ohler55 / agoo Public

Code · Issues 2 · Pull requests · Discussions · Actions · Security · Insights

8 Branches · 73 Tags · Go to file · Go to file · Code About ···

ohler55 Bump version ✓ 3242958 · 2 months ago 320 Commits

.github	Nil root okay (#131)	last year
bin	In a bind (#124)	2 years ago
example	Fix multi-threaded exit	last year
ext/agoo	A body should be allow…	2 months ago
lib	Bump version	2 months ago
misc	Nil root okay (#131)	last year
test	A body should be allow…	2 months ago
.gitignore	Symbol subjects	7 years ago
.yardopts	Nil root okay (#131)	last year
CHANGELOG.md	Release prep	2 months ago
Gemfile	Initial implementation.	8 years ago
LICENSE	Initial implementation.	8 years ago

A High Performance HTTP Server for Ruby

#ruby #rails #graphql #rack #fast #http #benchmark #web #server #ipv6 #clustering #webserver #https #unix-domain-socket #websocket #sse #graphql-server #rails5 #clusters #wabur

Readme

MIT license

Security policy

Activity

924 stars

16 watching

40 forks

Report repository

Releases 1

v2.15.14 Latest

README.md	Update benchmark link	last year
SECURITY.md	Nil root okay (#131)	last year
agoo.gemspec	Nil root okay (#131)	last year
notes	Nil root okay (#131)	last year
tea.yaml	Update tea.yaml	last year

README MIT license Security

Agoo

build passing gem version 2.15.14 downloads 202k lifted!

A High Performance HTTP Server for Ruby

Usage

Rack

```
require 'agoo'

Agoo::Server.init(6464, './root')

class MyHandler
```

on Sep 25

Sponsor this project

ohler55 Peter Ohler

tidelift.com/funding/github/**rubyge…**

Learn more about GitHub Sponsors

No packages published

Contributors 13

Languages

- ● **C** 85.2%
- ● **Ruby** 14.1%
- ● **Other** 0.7%

```
  def call(req)
    [ 200, { }, [ "hello world" ] ]
  end
end

handler = MyHandler.new
Agoo::Server.handle(:GET, "/hello", handler)
Agoo::Server.start()

# To run this example type the following then go to a browser and enter a URL
# of localhost:6464/hello.
#
# ruby hello.rb
```

GraphQL

```
require 'agoo'

class Query
  def hello
    'hello'
  end
end

class Schema
  attr_reader :query

  def initialize
    @query = Query.new()
  end
end

Agoo::Server.init(6464, './root', thread_count: 1, graphql: '/graphql')
Agoo::Server.start()
```

```
Agoo::GraphQL.schema(Schema.new) {
  Agoo::GraphQL.load(%^type Query { hello: String }^)
}
sleep

# To run this GraphQL example type the following then go to a browser and enter
# a URL of localhost:6464/graphql?query={hello}
#
# ruby hello.rb
```

Installation

```
gem install agoo
```

Using Agoo as a server for Rails

As agoo supports rack compatible apps you can use it for rails applications:

Add agoo to the Gemfile:

```
# Gemfile
gem 'agoo'
```

Install bundle:

```
$ bundle install
```

Start rails with agoo as server:

```
$ rails server -u agoo
```

Enjoy the increased performance!

What Is This?

Agoo is Japanese for a type of flying fish. This gem flies. It is a high performance HTTP server that serves static resource at hundreds of thousands of fetches per second. A simple hello world Ruby handler at over 100,000 requests per second on a desktop computer. That places Agoo at about 85 times faster than Sinatra and 1000 times faster than Rails. In both cases the latency was two orders of magnitude lower or more. Checkout the benchmarks.

Agoo supports the Ruby rack API which allows for the use of rack compatible gems such as Hanami and Rails. Agoo also supports WebSockets and SSE.

Agoo is not available on Windows.

Getting Started

Some examples can be found in the `example` directory. Some other documented walk through examples of using Agoo are in the `misc` directory. These examples are described in markdown files. Some like the song example are simple while others like the glue are more complex. The documented walk through examples are:

- flymd - realtime push
- glue - publish and subscribe
- optimize - optimizing static asset use
- push - push with Rack

- rails - faster rails
- song - fun and simple GraphQL

News

- Version 2.11.0 supports GraphQL subscriptions. TLS (SSL,HTTPS) support added. Examples for both. Related, the graphql-benchmark repo was given to the-benchmarker.
- Agoo has a new GraphQL module with a simple, easy to use API. Checkout the hello or song examples. An Instrumental Intro to GraphQL with Ruby is a walk through.
- Agoo takes first place as the highest throughput on web-frameworks benchmarks. Latency was not at the top but release 2.5.2 improves that. The Agoo-C benchmarks it at the top. The fastest web server across all languages.
- Clustered Agoo is ready. For slower application and a machine with multiple cores a significant improvement is performance is realized. The application must be stateless in that no data is shared between workers.
- WebSocket and SSE are supported and a PR has been submitted to updated the Rack spec. Go over to the proposed Rack extension and give it a look and a thumbs-up or heart if you like it.
- Agoo now serves Rails static assets more than 8000 times faster than the default Puma. Thats right, 8000 times faster.

Releases

See file:CHANGELOG.md

Releases are made from the master branch. The default branch for checkout is the develop branch. Pull requests should be made against the develop branch.

Support

Get supported Agoo with a Tidelift Subscription. Security updates are supported.

Links

- *Documentation*: http://rubydoc.info/gems/agoo or http://www.ohler.com/agoo/doc/index.html
- *GitHub repo*: https://github.com/ohler55/agoo
- *RubyGems repo*: https://rubygems.org/gems/agoo
- *WABuR repo*: https://github.com/ohler55/wabur has an option to use Agoo