

THE RISE AND FALL

OF THE GO WEB FRAMEWORK



**How I learned to abandon my
ego and embrace the
deliberate pursuit of less.**



**Which Go Web
Framework
should I use?**



1

Year Ago





Today





Antiframework



1. The Rise

2. The Fall

3. The Rebirth



Introduction

About Me







[go-martini/martini](#)

Classy web framework for Go

6,675 ★



[cli](#)

A small package for building command line ap...

2,454 ★



[negroni](#)

Idiomatic HTTP Middleware for Golang

2,291 ★



[gin](#)

Live reload utility for Go web servers

950 ★

GopherCasts

Learn Go



Deploying Web Apps to Heroku

#10 - Deploying Web Apps to



Auth: Part 3

#9 - Auth: Part 3



Auth: Part 2

#8 - Auth: Part 2



BUILDING WEB APPS WITH GO



Jeremy Saenz



```
// Use adds a middleware Handler
Middleware Handlers are invoked in the order that they are
func (m *Martini) Use(handler Handler) {
    validateHandler(handler)
    m.handlers = append(m.handlers, handler)
}

// for a Martini instance, useful if you want to control the
func (m *Martini) Use(handler Handler, request Request) {
```

Edit

Essential Go

Learn the essential features of the Go programming language through practical examples and creative exercises.

▶ Resume Course



7 / 22 Lessons Complete

 Lessons

 News



Introduction



Welcome



Latest News

8 March, 2015

Bonus! The Concurrency

Part 1

The Rise



Back in 2012...





On Distributing Command line Applications: Why I switched from Ruby to Go

Jul 21, 2013 · 4 minute read · [45 Comments](#)

[go](#) [golang](#) [cli](#) [ruby](#)

For as long as I have been using it, my go-to language for creating CLI applications was Ruby. After all, Ruby is a robust, mature language with many things to make us developers want to gyrate with joy (okay maybe that is just me).

To this day I still hold the opinion that Ruby provides the most elegant amount of expression when it comes to my everyday programming activities. Whether it is the language or the community that fosters clean code, I feel like an artisan every time I sit down to write an application or library in Ruby.

There is something so elegant and beautiful about creating a command line application with the following code:

```
desc "A todo application"
```



codegangsta/cli



What next?



codegangsta/inject



INJECT



ALL THE THINGS

memegenerator.net



Martini.

Classy Web Development in Go



Martini.

Classy web development in Go. Surprisingly simple. Incredibly productive.

CODE

DEMO

Star it, fork it, **Contribute** to it.



The Demo



Simple
Clean
Beautiful



```
func Auth(res http.ResponseWriter, req *http.Request) {
    if req.Header.Get("X-API-KEY") != "secret-key" {
        res.WriteHeader(401)
    }
}
```





Martini Contrib

Community contributed Martini handlers and packages

<http://martini.codegangsta.io> jeremy.saenz@gmail.com

Filters ▾

Find a repository...

+ New repository

encoder

Go ★ 15  14

Martini encoder service for rendering data in several formats and content negotiation.

Updated a day ago

csrf

Go ★ 25  6

CSRF protection for Martini applications

Updated 11 days ago

oauth2

Go ★ 90  42

Allows your Martini application to support user login via an OAuth 2.0 backend.

People

22 >



Invite someone

Teams

22 >

Jump to a team

Owners

2 members · 21 repositories



Part 2

The Fall



Stephen Searles



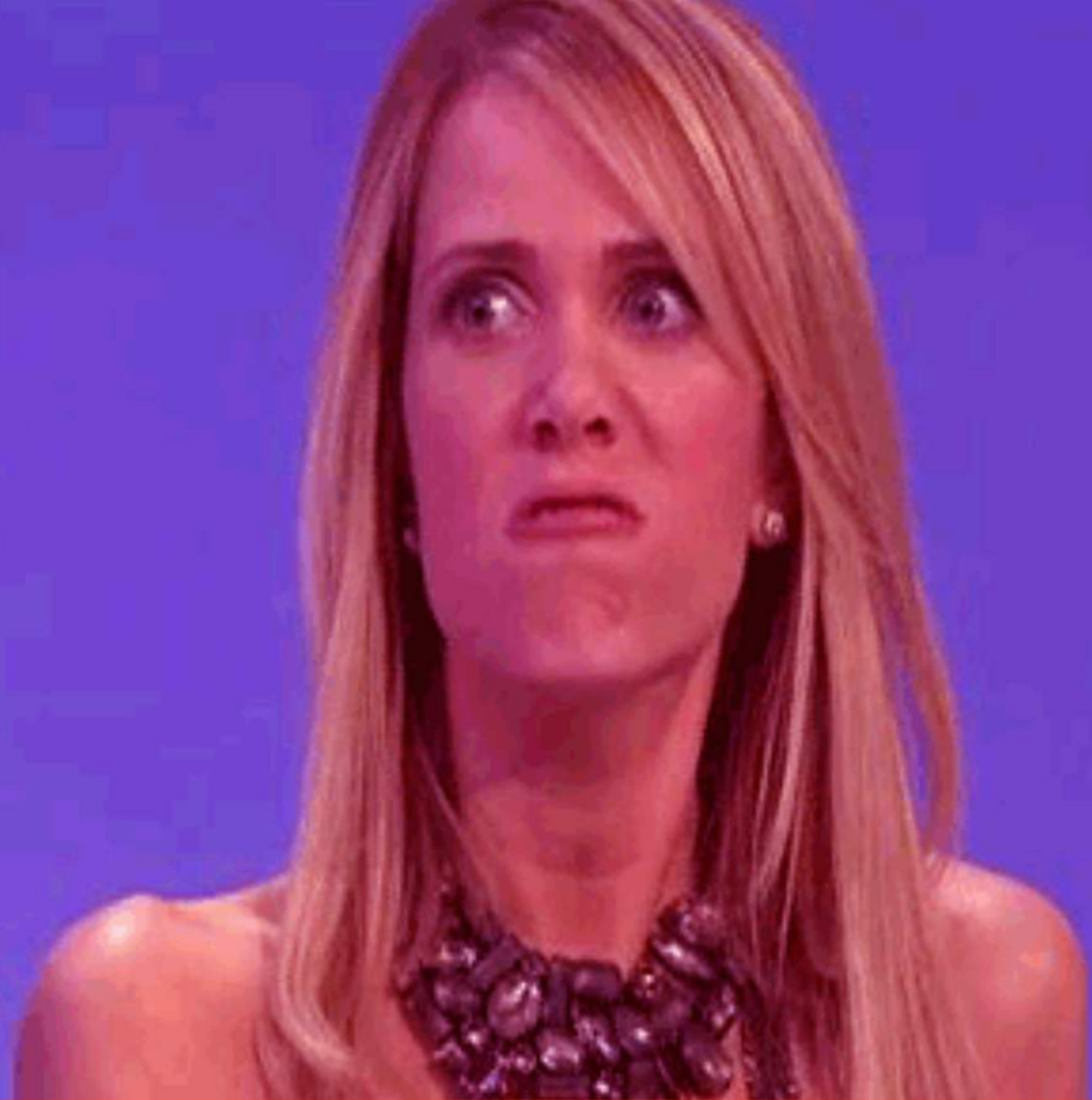
Three reasons you should not use Martini

TL;DR: Martini makes Go magic, but the whole point of Go is to be simple and not magic. It's the wrong direction for the language and ecosystem, so try something like [muxchain](#) instead.

Sorry codegangsta; I really tried to write this post without calling out Martini, but the point just wasn't coming across.

Martini has gained a tremendous amount of attention in a short time. I think it might have even gotten a few people to give Go a try. I think you shouldn't use it and I have several reasons.







Kelsey Hightower

@kelseyhightower

 Follow

Three reasons you should not use Martini, or why Go does not need web frameworks. Honest criticism. stephensearles.com/?p=254 #golang



RETWEETS
27

FAVORITES
31



7:32 AM - 18 May 2014



Matt Aimonetti @mattetti · May 18



My Thoughts on Martini

May 19, 2014 · 5 minute read · [24 Comments](#)

[golang](#) [web](#) [martini](#) [negroni](#)

It's never easy to take honest criticism, especially when the target of the criticism is considered your 'baby'. Earlier this week, [Stephen Searles posted an entirely honest review](#) of Martini and how, despite the popularity and hype, there are numerous reasons why Martini should not be used.

I have been asked by many people what my thoughts are regarding the reasons not to use Martini. I figure that it would be best to compile these thoughts into a blog post so I can shed some light on my opinions and how they may have changed over time. So here we go, my reflections on Martini...

Martini's Popularity was a HUGE surprise

Arguably, many people at the time were young Go developers.

This may be an obvious one. I was not expecting the positive attention that Martini



NEGRONI

An idiomatic approach to HTTP middleware in Go.

[GET THE CODE](#)



```
n := negroni.Classic()  
n.Use(Middleware)  
  
n.UseHandler(router)  
n.Run(":3000")
```



Lessons learned?



Humility

Empathy

Community Focus



Part 3

The Rebirth



Essentialism

The Disciplined Pursuit of Less





http.Handler

The missing link all along



Practical Examples



Negroni

Idiomatic HTTP Middleware



codegangsta/negroni



httprouter

A high performance HTTP request router that scales well



julienschmidt/httprouter



Render

Easily render JSON, XML, binary data, and HTML templates



unrolled/render



And More...

gorilla/context

gorilla/sessions

gorilla/schema

codegangsta/gin

markbates/goth



Takeaway?



Be Humble

Always Improve

Pursue Less



@codegangsta

