

# HTTP Requests

```
require 'net/http'
```

```
Net::HTTP.get('example.com', '/index.html') # => String
```

```
uri = URI('http://example.com/index.html?count=10')
```

```
Net::HTTP.get(uri) # => String
```

# HTTP Requests

```
uri = URI('http://example.com/index.html')  
params = { :limit => 10, :page => 3 }  
uri.query = URI.encode_www_form(params)  
  
res = Net::HTTP.get_response(uri)  
puts res.body if res.is_a?(Net::HTTPSuccess)
```

# HTTP Requests

```
uri = URI('http://www.example.com/search.cgi')  
res = Net::HTTP.post_form(uri, 'q' => 'ruby', 'max' => '50')  
puts res.body
```

```
uri = URI('http://www.example.com/search.cgi')  
res = Net::HTTP.post_form(uri, 'q' => ['ruby', 'perl'],  
  'max' => '50')  
puts res.body
```

# HTTP Requests

```
uri = URI('http://example.com/some_path?query=string')

Net::HTTP.start(uri.host, uri.port) do |http|
  request = Net::HTTP::Get.new uri

  response = http.request request # Net::HTTPResponse object
end
```

# HTTP Requests

```
uri = URI('http://example.com/index.html')
res = Net::HTTP.get_response(uri)
# Headers
res['Set-Cookie'] # => String
res.get_fields('set-cookie') # => Array
res.to_hash['set-cookie'] # => Array
puts "Headers: #{res.to_hash.inspect}"
# Status
puts res.code # => '200'
puts res.message # => 'OK'
puts res.class.name # => 'HTTPOK'

# Body
puts res.body if res.response_body_permitted?
```

```
def fetch(uri_str, limit = 10)

  raise ArgumentError, 'too many HTTP redirects' if limit == 0

  response = Net::HTTP.get_response(URI(uri_str))

  case response
  when Net::HTTPSuccess then
    response
  when Net::HTTPRedirection then
    location = response['location']
    warn "redirected to #{location}"
    fetch(location, limit - 1)
  else
    response.value
  end
end
```

# POST

```
uri = URI('http://www.example.com/todo.cgi')
req = Net::HTTP::Post.new(uri)
req.set_form_data('from' => '2005-01-01', 'to' => '2005-03-31')

res = Net::HTTP.start(uri.hostname, uri.port) do |http|
  http.request(req)
end

case res
when Net::HTTPSuccess, Net::HTTPRedirection
  # OK
else
  res.value
end
```

# POST

```
req = Net::HTTP::Post.new(uri)  
req.body = multipart_data  
req.content_type = 'multipart/form-data'
```



# Headers

```
uri = URI('http://example.com/cached_response')
file = File.stat 'cached_response'

req = Net::HTTP::Get.new(uri)
req['If-Modified-Since'] = file.mtime.rfc2822

res = Net::HTTP.start(uri.hostname, uri.port) { |http|
  http.request(req)
}

open 'cached_response', 'w' do |io|
  io.write res.body
end if res.is_a?(Net::HTTPSuccess)
```

# Basic Authentication

```
uri = URI('http://example.com/index.html?key=value')
```

```
req = Net::HTTP::Get.new(uri)  
req.basic_auth 'user', 'pass'
```

```
res = Net::HTTP.start(uri.hostname, uri.port) { |http|  
  http.request(req)  
}  
puts res.body
```

# HTTPS

```
uri = URI('https://secure.example.com/some_path?query=string')

Net::HTTP.start(uri.host, uri.port,
                :use_ssl => uri.scheme == 'https') do |http|
  request = Net::HTTP::Get.new uri

  response = http.request request # Net::HTTPResponse object
end
```

# Proxies

```
proxy_addr = 'your.proxy.host'  
proxy_port = 8080
```

```
Net::HTTP.new('example.com', nil,  
              proxy_addr,  
              proxy_port).start  
{ |http|  
  # always proxy via your.proxy.addr:8080  
}
```

# Что выбрать?

- `Net/Http` - да
- `Faraday` - точно да
- `multipart-post` - скорее нет
- `rest-client` - возможно
- `excon` - да
- `typhoeus` - да, если нужно параллельно выполнять множество запросов

HTTP client	Time	Implementation
curb (persistent)	2.519	libcurl wrapper
em-http-request	2.731	EM + http_parser.rb
Typhoeus	2.851	libcurl wrapper
StreamlyFFI (persistent)	2.853	libcurl wrapper
http.rb (persistent)	2.970	Ruby + http_parser.rb
http.rb	3.588	Ruby + http_parser.rb
HTTParty	3.931	Net::HTTP wrapper
Net::HTTP	3.959	Pure Ruby
Net::HTTP (persistent)	4.043	Pure Ruby
open-uri	4.479	Net::HTTP wrapper
Excon (persistent)	4.618	Pure Ruby
Excon	4.701	Pure Ruby
RestClient	26.838	Net::HTTP wrapper

Benchmarks performed using excon's benchmarking tool

DISCLAIMER: Most benchmarks you find in READMEs are crap, including this one. These are out-of-date. If you care about performance, benchmark for yourself for your own use cases!

# OAuth

- OAuth — открытый протокол авторизации, который позволяет предоставить третьей стороне ограниченный доступ к защищённым ресурсам пользователя без необходимости передавать логин и пароль.

# Rack

- Middleware – связующее программное обеспечение.

```
$ rails middleware
```



```
use Rack::Sendfile
use ActionDispatch::Static
use ActionDispatch::Executor
use ActiveSupport::Cache::Strategy::LocalCache::Middleware
use Rack::Runtime
use Rack::MethodOverride
use ActionDispatch::RequestId
use ActionDispatch::RemoteIp
use Sprockets::Rails::QuietAssets
use Rails::Rack::Logger
use ActionDispatch::ShowExceptions
use WebConsole::Middleware
use ActionDispatch::DebugExceptions
use ActionDispatch::Reloader
use ActionDispatch::Callbacks
use ActiveRecord::Migration::CheckPending
use ActionDispatch::Cookies
use ActionDispatch::Session::CookieStore
use ActionDispatch::Flash
use Rack::Head
use Rack::ConditionalGet
use Rack::ETag
use Warden::Manager
use OmniAuth::Builder
run Rndgd::Application.routes
```

# Настройка Rack

```
# config/application.rb
```

```
config.middleware.use Rack::BounceFavicon
```

```
config.middleware.insert_after ActionDispatch::Executor,  
                               Lifo::Cache, page_cache: false
```

```
config.middleware.swap ActionDispatch::ShowExceptions, Lifo::ShowExceptions
```

```
config.middleware.delete "Rack::Runtime"
```

# Конфигурация

```
Rails.application.config.middleware.use OmniAuth::Builder do  
  provider :developer unless Rails.env.production?  
  provider :bnet, ENV['BATTLE_NET_ID'], ENV['BATTLE_NET_SECRET']  
  provider :github, ENV['GITHUB_ID'], ENV['GITHUB_SECRET']  
end
```

## Register a new OAuth application

---

**Application name**

Something users will recognize and trust

**Homepage URL**

The full URL to your application homepage

**Application description**

This is displayed to all users of your application

**Authorization callback URL**

Your application's callback URL. Read our [OAuth documentation](#) for more information.

---

[Register application](#)

[Cancel](#)

---



## Authorize rndgd



**rndgd** by [Wolwer1nE](#)

wants to access your Wolwer1nE account



**Public data only**

Limited access to your public data [...](#)

Authorize Wolwer1nE

Authorizing will redirect to  
<https://rndgd.herokuapp.com>

Not owned or operated by GitHub

Created day ago

Fewer than 10 GitHub users

[Learn more about OAuth](#)

/auth/github/callback?code=...&state=...

```
get '/auth/:provider/callback', to: 'sessions#create'
```

```
class SessionsController < ApplicationController
  def create
    render text: request.env['omniauth.auth'].to_yaml
  end
end
```

```
class PeopleController < ActionController::Base
  # This will raise an ActiveRecord::ForbiddenAttributesError exception
  # because it's using mass assignment without an explicit permit
  # step.
  def create
    Person.create(params[:person])
  end

  # This will pass with flying colors as long as there's a person key
  # in the parameters, otherwise it'll raise a
  # ActionController::ParameterMissing exception, which will get
  # caught by ActionController::Base and turned into a 400 Bad
  # Request error.
  def update
    person = current_account.people.find(params[:id])
    person.update!(person_params)
    redirect_to person
  end

  private
  # Using a private method to encapsulate the permissible parameters
  # is just a good pattern since you'll be able to reuse the same
  # permit list between create and update. Also, you can specialize
  # this method with per-user checking of permissible attributes.
  def person_params
    params.require(:person).permit(:name, :age)
  end
end
```

# Секреты приложения

```
development:
  secret_key_base: fe904cab216cff3e3c0773f7b9a085a3500adab34ad929c0049daf68e6fa0b011b6bba6241aa5cd7b8700011299d2852435509
  battle_net_id: 72216ea01ee0453e85edb7d761a6efe0
  battle_net_secret:

test:
  secret_key_base: 973f22cc7c965af388779e15f47a61f17e0dce7f319b2f56f31ddladd87d8ddb8446aa47cf47899356d8a7ffdaf97ae384d7ff

# Do not keep production secrets in the unencrypted secrets file.
# Instead, either read values from the environment.
# Or, use `bin/rails secrets:setup` to configure encrypted secrets
# and move the `production:` environment over there.

production:
  secret_key_base: <%= ENV["SECRET_KEY_BASE"] %>
  battle_net_id: 72216ea01ee0453e85edb7d761a6efe0
  battle_net_secret:
```





You don't have any apps yet

Every app and pipeline you create or become a collaborator on will appear here

Create new app

Looking for help getting started?

Get started by reading one of our language guides in the Dev Center

Choose a language guide... ▾

App name

rndgd



rndgd is available

Choose a region



Europe



Add this app to a pipeline

Pipelines form a continuous deployment workflow and enable additional features. [Learn more.](#)



Choose a pipeline



Create app

## Deployment method



Heroku Git  
Use Heroku CLI



GitHub  
Connect to GitHub



Container Registry  
Use Heroku CLI

## Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

 Wolwer1nE

rdgd

Search

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

 Wolwer1nE/rdgd

Connect

## Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

Automatic deploys from `master` are enabled

Every push to `master` will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. [Learn more.](#)

Wait for CI to pass before deploy

Only enable this option if you have a [Continuous Integration](#) service configured on your repo.

Disable Automatic Deploys



---

## Manual deploy

Deploy the current state of a branch to this app.


Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

`master`

Deploy Branch













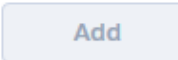


## Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

### Config Vars

Hide Config Vars

LANG	en_US.UTF-8		
RACK_ENV	production		
RAILS_ENV	production		
RAILS_LOG_TO_STDOUT	enabled		
RAILS_SERVE_STATIC_FILES	enabled		
SECRET_KEY_BASE	584cf419604e4df85f4408cfb0bc419e8635989		
KEY	VALUE		

<https://devcenter.heroku.com/articles/getting-started-with-rails5>

