# Practical ngx_mruby

## Middleware as a Code

GMOペパボ株式会社



# self.introduce



=>
{
  name: “SHIBATA Hiroshi”,
  nickname: “hsbt”,
  title: “Chief engineer at GMO Pepabo, Inc.”,
  commit_bits: [“ruby”, “rake”, “rubygems”, “rdoc”, “tdiary”, “hiki”, “railsgirls”, “railsgirls-jp”, …],
  sites: [“hsbt.org”, ruby-lang.org”, “rubyci.org”, “railsgirls.com”, “railsgirls.jp”],
}




最新ニュース | セミナー・イベント情報 | IRニュース

2015.12.02 お知らせ GMOペパボ

「J-Stock」銘柄選定に関するお知らせ

グローバルサイン認証サイト
SSL secured
クリックして確認
GlobalSign by GMO

株価情報

pepabo.com

# ngx_mruby

# Introduction to ngx_mruby

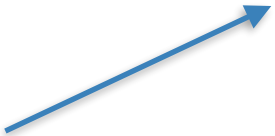
**“ngx_mruby is A Fast and Memory-Efficient Web Server Extension Mechanism Using Scripting Language mruby for nginx.”**

https://github.com/matsumoto-r/ngx_mruby#whats-ngx_mruby

```
location /hello {
    mruby_content_handler /path/to/hello.rb cache;
}
```



In “nginx.conf”!!!

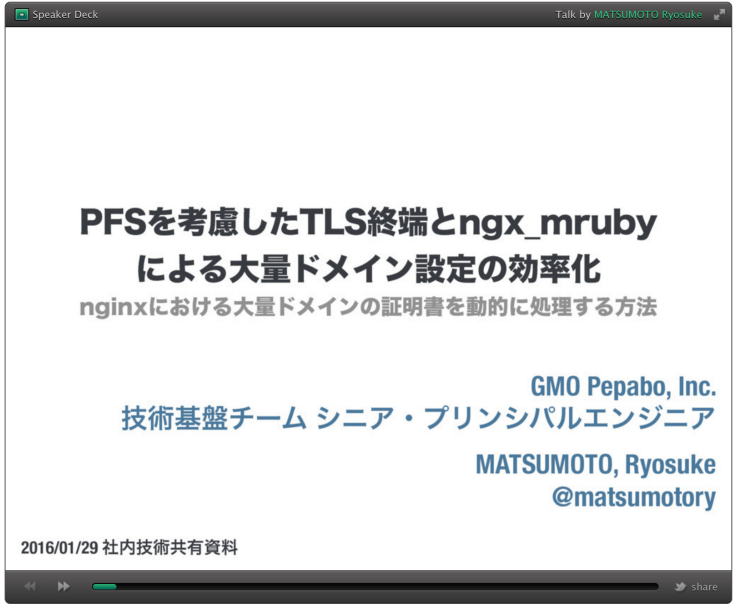


```
location /proxy {
    mruby_set_code $backend '
        backends = [
          "test1.example.com",
          "test2.example.com",
          "test3.example.com",
        ]
        backends[rand(backends.length)]
    ';
}
```

# Why mruby?

See http://hb.matsumoto-r.jp/entry/2016/02/05/140442



# How to build ngx_mruby (and mruby)

**I suggest to try it on OS X or Linux environment. You can change embedded mgem via “build_config.rb” in ngx_mruby repository.**

```
$ git clone https://github.com/matsumoto-r/ngx_mruby
$ git clone https://github.com/nginx/nginx
$ cd ngx_mruby
$ git submodule init && git submodule update

comment-out mruby-redis and mruby-vedis

$ ./configure —with-ngx-src-root=../nginx
$ make build_mruby
$ make
$ cd ../nginx
$ ./objs/nginx -V
```

# Build on docker

## You can get rpm and deb packages via docker and docker-compose

```
$ docker-compose build centos7
$ docker-compose run centos7
=> nginx-ngx_mruby-1.9.3-1.el7.centos.ngx.x86_64.rpm
```

```
$ docker-compose build ubuntu14.04
$ docker-compose run ubutnu14.04
=> nginx-ngx_mruby_1.9.4-1~trusty_amd64.deb
```

https://github.com/hsbt/ngx_mruby-package-builder

## You can install via default package management tool like yum and apt-get above packages.

# Practical example code

https://github.com/hsbt/nginx-tech-talk

# p.228 Return to summation of argument

```
location /sum {
  mruby_set_code $sum '
    result = 0
    for i in 0..Nginx::Var.new.arg_n.to_i do
      result += i
    end
    result
  ';
  return 200 $sum;
}
```

`mruby_set_code` sets return variable to nginx variable.

You can access via Nginx::Var instance and arg_* methods.

# p.234 logging with log_hander

```
location / {
  mruby_content_handler_code '
    Nginx.echo "Hello"
  ';
  mruby_log_handler_code '
    if Nginx::Var.new.status == Nginx::HTTP_OK
      Nginx.errlogger(Nginx::LOG_INFO, "Success")
    else
      Nginx.errlogger(Nginx::LOG_ERR, "Error")
    end
  ';
}
```

log_handler invokes after content_handler

ngx_mruby provides status code with Nginx::* constants.

ngx_mruby provides errlogger method for user logging.

# p.244 internal redirect with ngx_mruby

```
location /internal_redirect {
  mruby_content_handler_code '
    Nginx.redirect "/internal"
  ';
}

location /internal {
  return 200 "internal redirection";
}
```

ngx_mruby provides redirect method for internal redirect.

ngx_mruby can’t use `@internal` style.

# p.246 Rewrite URI using ngx_mruby

```
location /image/ {
  mruby_rewrite_handler_code '
    file = Nginx::Var.new.uri.match(/^\/image\/(.+\.jpg)$/)
    if !file
      return Nginx::HTTP_FORBIDDEN
    end
    url = "/image/jpg/" + file[1]
    Nginx.redirect url
  ';
}
```

You can rewrite uri with rewrite_handler phase.

String#match provides Regexp matcher.

# p.248 Assign to variables with ngx_mruby

```
location / {
  set $data1 "foo1";
  mruby_set_code $data2 "bar1";
  mruby_content_handler_code '
    v = Nginx::Var.new
    v.data1 = "foo2"
    v.data2 = "bar2"
    v.data3 = "buzz2"
    Nginx.rputs "#{v.data1}, #{v.data2}, #{v.data3}"
  ';
}
```

It's same behavior with ngx_lua.

You can use interpolation like Ruby.

# p.250 Reference with request arguments

```
location / {
  mruby_content_handler_code '
    args = Hash[*Nginx::Request.new.args.split("&").map{|arg| arg.split("=")}.flatten]
    args.each do |k, v|
      Nginx.echo "#{k}:#{v}"
    end
  ';
}
```

```
args = Nginx::Request.new.get_uri_args
```

I added helper method like ngx_lua

You can access Nginx::Request#args and modified it for Hash access.

# p.251 Assign variables to arguments

```
location / {
  mruby_content_handler_code '
    args = {"pass" => "ngx_lua"}
    args = args.map{|k,v| "#{k}=#{v}"}.join("&")
    r = Nginx::Request.new
    r.args = args

    Nginx.echo(r.args)
  ';
}
```

You can assign local variable using Nginx::Request#args=

```
r.set_uri_args(args)
```

I added helper method like ngx_lua

# p.253 Reference with post parameters

```
location / {
  mruby_content_handler_code '
    r = Nginx::Request.new
    args = Nginx::Request.new.body

    if !args
      Nginx.echo "failed to get post args."
      return
    end

    Hash[*args.split("&").map{|arg| arg.split("=")}.flatten]
    args.each do |k, v|
      Nginx.echo "#{k}:#{v}"
    end
  ';
}
```

You can access using Nginx::Request#body when post request.

```
args = r.get_post_args
```

I added helper method like ngx_lua

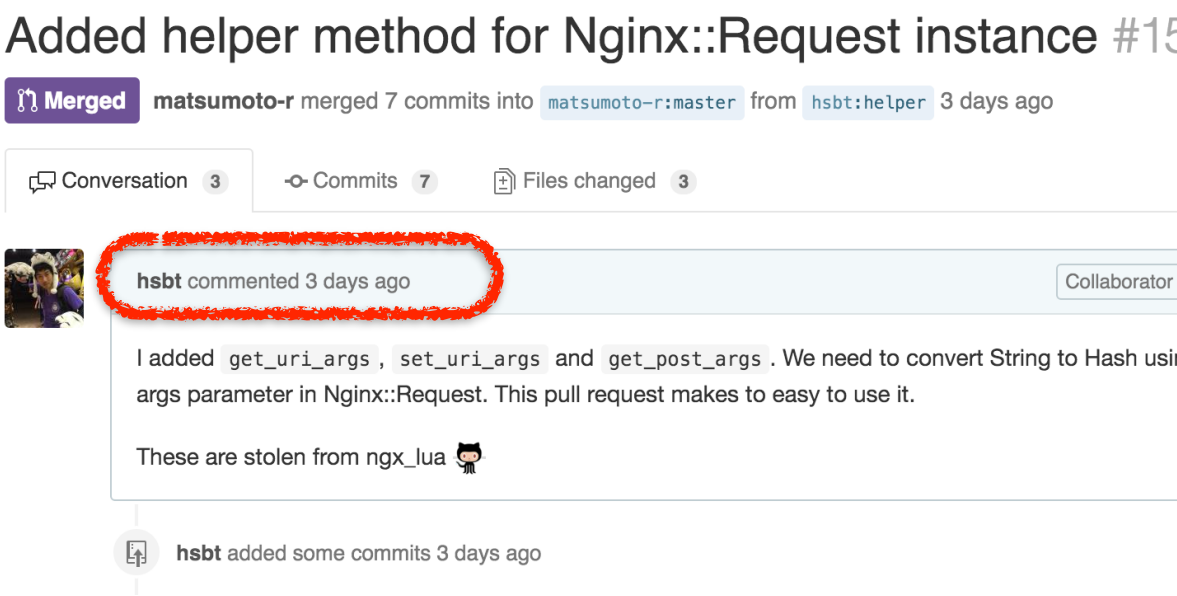
GMOペパボ株式会社

# Rubyists friendly

```
r.get_uri_args
r.set_uri_args(args)
r.get_post_args
```

↓

```
r.uri_args
r.uri_args=(args)
r.post_args
```



# p.254 Regular expression

**Use mrbgem based onigumo. It’s embedded with mruby-core box.**

https://github.com/mattn/mruby-onig-regexp

# p.256 Data sharing with request phase

```
location / {
  mruby_rewrite_handler_code '
    Userdata.new.phases = ["rewrite"]
    Nginx.return Nginx::DECLINED
  ';
  mruby_access_handler_code '
    Userdata.new.phases << "access"
    Nginx.return Nginx::DECLINED
  ';
  mruby_content_handler_code '
    Userdata.new.phases << "content"

    Userdata.new.phases.each do |phase|
      Nginx.echo phase
    end
  ';
}
```

You can share local variables use `mruby-userdata` gem across handlers.

You need to return `Nginx::DECLINED` when finished to handler processes. It's specification of ngx_mruby.

# p.258 Data sharing with worker processes

```
mruby_init_code '
  c = Cache.new :namespace => "writing"
  c["publisher"] = "技術評論社"
  c["book"] = "nginx実践入門"
';
server {
  listen       3000;

  location / {
    mruby_content_handler_code '
      c = Cache.new :namespace => "writing"
      Nginx.echo "出版社: #{c["publisher"]}"
      Nginx.echo "書籍名: #{c["book"]}"
    ';
  }
}
```

You can share variables use `mruby-cache` gem across master and worker processes.

`mruby-cache` handles only string keys. It's limitation.

# p.260 Processing with non blocking

Nothing. Stay tune…

Or You can implement it.

# p.262 Authorization using ngx_mruby

```
set $publickey "nginx";
set $privatekey "mruby";

location / {
  mruby_content_handler_code '
    arg_publickey = Nginx::Var.new.arg_publickey
    arg_signature = Nginx::Var.new.arg_signature
    arg_expires  = Nginx::Var.new.arg_expires

    expires = arg_expires.to_i

    publickey  = Nginx::Var.new.publickey
    privatekey = Nginx::Var.new.privatekey

    if arg_publickey && arg_publickey.match(publickey) || !arg_expires
      Nginx.return Nginx::HTTP_FORBIDDEN
    end
```

# p.262 Authorization using ngx_mruby

```
    plaintext = "#{Nginx::Var.new.request_method}#{Nginx::Var.new.uri}#{arg_expires}#{publickey}"
    hmac_sha1 = Digest::HMAC.digest(plaintext, privatekey, Digest::SHA1)
    signature = Base64::encode(hmac_sha1)

    if expires < Time.now.to_i
      Nginx.return Nginx::HTTP_FORBIDDEN
    end

    if signature == arg_signature
      Nginx.echo "Sucess"
    else
      Nginx.return Nginx::HTTP_FORBIDDEN
    end
  ';
}
```

matsumoto-r / ngx_mruby

Unwatch 39 Unstar

Code Issues 5 Pull requests 1 Wiki Pulse Graphs

## Add hsbt as collaborator to ngx_mruby AUTHORS

master

matsumoto-r committed 6 hours ago 1 parent f9e0dc9 commit 231d5786596a87de54aa0d

Showing 1 changed file with 1 addition and 0 deletions.

1 AUTHORS

```
...  ...  @@ -1,3 +1,4 @@
1    1     Original Authors "ngx_mruby developers" are:
2    2        MATSUMOTO Ryosuke
3    3        Tatsuhiko Kubo
     4  +     SHIBATA Hiroshi
```

0 comments on commit 231d578

Lock conversation

https://github.com/matsumoto-r/ngx_mruby