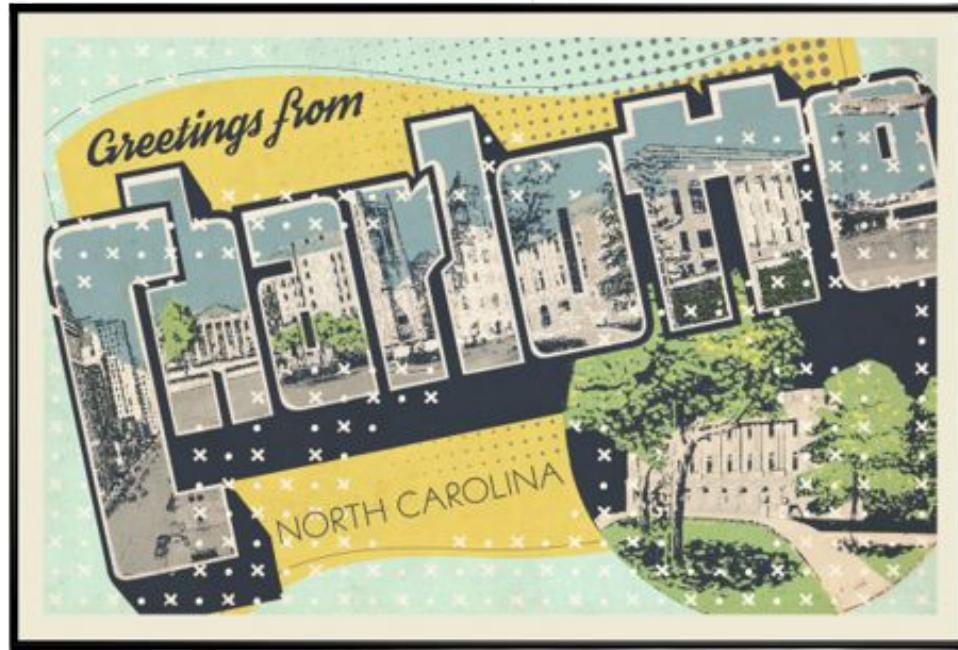
Hi, I'm Mike Bifulco



Mike Bifulco

Developer Advocate, Stripe

@irreverentmike



kylemosher.com



Background

I ❤ OSS

APIs you won't hate

Largest community of API Developers on the web.
We publish a podcast, books, and OSS tools,
articles, and resources for API developers.

apisyouwonthate.com



@irreverentmike

APIs
you won't hate

FEATURED ARTICLE

Testing Spectral Style Guides with Jest

written by Phil Sturgeon

Make sure your automated API style guide actually works, and doesn't break as you improve it over time.

[Read more →](#)

BOOKS FOR API DEVELOPERS

Build APIs You Won't Hate

Everyone and their dog wants an API, so you should probably learn how to build them.

[Get the book](#)



Surviving Other People's APIs

Designing the world API is only half the story; the other half needs to interact with it.

[Get the book](#)

GYMNASIUM

Free, online courses for digital, creative, and communications professionals to equip them with in-demand skills and advance their careers.

thegymnasium.com



@irreverentmike

The screenshot shows the Gymnasium website's homepage. At the top right, there are navigation links for 'Courses', 'Jobs', 'About', and a blue 'Log In' button. Below the navigation is a dark sidebar containing a 'NEW ARTICLE' section. The article title is 'ADVOCATING FOR ACCESSIBILITY WITH AARON GUSTAFSON'. Below the title, a blurb reads: 'If you're not sure where to get started with creating a more inclusive web, you're not alone. (Hint: It starts with awareness.) Learn the fundamentals of accessibility from Aaron Gustafson of Microsoft in our new article.' A blue 'Read Now' button is located at the bottom of this section. The main content area below the sidebar has a white background and features a section titled 'HOW GYMNASIUM WORKS' with three columns: 'New Skills', 'Career Opportunities', and 'Lifelong Learning'. Each column contains a brief description and a small image. At the very bottom of the page, there is a small legal notice about cookie usage.

our experience. By continuing to visit this site you agree to our use of cookies. Find out more about how we use cookies in our [Privacy Policy](#).

mikebifulco.com

Articles, podcast & newsletter for react devs & product designers building great things.

mikebifulco.com

stripe

@irreverentmike

co

Blog Work About Newsletter

growth hack: Publish newsletters to your Next.js site with ConvertKit API

September 22, 2022

#next.js #convertkit #tutorial #api

ConvertKit API
Publish past newsletters to your Next.js site

mikebifulco.com



I publish a newsletter called Tiny Improvements

[My little newsletter](#) is all about making small, incremental improvements to your work

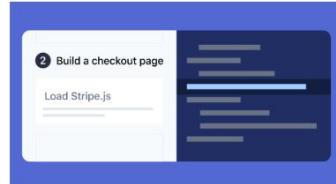
Developer essentials

```
$ stripe payment_intents create --amount=100 --currency=usd
{
  "id": "pi_1G1Z7tIuK0OPCijjmY8yMM",
  "object": "payment_intent",
  "amount": 100,
  "amount_capturable": 0,
  "amount_received": 0,
  "application": null
}
```

STRIPE COMMAND LINE (CLI)

Build, test, and manage your integration with Stripe directly from your terminal. With the CLI you can:

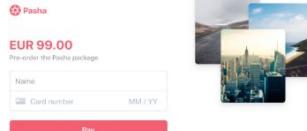
- Create, retrieve, update, and delete API objects
- Tail API request logs in real time
- Securely test webhooks without third-party tunneling software

[INSTALL THE CLI](#)

DOCUMENTATION

Explore our guides and examples to integrate Stripe. Our documentation covers the essentials:

- **Payments** - Build an integration to accept payments online or in person
- **Business operations** - Monitor, protect, and report on your money
- **Financial services** - Move, control, and borrow money with Stripe

[READ THE DOCS](#)

STRIPE SAMPLES

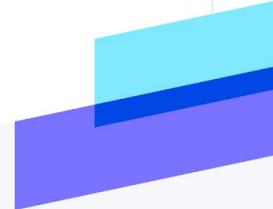
Build with sample code designed to help you get started with your Stripe integration. Learn how to:

- Quickly [accept one-time payments](#) with Stripe Checkout
- Combine Checkout and Billing for [fast subscription pages](#)
- Accept a one-time payment on [web](#), iOS, or Android.

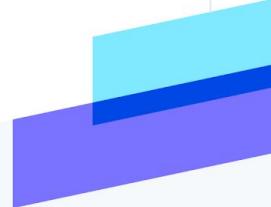
[EXPLORE SAMPLES](#)

Agenda

- 1 History of Stripe docs
- 2 The Markdoc story
- 3 Using Markdoc
- 4 How it all works
- 5 Contributing



stripe



[Dashboard](#)[Documentation](#)[Help & Support](#)[Sign In](#)

Payments for developers

Stripe makes it easy to start accepting credit cards on the web today.

[Get Started with Stripe](#)

Stripe Connect! We are excited to announce that we've launched Stripe Connect. [Read more ▾](#)

Why you'll love using Stripe



Full-stack payments

You don't need a merchant account or gateway. Stripe handles everything, including storing cards, subscriptions, and direct payouts to your bank account.

[Dashboard](#) [Documentation](#) [Help & Support](#) [Sign In](#)

GETTING STARTED

- [Payment Forms](#)
- [Creating Charges](#)
- [Subscriptions](#)
- [Your Account](#)

REFERENCES

- [Stripe.js](#)
- [Button](#)
- [Webhooks](#)
- [Subscription Billing](#)
- [Testing](#)
- [Examples](#)
- [API Libraries](#)
- [Full API Reference](#)

APPS

- [Overview](#)
- [Getting Started](#)
- [Connecting with OAuth](#)
- [OAuth Reference](#)

FAQ

- [General](#)
- [Pricing](#)
- [Disputes](#)
- [...](#)

Documentation

Below you'll find tutorials that will teach you how to use Stripe, and reference documentation for all the moving parts.

Get started with Stripe

Start accepting payments in minutes.



Financial infrastructure for the internet

Millions of companies of all sizes—from startups to Fortune 500s—use Stripe's software and APIs to accept payments, send payouts, and manage their businesses online.

[Start now >](#)[Contact sales >](#)

ROCKET RIDES

Search

Today

Gross volume **\$3,528,198.72** August 18 **\$2,931,556.34**
1:00 PM 1:00 PM

12:00 AM Now

Reports summary
Last 4 weeks **+4.6%** July 18 - August 18 Compared to previous

Gross volume **\$4,542,345.45** Net volume from sales **\$4,180,332.54**
\$4,062,124.33 Jul 18 Aug 18 Jul 18

Finance infrastructure for the internet

Millions of companies of all sizes—from startups to Fortune 500s—use Stripe's software and APIs to accept payments, send payouts, and manage their businesses online.

[Start now >](#)[Contact sales >](#)

Global Payments

Accept payments online, in person, or through your platform.

Revenue and Financial Management

Automate revenue collection and finance.

Banking-as-a-Service

Embed financial services in your platform or product.



Payments

Online payments



Checkout

Pre-built payments page



Elements

Customizable payments UIs



Payment Links

No-code payments



Radar

Fraud & risk management



Connect

Payments for platforms



Billing

Subscription management



Invoicing

Online invoices



Terminal

In-person payments

 Q. Search

Today

Gross volume

\$3,528,198.72

1:00 PM

August 18

\$2,931,556.34

1:00 PM

12:00 AM

Now

Reports summary

Last 4 weeks July 18 - August 18

Compared to previous

Gross volume +4.6%

\$4,542,345.45

\$4,062,124.33

Net volume from sales

\$4,180,332.54

Card Information

Number				
MM / YY				

Country or region

United States

ZIP

Pay

Feedback +



Overview ←

▶ Online payments

▶ Products and prices

▶ Invoicing

▼ Subscriptions

Overview

How subscriptions work

Recurring pricing models

GET STARTED

Quickstart

Integrate a SaaS business

Embed a pricing table

Design an integration

Build a subscriptions integration

Migrate subscriptions to Stripe

BILLING RESOURCES

Coupons

Customers

Subscriptions

Subscription invoices

MANAGE SUBSCRIPTIONS

Change subscriptions

Usage-based billing

Use trial periods

United States

English (United States)

1 Set up Stripe

Install the Stripe client of your choice:

[Ruby](#) [Python](#) [PHP](#) [Java](#) [Node](#) [Go](#) [.NET](#)

Command Line

```
# For detailed setup, see our quickstarts at https://stripe.com/docs/development/quickstart
npm install stripe --save
```

Install the Stripe CLI (optional). The CLI provides [webhook](#) testing, and you can run it to create your products and prices.

[homebrew](#) [Install from source](#)

Command Line

```
brew install stripe/stripe-cli/stripe
```

```
# Connect the CLI to your dashboard
stripe login
```

For additional install options, see [Get started with the Stripe CLI](#).

2 Create the pricing model

Dashboard or Stripe CLI

Create your [products](#) and their [prices](#) in the Dashboard or with the Stripe CLI.

This example uses a fixed-price service with two different service-level options: Basic and Premium. For each service-level option, you need to create a [product](#) and a recurring [price](#).

[Code quickstart →](#)

ON THIS PAGE

What you'll build

[Set up Stripe](#)

Create the pricing model

Create a Checkout Session

Provision and monitor subscriptions

Configure the customer portal

Create a portal Session

Send customers to the customer portal

Test your integration

[Overview](#)[Online payments](#)[Products and prices](#)[Invoicing](#)[Subscriptions](#)[Overview](#)[How subscriptions work](#)[Recurring pricing models](#)**GET STARTED**[Quickstart](#)[Integrate a SaaS business](#)[Embed a pricing table](#)[Design an integration](#)**Build a subscriptions integration**[Migrate subscriptions to Stripe](#)**BILLING RESOURCES**[Coupons](#)[Customers](#)[Subscriptions](#)[Subscription invoices](#)**MANAGE SUBSCRIPTIONS**[Change subscriptions](#)[Usage-based billing](#)[Use trial periods](#)

United States

English (United States)

1 Set up Stripe

Install the Stripe client of your choice:

Ruby Python PHP Java Node Go .NET

Command Line

```
# For detailed setup, see our quickstarts at https://stripe.com/docs/development/quickstart
npm install stripe --save
```

Install the Stripe CLI (optional). The CLI provides [webhook](#) testing, and you can run it to create your products and prices.

homebrew Install from source

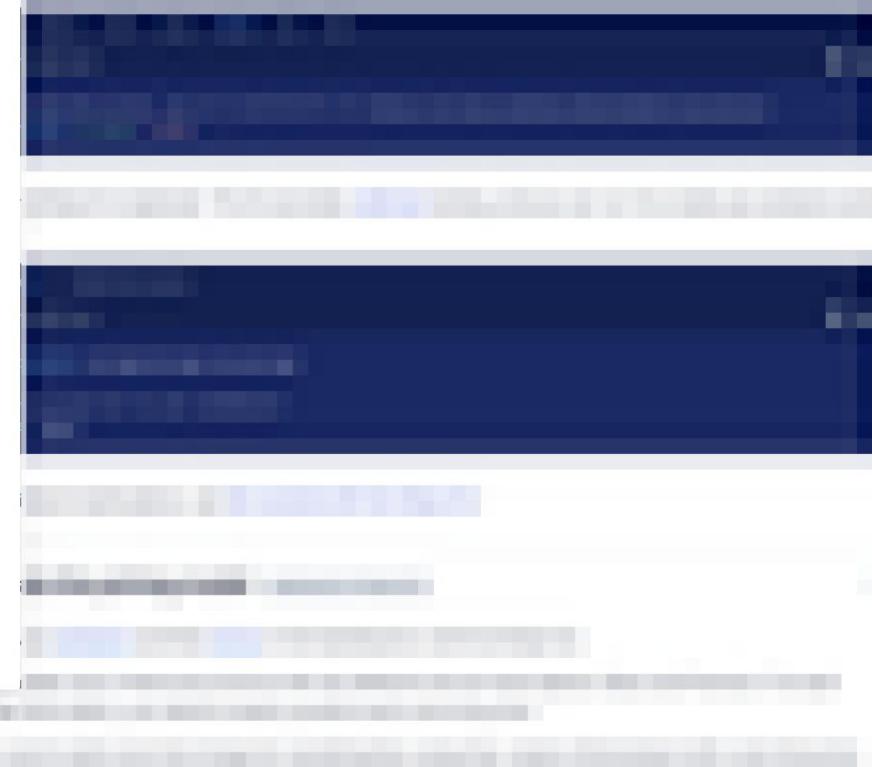
Command Line

```
brew install stripe/stripe-cli/stripe

# Connect the CLI to your dashboard
stripe login
```



Australia	Austria
Belgium	Brazil
Bulgaria	Canada
Croatia	Cyprus
Czech Republic	Denmark
Estonia	Finland
France	Germany
Gibraltar	Greece
Hong Kong	Hungary
India	Ireland
Italy	Japan
Latvia	Liechtenstein
Lithuania	Luxembourg
Malaysia	Malta
Mexico	Netherlands
New Zealand	Norway
Poland	Portugal
Romania	Singapore
Slovakia	Slovenia
Spain	Sweden
Switzerland	Thailand
United Arab Emirates	United Kingdom
United States	
United States	



Available for businesses in 47 countries.
Accept payments from anywhere in the world.

 Australia	 Ireland	 Spain
 Austria	 Italy	 Sweden
 Belgium	 Japan	 Switzerland
 Brazil	 Latvia	 Thailand
 Bulgaria	 Liechtenstein	 United Arab Emirates
 Canada	 Lithuania	 United Kingdom
 Croatia	 Luxembourg	 United States
 Cyprus	 Malaysia	
 Czech Republic	 Malta	
 Denmark	 Mexico	
 Estonia	 Netherlands	
 Finland	 New Zealand	
 France	 Norway	
 Germany	 Poland	
 Gibraltar	 Portugal	
 Greece	 Romania	
 Hong Kong	 Singapore	
 Hungary	 Slovakia	
 India	 Slovenia	

Request an invite

Sign up to be notified about your country.

We'll email you as soon as we launch.

Share your email so Stripe can send you product updates.

your@email.com

Select countr... ▾

You can unsubscribe at any time. [Read our privacy policy](#)

NOTIFY ME

Stripe Atlas

No matter what country you're based in, you can use [Atlas](#) to easily incorporate a U.S. company. Once you form a U.S. company, you can set up a U.S. bank account and activate payments with Stripe.

[Get started ▶](#)

Overview



✓ Online payments

Overview

How cards work

Quickstart

Accept a payment

> Share payment links

> Faster checkout with Link

> More payment scenarios

> U.S. and Canadian cards

> Products and prices

> Invoicing

> Subscriptions

> In-person payments

> Multiparty payments

> After the payment

> Add payment methods

> Stripe Checkout

> Stripe Elements

> About the APIs

> Regulation support

> Implementation guides

United States

English (United States)

Checkout Sessions expire 24 hours after creation.

After creating a Checkout Session, redirect your customer to the URL [API](#) returned in the response.

Ruby Python PHP Java Node Go .NET

```
1 // This example sets up an endpoint using the Express framework.
2 // Watch this video to get started: https://youtu.be/rPR2aJ6XnAc.
3
4 const express = require('express');
5 const app = express();
6 const stripe = require('stripe')('sk_test_4eC39HqLyjWDarjtT1zdp7dc')
7
8 app.post('/create-checkout-session', async (req, res) => {
9   const session = await stripe.checkout.sessions.create({
10     line_items: [
11       {
12         price_data: {
13           currency: 'usd',
14           product_data: {
15             name: 'T-shirt',
16           },
17           unit_amount: 2000,
18         },
19         quantity: 1,
20       },
21     ],
22     mode: 'payment',
23     success_url: 'https://example.com/success',
24     cancel_url: 'https://example.com/cancel',
25   });
26
27   res.redirect(303, session.url);
28 });
29
30 app.listen(4242, () => console.log(`Listening on port ${4242}!`));
```

Code quickstart →

ON THIS PAGE

Set up Stripe

Redirect your customer to Stripe Checkout

Show a success page

Additional testing resources

Create products and prices upfront

Existing customers

Prefill customer data

Save payment method details

Separate authorization and capture

Let customers manage their account

RELATED GUIDES

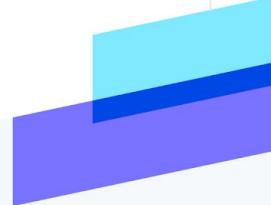
More Payment Scenarios

How Cards Work

```
1 // This example sets up an endpoint using the Express framework.
2 // Watch this video to get started: https://youtu.be/rPR2aJ6XnAc.
3
4 const express = require('express');
5 const app = express();
6 const stripe = require('stripe')('sk_test_4eC39HqLyjWDarjtT1zdp7dc')
7
8 app.post('/create-checkout-session', async (req, res) => {
9   const session = await stripe.checkout.sessions.create({
10     line_items: [
11       {
12         price_data: {
13           currency: 'usd',
14           product_data: {
15             name: 'T-shirt',
16           },
17             unit_amount: 2000,
18           },
19             quantity: 1,
20           },
21         ],
22         mode: 'payment',
23         success_url: 'https://example.com/success',
24         cancel_url: 'https://example.com/cancel',
25       }):

```

There's a lot here.



Early days: ERB

```
require "erb"

# Create template.
template = %q{
  From: Lorem Ibsen <lorem@example.com>
  To: <%= to %>
  Subject: Addressing Needs

  <%= to[/\w+/] %>:
```

Just wanted to send a quick note assuring that your needs are
beaddressed.

I want you to know that my team will keep working on the issues,
especially:

```
<%# ignore numerous minor requests -- focus on priorities %>
% priorities.each do |priority|
  * <%= priority %>
% end
```

Thanks for your patience.

From: Lorem Ibsen <lorem@example.com>
To: Community Spokesman <spokesman@ruby_community.org>
Subject: Addressing Needs

Community:

Just wanted to send a quick note assuring that your needs are being

I want you to know that my team will keep working on the issues, es

- * Run Ruby Quiz
- * Document Modules
- * Answer Questions on Ruby Talk

Thanks for your patience
Lorem Ibsen

What we learned from ERB

Benefits

Access to data models on page was easy

All Ruby namespaces available

Dynamic code examples

Cons

Scalability was hard

Application views vs declarative content

Complex pages easily broken

Hard to debug

Each page implemented differently

Technical Writers don't want to debug

Goal: Rebuild docs with these learnings

Goals for rebuilding stripe.com/docs

New Architecture

Learn from past experiences

Make debugging easier

Simplify

Simplify DevX for Docs team

Authoring process for Tech Writers

Maintain dynamic features

Separate content from Ruby logic

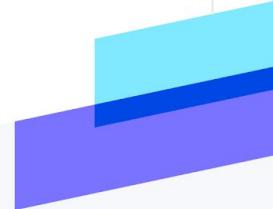
Gain Sorbet typing & static analysis

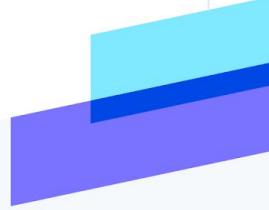
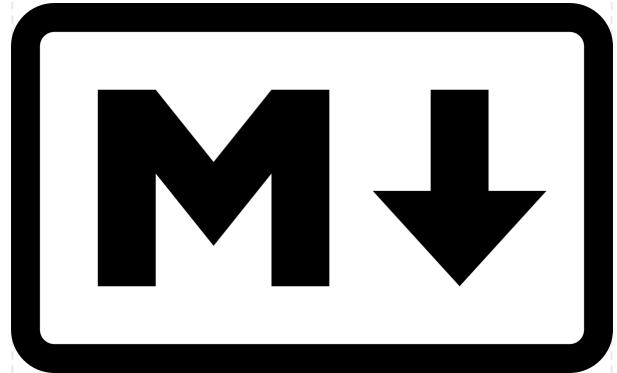
All ruby code in ruby-land

Content independent from ruby*

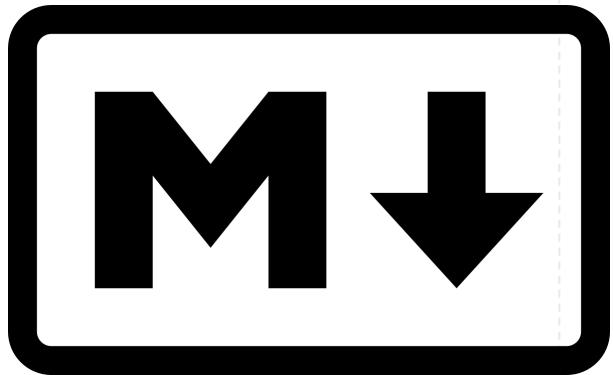
Agenda

- 1 History of Stripe docs
- 2 **The Markdoc story**
- 3 Using Markdoc
- 4 How it all works
- 5 Contributing





Built for a technical team



Markdown
=====

Version 1.0.1 - Tue 14 Dec 2004

by John Gruber
<http://daringfireball.net/>

Introduction

Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).

Thus, "Markdown" is two things: a plain text markup syntax, and a software tool, written in Perl, that converts the plain text markup to HTML.

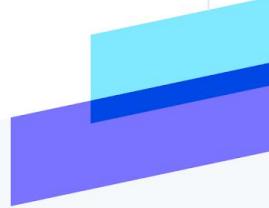
Markdown works both as a Movable Type plug-in and as a standalone Perl script -- which means it can also be used as a text filter in BBEdit (or any other application that supports filters written in Perl).

Full documentation of Markdown's syntax and configuration options is available on the web: <<http://daringfireball.net/projects/markdown/>>. (Note: this readme file is formatted in Markdown.)

Installation and Requirements

Markdown requires Perl 5.6.0 or later. Welcome to the 21st Century. Markdown also requires the standard Perl library module `Digest::MD5`.

MARK **DOC**





Ryan Paul
@segphault · [Follow](#)



We created Markdoc to provide a great authoring experience for technical writers and documentation contributors. Today on the [@stripe blog](#), I've shared details about how Markdoc works and how we arrived at some of our design decisions:

stripe

How Stripe builds interactive docs with Markdoc

stripe.com

How Stripe builds interactive docs with Markdoc

Markdoc delivers a good user experience without compromising the authoring experience. It enables writers to express interactivity and ...

12:56 PM · Sep 13, 2022



264



Reply



Copy link

[Read 2 replies](#)

stripe

How Stripe builds interactive docs with Markdoc

September 13, 2022



Ryan Paul
Docs Product

At Stripe, our [product docs](#) are designed to feel like an application rather than a traditional user manual. For example, we incorporate a user's own API test key into code samples, making it possible to copy and paste code that seamlessly works with the user's own account. We have client-side interactivity, like checklists and collapsible sections. We tailor the content to the individual user, conditionally displaying content based on their location or the Stripe features they use. These features result in a high-quality user experience that reduces friction and contributes to the success of developers.

For these capabilities to have the desired impact we have to make it easy for writers to use them in their content. Delivering a good user experience without compromising the authoring experience required us to develop an authoring format that enables writers to express interactivity and simple page logic without mixing code and content.

Over several years, we learned how to balance interactivity, customization, and authoring productivity while undertaking a major overhaul of our documentation platform.

MARK DOC

MARK DOC

≡

Markdoc is a powerful, flexible, Markdown-based authoring framework.

From personal blogs to massive documentation sites, Markdoc is a content authoring system that grows with you.

[View docs →](#)

Open source

Maintain full control over your code and content. Markdoc is open-source and

In a nutshell

A Markdown-based syntax and toolchain for creating custom documentation sites.

Developer & writer friendly

Supporting body copy goes here. Lorem ipsum dolor sit amet.

More than just docs

Markdoc is a flexible framework for authoring, and can be used for many needs

stripe

Markdoc

Core Concepts

Core Concepts

Nodes

Tags

Attributes

Variables

Functions

Validation

Core Concepts

Nodes

Tags

Attributes

Variables

Functions

Validation

```
# Headers  
  
**Bold**  
  
_Italic_  
  
[Links](/docs/nodes)  
  
![Images](/logo.svg)  
  
Lists  
- Item 1  
- Item 1  
- Item 1  
  
> Quotes  
  
'Inline code'  
  
...  
  
Code fences  
```
```

## Headers

### Bold

### *Italic*

### Links

### Lists

- Item 1
- Item 1
- Item 1

### Quotes

### Inline code

### Code fences

# Core Concepts

Nodes



Tags

```
{% if true %}
```

Attributes

```
{% callout type="note" %}
Tags are composable!
{% /callout %}
```

Variables

```
{% else /%}
```

Functions

```
{% callout type="warning" %}
Tags aren't composable!
{% /callout %}
```

Validation

```
{% /if %}
```

# Core Concepts

Nodes



Tags

```
{% city
 index=0
 name="San Francisco"
 deleted=false
 coordinates=[1, 4, 9]
 meta={id: "id_123"}
 color=$color /%}
```

Attributes

Variables

Functions

Validation

# Core Concepts

Nodes

Tags

Attributes

**Variables**

Functions

Validation



SDK: {`% $sdk %`}

Version: {`% $version %`}

# Core Concepts

Nodes



Tags

```
{% if equals($myVar, "test") %}
The variable $myVar is equal to the string "test".
{% /if %}
```

Attributes

Variables

Functions

Validation

# Core Concepts

Nodes

Tags

Attributes

Variables

Functions

Validation



```
const uppercase = {
 transform(parameters) {
 const string = parameters[0];

 return typeof string === 'string' ? string.toUpperCase() : string;
 }
};

const config = {
 functions: {
 uppercase
 }
};

const content = Markdoc.transform(ast, config);
```

# Core Concepts

Nodes

Tags

Attributes

Variables

Functions

Validation

```
const Banner = ({children, type}) => (
 <aside className={type}>
 <blockquote>{children}</blockquote>
 </aside>
)

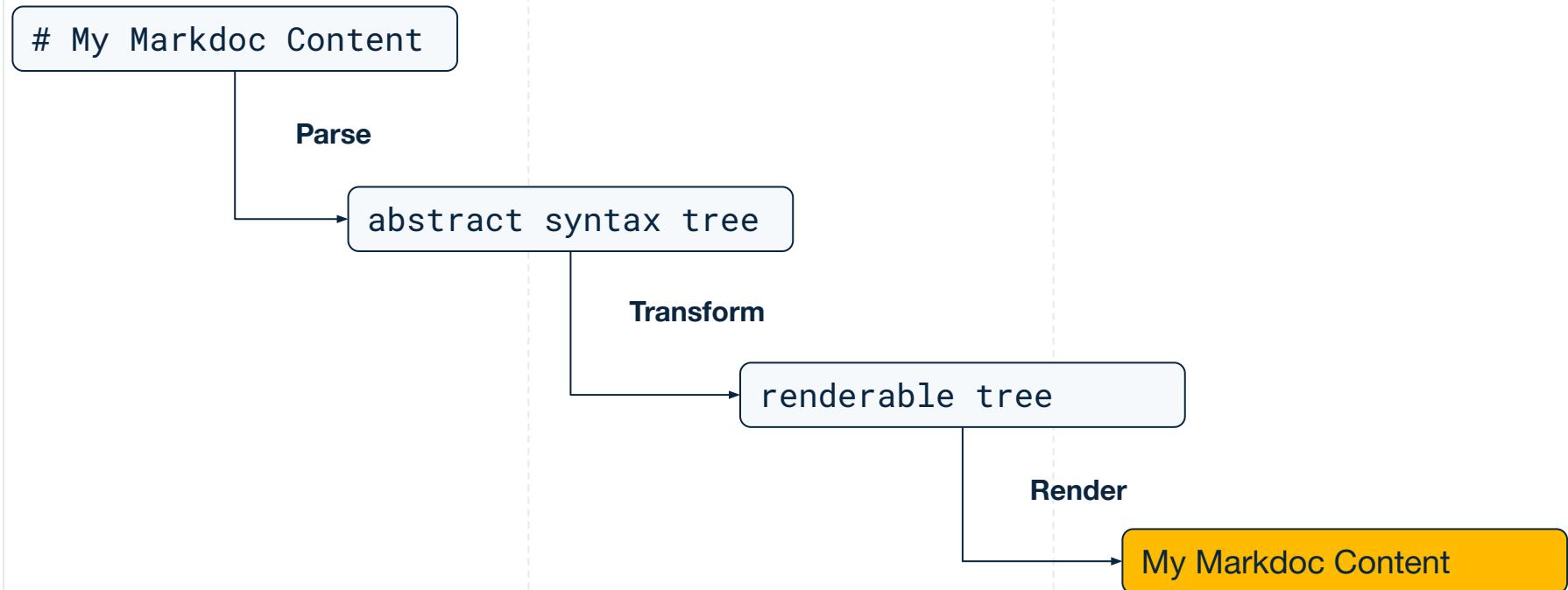
const banner = {
 render: "Banner",
 attributes: {
 type: {
 type: String,
 required: true,
 matches: ["warning", "info", "error"]
 }
 }
}

const config = {
 tags: {
 banner,
 }
}

const ast = Markdoc.parse(doc);
const errors = Markdoc.validate(ast, config);

if (errors) {
 console.log(errors);
}
```

# Rendering



# Putting it all together

```
import React, { useRef } from "https://esm.sh/react@18";
import ReactDOM from "https://esm.sh/react-dom@18";
import Markdoc from 'https://esm.sh/@markdoc/markdoc';

const doc = `
Hello world from Markdoc
You can learn more at [Markdoc.dev](https://markdoc.dev)
`;

const config = {};

const ast = Markdoc.parse(doc);
const content = Markdoc.transform(ast, config);
const children = Markdoc.renderers.react(content, React);

const root = document.getElementById("root");
ReactDOM.createRoot(root).render(children);
```

## Get started

What is Markdoc?

Installation

FAQ

Try it out

## Core concepts

Syntax and schema

Nodes

Tags

Attributes

Variables

Functions

Rendering

Validation

## Integration guides

Common examples

Using with HTML

[Using with Next.js](#)

Using with React

## Advanced concepts

Frontmatter

Partials

# Using Markdoc with Next.js

Using the `@markdoc/next.js` package/plugin allows you to create custom `.md` and `.mdoc` pages in your Next.js apps, and automatically render them with Markdoc.

To get started right away, check out [this starter repo](#). The quickest way to deploy your own version of the starter is by deploying it with [Vercel](#) or [Netlify](#) by clicking one of the buttons below.

▲ Deploy

❖ Deploy to Netlify

## Setup

This guide assumes that you already have Next.js installed. If you're starting from scratch, follow these steps for [getting started with Next.js](#).

Follow these steps to get started with `@markdoc/next.js`.

1. Install `@markdoc/next.js` and `@markdoc/markdoc`:

```
npm install @markdoc/next.js @markdoc/markdoc
```

2. Update your `next.config.js`

```
const withMarkdoc = require('@markdoc/next.js');

module.exports = withMarkdoc({/* options */}){
 pageExtensions: ['md', 'mdoc', 'js', 'jsx', 'ts', 'tsx']
};
```

3. Create a new `.md` file in within `/pages/`, such as `getting-started.md`

```
pages
```

Setup

Options

Schema customization

Tags

Nodes

Functions

Advanced

Frontmatter

Partials

Layouts

Built-in Next.js tags

Comment

Head

Link

Script



Search or jump to...

/ Pull requests Issues Marketplace Explore

Follow

# Markdoc

MARK DOC

The Markdoc toolchain

87 followers

<https://markdoc.dev>

@StripeDev

Verified

Overview

Repositories 5

Projects 1

Packages

People

README.md

MARK DOC

## People

This organization has no public members.  
You must be a member to see who's a part  
of this organization.

## Top languages

JavaScript TypeScript

## Most used topics

markdoc documentation markdown

[Report abuse](#)

+ Feedback



Search

Pull requests Issues Marketplace Explore

Bell icon + User icon

markdoc / markdoc

Public

Watch 23

Fork 92

Starred 5.5k

Code

Issues 7

Pull requests

Discussions

Actions

Projects

Security

Insights

main

4 branches

15 tags

Go to file

Add file

mfix-stripe 0.2.0 (#283)

411e84e 15 hours ago 93 commits

.github

Fix test failures from #49 (#51)

5 months ago

.vscode

Patch parseTags to avoid crashing on unopened tag (#94)

5 months ago

patches

Initial commit

6 months ago

spec

Adds comment syntax to Markdoc (#198)

last month

src

Fix bugs in formatter (#280)

4 days ago

.eslintrc.json

Initial commit

6 months ago

.gitignore

update package-lock.json (#52)

5 months ago

.prettierrc

Initial commit

6 months ago

LICENSE

Initial commit

6 months ago

README.md

Fix up dependencies to use optional and peer correctly (#281)

4 days ago

build.js

add prebuild and postgrammar scripts (#37)

6 months ago

global.d.ts

fix delimiters bug (#48)

5 months ago

index.ts

make format public (#284)

15 hours ago

package-lock.json

0.2.0 (#283)

15 hours ago

package.json

0.2.0 (#283)

15 hours ago

## About

A powerful, flexible, Markdown-based authoring framework.

markdoc.dev

react toolchain markdown  
documentation authoring markdoc

Readme

MIT license

Code of conduct

Security policy

5.5k stars

23 watching

92 forks

## Releases 15

0.2.0 Latest

15 hours ago

+ 14 releases

## Used by 432

[For more information](#)

# Recommended Reading

[markdoc.dev](#)

[Stripe.dev](#)

[stripe.com/blog/markdoc](#)

[daringfireball.net/projects/markdown](#)

[github.com/markdoc](#)

# Thank you!



**Mike Bifulco**

Developer Advocate, Stripe

@irreverentmike

**stripe**